

**RANLP**

**INTERNATIONAL CONFERENCE**

**RECENT ADVANCES IN**

**NATURAL LANGUAGE PROCESSING**

**PROCEEDINGS**

Edited by  
Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, Nicolas Nicolov, Nikolai Nikolov

**27-29 September 2007, Borovets, Bulgaria**

**XEROX.**  
Research Centre Europe



**INTERNATIONAL CONFERENCE**

**RECENT ADVANCES IN**

**NATURAL LANGUAGE PROCESSING**

**P R O C E E D I N G S**

Edited by  
Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, Nicolas Nicolov, Nikolai Nikolov

Borovets, Bulgaria

27-29 September 2007

INTERNATIONAL CONFERENCE  
RECENT ADVANCES IN  
NATURAL LANGUAGE PROCESSING' 2007

**PROCEEDINGS**

Borovets, Bulgaria  
27-29 September 2007

ISBN 978-954-91743-7-3

Designed and Printed by INCOMA Ltd.  
Shoumen, BULGARIA

# ORGANISERS AND SPONSORS

## **The International Conference RANLP–2007 is organised by:**

Linguistic Modelling Department, Institute for Parallel Processing (IPP),  
Bulgarian Academy of Sciences (BAS)

and

Association for Computational Linguistics - Bulgaria

## **RANLP–2007 is partially supported by:**

The European Commission via the project BIS-21++,  
FP6 contract no. INCO-CT-2005-016639

IPP-BAS (BIS-21 Centre of Excellence)

Xerox Research Centre Europe

Association for Computational Linguistics - Bulgaria

## **The team behind RANLP–2007:**

<b>Galia Angelova</b>	Bulgarian Academy of Sciences, Bulgaria, OC Chair
<b>Kalina Bontcheva</b>	University of Sheffield, U.K.
<b>Ruslan Mitkov</b>	University of Wolverhampton, U.K., PC Chair
<b>Nicolas Nicolov</b>	Umbria, Inc., Boulder, U.S.A.
<b>Nikolai Nikolov</b>	INCOMA Ltd., Shoumen, Bulgaria
<b>Kiril Simov</b> coordinator)	Bulgarian Academy of Sciences, Bulgaria (workshop



## PROGRAMME COMMITTEE CHAIR

Ruslan Mitkov (University of Wolverhampton, UK)

## PROGRAMME COMMITTEE

Eneko Agirre (*Basque Country University, Spain*)  
Elisabeth Andre (*Univ. of Augsburg, Germany*)  
Laura Alonso i Alemany (*Univ. de la República, Uruguay & Univ. Nacional de Córdoba, Argentina*)  
Galia Angelova (*BAS, Bulgaria*)  
Amit Bagga (*IAC Search & Media, USA*)  
Marco De Boni (*Unilever, UK*)  
Branimir Boguraev (*IBM, USA*)  
Kalina Bontcheva (*Univ. of Sheffield, UK*)  
Antonio Branco (*Univ. of Lisbon, Portugal*)  
Kevin Bretonnel Cohen (*Univ. of Colorado, USA*)  
Sylviane Cardey (*Univ. of Franche-Comté, France*)  
Dan Cristea (*Al.I. Cuza Univ. of Iasi, Romania*)  
Hamish Cunningham (*Univ. of Sheffield, UK*)  
Walter Daelemans (*Univ. of Antwerp, Belgium*)  
Ido Dagan (*Bar-Ilan Univ., Israel*)  
Robert Dale (*Macquarie Univ., Australia*)  
Dekai Wu (*Hong Kong Univ. S&T, Hong Kong*)  
Rodolfo Delmonte (*Ca' Foscari Univ., Italy*)  
Gael Dias (*Univ. of Beira Interior, Portugal*)  
Robert Gaizauskas (*Univ. of Sheffield, UK*)  
Alexander Gelbukh (*Nat. Polytechnic Inst., Mexico*)  
Gregory Grefenstette (*LIC2M, CEA-LIST, France*)  
Johann Haller (*IAI, Saarbrücken, Germany*)  
Catalina Hallett (*The Open Univ., UK*)  
Patrick Hanks (*Masaryk Univ., Czech Republic*)  
Michael Hess (*Univ. of Zürich, Switzerland*)  
Erhard Hinrichs (*Eberhard Karls Univ., Germany*)  
Veronique Hoste (*Univ. College Ghent, Belgium*)  
Gerhard van Huyssteen (*North-West Univ., S. Africa*)  
Diana Inkpen (*Univ. of Ottawa, Canada*)  
Hitoshi Isahara (*NICT, Japan*)  
Graeme Hirst (*Univ. of Toronto, Canada*)  
Frances Johnson (*Manchester Metrop. Univ., UK*)  
Mijail A. Kabadjov (*Univ. of Edinburgh, UK*)  
Asanee Kawtrakul (*Kasetsart University, Thailand*)  
Dimitar Kazakov (*University of York, UK*)  
Alma Kharrat (*Microsoft, USA*)  
Richard Kittredge (*CoGenTex, USA*)  
Steven Krauwer (*Univ. of Utrecht, The Netherlands*)  
Hristo Krushkov (*Plovdiv Univ., Bulgaria*)  
Udo Kruschwitz (*Univ. of Essex, UK*)  
Sandra Kuebler (*Indiana Univ., USA*)  
Lori Lamel (*LIMSI - CNRS, France*)  
Mirella Lapata (*Univ. of Edinburgh, UK*)  
Shalom Lappin (*King's College, UK*)  
Anke Ludeling (*Humboldt Univ., Germany*)  
Bernardo Magnini (*FBK-irst, Italy*)  
Inderjeet Mani (*Georgetown Univ., USA*)  
Montserrat M. Anglada (*Basque Country Univ., Spain*)  
Patricio Martinez-Barco (*Univ. of Alicante, Spain*)  
Yuji Matsumoto (*NAIST, Japan*)  
Wolfgang Menzel (*Univ. of Hamburg, Germany*)  
Rada Mihalcea (*Univ. of North Texas, USA*)  
Andrei Mikheev (*Infogistics Ltd & Daxtra Technol. Ltd, UK*)  
Leonel Miyares (*Centre for Applied Linguistics, Cuba*)  
Dunja Mladenic (*Josef Stefan Inst., Slovenia*)  
Andres Montoyo (*Univ. of Alicante, Spain*)  
Rafael Munoz Guillena (*Univ. of Alicante, Spain*)  
Masaki Murata (*NICT, Japan*)  
Makoto Nagao (*National Diet Library, Japan*)  
Preslav Nakov (*Univ. of California, USA*)  
Vivi Nastase (*EML Research, Germany*)  
Roberto Navigli (*Univ. di Roma La Sapienza, Italy*)  
Ani Nenkova (*Univ. of Pennsylvania, USA*)  
Nicolas Nicolov (*Umbria Inc., USA*)  
Michael Oakes (*Univ. of Sunderland, UK*)  
Kemal Oflazer (*Sabancı Univ., Turkey*)  
Constantin Orasan (*Univ. of Wolverhampton, UK*)  
Petya Osenova (*BAS, Bulgaria*)  
Manuel Palomar (*Univ. of Alicante, Spain*)  
Viktor Pekar (*Univ. of Wolverhampton, UK*)  
Stelios Piperidis (*ILSP, Greece*)  
Aurora Pons (*Univ. de Oriente, Cuba*)  
Oana Postolache (*Univ. Southern California, USA*)  
John Prager (*IBM, USA*)  
Gabor Proszeky (*MorphoLogic, Hungary*)  
Stephen Pulman (*Oxford Univ., UK*)  
Allan Ramsay (*Univ. of Manchester, UK*)  
Ellen Riloff (*Univ. of Utah, USA*)  
Horacio Rodriguez (*Technical Univ. of Catalonia, Spain*)  
Anne De Roeck (*The Open Univ., UK*)  
Horacio Saggion (*Univ. of Sheffield, UK*)  
Christer Samuelsson (*Umbria Inc., USA*)  
Frederique Segond (*Xerox Research Centre, France*)  
Khaled Shaalan (*British Univ. in Dubai, U. Arab Emirates*)  
Kiril Simov (*BAS, Bulgaria*)  
Ralf Steinberger (*EC Joint Research Centre, Italy*)  
Keh-Yih Su (*Behavior Design Corporation, Taiwan*)  
Jana Sukkarieh (*ETS, USA*)  
John Tait (*Univ. of Sunderland, UK*)  
Mike Thelwall (*Univ. of Wolverhampton, UK*)  
Kristina Toutanova (*Microsoft, USA*)  
Harald Trost (*Medical Univ. of Vienna, Austria*)  
Dan Tufis (*Research Institute for AI, Romania*)  
L. Alfonso Urena Lopez (*Univ. of Jaen, Spain*)  
Karin Verspoor (*Los Alamos National Lab., USA*)  
Manuel Vilares Ferro (*Univ. of Corunna, Spain*)  
Aline Villavicencio (*Fed. Univ. of Rio Grande do Sul, Brazil*)  
Yorick Wilks (*Univ. of Sheffield, UK*)  
Piek Vossen (*Irion Technologies BV, The Netherlands*)  
Michael Zock (*LIF, CNRS, France*)

## REVIEWERS

In addition to the members of the Programme Committee, the following colleagues were involved in the reviewing process

Naveed Afzal (*Univ. of Wolverhampton, UK*)  
Hanady Ahmed (*Univ. of Alexandria, Egypt*)  
Amparo Alcina (*Univ. Jaume I, Spain*)  
Yafa Al-Raheb (*Dublin City Univ., Ireland*)  
Rania Al-Sabbagh (*Ain Shams Univ., Egypt*)  
Elsa Alves (*Univ. of Beira Interior, Portugal*)  
Todor Arnaudov (*Plovdiv Univ., Bulgaria*)  
Svetla Boycheva (*Univ. of Sofia, Bulgaria*)  
Boryana Bratanova (*Univ. of V. Turnovo, Bulgaria*)  
Iria da Cunha Fanego (*Pompeu Fabra Univ., Spain*)  
Maarten de Rijke (*U. Amsterdam, The Netherlands*)  
Rachel Dugdale (*GCHQ, UK*)  
Richard Evans (*Univ. of Wolverhampton, UK*)  
Lisette Garcia Moya (*Univ. de Oriente, Cuba*)  
Laura Hasler (*Univ. of Wolverhampton, UK*)  
Milen Kouylekov (*FBK-irst, Italy*)  
Zornitsa Kozareva (*Univ. of Alicante, Spain*)  
Le An Ha (*Univ. of Wolverhampton, UK*)  
Yaoyong Li (*Sheffield Univ., UK*)

Irina Matveeva (*Univ. of Chicago, USA*)  
Diana Maynard (*Univ. of Sheffield, UK*)  
Dalila Mekhaldi (*Univ. of Wolverhampton, UK*)  
Paul Morarescu (*Univ. of Texas, USA*)  
Andrea Mulloni (*Univ. of Wolverhampton, UK*)  
Shiyan Ou (*Univ. of Wolverhampton, UK*)  
Paul Piwek (*The Open Univ., UK*)  
Georgiana Puscasu (*Univ. of Wolverhampton, UK*)  
Yamile Ramirez-Safar (*Saarland Univ., Germany*)  
Doaa Samy (*Cairo University, Egypt*)  
Miriam Seghiri (*Univ. of Malaga, Spain*)  
Violeta Seretan (*Univ. of Geneva, Switzerland*)  
Smriti Singh Singh (*Indian Institute of Technology, India*)  
Mariona Taule (*Univ. of Barcelona, Spain*)  
Irina Temnikova (*Univ. of Wolverhampton, UK*)  
Sandra Williams (*The Open Univ., UK*)  
Alistair Willis (*The Open Univ., UK*)  
Imed Zitouni (*IBM Research, USA*)

## PROGRAMME COMMITTEE COORDINATOR

Ivelina Nikolova (*Sofia University "St. Kl. Ohridski" and Bulgarian Academy of Sciences, Bulgaria*)

# TABLE OF CONTENTS

Nahed ABUL-HASSAN <i>Improving Tokenization of Clitics in Some Statistical Processing Tools for Arabic: AlwAw Coordinating Conjunction as a Case Example</i> .....	1
Sisay Fissaha ADAFRE, Maarten de RIJKE and Erik Tjong Kim SANG <i>Entity Retrieval</i> .....	5
Stergos AFANTENOS <i>Some Reflections on the Task of Content Determination in the Context of Multi-Document Summarization of Evolving Events</i> .....	12
Rodrigo AGERRI, John A. BARNDEN, Mark G. LEE and Alan M. WALLINGTON <i>Metaphor, Inference and Domain Independent Mappings</i> .....	17
Rayner ALFRED, Dimitar KAZAKOV, Mark BARTLETT, Elena PASKALEVA <i>Hierarchical Agglomerative Clustering of Bulgarian-English Parallel Corpora</i> .....	24
Laura ALONSO, Irene CASTELLÓN, Nevena Tinkova TINCHEVA <i>Obtaining Coarse-Grained Classes of Subcategorization Patterns for Spanish</i> .....	30
Miguel Molinero ÁLVAREZ, Fco. Mario Barcala RODRÍGUEZ, Juan Otero POMBO, Jorge Graña GIL <i>Practical Application of One-Pass Viterbi Algorithm in Tokenization and Part-of-Speech Tagging</i> .....	35
Ana-Maria BARBU, Emil IONESCU <i>Designing a Valence Dictionary for Romanian</i> .....	41
Roberto BASILI, Alfio Massimiliano GLIOZZO, Marco PENNACCHIOTTI <i>Harvesting Ontologies from Open Domain Corpora: a Dynamic Approach</i> .....	46
Fernando BATISTA, Nuno MAMEDE, Diamantino CASEIRO, Isabel TRANCOSO <i>A Lightweight on-the-fly Capitalization System for Automatic Speech Recognition</i> .....	52
José-Miguel BENEDÍ, Joan-Andreu SÁNCHEZ, Alberto SANCHIS <i>Confidence Measures for Stochastic Parsing</i> .....	58
Chris BIEMANN, Claudio GIULIANO, Alfio GLIOZZO <i>Unsupervised Part-Of-Speech Tagging Supporting Supervised Methods</i> .....	64
Patrick BLACKBURN, Sébastien HINDERER <i>Generating Models for Temporal Representations</i> .....	69
Victoria BOBICEV <i>Comparison of Word-Based and Letter-Based Text Classification</i> .....	76
Florian BOUDIN, Juan-Manuel TORRES-MORENO <i>A Cosine Maximization-Minimization Approach for User-Oriented Multi-Document Update Summarization</i> .....	81
Amanda BOUFFIER, Thierry POIBEAU <i>Re-engineering Free Texts to Obtain XML Documents: a Discourse Based Approach</i> .....	88

Christopher BREWSTER, José IRIA, Ziqi ZHANG, Fabio CIRAVEGNA, Louise GUTHRIE, Yorick WILKS <i>Dynamic Iterative Ontology Learning</i> .....	93
Conor CAFFERKEY, Deirdre HOGAN, Josef, GENABITH <i>Multi-Word Units in Treebank-Based Probabilistic Parsing and Generation</i> .....	98
Sander CANISIUS, Antal van den BOSCH <i>Recompiling a Knowledge-Based Dependency Parser into Memory</i> .....	104
Maria Fernanda CAROPRESO, Stan MATWIN <i>Incorporating Syntax and Semantics in the Text Representation for Sentence Selection</i> .....	109
Atanas CHANEV, Kiril SIMOV, Petya OSENOVA, Svetoslav MARINOV <i>The BulTreeBank: Parsing and Conversion</i> .....	114
John CHEN, Laurie CRIST, Len ENYON, Cassandre CRESWELL, Amit MHATRE, Rohini SRIHARI <i>Confidence Measures and Thresholding in Coreference Resolution</i> .....	121
Henning CHRISTIANSEN, Christian Theil HAVE, Knut TVEITANE <i>From Use Cases to UML Class Diagrams Using Logic Grammars and Constraints</i> .....	128
Grzegorz CHRUPALA, Nicolas STROPPIA, Josef van GENABITH, Georgiana DINU <i>Better Training for Function Labeling</i> .....	133
Montse CUADROS, German RIGAU, Mauro CASTILLO <i>Evaluating Large-Scale Knowledge Resources across Languages</i> .....	139
Turhan DAYBELGE, Ilyas CICEKLI <i>A Rule-Based Morphological Disambiguator for Turkish</i> .....	145
Seniz DEMIR, Sandra CARBERRY, Stephanie ELZER <i>Effectively Realizing the Inferred Message of an Information Graphic</i> .....	150
Mona DIAB <i>Towards an Optimal POS Tag Set for Arabic Processing</i> .....	157
Mona DIAB, Alessandro MOSCHITTI <i>Semantic Parsing of Modern Standard Arabic</i> .....	162
Markus DICKINSON <i>Determining Ambiguity Classes for Part-of-Speech Tagging</i> .....	167
Georgiana DINU, Sandra KÜBLER <i>Sometimes Less Is More: Romanian Word Sense Disambiguation Revisited</i> .....	173
Asif EKBAL, Sivaji BANDYOPADHYAY <i>Recognition and Transliteration of Bengali Named Entities: A Computational Approach</i> .....	178
Natalia ELITA, Monica GAVRILA, Cristina VERTAN <i>Experiments with String Similarity Measures in the EBMT Framework</i> .....	183
Sergio FERRÁNDEZ, Óscar FERRÁNDEZ, Antonio FERRÁNDEZ, Rafael MUÑOZ <i>The Importance of Named Entities in Cross-Lingual Question Answering Scenarios</i> .....	188

Debora FIELD, Allan RAMSAY <i>Minimal Sets of Minimal Speech Acts</i> .....	193
Fumiyo FUKUMOTO, Yoshimi SUZUKI <i>Integrating Cross-Language Hierarchies by Text Classification</i> .....	200
Kotaro FUNAKOSHI, Mikio NAKANO, Yuji HASEGAWA, Hiroshi TSUJINO <i>Semantic Interpretation Supplementarily Using Syntactic Analysis</i> .....	205
Lisette GARCÍA-MOYA, Aurora PONS-PORRATA, Leonel RUIZ-MIYARES <i>A Proposal of a Morphological Tagger for Spanish Based on Cuban Corpora</i> .....	210
Guillem GASCÓ, Joan Andreu SÁNCHEZ <i>A* Parsing with Large Vocabularies</i> .....	215
Milagros Fernández GAVILANES, Eric Villemonte de la CLERGERIE, Manuel VILARES-FERRO <i>Knowledge Acquisition through Error-Mining</i> .....	220
Luiz GENOVES Jr, Richard LIZOTTE, Ethel SCHUSTER, Carmen DAYRELL, Sandra ALUÍSIO <i>A Two-Tiered Approach to Detecting English Article Usage: an Application in Scientific Paper Writing Tools</i> .....	225
Olga GERASSIMENKO, Riina KASTERPALU, Mare KOIT, Andriela RÄÄBIS, Krista STRANDSON <i>Initial Requests in Institutional Calls: Corpus Study</i> .....	230
Emiliano GIOVANNETTI, Simone MARCHI, Simonetta MONTEMAGNI, Roberto BARTOLINI <i>Ontology-based Semantic Annotation of Product Catalogues</i> .....	235
Felix GOLCHER <i>A Stable Statistical Constant Specific for Human Language Texts</i> .....	240
Carlos GÓMEZ-RODRÍGUEZ, Jesús VILARES, Miguel A. ALONSO <i>Prototyping Efficient Natural Language Parsers</i> .....	246
Hendrik Johannes GROENEWALD, Gerhard Beukes van HUYSSSTEEN, Martin Johannes PUTTKAMMER <i>Evaluating Wrapped Progressive Sampling for Automatic Algorithmic Parameter Optimisation</i> .....	251
Le An HA <i>Exploiting Glossaries for Automatic Terminology Processing</i> .....	256
Catherine HAVASI, Robert SPEER, Jason ALONSO <i>ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge</i> .....	261
Chung-Chi HUANG, Wei-Teh CHEN, Jason S. CHANG <i>Improving Word Alignment Based on Extended Inversion Transduction Grammar</i> .....	268
Tomoya IAWKURA, Seishi OKAMOTO <i>Fast Training Methods of Boosting Algorithms for Text Analysis</i> .....	274
Arantza Díaz de ILARRAZA, Koldo GOJENOLA, Maite ORONOZ <i>Reusability of a Corpus and a Treebank to Enrich Verb Subcategorisation in a Dictionary</i> .....	280

Ozlem ISTEK, Ilyas CICEKLI <i>A Link Grammar for an Agglutinative Language</i> .....	285
Aminul ISLAM, Diana INKPEN <i>Semantic Similarity of Short Texts</i> .....	291
Rubén IZQUIERDO, Armando SUÁREZ, German RIGAU <i>Exploring the Automatic Selection of Basic Level Concepts</i> .....	298
Heng JI, Ralph GRISHMAN <i>Collaborative Entity Extraction and Translation</i> .....	303
Sittichai JIAMPOJAMARN, Grzegorz KONDRAK, Colin CHERRY <i>Biomedical Term Recognition Using Discriminative Training</i> .....	310
Kate H. KAO, James M. LEE, Richard Y. CHANG, Jason S. CHANG <i>Giving Semantic Structure to Verbs in the Context of VN Collocations</i> .....	317
Manfred KLENNER <i>Enforcing Consistency on Coreference Sets</i> .....	323
Zornitsa KOZAREVA, Sonia VAZQUEZ, Andres MONTOYO <i>Discovering the Underlying Meanings and Categories of a Name through Semantic and Domain Information</i> .....	329
Cornelis H.A. KOSTER, Marc SEUTTER, Olaf SEIBER <i>Parsing the Medline Corpus</i> .....	335
Ralf KRESTEL, Rene WITTE, Sabine BERGLER <i>Processing of Beliefs Extracted from Reported Speech in Newspaper Articles</i> .....	340
Abolfazl Keighobadi LAMJIRI, Leila KOSSEIM, Thiruvengadam RADHAKRISHNAN <i>A Syntactic Candidate Ranking Method for Answering Questions with a Main Content Verb</i> .....	345
Jianguo LI, Chris BREW <i>Disambiguating Levin Verbs Using Untagged Data</i> .....	351
Berenike LOOS, Hans-Peter ZORN <i>Combining Information Extraction and Knowledge Acquisition for Spoken Dialog Systems</i> .....	357
Tanja MATHIS, Dennis SPOHR <i>Corpus-Driven Enhancement of a BCI Spelling Component</i> .....	362
Lukas MICHELbacher, Stefan EVERT, Hinrich SCHÜTZE <i>Asymmetric Association Measures</i> .....	367
Daniel MICOL, Óscar FERRÁNDEZ, Rafael MUÑOZ, Manuel PALOMAR <i>A Semantic-Less Approach for the Textual Entailment Recognition Task</i> .....	373
Karo MOILANEN, Stephen PULMAN <i>Sentiment Composition</i> .....	378
Rumen MORALIYSKI, Gaël DIAS <i>One Sense Per Discourse for Synonym Detection</i> .....	383

Roser MORANTE, Antal van den BOSCH <i>Memory-Based Semantic Role Labeling of Catalan and Spanish</i> .....	388
Pilar López MORENO, Antonio FERRÁNDEZ, Sandra ROGER, Sergio FERRÁNDEZ <i>The Problems in a Question Answering System in the Academic Domain</i> .....	395
Preslav NAKOV, Svetlin NAKOV, Elena PASKALEVA <i>Improved Word Alignments Using the Web as a Corpus</i> .....	400
Vivi NASTASE, Marina SOKOLOVA, Jelber Sayyad SHIRABAD <i>Do Happy Words Sound Happy? A Study of Relations between Form and Meaning for English Words Expressing Emotions</i> .....	406
Costanza NAVARRETTA <i>Semi-Automatic Construction of Training Data for Tagging Non-Contemporary Literary Texts</i> .....	411
Yun NIU, Graeme HIRST <i>Identifying Cores of Semantic Classes in Unstructured Text with a Semi-supervised Learning Approach</i> .....	418
Elisa NOGUERA, Fernando LLOPIS, Antonio FERRÁNDEZ, Alberto ESCAPA <i>Exploring New Measures for Open-Domain Question Answering Evaluation within a Time Constraint</i> .....	425
Constantin ORĂSAN <i>Pronominal Anaphora Resolution for Text Summarisation</i> .....	430
Constantin ORĂSAN, Laura HASLER <i>Computer-Aided Summarisation: How Much Does It Really Help?</i> .....	437
Shiyan OU, Christopher S. G. KHOO, Dion H. GOH <i>Multi-document Summarizing Focusing on Extracting and Integrating Similarities and Differences among Documents</i> .....	442
Lilja ØVRELID, Joakim NIVRE <i>When Word Order and Part-of-Speech Tags Are not Enough -- Swedish Dependency Parsing with Rich Linguistic Features</i> .....	447
Marco PENNACCHIOTTI, Roberto BASILI, Diego De CAO, Paolo MAROCCO <i>Learning Selectional Preferences for Entailment or Paraphrasing Rules</i> .....	452
Marco PENNACCHIOTTI, Fabio Massimo ZANZOTTO <i>Learning Shallow Semantic Rules for Textual Entailment</i> .....	458
Guy PERRIER <i>A French Interaction Grammar</i> .....	463
William PHILLIPS, Ellen RILOFF <i>Exploiting Role-Identifying Nouns and Expressions for Information Extraction</i> .....	468
Jakub PISKORSKI <i>On Some Aspects of Implementing a Pattern Engine Based on Regular Expressions over Feature Structures</i> .....	474



Natalia PONOMAREVA, Paolo ROSSO, Ferran PLA, Antonio MOLINA <i>Conditional Random Fields vs. Hidden Markov Models in a Biomedical Named Entity Recognition Task</i> .....	479
Marius POPESCU, Liviu P. DINU <i>Kernel Methods and String Kernels for Authorship Identification: The Federalist Papers Case</i> .....	484
Bruno POULIQUEN, Ralf STEINBERGER, Clive BEST <i>Automatic Detection of Quotations in Multilingual News</i> .....	487
Georgiana PUȘCAȘU <i>Discovering Temporal Relations with TICTAC</i> .....	493
Catherine RECANATI, Nicoleta ROGOVSCHI, Younès BENNANI <i>Sequencing of Verb - a Study on Tense and Aspect Using Unsupervised Learning</i> .....	499
Marta RECASENS, M. Antònia MARTÍ, Marionna TAULÉ <i>Where Anaphora and Coreference Meet. Annotation in the Spanish CESS-ECE Corpus</i> .....	504
Georg REHM, Richard ECKART, Christian CHIARCOS <i>An OWL- and XQuery-Based Mechanism for the Retrieval of Linguistic Patterns from XML-Corpora</i> ...	510
Jean ROYAUTÉ, Elisabeth GODBERT and Mohamed Mahdi MALIK <i>Identifying Relations between Scientific Objects within Predicate Structures</i> .....	515
Michael SCHIEHLEN <i>Global Learning of Context Weights from GermaNet</i> .....	520
Khaled SHAALAN, Hitham M. Abo BAKR, Ibrahim ZIEDAN <i>Transferring Egyptian Colloquial Dialect into Modern Standard Arabic</i> .....	525
Vera SHEINMAN, Takenobu TOKUNAGA <i>WordSets: Finding Lexically Similar Words for Second Language Acquisition</i> .....	530
Anil Kumar SINGH, Samar HUSAIN, Harshit SURANA, Jagadeesh GORLA, Dipti Misra SHARMA, Chinnappa GUGGILLA <i>Disambiguating Tense, Aspect and Modality Markers for Correcting Machine Translation Errors</i> .....	536
Ielka van der SLUIS, Albert GATT, Kees van DEEMTER <i>Evaluating Algorithms for the Generation of Referring Expressions: Going Beyond Toy Domains</i> .....	541
Anders SØGAARD, Timm LICHTTE, Wolfgang MAIER <i>The Complexity of Linguistically Motivated Extensions of Tree-Adjoining Grammar</i> .....	548
Kristina SPRANGER <i>Combining Deterministic Processing with Ambiguity-Awareness</i> .....	553
Idan SZPEKTOR, Ido DAGAN <i>Learning Canonical Forms of Entailment Rules</i> .....	558
Martha Yifiru TACHBELIE, Wolfgang MENZEL <i>Sub-word Based Language Modeling for Amharic</i> .....	564

Olivier TARDIF, Grégory SMITS <i>Resolving Coreference Using an Outranking Approach</i> .....	571
Jörg TIEDEMANN <i>Comparing Document Segmentation Strategies for Passage Retrieval in Question Answering</i> .....	576
Jörg TIEDEMANN <i>Improved Sentence Alignment for Movie Subtitles</i> .....	582
Nevena Tinkova TINCHEVA, Irene Castellón MASALLES <i>A Comparative Study of Parsers Outputs for Spanish</i> .....	589
Amalia TODIRASCU, Cristopher GLEDHILL, Dan STEFANESCU <i>Extracting Collocations in Context: the Case of Romanian VN Constructions</i> .....	594
Antonio TORAL, Monica MONACHINI <i>Formalising and Bottom-up Enriching the Ontology of a Generative Lexicon</i> .....	599
Antonio TORAL, Rafael MUÑOZ <i>Towards a Named Entity WordNet (NEWN)</i> .....	604
François TROUILLEUX <i>Specifying Properties of a Language with Regular Expressions</i> .....	609
Andrejs VASILJEVS, Signe RIRDANCE <i>Consolidation and Unification of Dispersed Multilingual Terminology Data</i> .....	614
Marc VILAIN, Jonathan GIBSON, Rob QUIMBY <i>Table Classification: an Application of Machine Learning to Web-hosted Financial Texts</i> .....	619
Jesus VILARES, Michael P. OAKES, Manuel VILARES <i>A Knowledge-Light Approach to Query Translation in Cross-Language Information Retrieval</i> .....	624
René WITTE, Sabine BERGLER <i>Next-Generation Summarization: Contrastive, Focused and Update Summaries</i> .....	631
René WITTE, Ting TANG <i>Task-Dependent Visualization of Coreference Resolution Results</i> .....	637
Markus WEIMER, Iryna GUREVYCH <i>Predicting the Perceived Quality of Web Forum Posts</i> .....	643
Davy WEISSENBACHER, Adeline NAZARENKO <i>A Bayesian Approach Combining Surface Clues and Linguistic Knowledge: Application to the Anaphora Resolution Problem</i> .....	649
Xiaohong WU, Yujie ZHANG, Hitoshi ISAHARA, Sylviane CARDEY <i>Error Analysis to Translations by MT Systems</i> .....	654
Katerina ZDRAVKOVA, Aleksandar PETROVSKI <i>Derivation of Macedonian Verbal Adjectives</i> .....	661

# Improving Tokenization of Clitics in Some Statistical Processing Tools for Arabic: AlwAw Coordinating Conjunction as a Case Example

Nahed Abul-Hassan  
Ain Shams University, Faculty of Alsun  
Egypt, Cairo  
[nahed.salma@yahoo.com](mailto:nahed.salma@yahoo.com)

## Abstract

Morphological segmentation of clitics is a key first step in syntactic disambiguation in Arabic. Therefore, in this paper, we present a method for improving morphological segmentation, and hence POS tagging, of Arabic words containing the ambiguous form الواو /AlwAw/ ('and'), using *ASVMTools*. Our hypothesis enhances accuracy rate to 97.4% by a single preprocessing step in input text.

**Index terms:** Morphological Segmentation, POS Tagging, Clitics, Coordinating Conjunctions, *ASVMTools*.

## 1 Introduction

Morphological Segmentation is the process of segmenting clitics from stems. Prepositions, conjunctions, and some pronouns are cliticized onto stems in Arabic [3]. This paper focuses on the morphological segmentation of الواو /AlwAw/ ('and') (see appendix 1 for transliteration convention) as a case example. الواو /AlwAw/ ('and') is the most commonly used coordinating conjunction in Arabic and a common source of morphological ambiguity. According to a manual evaluation of a random sample of 100k Arabic word tokens derived from newswire articles<sup>1</sup>(2006), it has been found that الواو /AlwAw/ ('and') alone accounts for approximately 8.6% of any written text.

Unlike the English coordinator *and*, الواو /AlwAw/ can be morphologically ambiguous: it can function as a coordinating conjunction or as part of a word. For example, وحدة /whdp/ can be either وحدة /whdp/ ('unity') or وحدة + و /w + hdp/ ('and

intensity'). It is worth noting that الواو /alwaw/ ('and') can be distinguished phonologically to be part of the word or a coordinating conjunction. However, when dealing with written text ambiguity arises.

The rest of this paper is divided as follows. Section 2 gives a brief background about different approaches to Arabic morphological segmentation. The hypothesis and our tools are given in section 3. Section 4 presents an evaluation of our work according to standard evaluation metrics. The conclusion and further suggestions for future work are given in section 5.

## 2 Related Work

This section represents a literature survey of different approaches to Arabic morphological segmentation and POS tagging, with an emphasis on Automatic Tagging of Arabic Text Using SVM (*ASVMTools*), upon which this work is based.

### 2.1 AraMorph

Buckwalter (2002) has introduced AraMorph<sup>2</sup> which applies a dictionary-based approach to Arabic morphological segmentation and POS tagging. In AraMorph, morphological analysis depends on a dictionary of prefixes, a dictionary of suffixes, a stem dictionary, and three checking tables for testing the validity of a word analysis. The system uses Latin characters, as input Arabic words are transliterated, and the linguistic data inside the system are represented in Latin characters as well (using Buckwalter transliteration system) [1].

### 2.2 Language Model Based Arabic Word Segmentation

Lee et al (2003) have presented a statistical approach for Arabic morphological analysis. They segment a

<sup>1</sup> Alahram Newspaper: <http://www.ahram.org.eg>

<sup>2</sup> <http://www.nongnu.org/aramorph/english/download.html>

word into prefix- stem-suffix sequence. This system depends on three linguistic resources: a small corpus manually segmented, a large unsegmented corpus, and a table of Arabic prefixes and suffixes. The authors choose to use the stem, not the root, in their approach. They believe that the stem as a morpheme is more suitable than the root in their applications (information retrieval and translation). A trigram language model is used to segment a word into its component. Their Arabic word segmentation system has achieved an accuracy rate of 97% on a test corpus containing 28,449 word tokens provided by LDC Arabic Treebank<sup>3</sup> [5].

### 2.3 Nizar and Rambow

Nizar and Rambow (2005) have presented an approach in which they use a morphological analyzer for morphological segmentation and POS tagging of Arabic words. In this approach, morphological segmentation and POS tagging are considered the same operation, which consists of three phases. First, they obtain from their morphological analyzer (*i.e. Almorgeana*) a list of all possible analyses for the words in a given sentence. Then, they apply classifiers for ten morphological features to the words of the text. Then, they choose among the analyses returned by the morphological analyzer by using the output of the classifier [4]. It has been reported that this approach achieves a precision rate of 98.6% (token-based) in morphological segmentation and 94.3% (word-based) in POS tagging.

### 2.4 Automatic Tagging of Arabic Text using SVM (*ASVMTools*)

Developed by Diab et al (2004), *ASVMTools* provide solutions to fundamental NLP problems such as Morphological Segmentation, Part-Of-Speech (POS) Tagging and Base Phrase (BP) Chunking. Morphological Segmentation (section I) is the process of segmenting clitics from stems, such as separating "ها" /ha/ ('her') from "كتابها" /kitAbahA/ ('her book'). In POS tagging, segmented words have been annotated with parts of speech drawn from the "collapsed" Arabic Penn Treebank POS tag set. This collapsed tag set is as follows: {*CC, CD, CONJ+NEG\_PART, DT, FW, IN, JJ, NN, NNP, NNPS, NO-FUNC, NUMERIC\_COMMA, PRP, PRP\$, PUNC, RB, UH, VBD, VBN, VBP, WP, WRB*}<sup>4</sup>. BP chunking is the process of creating non-recursive base phrases such as noun phrases, adjectival phrases, etc.

Diab et al have adopted a statistical approach using a language- independent algorithm trained on Arabic Penn Treebank. Arabic Penn Treebank is a modern standard Arabic corpus containing 734 news articles from *Agence France Presse* and covering various topics such as sports, politics, news, etc. Using standard evaluation metrics, they have reported that the Morphological Segmentation has achieved an accuracy of 99.12%, the POS Tagger yields 95.49%, and the BP Chunker has a precision of 92.08%. Morphological ambiguity is not taken into consideration during evaluation.

To the best of the author's knowledge, *ASVMTools* are significant for a number of reasons. First, like most non-European languages, Arabic is lacking in annotated resources and tools. Second, Arabic processing tools are fundamental for almost all NLP applications, such as machine translation (MT), text-to-speech, text summarization, etc. Third, they are for public use<sup>5</sup>.

*ASVMTools* have achieved a precision rate of 83.5% in the morphological segmentation of الواو /AlwAw/ ('and'). This is according to a random sample consisting of 3k Arabic word tokens extracted from newswire articles (1999) and processed by *ASVMTools*. See the following example;

#### Coordinator 2nd conjunct

Arabic: اعتقد و  
 Translit: /w/ /AEtqd/  
 Gloss: and he thought  
*ASVMTools*' output: < wAEtqd/JJ>

In fact, incorrect morphological segmentation produces incorrect part-of-speech tags.

## 3 Experimental Setup

We assume that by segmenting clitics in input text before being submitted to the *ASVMTools*, we improve both morphological segmentation and POS tagging. This assumption has been tested on الواو /AlwAw/ clitic. Using *Perl* script language, we separate every initial واو /wAw/ in input text, except those that are in lexica. Our hypothesis is that every واو /wAw/ is a coordinating conjunction unless it is part of an entry in lexica, such as الواو /AlwAw/ in وفاة /wfAp/ ('death'), for instance.

The lexica utilized are:

#### A. Al-mawrid Lexicon:

<sup>3</sup> <http://www ldc.uppen.edu>

<sup>4</sup> <http://www ircs.upenn.edu/arabic/manuals/tagguide.pdf>

<sup>5</sup> <http://www1.cs.columbia.edu/~mdiab/>

It contains 13553 stems including those for الواو /AlwAw/. It is found within Buckwalter's package for morphological segmentation (2002). Short vowels and diacritics are included in this lexicon.

## B. A Lexicon of proper names & country names:

The lexicon of proper names is extracted from Al-asmaa website<sup>6</sup> and consists of a list of 1682 male and female names which are alphabetically arranged. Regarding that of country names, it is acquired through a second language (English). First, it is extracted from a geography website<sup>7</sup>. Then, the output is submitted to Golden Al-Wafi<sup>8</sup> English-Arabic Machine Translation system, resulting in 477 possible country names.

## 4 Evaluation

Table 1 presents the results obtained using our hypothesis, compared against Diab's. Our test set is a random sample of 10k tokens derived from newswire articles (1998) and in which 832 instances of الواو /AlwAw/ are found. Standard metrics of Precision (Prec), Recall (Rec), and the F-measure,  $F_B$ , on the test set are utilized. We employ ten-fold cross-validation to ensure that any statistics obtained from our data are not biased. We have performed it manually.

Improving morphological segmentation has reduced error rate in POS tagging by approximately 7%. Examining errors in our output, we have found that they are due to the fact that Al-mawrid lexicon does not include all word's derivatives. For example, it does not contain the broken plural وزراء /wzrA}/ ('ministers'), although it includes the single form وزير /wzyr/ ('minister').

	Prec	Rec	$F_B$
Diab's Tokenize	83.5%	100%	91%
Our hypothesis	97.4%	100%	98.7%
Diab's POS Ta	87.2 %	100%	93.2%
Our hypothesis	93.6%	100%	96.7%

Table 1: Results of our hypothesis compared against Diab's

## 5 Conclusion and Future Directions

<sup>6</sup> <http://www.alasmaa.com>

<sup>7</sup> <http://geography.about.com/od/countryinformation/a/capital.htm>

<sup>8</sup> <http://www.atasoft.com>

In this paper, we introduce a preprocessing procedure that would help improve the processing of Arabic. It focuses on the identification of الواو /AlwAw/ through a morphological segmentation of this clitic. Our hypothesis is that every واو /wAw/ is a coordinating conjunction unless it is part of a word that is found in a dictionary of words or of proper names. For future work, we suggest applying this hypothesis to other clitics, such as other coordinating conjunctions, prepositions, pronouns, etc. Moreover, a comparison with other morphological analyzers developed for Arabic can be provided.

## 6 References

- 1 Anbar, T. **Current Trends in Processing Arabic Morphology**. In the *Proceedings of the Sixth Conference of Language Engineering*, pp.1-15, Cairo, December 2006.
- 2 Buckwalter, T. **Arabic Morphology Analysis**. <http://www.qamus.org/morphology.html>, 2002.
- 3 Diab, M., Hacioglu, K., Jurafsky, D. **Automatic Tagging of Arabic Text: From Raw text to Base Phrase Chunks**. In the *Proceedings of Human Language Technology/North American chapter of the Association for Computational Linguistics*, pp. 1-4, Boston, May 2004.
- 4 Habash, N., Rambow, O. **Arabic Tokenization, Morphological Analysis, and Part-of-Speech tagging in One Fell Swoop**. In *Proceedings of the 43<sup>rd</sup> Annual Meeting of the Association for Computational Linguistics*, pp. 573-580, Ann Arbor, June 2005.
- 5 Lee, Y., Papineni, K., Roukos, S., Emam, O., Hassan, H. **Language Model Based Arabic Word Segmentation**. In the *proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 399-406, Ann Arbor, June 2003.

Arabic Alphabet	Transliteration	Arabic Alphabet	Transliteration
أ	>	ش	\$
إ	<	م	m
ا	A	ك	k
ب	b	ن	n
ت	t	ف	f
ث	t	ق	q
ج	j	ه	h
ح	H	ة	p
خ	x	و	w
د	d	ل	l
ذ	*	ي	y
ر	r	ى	Y

ز ك ل م ن هـ و ط ظ ع غ ف	z s d t z e g s	ق ح ع ء أ إ أ أ أ	& } ' u a  i  ~
---	--------------------------------------	---	---

Appendix 1: Buckwalter Arabic Transliteration

# Entity Retrieval

Sisay Fissaha Adafre  
School of Computing, DCU  
Dublin 9, Ireland  
sadafre@computing.dcu.ie

Maarten de Rijke and Erik Tjong Kim Sang  
ISLA, University of Amsterdam  
Kruislaan 403, 1098SJ Amsterdam, The Netherlands  
mdr,erikt@science.uva.nl

## Abstract

Generalizing recent attention to retrieving entities and not just documents, we introduce two entity retrieval tasks: list completion and entity ranking. For each task, we propose and evaluate several algorithms. One of the core challenges is to overcome the very limited amount of information that serves as input—to address this challenge we explore different representations of list descriptions and/or example entities, where entities are represented not just by a textual description but also by the description of related entities. For evaluation purposes we make use of the lists and categories available in Wikipedia. Experimental results show that cluster-based contexts improve retrieval results for both tasks.

## Keywords

Entity retrieval, Wikipedia, language modeling

## 1 Introduction

Both commercial systems and the information retrieval community are displaying an increasing interest in not just returning web pages or other documents in response to a user’s query but “objects,” “entities” or their properties. E.g., various web search engines recognize specific types of entity (such as books, CDs, restaurants), and list these separately from the standard document-oriented hit list. Enterprise search provides another example [5], as has also been recognized within the TREC Enterprise track. In its 2005 and 2006 editions, the track featured an expert finding task [6] where systems return a list of entities (people’s names) who are knowledgeable about a certain topic (e.g., “web standards”).

This emerging area of *entity retrieval* differs from traditional document retrieval in a number of ways. Entities are not represented directly (as retrievable units such as documents), and we need to identify them “indirectly” through occurrences in documents. Entity retrieval systems may initially retrieve documents (pertaining to a given topic or entity) but they must then extract and process these documents in order to return a ranked list of entities [20]. In order to understand the issues at hand, we propose two entity retrieval tasks (building on a proposal launched in the run-up to INEX 2006 [7] and scheduled to be implemented at INEX 2007): *list completion* and *entity ranking*.

The *list completion* task is defined as: given a topic text and a number of examples, the system has to produce further examples. I.e., given a topic description, a set of entities  $S$  and a number of example entities  $e_1, \dots, e_n$  in  $S$  that fit the description, return “more examples like  $e_1, \dots, e_n$ ” from  $S$  that fit the description. E.g., given the short description *tennis players* and two example entities such as *Kim Clijsters* and *Martina Hingis*, entities such as tennis tournaments or coaches are not relevant. Instead, the expected set should include only individuals who are or have been professional tennis players. In the *entity ranking* task, a system has to return entities that satisfy a topic described in natural language text. I.e., given a set of entities  $S$  and a topic statement  $t$ , return elements of  $S$  that satisfy  $t$ . For example, let  $S$  denote a set of Dutch people; then “Dutch actors,” “Dutch politicians,” “Dutch artists,” etc., are some of the typical topic statements  $t$  that we envisage for this task.

The main research questions we address concern the ways in which we represent entities and in which we match topics and entities. As we will see, providing a sufficiently rich description of both topics and entities to be able to rank entities in an effective manner, is one of the main challenges. We address this challenge by using several contextual models.

For evaluation purposes we make use of Wikipedia, the online encyclopedia. The decision for using Wikipedia for this task is based on practical and theoretical considerations. Wikipedia contains a large set of lists that can be used for generating the necessary test data, and also assessing the outputs of our methods. Also, with its rich structure Wikipedia offers an interesting experimental setting where we can experiment with different features, both content-based and structural. Finally, by using Wikipedia’s lists, we can avoid the information extraction task of *identifying entities* in documents and focus on the retrieval task itself, instead. Below, we will only consider entities available in Wikipedia, and we will identify each entity with its Wikipedia article.<sup>1</sup>

The remainder of the paper is organized as follows. First, we provide background material and related work on working with Wikipedia, list questions, and contextual models. After that we turn to the list completion task, proposing and evaluating a number of algorithms. We then do the same for the entity ranking task before concluding the paper.

<sup>1</sup> We used the XML version of the English Wikipedia corpus made available by Denoyer and Gallinari [8]. It contains 659,388 articles, and has annotations for common structural elements such as article title, sections, paragraphs, sentences, and hyperlinks.



## 2 Background

**Mining/Retrieval against Wikipedia** Wikipedia has attracted interest from researchers in disciplines ranging from collaborative content development to language technology, addressing aspects such as information quality, users motivation, collaboration pattern, network structures, e.g., [25]. Several publications describe the use of Wikipedia as a resource for question answering and other types of IR systems; see e.g., [1, 10, 17]. Wikipedia has been used for computing word semantic relatedness, named-entity disambiguation, text classification, and as a document collection in various retrieval and knowledge representation tasks, e.g., [11].

**Entity Retrieval** List queries are a common types of web queries [22]. The TREC Question Answering track has recognized the importance of list questions [23]; there, systems have to return two or more instances of the class of entities that match the description in the list question. List questions are often treated as (repeated) factoids, but special strategies are called for as answers may need to be collected from multiple documents [4].

Recognizing the importance of list queries, Google Sets allows users to enter some instances of a concept and retrieve others that closely match the examples provided [13]. Ghahramani and Heller [12] developed an algorithm for completing a list based on examples using machine learning techniques. A proposed INEX entity retrieval task, with several tasks will likely be run during 2007 [7].

Our entity retrieval tasks are related to ontological relation extraction [14], where a combination of large corpora with simple manually created patterns are often used. Wikipedia, as a corpus, is relatively small, with much of the information being presented in a concise and non-redundant manner. Therefore, pattern-based methods may have limited coverage for the entity retrieval tasks that we consider.

**Document expansion and contextual IR** Enriching the document representation forms an integral part of the approach we propose in this paper. Though, in the past, application of document expansion techniques, particularly document clustering, has shown mixed results in document retrieval settings, recent studies within the language modelling framework provide new supporting evidence of the advantages of using document clusters [19]. Due to the nature of the tasks defined in this paper, the cluster hypothesis which states that “closely associated documents tend to be relevant to the same request” [16] provides for an intuitive starting point in designing our methods. Specifically, for each entity (or article) a precomputed cluster will be used to supply it with contextual information, much in the spirit of the work done by Azopardi [2] and Liu and Croft [19].

## 3 Task 1: List Completion

The main challenge of the list completion task is that the topic statement, example entity descriptions, and, more generally, entity descriptions in Wikipedia, tend

to be very short. Therefore, a straightforward retrieval baseline may suffer from poor recall. Hence, in our modeling we will address several ways of representing the topic statement and example entities.

We model the list completion task as follows: *what is the probability of a candidate  $e$  belonging to the list defined by the topic statement  $t$  and example entities  $e_1, \dots, e_n$ ?* We determine  $p(e|t, e_1, \dots, e_n)$  and rank entities according to this probability. To estimate  $p(e|t, e_1, \dots, e_n)$ , we proceed in two steps: (1) select candidate entities, and (2) rank candidate entities. More formally,

$$p(e|t, e_1, \dots, e_n) \propto \chi_C \cdot \text{rank}(e; t, e_1, \dots, e_n),$$

where  $\chi_C$  is a characteristic function for a set of selected candidate entities  $C$  and  $\text{rank}(\cdot)$  is a ranking function. Below, we consider alternative definitions of the function  $\chi_C$  and we describe two ranking functions. First, though, we define so-called entity neighborhoods that will be used in the candidate selection phase: to each individual entity  $e$  they associate additional entities based on  $e$ 's context, both in terms of link structure and contents.

### 3.1 Entity Neighborhoods

In the context of a hypertext documents, identification of a cluster typically involves searching for graph structures, where co-citations and bibliographic couplings provide importance features. Fissaha Adafre and de Rijke [9] describe a Wikipedia specific clustering method called *LTRank*. Their clustering method primarily uses the co-citation counts. We provide a slight extension that exploits the link structure (both incoming and outgoing links), article structure, and content. In Wikipedia, the leading few paragraphs contain essential information about the entity being described in the articles serving as summary of the content of the article; we use the first five sentences of the Wikipedia article as a representation of the content of the article. Our extension of the *LTRank* method for finding the neighborhood  $\text{neighborhood}(e)$  of an entity  $e$  is summarized in Figure 1. With this definition we can return to the first phase in our approach: *candidate entity selection*.

### 3.2 Candidate Entity Selection

To perform the candidate entity selection step, we use a two part representation of entities (Wikipedia articles). Each entity  $e$  is represented using (1) the textual content of the corresponding article  $a_e$ , and (2) the list of all entities in the set of  $\text{neighborhood}(e)$  defined above. We propose four candidate entity selection methods, that exploit this representation in different ways.

**B-1. Baseline: Retrieval** Here we rank entities by the similarity of their content part to a query consisting of the topic statement  $t$  and the titles  $t_{e_1}, \dots, t_{e_n}$  of the example entities. We used a simple vector space retrieval model for computing the similarity. The top  $n$  retrieved documents constitute the baseline candidate set  $C_1$ .

- Given a Wikipedia article  $a_e$  of an entity  $e$ , collect the titles of pages with links to or from  $a_e$ , as well as the words in the first five sentences of  $a_e$ . Let  $long(a_e)$  be the resulting bag of terms; this is the *long* representation of  $a_e$ .
- Given a Wikipedia article  $a_e$ , rank all articles w.r.t. their content similarity to  $long(a_e)$ ; we use a simple vector space model for the ranking. This produces a ranked list  $L_{a_e} = a_{e_1}, \dots, a_{e_n}, \dots$ .
- Given a Wikipedia article  $a_e$ , consider the titles  $t_1, \dots, t_k$  of the top  $k$  articles in the list  $L_{a_e}$ . Represent  $a_e$  as the bag of terms  $short(a_e) = \{t_1, \dots, t_k\}$ ; we call this the *short* representation of  $a_e$ .
- For each Wikipedia article  $a_e$ , rank the short representations of other Wikipedia articles w.r.t. their content similarity to  $short(a_e)$ ; again, we use a simple vector space model for the ranking. This produces a ranked list  $L'_{a_e}$ . The  $neighborhood(e)$  is defined to be the set of top  $l$  articles in  $L'_{a_e}$  whose similarity score is above some threshold  $\alpha$ .

**Fig. 1:** An extension of LTRank [9]. Our extension is in the first step, where we add outgoing links and the first 5 sentences of  $a_e$ . For the experiments in this paper, we took  $k = 10$ ,  $l = 100$ , and  $\alpha = 0.3$ .

**B-2. Neighborhood search** Our second candidate selection method matches the titles of the example entities against the neighborhoods of Wikipedia articles.

$$C_2 = \{e \mid \bigvee_i (e_i \in neighborhood(e))\}$$

**B-3. Neighborhood and Topic statement search** Here we take the union of the entities retrieved using the topic statement, and method B-2 described above. First, we rank entities by the similarity of their content part to a query which corresponds to the topic statement  $t$ . Here again, we used a simple vector space similarity measure to compute the similarity. We take the top  $k$  entities ( $k = 200$  in this paper) which constitute the first set,  $C_{3.1}$ . We then take all entities that contain at least one example entity in their neighborhood as with B-2, i.e.,

$$C_{3.2} = \{e \mid \bigvee_i (e_i \in neighborhood(e))\}.$$

The final candidate set is simply the union of these two sets, i.e.,  $C_3 = C_{3.1} \cup C_{3.2}$ .

**B-4. Neighborhood and Definition search** This method is similar to the method B-3. But instead of taking the topic statement  $t$  as a query for ranking entities (in the set  $C_{3.1}$  above), we take the definitions of the example entities  $e_1, \dots, e_n$ , where the first sentence of the Wikipedia article  $a_e$  of an entity  $e$  to be its definition; stopwords are removed.

### 3.3 Candidate Entity Ranking

We compare two methods that make use of the content of articles for ranking the entities generated by the previous step. Particularly, we apply the following two methods: Bayesian inference [12] and relevance-based language models [18]. Both methods provide a

mechanism for building a model of the concept represented by the example set. These two algorithms are developed for a task which closely resembles our task definitions, i.e., given a limited set of examples, find other instances of the concept represented by the examples. In the next paragraphs, we briefly discuss these methods.

**C-1. Bayesian Inference** Ghahramani and Heller [12] addressed the entity ranking task in the framework of Bayesian Inference. Given  $n$  example entities,  $e_1, \dots, e_n$ , and candidate entity  $e$ , the ranking algorithm is given by

$$score(e) = \frac{P(e, e_1, \dots, e_n)}{P(e) P(e_1, \dots, e_n)}. \quad (1)$$

To compute Eq. 1, a parameterized density function is posited. We list all terms  $t_{e_{1,1}}, \dots, t_{e_{1,k_1}}, \dots, t_{e_{n,k_n}}$  occurring in the example entities. Then, each candidate entity  $e$  is represented as a binary vector where vector element  $e_{i,j}$  corresponds to the  $j$ -th term from article  $a_{e_i}$  of the  $i$ -th example instance and assumes 1 if  $t_{e_{i,j}}$  appears in the article for the entity  $e$  and 0 otherwise. It is assumed that the terms  $e_{i,j}$  are independent and have a Bernoulli distribution  $\theta_j$  with parameters  $\alpha_j$  and  $\beta_j$ ; see [12]. In sum, Eq. 1 is rewritten to:

$$score(e) = c + \sum_{j=1}^N q_j e_{.,j},$$

where the summation ranges over the binary vector representation of  $e$ , and

$$c = \sum_j (\log(\alpha_j + \beta_j) - \log(\alpha_j + \beta_j + n) + \log(\beta_j + n - \sum_{i=1}^n e_{i,j}) - \log(\beta_j)),$$

while

$$q_j = \log(\alpha_j + \sum_{i=1}^n e_{i,j}) - \log(\alpha_j) + \log(\beta_j) - \log(\beta_j + n - \sum_{i=1}^n e_{i,j})$$

For given values of  $\alpha_j$  and  $\beta_j$ , the quantity  $q_j$  assigns more weights to terms that occur in most of the example entities. Therefore, a candidate instance  $e_i$  will be ranked high if it contains many terms from the example instances and the  $e_{i,j}$  receive high weights from the  $q_j$ s.

**C-2. Relevance Models** Lavrenko and Croft [18] proposed so-called relevance-based language models for information retrieval. Given  $n$  example entities,  $e_1, \dots, e_n$ , and the candidate  $e$  from the candidate set  $C$ , the ranking function is given by the KL-divergence between two relevance models:

$$score(e) = KL(P_{e_1, \dots, e_n} \| P_e),$$

where  $P_{e_1, \dots, e_n}$  is the relevance model of the example entities, and  $P_e$  is the language model induced from the Wikipedia article for entity  $e$ . The relevance models are given by

$$\begin{aligned} P(w|e_1, \dots, e_n) &= \sum_{e \in W} P(w|e) \cdot P(e|e_1, \dots, e_n) \\ P(e|e_1, \dots, e_n) &= \begin{cases} 1/n & \text{if } e \in \{e_1, \dots, e_n\} \\ 0 & \text{otherwise} \end{cases} \\ P(w|e) &= \frac{\#(w, e)}{|e|}, \end{aligned}$$

where  $W$  is the collection (Wikipedia), and  $w$  represents the terms in the Wikipedia article for entity  $e$ . The KL divergence will be small for entities that more closely resemble the example entities in terms of their descriptions.

**Summary** Both of the ranking methods outlined above return a ranked list of candidate entities. We normalize the scores using

$$\text{score}_{\text{norm}} = \frac{\text{score}_{\text{MAX}} - \text{score}}{\text{score}_{\text{MAX}} - \text{score}_{\text{MIN}}},$$

and take those candidate entities for which the normalized score lie above empirically determined threshold ( $\text{score}_{\text{norm}} > 0.5$ ). The resulting set will be assessed.

### 3.4 Experimental Set-up

The performance of our approach to the list completion task depends on the performance of the two sub-components: candidate selection and candidate ranking. We conduct two sets of experiments, one to determine the effectiveness of the candidate selection methods, and a second to determine the effectiveness of the overall approach. We are especially interested in the contribution of using the neighborhoods of entities.

The Wikipedia lists serve as our gold standard. We selected a random sample of 30 lists (the topics) from Wikipedia. We chose relatively homogeneous and complete lists, and excluded those that represent a mixture of several concepts. We take 10 example sets for each topic. Each example set consists of a random sample of entities from the Wikipedia list for the topic. We run our system using each of these 10 example sets as a separate input. The final score for each topic is then the average score over the ten separate runs. In the experiments in this section, we assume that each example set contains two example instances. This choice is mainly motivated by our assumption that users are unlikely to supply many examples.

The results are assessed based on the following scores:  $P@20$  (number of correct entities that are among the top 20 in the ranked list), *precision* (P; number of correct entities that are in the ranked list, divided by size of the ranked list), *recall* (R; number of correct entities that are in the ranked list, divided by the number of entities in the Wikipedia list) and *F-scores* (F; harmonic mean of the recall and precision values).

In order to test if the differences among the methods measured in terms of F-scores is statistically significant, we applied the two-tailed Wilcoxon matched pair signed-ranks test (for  $\alpha = 0.05$  and  $\alpha = 0.005$ ).

### 3.5 Results

First, we assess the methods we used for candidate selection. Following this, we present the evaluation results of the overall system.

**Candidate selection** Table 1 shows results of the evaluation of the candidate selection module. The figures are averages over all topics and all sets of example entities. The values are relatively low. Retrieving additional candidates using terms derived either from the

Selection method	P	R	Result set size
B-1 (Top $k = 500$ )	0.042	0.235	500
B-2	0.142	0.236	206
B-3	0.089	0.311	386
B-4	0.093	0.280	367

**Table 1:** Performance on the candidate selection sub-task.

Candidate selection	Candidate ranking	P	R	F	P@20
		B-1	C-1	0.100	0.068
	C-2	0.203	0.046	0.060	0.144
B-2	C-1	0.172	0.163	0.136	0.205
	C-2	<b>0.227</b>	0.142	0.137	0.231
B-3	C-1	0.121	<b>0.236</b>	0.136	0.196
	C-2	0.188	0.210	0.151	<b>0.249</b>
B-4	C-1	0.140	0.202	0.142	0.201
	C-2	0.204	0.209	<b>0.158</b>	0.248

**Table 2:** Performance on the entire list completion task. Best scores per metric in boldface.

definition of the entities or topic statement improves recall to some extent. The recall values for method B-3 are the best. This suggests that the terms in the topic are more accurate than the terms automatically derived from the definitions.

The neighborhood-based methods achieve better recall values while returning fewer number of candidates (cf. the last column of Table 1).

**Overall results** Table 2 shows the scores resulting from applying the two ranking methods C-1 and C-2 on the output of different candidate selection methods. The first column of Table 2 shows the different candidate selection methods; the second column shows the ranking methods.

The neighborhood-based combinations outperform the baselines at the  $\alpha = 0.005$  significance level (when considering F-scores). The combination of C-2 (*Relevance model*) with B-4 (*Neighborhood plus Definition Terms*) input outperforms both the B-2 + C-1 and B-2 + C-2 combinations at the  $\alpha = 0.05$  significance level. Generally, the C-2 ranking method has a slight edge over the C-1 method on most inputs. Furthermore, retrieving additional candidates using either the topic statement or the definition terms improves results, especially when used in combination with the C-2 ranking method.

### 3.6 Error Analysis

A closer look at the results for the individual topics reveals a broad range of recall values. The recall values for the topics *North European Jews*, *Chinese Americans*, *French people*, and *Miami University alumni* are very low. On the other hand, the topics *Indian Test cricketers*, *Revision control software*, *Places in Norfolk*, and *Cities in Kentucky* receive high recall scores. For the neighborhood-based methods, there is some correlation between the composition of the neighborhoods corresponding to the example entities and the results obtained. For example, the neighborhoods corresponding to the example entities for the topic *Indian Test cricketers* contain Indian cricket players. On the

other hand, the neighborhoods corresponding to the example entities for the topic *Chinese Americans* contain individuals from the USA, most of whom are not Chinese Americans, and have very little in common except for the features identified by the topic titles, which are too specific.

## 4 Task 2: Entity Ranking

The goal of the entity ranking task is to retrieve a subset of a given set of entities that satisfy a topic statement. More formally, let  $E$ , a set of entities, be given. We rank entities according to the probability  $p(t|e)$ , where  $e$  ranges over elements of  $E$  and  $t$  is a topic statement. We present different methods of estimating  $p(t|e)$ . These methods are organized along two dimensions; along one we consider richer representations of the topic statement  $t$ , along the other we consider different ways of representing entities.

### 4.1 Topic Representations

We compare two types of topic representation which we describe below.

**F-1. Baseline** As our baseline, we only remove stopwords from the topic statements. No further processing is done on the topic statement.

**F-2. Topic expansion** In addition to removing stopwords, we enrich the topic by incorporating additional terms based on the method proposed in [21]. We assume the top  $n$  ( $n = 5$ ) articles returned using the *Collection smoothing method* (see below) with  $\lambda = 0.9$  as being relevant. Extra terms are added based on the log ratio of their likelihood in terms of the model for relevant articles to their likelihood in terms of the model for whole entity set.

### 4.2 Entity Representations

We now introduce several ways of representing entities, all in terms of two or three part mixture models. We start with our baseline approach.

**G-1. Baseline** As explained in the introduction, the entities we consider are titles of Wikipedia articles. Hence, the simplest representation of an entity  $e$  is its associated Wikipedia article  $a_e$ . As usual, the topic  $t$  is represented by a set of terms:  $t = \{t_1, \dots, t_k\}$ ; we write  $c(t_i, a_e)$  to indicate the number of times  $t_i$  occurs in  $a_e$ . Each topic term is assumed to be generated independently, and so the topic likelihood is obtained by taking the product across all the terms in the topic:

$$p(t|e) = \prod_{t_i \in t} p(t_i|a_e)^{c(t_i, t)}.$$

In our baseline approach, we estimate  $p(t_i|a_e)$  by taking the maximum likelihood estimate of  $t_i$  in  $a_e$ :

$$p_{baseline}(t_i|a_e) = p_{MLE}(t_i|a_e) = \frac{c(t_i, e)}{|a_e|},$$

where  $|a_e|$  the total number of term occurrences in  $a_e$ .

**G-2. Collection smoothing** Since  $p_{MLE}(t_i|a_e)$  may contain zero probabilities it is standard to employ smoothing [24]. Therefore, we smooth the maximum likelihood estimate, i.e.,  $p_{MLE}(t_i|e)$ , against a general model estimated from the whole Wikipedia collection as follows:

$$p(t_i|a_e) = \lambda \cdot p_{MLE}(t_i|a_e) + (1 - \lambda) \cdot p_{MLE}(t_i|W), \quad (2)$$

where the latter is the maximum likelihood estimate of  $t_i$  in  $W$ , the entire Wikipedia corpus.

**G-3. Context models 1: A generic approach** In this paragraph and the next, we introduce two context models, both give rise to three part mixture models, involving the entity, the context, and the collection. The intuition behind these models is that a more focused context should be more accurate in capturing the topic of the entity, thus producing a more meaningful representation of the entity than the entire collection. The first context model we consider is generic, and does not exploit special features of the Wikipedia corpus. Specifically, we use probabilistic latent semantic analysis (PLSA, [15]) to induce a context for every entity  $e$ . Given an entity  $e$ , a latent class  $z$  is selected with probability  $p(z|e)$ , and given the class  $z$ , terms  $t_i$  are generated with probability  $p(t_i|z)$ . Then the following context model is obtained:

$$p_{PLSA}(t_i|e) = \sum_{z \in Z} p(t_i|z) \cdot p(z|e), \quad (3)$$

where  $Z$  is the set of latent variables considered (in our experimental evaluation we fix  $|Z| = 20$ ). The probabilities  $p(t_i|z)$  and  $p(z|e)$  are estimated using the EM algorithm as described in [15]. Putting Eq. 3 together with the smoothed model (Eq. 2), we obtain the following:

$$p_{TOPIC}(t_i|e) = \lambda_1 \cdot p_{MLE}(t_i|e) + \lambda_2 \cdot p_{PLSA}(t_i|e) + (1 - \lambda_1 - \lambda_2) \cdot p_{MLE}(t_i|W), \quad (4)$$

where  $\lambda_1, \lambda_2 \in [0, 1]$  and  $\lambda_1 + \lambda_2 \leq 1$ .

**G-4. Context models 2: A Wikipedia-specific approach** The second context model we consider in this paper exploits specific features of the Wikipedia corpus. We use the method summarized in Figure 1 for estimating the Wikipedia specific context model. Specifically, given an entity  $e$ , consider the neighborhood of  $e$  as produced by the algorithm in Figure 1. Assume  $neighborhood(e) = d_1(e), \dots, d_k(e)$  for  $e$ . Then,

$$p_{WIKI}(t_i|e) = \lambda_1 \cdot p_{MLE}(t_i|e) + \lambda_2 \cdot p_{LTS}(t_i|d(e)_1, \dots, d(e)_k) + (1 - \lambda_1 - \lambda_2) \cdot p_{MLE}(t_i|W), \quad (5)$$

where, as before,  $\lambda_1, \lambda_2 \in [0, 1]$  and  $\lambda_1 + \lambda_2 \leq 1$ .  $p_{LTS}(t_i|d_1, \dots, d_k)$  is the context model, which gives the likelihood of the term  $t_i$  in the cluster consisting of the context documents,  $d_1, \dots, d_k$ .

### 4.3 Experimental Set-up

The experiments in this section are aimed at gaining insight into the contributions (for the *Entity Ranking* task) of the different topic and document representation methods introduced previously. We used

Document representation	Parameters	Topic representation			
		F-1		F-2	
		P@10	R-Prec	P@10	R-Prec
G-1	–	0.587	0.399	0.217	0.211
G-2	$\lambda = 0.9$	0.567	0.428	0.567	0.413
G-3	$\lambda_1 = 0.7, \lambda_2 = 0.2$	0.583	0.448	0.570	0.426
G-4	$\lambda_1 = 0.7, \lambda_2 = 0.2$	<b>0.623</b>	<b>0.476</b>	<b>0.580</b>	<b>0.464</b>

**Table 3:** *Entity ranking results: average values over all topics.*

Wikipedia’s hierarchical categories for generating the data for evaluating the methods. We selected a random sample of 30 Wikipedia lists, i.e., *main entity sets*. For each *main entity set*, we selected a subset of entities and the associated topic. Each of the alternative approaches presented in this section rank entities in the *main entity set*. The ranked list is assessed based on the following precision scores: R-Precision (the fraction of the number of correct entities for each topic that are among the top  $n$  entities returned, where  $n$  is the size of the sublist we are seeking), and p@10 (number of correct entities for each topic that are among the top 10 entities returned).

We applied the two-tailed Wilcoxon matched pair signed-ranks test to determine whether the differences among the methods as measured in terms of R-Precision scores are statistically significant ( $\alpha = 0.05$ ).

## 4.4 Results

Table 3 shows the result of the different runs. In the tables, the columns *Parameters*, *p@10* and *R-Prec* correspond to the parameter settings, precision at 10, and R-Precision. The parameter settings are the optimal mixing values for the given model. As the results show, the baseline method, which uses the maximum likelihood estimation without term expansion (F-1 + G-1), performs relatively well. However, term expansion hurts performance of the baseline method due to the MLE estimation (the extended topic tends to be assigned zero probability). All methods outperform the F-2 + G-1 combination. The ranking method that uses the *Wikipedia Specific Context model* (G-4) outperforms the *Collection-based context* and the MLE method at a significance level of  $\alpha = 0.05$ . G-4 performs better than G-3 at the significance level of  $\alpha = 0.1$ . Term expansion tends to hurt performance as can be seen from the general pattern in Table 3.

## 5 Discussion

### Entity retrieval vs information extraction

The tasks considered in this paper, i.e., *list completion* and *entity ranking*, share a common overall goal. They both aim at identifying entities that share certain characteristics. In this respect, they resemble tasks commonly addressed in Information Extraction (IE), such as *named entity recognition* and *relation extraction*. However, there are important distinctions between traditional IE and the entity retrieval tasks we consider. First, in typical IE scenarios, the entities are embedded in a text, and the aim is to extract or recognise

occurrences of these entities in the text. Systems commonly use surrounding contextual information, and redundancy information to recognise the entities in the text. The inputs to these systems are documents that may contain one or more occurrences of the target entities. In contrast, in the entity retrieval tasks that we consider, the entities are represented by documents which provide descriptive information about them—typically, there is a one-to-one relation between the entities and the documents. In our setting, then, we abstract away from the recognition phase so that we are able to zoom in on the retrieval task only—unlike, e.g., the expert finding scenarios currently being explored at TREC, that do require participating systems to create effective combinations of extraction and retrieval [3].

**One or two tasks?** Although the list completion and entity ranking tasks are similar at an abstract level, a closer look at the specific details reveals important differences which necessitated task-specific approaches. One aspect concerns the size of the input; for the list completion task, the inputs are example entities with/without topic statements, and the candidates are all Wikipedia entries. On the other hand, the inputs for the entity ranking task consist of the topic statements only, and the candidates are entities in a particular Wikipedia list, such as, e.g., the List of Countries, which is obviously much smaller and more homogeneous than the entire Wikipedia collection.

The result of the list completion task shows that traditional information retrieval methods significantly underperform for selecting initial candidates from all of Wikipedia. This affects the overall score of the method as subsequent processing makes use of the output of this step. On the other hand, preclustering of Wikipedia articles led to much better performance. The re-ranking methods showed comparable performance results, with the relevance feedback method having a slight edge over the Bayesian method.

In the entity ranking task, we compared different ways of enriching the topic statements and document representations. As to the former, we added more terms to the topic description, and in the latter, we applied document modeling techniques that capture natural groupings that may exist in the target list. The results show that automatic addition of terms using relevance feedback methods seems to hurt performance. Here again, our notion of neighbourhood seems to capture the natural groupings in the target list better than the topic modeling method we considered in this paper.

By comparing the absolute scores of the two tasks, it seems safe to conclude that the richer input used for the entity ranking task (working with a specific list rather than all of Wikipedia) leads to higher scores.

## 6 Conclusion

We described, and proposed solutions for, two types of entity retrieval tasks, *list completion* and *entity ranking*. We conducted two sets of experiments in order to assess the proposed methods, which focused on enriching the two key elements of the retrieval tasks, i.e.,

*Topic statements and Example entities.*

For the list completion task, the methods that used the titles of the example entities and the topic statements or definition terms performed better. All methods that used a context set consisting of related articles significantly outperformed a simple document-based retrieval baseline that does not use the related articles field.

For the entity ranking task, the method that used a context set of related articles also performed better than most of the alternatives we considered. Here, we used the related articles to provide contextual information for the entity description when computing the similarity between the topic statement and entity description. Our notion of related articles improves results when used both as a means of retrieving initial candidates and for providing contextual information during similarity computations.

Our results are limited in a number of ways. For example, entities are represented primarily by the combination of the content of their Wikipedia articles (as a bag of words) and a precomputed set of related articles. We need to explore other—rich—representations of the content, e.g., phrases or anchor text, and also other concepts of relatedness, e.g., the Wikipedia categories.

## Acknowledgments

This research was supported by the Netherlands Organisation for Scientific Research (NWO) under project numbers 017.001.190, 220-80-001, 264-70-050, 354-20-005, 600.065.120, 612-13-001, 612.000.-106, 612.066.302, 612.069.006, 640.001.501, 640.002.-501, and by the E.U. IST programme of the 6th FP for RTD under project MultiMATCH contract IST-033104.

## References

- [1] K. Ahn, J. Bos, J. R. Curran, D. Kor, M. Nissim, and B. Webber. Question answering with QED at TREC-2005. In *Proceedings of TREC 2005*, 2005.
- [2] L. Azzopardi. Incorporating context within the language modeling approach for ad hoc information retrieval. *SIGIR Forum*, 40(1):70–70, 2006.
- [3] K. Balog and M. de Rijke. Finding experts and their details in e-mail corpora. In *15th International World Wide Web Conference (WWW2006)*, 2006.
- [4] J. Chu-Carroll, K. Czuba, J. Prager, A. Ittycheriah, and S. Blair-Goldensohn. IBM's PIQUANT II in TREC 2004. In *Proceedings TREC 2004*, 2004.
- [5] N. Craswell, D. Hawking, A. M. Vercoustre, and P. Wilkins. P@noptic expert: Searching for experts not just for documents. In *Ausweb*, 2001.
- [6] N. Craswell, A. de Vries, and I. Soboroff. Overview of the TREC 2005 Enterprise Track. In *Proceedings of TREC 2005*, 2006.
- [7] A. de Vries and N. Craswell. XML entity ranking track, 2006. URL: <http://inex.is.informatik.uni-duisburg.de/2006/xmlSearch.html>.
- [8] L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 2006.
- [9] S. Fissaha Adafre and M. de Rijke. Discovering missing links in Wikipedia. In *Proceedings of LinkKDD-2005 Workshop*, 2005.
- [10] S. Fissaha Adafre and M. de Rijke. Estimating importance features for fact mining (with a case study in biography mining). In *RIAO 2007*, 2007.
- [11] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI '06*, 2006.
- [12] Z. Ghahramani and K. A. Heller. Bayesian sets. In *NIPS 2005*, 2005.
- [13] Google, 2006. GoogleSets. URL: <http://labs.google.com/sets>, accessed on 04-10-2006.
- [14] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING 1992*, pages 539–545, 1992.
- [15] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence, UAI'99*, 1999.
- [16] N. Jardine and C. J. van Rijsbergen. The use of hierarchic clustering in information retrieval. *Information Storage Retrieval*, 7(5):217–240, 1971.
- [17] V. Jijkoun and M. de Rijke. WiQA: Evaluating Multi-lingual Focused Access to Wikipedia. In *EVIA 2007*, 2007.
- [18] V. Lavrenko and W. B. Croft. Relevance models in information retrieval. In *Language Modeling for Information Retrieval*. Kluwer Academic Publishers, 2003.
- [19] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *SIGIR '04*, pages 186–193, 2004.
- [20] D. Petkova and W. B. Croft. Hierarchical language models for expert finding in enterprise corpora. In *Proceedings ICTAI 2006*, pages 599–608, 2006.
- [21] J. M. Ponte. Language models for relevance feedback. In *Advances in Information Retrieval*, pages 73–96. 2000.
- [22] D. E. Rose and D. Levinson. Understanding user goals in web search. In *WWW '04*, pages 13–19, 2004.
- [23] E. Voorhees. Overview of the TREC 2004 question answering track. In *Proceedings of TREC 2004*, 2005. NIST Special Publication: SP 500-261.
- [24] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01*, pages 334–342, 2001.
- [25] V. Zlatić, M. Božičević, H. Štefančić, and M. Domazet. Wikipedias: Collaborative web-based encyclopedias as complex networks. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 74(1), 2006.

# Some Reflections on the Task of Content Determination in the Context of Multi-Document Summarization of Evolving Events

Stergos D. Afantenos

Laboratoire d'Informatique Fondamentale de Marseille  
Centre National de la Recherche Scientifique (LIF - CNRS - UMR 6166)  
Université de la Méditerranée, Faculté des Sciences de Luminy  
163, Avenue de Luminy - Case 901, 13288 Marseille Cédex 9 - France  
stergos.afantenos@lif.univ-mrs.fr

## Abstract

Despite its importance, the task of summarizing evolving events has received small attention by researchers in the field of Multi-document Summarization. In a previous paper [5] we have presented a methodology for the automatic summarization of documents, emitted by multiple sources, which describe the evolution of an event. At the heart of this methodology lies the identification of similarities and differences between the various documents, in two axes: the synchronic and the diachronic. This is achieved by the introduction of the notion of *Synchronic and Diachronic Relations*. Those relations connect the messages that are found in the documents, resulting thus in a graph which we call *grid*. Although the creation of the grid completes the Document Planning phase of a typical NLG architecture, it can be the case that the number of messages contained in a grid is very large, exceeding thus the required compression rate. In this paper we provide some initial thoughts on a probabilistic model which can be applied at the Content Determination stage, and which tries to alleviate this problem.

**Keywords :** *summarization of evolving events, multi-document summarization, natural language generation*

## 1 Introduction

It wouldn't be an exaggeration to claim that human beings live engulfed in an environment full of information. Information which, metaphorically speaking, vie with each other in order to gain our attention, to gain an almost exclusive control of the precious resources which are our brains. This is most evident in the medium of Internet in which so many people are spending nowadays a considerable amount of their time. Information in this medium is constantly flowing in front of our screens, making the assimilation of such a plethora no longer feasible. In such an environment, information which is presented in brief and concise manner—*i.e.* summarized information—stand more chances of retaining our attention, in relation to information presented in long and fragmented pieces of text. We can claim then, with a certain degree of certainty, that the task of automatic text summarization can prove to be very useful.

To provide a concrete example, we can imagine the case of a person who would like to keep track of the information related to an event as the event is evolving through time.

What will usually happen in such cases is that, firstly, there will be more than one sources which will provide an account of the event, and secondly, most of the sources will provide more than one descriptions, in the sense that they will most probably follow the evolution of the event and provide updates as the event evolves through time. This can easily result in hundreds or even thousands of related articles which will describe the evolution of the same event, rendering it thus almost impossible for the interested person to read through its evolution comparing along the way the points in which the sources agree, disagree or present the information from a different point of view. A simple visit to a news aggregator, such as for example Google News,<sup>1</sup> can make this point very clear.

As we have hinted before, a solution to this problem might be the automatic creation of summaries. In this paper we will present a methodology which aims at exactly that, *i.e.* the automatic creation of text summaries from documents emitted by multiple sources which describe the evolution of a particular event. In Section 2 we will briefly present this methodology, at the heart of which lies the notion of *Synchronic and Diachronic Relations* (SDRs) whose aim is the identification of the similarities and differences that exist between the documents in the synchronic and diachronic axes. The end result of this methodology is a graph whose vertices are the SDRs and whose nodes are some structures which we call *messages*. The creation of this graph can be considered as completing—as we have previously argued [5]—the *Document Planning* phase of a typical architecture of a Natural Language Generation (NLG) system [20]. Nevertheless, this graph can prove to be very large and thus the resulting summary can easily exceed the desired compression rate. In Section 4 we will present a brief sketch of a probabilistic model for the selection of the appropriate information—*i.e.* messages—to be included in the final summary, so that the desired compression rate will not be violated. In other words, we will propose a model for the *Content Determination* stage of the Document Planning phase. This model will be based on certain remarks concerning the way with which information overlap between multiple documents which we present in Section 3. The conclusions of this paper are presented in Section 5.

<sup>1</sup> <http://news.google.com/>



## 2 A Methodology for Summarizing Evolving Events<sup>2</sup>

At the heart of Multi-document Summarization (MDS) lies the process of identifying the similarities and differences that exist between the input documents. Although this holds true for the general case of Multi-document Summarization, for the case of summarizing *evolving events* the identification of the similarities and differences should be distinguished, as we have previously argued [1, 2, 4, 5, 6] between two axes: the *synchronic* and the *diachronic* axes. In the synchronic axis we are mostly concerned with the degree of agreement or disagreement that the various sources exhibit, for the same time frame, whilst in the diachronic axis we are concerned with the actual evolution of an event, as this evolution is being described by one source.

The initial inspiration for the SDRs was provided by the *Rhetorical Structure Theory* (RST) of Mann & Thompson [15, 16]. Rhetorical Structure Theory—which was initially developed in the context of “computational text generation”<sup>3</sup> [15, 16, 22]—is trying to connect several *units of analysis* with relations that are semantic in nature and are supposed to capture the intentions of the author. As “units of analysis” today are used, almost ubiquitously, the clauses of the text. In our case, as units of analysis for the SDRs we are using some structures which we call *messages*, inspired from the research in the NLG field. Each message is composed of two parts: its *type* and a list of *arguments* which take their values from an *ontology* for the specific domain. In other words, a message can be defined as follows:

```
message_type ( arg1, ... , argn )
  where argi ∈ Domain Ontology
```

The message type represents the type of the action that is involved in an event, whilst the arguments represent the main entities that are involved in this action. Additionally, each message is accompanied by information on the source which emitted this message, as well as its publication and referring time.

Concerning the SDRs, in order to formally define a relation the following four fields ought to be defined (see also [5]):

1. The relation’s type (*i.e.* Synchronic or Diachronic).
2. The relation’s name.
3. The set of pairs of message types that are involved in the relation.
4. The constraints that the corresponding arguments of each of the pairs of message types should have. Those constraints are expressed using the notation of first order logic.

The name of the relation carries *semantic* information which, along with the messages that are connected with the relation, are later being exploited by the NLG component (see [5]) in order to produce the final summary.

<sup>2</sup> Due to space limitations this section contains a very brief introduction to a methodology for the creation of summaries from evolving events that we have earlier presented [5]. The interested reader is encouraged to consult [1, 2, 4, 5, 6] for more information.

<sup>3</sup> Also referred to as Natural Language Generation (NLG).

The methodology we propose consists of two main phases, the *topic analysis phase* and the *implementation phase*. The topic analysis phase is composed of four steps, which include the creation of the ontology for the topic and the providing of the specifications for the messages and the SDRs. The final step of this phase, which in fact serves as a bridge step with the implementation phase, includes the annotation of the corpora belonging to the topic under examination that have to be collected as a preliminary step during this phase. The annotated corpora will serve a dual role: the first is the training of the various Machine Learning algorithms used during the next phase and the second is for evaluation purposes. The implementation phase involves the computational extraction of the messages and the SDRs that connect them in order to create a directed acyclic graph (DAG) which we call *grid*. The architecture of the summarization system is shown in Figure 1.

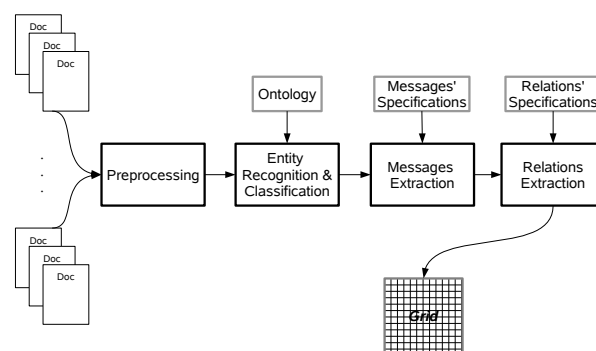


Fig. 1: The summarization system.

We applied our methodology in two different case studies. The first case study concerned the description of football matches, a topic which evolved linearly and exhibited synchronous emission of reports, while the second case study concerned the description of terroristic incidents with hostages, a topic which evolved non-linearly and exhibited asynchronous emission of reports.<sup>4</sup> The preprocessing stage involved tokenization and sentence splitting in the first case study and tokenization, sentence splitting and part-of-speech tagging in the second case study. For the task of the *entities recognition and classification* in the first case the use of simple gazetteer lists proved to be sufficient. In the second case study this was not the case and thus we opted for using what we called a *cascade of classifiers* which contained three levels. At the first level we used a binary classifier which determines whether a textual element in the input text is an instance of an ontology concept or not. At the second level, the classifier takes the instances of the ontology concepts of the previous level and classifies them under the top-level ontology concepts (e.g. *Person*). Finally at the third level we had a specific classifier for each top-level ontology concept, which classifies the instances in their appropriate sub-concepts; for example, in the *Person* ontology concept the specialized classifier classifies the instances into *Offender*, *Hostage*, etc. For the third stage of the messages’ extraction we use in

<sup>4</sup> On the distinction between linearly/non-linearly events and synchronous/asynchronous emission of reports the interested reader is encouraged to consult [1, 4, 5, 6].

both case studies lexical and semantic features. As lexical features in the first case we used the words of the sentences (excluding low frequency words and stop-words) while in the second case study we used only the verbs and nouns of the sentences as lexical features. As semantic features in the first case study we used the number of the top-level ontology concepts that appear in the sentence, while in the second case study we enriched that with the appearance of certain trigger words in the sentence. Finally, the extraction of the SDRs is the most straightforward task, since the only thing that is needed is the translation of the relations' specifications into an appropriate algorithm which, once applied to the extracted messages, will provide the relations that connect the messages, effectively thus creating the grid. In Table 1 we present the statistics of the final messages and SDRs extraction stages for both case studies.<sup>5</sup>

	Case Study I	Case Study II
Messages	Pr : 91.12% Rc : 67.79% FM : 77.74%	Pr : 42.96% Rc : 35.91% FM : 39.12%
SDRs	Pr : 89.06% Rc : 39.18% FM : 54.42%	Pr : 30.66% Rc : 49.12% FM : 37.76%

**Table 1:** Precision, Recall and F-Measure for the extraction of the Messages and SDRs for both case studies.

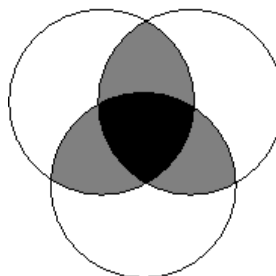
The creation of the grid can be considered as completing—as we have previously argued [5]—the *Document Planning* phase of a typical architecture of an NLG system [20]. Nevertheless, this graph can prove to be very large and thus the resulting summary can easily exceed the desired compression rate. In the following two sections we will present a brief sketch of a probabilistic model which can operate on the Content Determination stage of the Document Planning phase in order to select the appropriate content so that the compression rate of the summary will be respected.

### 3 The White, Grey, and Black Areas of MDS

Not too distant in time from the dawn of Artificial Intelligence in the early 1950's, the first seeds of automatic text summarization appeared with the seminal works of Luhn [12] and Edmundson [7]. Those early works, as well as the works on summarization that would follow in the next decades, were mostly concerned with the creation of summaries from single documents. Most of them were focusing on the verbatim extraction of important textual elements, usually sentences or paragraphs, from the input document in order to create the final summary. The methods used for the identification of the most salient sentences or paragraphs vary from a mixture of locational criteria with statistics [7, 12, 19] to statistical based graph creation methods [21] to RST based methods [17].

Multi-document Summarization would not be actively pursued by researchers up until the mid 1990's, since when

it is a quite active area of research.<sup>6</sup> The main difference that seems to exist between the summarization of a single document and the summarization of multiple (related) documents, seems to be the fact that the ensemble of the related documents, in most of the cases, creates *informational redundancy*, as well as what—for a lack of better term—we will call *informational isolation*. In the case of informational redundancy more than one document contain the same information, while in the case of informational isolation only one document contains a specific piece of information. This is graphically depicted in Figure 2, in which each circle represents the information that is contained in a different document. The black and grey areas of the figure represent the information redundancy that exists between the documents. More specifically, the black area represents information which is common to all of the documents, while the grey areas represent information which are common between some articles but not all of them. The white areas, on the other hand, represent what we have called the informational isolation of certain portions of texts, in the sense that the information contained therein is not found anywhere else in the collection of documents.



**Fig. 2:** Information redundancy and information isolation.

Of course, one could imagine many more ways in which the circles could be arranged. For example, a circle could be contained inside two other circles, which would imply that the corresponding document is informationally subsumed by the other two. More extreme cases can involve circles arranged in a way that only gray areas exist, which would imply that the documents of the collection are only very loosely related, or cases in which one or more circles are completely white, meaning that the documents which are represented by those circles are completely unrelated with the rest of the documents. Such cases though, one could argue, violate the premises of MDS which require a set of *related documents* that will be informationally condensed by the end of the process.

Despite those extreme cases, it is fair to assume that the configuration depicted in Figure 2 represents a fairly common situation in most of the MDS scenarios. Of course we have to bare in mind that in most of the cases we will not have just three documents to be summarized, but most possibly many more. This will have the consequence that the grey areas will not have a single shade of greyness but in-

<sup>5</sup> For more details, critique of those results and comparison with related work the interested reader is encouraged to consult [1, 5].

<sup>6</sup> For a general overview of summarization the interested reader is encouraged to consult [13]. Mani & Maybury [14] provide a wonderful collection of papers on summarization spanning most of the research sub-fields of this area. Afantenos *et al.* [3] provide an overview as well, focusing mostly on the summarization from medical documents. Finally, [8] contains an excellent account of the *cognitive processes* that are involved during the task of single document summarization by professionals, as well a brief overview of the field of summarization.

stead they will range from light grey to dark grey depending on the degree of information overlap that will exist between the various sources.

## 4 What Should Be Included in a Multi-Document Summary of Evolving Events?

Having made the above distinction between the different levels of information overlap, the question that arises at this point is which pieces of information should finally be included in the text that will summarize the multiple documents. The obvious answer to this question would be that such a summary should include the information that are contained in the input documents in decreasing order of their importance, until the length of the summary reaches the required compression rate of the total length of the input documents. In other words, a summary should contain the black areas of Figure 2, then the darker to the lighter grey areas, until the length of the summary reaches the required compression rate.

In mathematical terms this can be expressed as follows. If  $P(i)$  is the probability that a piece of information will be included in the final summary, then we can claim that:

$$P(i) = \frac{\sum_{k=1}^n d_{ki}}{n}$$

where  $n$  represents the total number of documents,  $d_k$  the  $k$ -th document, and:

$$d_{ki} = \begin{cases} 1 & \text{if } d_k \text{ contains information } i \\ 0 & \text{if } d_k \text{ does not contain information } i \end{cases}$$

Additionally, if  $c$  is the desirable compression rate, then the final summary  $S$  should confront to the following constraint:

$$\text{length}(S) \leq c \sum_{k=1}^n \text{length}(d_k)$$

### 4.1 Objections to the Proposed Model for the General Case of MDS

Now, the above model is really a simplistic one and a host of objections could be raised concerning its usefulness in the general case of MDS, something that we do acknowledge. One could for example claim that the information that will be contained in the black areas will tend to be trivial information, in the sense that they can be characterized as representing “common knowledge”. This objection can be balanced by two arguments. The first is that the authors of the original documents will most possibly not contain in their articles such common knowledge, unless it is necessary, in which case it might be a good idea to be included in a summary. The second argument is that if the summarization system uses knowledge representation methods—an ontology for example—then such trivial information will tend not to be included in this knowledge representation. Of course, if the system uses purely statistical methods, then the last argument does not hold.

The second objection concerns the white or light grey areas. In the proposed model such areas will have a small

probability of being included in the final summary. Nevertheless, it can be argued that under certain circumstances it can be the case that a piece of information which is mentioned only by one or very few sources might turn out to be very important. For example, a prominent source might have an exclusive piece of information that other sources do not have which might prove to be important for inclusion in the final summary. In such case the proposed model, indeed, will fail to include this piece of information in the final summary.

### 4.2 Why the Proposed Model Can Be Considered as a Good Starting Point for the Case of MDS for Evolving Events

The above discussion outlines some of the objections that might arise when the proposed model is applied under the prism of the general case of Multi-document Summarization. Despite those objections, we make the claim in this paper that the proposed model can nevertheless be considered as a good starting point for the case of Multi-document Summarization of Evolving Events, at least in the framework we have described in Section 2.

Concerning the first objection—*i.e.* the claim that the same trivial information might be contained in all the documents and thus such trivial information will have a high probability of being included in the final summary—this claim is rebuffed by the nature of the methodology that we have briefly presented in Section 2 and more fully exposed in [1] and [5]. The use of an ontology and especially the use of the messages guarantee that the system will try to extract information whose nature, we know beforehand, will be non-trivial. Of course, this beneficial situation has its drawbacks as well. As we have argued in [5] the creation of the ontology and the specifications of the messages require a considerable amount of human labor. Nevertheless, in Section 9 of [5] we present specific propositions of how this problem can be alleviated.

Let us now come to the second objection. According to this objection, it can be the case that a piece of information while mentioned by only one or very few sources (which implies that this piece of information stands very few chances of being included in the summary, according to the proposed model of Section 4) it might nevertheless be mentioned by a prominent source and thus ought finally to be included in the summary. Although this could be the case, we have to note as well that such prominent sources are usually highly influential ones as well. This has the implication that if a piece of information—which was initially exclusively mentioned by one source only—is indeed an important one for the description of the event’s evolution, then, almost surely, the rest of the sources will sooner or later follow the initial source in mentioning this information. Thus what was initially a light grey area, according to the discussion of Section 3, will tend to become darker grey, or even black, as time goes by, if indeed the mentioned piece of information is important and thus worthy of inclusion in the final summary of the event’s evolution.

This leaves us with the conclusion that the afore presented model can indeed serve as a nice starting point for the Content Determination stage, in the case that the grid contains more messages than the required compression rate requires.<sup>7</sup>

<sup>7</sup> It would be fair to mention that the above conclusion is valid in the case

## 5 Conclusions

In [1] and [5] we thoroughly presented a methodology (and applied it in two different case studies) which aims towards the creation of summaries from descriptions of evolving events which are emitted from multiple sources. The end result of this methodology is the computational extraction of a structure, which we called a grid. This structure is a directed acyclic graph (DAG) whose nodes are the messages extracted from the input documents and whose vertices are the Synchronic and Diachronic Relations that connect those messages. The creation of the grid, as we have argued, completes the Document Planning stage of a typical NLG architecture.

Nevertheless, it can be the case that the created grid can prove to be large enough in order for the final summary to exceed the required compression rate. In this paper we have presented a probabilistic model which can be applied to the Content Determination stage of the Document Planning phase. The application of that model<sup>8</sup> to the extracted grid will have the effect of creating a *subset* of the original grid (a *sub-grid* in other words) which will contain just the messages that confront to this model as well as the SDRs that connect *only* the selected messages.

From the discussion in this paper, as well as from the general literature in the area of Multi-document Summarization, we can conclude that the identification of similarities and differences is an essential component for any MDS system. Digressing a little bit at this point, we would like to note that spotting similarities between even disparate situations or objects, is something that human beings effortlessly and continuously perform all the time, and thus the study of this phenomenon is of paramount importance for the understanding of the human cognitive functioning. The mechanism of identifying “sameness”—despite its subtlety [9]—is an essential component for the task of analogy-making which lies at the core of cognition as [11] has claimed.

Closing this digression on the fascinating topic of analogy-making<sup>9</sup> we would like to note that with respect to MDS, to the best of our knowledge, there are no empirical studies as to how human beings proceed in order to create a summary from multiple documents—be they documents that describe evolving events, or not. We do not even have sufficient corpora of summaries from multiple documents which will provide us with an insight as to what can be considered a “good” multi-document summary. This comes in contrast with the area of Single Document Summarization (SDS) in which, of course, we do have such corpora. Moreover, in SDS we do have at least one substantial research from the perspective of Cognitive Science [8] which studies the cognitive mechanisms—or “strategies” as they are called in that book—of professional summarizers during the process of creating a summary from a single document. It is our personal belief that the performance of more such studies from the cognitive science perspective, for SDS and

---

that we do have the final set of documents which describe the evolution of the event. In case that the evolution is still on-going and this set is not yet finalized, then it might be the case that the second objection still holds.

<sup>8</sup> Although the probabilistic model presented in Section 4 talks about “pieces of information” the substitution of this abstract notion with the more concrete concept of *messages* makes that model ready for use in our methodology.

<sup>9</sup> The interested reader is encouraged to consult [9, 10] and [18] for more information on this topic.

MDS alike, will be beneficial for the advancement of our understanding not only of how we do create summaries, but for the understanding of how we spot similarities and differences; a task which lies at the heart of analogy-making as well.

## References

- [1] S. D. Afantenos. *Automatic Text Summarization from Multiple Sources for Time Evolving Events*. PhD thesis, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Athens, Greece, Dec. 2006.
- [2] S. D. Afantenos, I. Doura, E. Kapellou, and V. Karkaletsis. Exploiting cross-document relations for multi-document evolving summarization. In G. A. Vouros and T. Panayiotopoulos, editors, *Methods and Applications of Artificial Intelligence: Third Hellenic Conference on AI, SETN 2004*, volume 3025 of *Lecture Notes in Computer Science*, pages 410–419, Samos, Greece, May 2004. Springer-Verlag Heidelberg.
- [3] S. D. Afantenos, V. Karkaletsis, and P. Stamatopoulos. Summarization from medical documents: A survey. *Journal of Artificial Intelligence in Medicine*, 33(2):157–177, Feb. 2005.
- [4] S. D. Afantenos, V. Karkaletsis, and P. Stamatopoulos. Summarizing reports on evolving events; part i: Linear evolution. In G. Angelova, K. Bontcheva, R. Mitkov, N. Nicolov, and N. Nikolov, editors, *Recent Advances in Natural Language Processing (RANLP 2005)*, pages 18–24, Borovets, Bulgaria, Sept. 2005. INCOMA.
- [5] S. D. Afantenos, V. Karkaletsis, P. Stamatopoulos, and C. Halatsis. Using synchronic and diachronic relations for summarizing multiple documents describing evolving events. *Journal of Intelligent Information Systems*, 2007. Accepted for Publication.
- [6] S. D. Afantenos, K. Liantou, M. Salapata, and V. Karkaletsis. An introduction to the summarization of evolving events: Linear and non-linear evolution. In B. Sharp, editor, *Proceedings of the 2nd International Workshop on Natural Language Understanding and Cognitive Science, NLUCS 2005*, pages 91–99, Miami, Florida, USA, May 2005. INSTICC Press.
- [7] H. P. Edmundson. New methods in automatic extracting. *Journal for the Association for Computing Machinery*, 16(2):264–285, 1969. Also in [14].
- [8] B. Endres-Niggemeyer. *Summarizing Information*. Springer-Verlag, Berlin, 1998.
- [9] R. M. French. *The Subtlety of Sameness: A Theory and Computer Model of Analogy-Making*. A Bradford Book. The MIT Press, Cambridge, Massachusetts, 1995.
- [10] D. Gentner, K. J. Holyoak, and B. N. Kokinov, editors. *The Analogical Mind: Perspectives from Cognitive Science*. The MIT Press, Cambridge, Massachusetts, 2001.
- [11] D. R. Hofstadter. Analogy as the core of cognition. In D. Gentner, K. J. Holyoak, and B. N. Kokinov, editors, *The Analogical Mind: Perspectives from Cognitive Science*, chapter 15, pages 499–538. The MIT Press, Cambridge, Massachusetts, 2001.
- [12] H. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research & Development*, 2(2):159–165, 1958. Also in [14].
- [13] I. Mani. *Automatic Summarization*, volume 3 of *Natural Language Processing*. John Benjamins Publishing Company, Amsterdam/Philadelphia, 2001.
- [14] I. Mani and M. T. Maybury, editors. *Advances in Automatic Text Summarization*. The MIT Press, 1999.
- [15] W. C. Mann and S. A. Thompson. Rhetorical structure theory: A framework for the analysis of texts. Technical Report ISI/RS-87-185, Information Sciences Institute, Marina del Rey, California, 1987.
- [16] W. C. Mann and S. A. Thompson. Rhetorical structure theory: Towards a functional theory of text organization. *Text*, 8(3):243–281, 1988.
- [17] D. Marcu. *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press, 2000.
- [18] M. Mitchell. *Analogy Making as Perception: A Computer Model*. The MIT Press, Cambridge, Massachusetts, 1993.
- [19] C. D. Paice. The automatic generation of literature abstracts: An approach based on the identification of self-indicating phrases. In R. N. Oddy, S. E. Robertson, C. J. van Rijsbergen, and P. W. Williams, editors, *Information Retrieval Research*, pages 172–191. Butterworth, London, 1981.
- [20] E. Reiter and R. Dale. *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press, 2000.
- [21] G. Salton, A. Singhal, M. Mitra, and C. Buckley. Automatic text structuring and summarization. *Information Processing and Management*, 33(2):193–207, 1997. Also in [14].
- [22] M. Taboada and W. C. Mann. Rhetorical structure theory: Looking back and moving ahead. *Discourse Studies*, 8(3):423–459, June 2006.

# Metaphor, Inference and Domain Independent Mappings

Rodrigo Agerri, John Barnden, Mark Lee and Alan Wallington  
School of Computer Science, University of Birmingham  
B15 2TT Birmingham, UK  
*r.agerri@cs.bham.ac.uk*

## Abstract

This paper focuses on the interpretation of metaphor in discourse. We build on previous work [1] in which we provide a formalization in a computationally-oriented formal semantic framework of a set of mappings that we claim are required for the interpretation of *map-transcending* metaphor. Such mappings are domain-independent and are identified as invariant adjuncts to any conceptual metaphor. In this paper we claim that the invariant adjunct mappings allow us to account for metaphors where inferring discourse structure is not sufficient. Moreover, these mappings interact with rhetorical relations in order to explain cases in which metaphor affects discourse structure.

## Keywords

Metaphor Interpretation, Inference, Computational Semantics, Discourse Structure.

## 1 Introduction

We assume the general view that metaphor understanding involves some notion of events, properties, relations, etc. that are transferred from a source domain into a target domain. In this view, a metaphorical utterance conveys information about the target domain. We are particularly interested in a type of metaphorical utterances that we call *map-transcending*. A characteristic of *map-transcending* metaphor is that finding a target correspondent for every aspect of the source domain is a difficult task which, in some cases, seems to be plainly impossible. Thus, this type of metaphor poses great difficulties for correspondence-based approaches [11] which require to establish a parallelism between the source and target domains to explain metaphor.

We believe that an account of metaphor interpretation ought to explain what extra information *map-transcending* entities convey and it should provide a viable (computational) mechanism to explain how this transfer of information occurs. Moreover, it should do so by taking into account the fact that metaphor is a highly contextual phenomenon.

This paper addresses these two issues: Firstly, it builds on Agerri *et al.* [1] to provide a formal set of invariant mappings that we call View-Neutral Mappings Adjuncts (VNMA) for the interpretation of *map-transcending* metaphor. Secondly, it grounds the

invariant mappings on a (modified) computationally-oriented formal semantic framework for the interpretation of metaphor in discourse [3].

In order to do so, we first discuss the problems of correspondence approaches to deal with *map-transcending* metaphor. In section 3 we argue that inferring discourse structure is not sufficient to interpret certain metaphors. Sections 4 and 5 briefly describe our approach to metaphor interpretation. Section 6 describes a number of VNMA that are particularly useful to interpret *map-transcending* metaphor. In section 7 we propose to adapt Segmented Discourse Representation Theory (SDRT) [3] to our purposes of providing a formal account of metaphor interpretation based on the ATT-Meta approach. Finally, in section 8 we present some conclusions and discussion on further work.

## 2 Missing Correspondents

We do not address in this paper the issue of when an utterance is to be considered metaphorical. Instead, we aim to offer an explanation of how a metaphorical utterance such as (1) can be interpreted.

- (1) “McEnroe starved Connors to death.”

If we infer, using our knowledge about McEnroe and Connors, that (1) is used to describe a tennis match, it can be understood as an example of the conceptual metaphors (or, in our terminology, ‘metaphorical views’) DEFEAT AS DEATH and NECESSITIES AS FOOD. However, these metaphorical views would not contain any relationship that maps the specific *manner* of dying that constitutes *being starved to death* (we say that “starving” is a *map-transcending* entity as it goes beyond known mappings). Yet one could argue that the *manner* of Connors’s death is a crucial part of the informational contribution of (1).

A possible solution would be to create a new view-specific mapping that goes from the form of killing involved in *starving to death* to some process in sport, but such enrichment of mappings would be needed for many other verbs or verbal phrases that refer to other *ways* in which death is brought about, each requiring a specific mapping when occurring in a metaphorical utterance. Thus, finding adequate mappings could become an endless and computational intensive process. Moreover, there are even cases in which we may not find a plausible mapping. Consider the following description of the progress of a love affair:

(2) “We are spinning our wheels. ”

It is not very clear what could be a target correspondent for ‘wheels’; the unavailability of a correspondent would therefore prevent the source to target transfer of information needed for the interpretation of the metaphorical utterance. Thus, an account of metaphor ought to explain what extra information map-transcending entities provide. Furthermore, how the transfer of information occurs should be accounted for in a viable computational manner.

### 3 Metaphor in Discourse

Consider the following example:

(3) Sam is a pebble.

Asher and Lascarides [2] claim that it is not possible to calculate the meaning of an utterance such as (3) on the basis of the domain information about pebbles, but that it is possible to process it if it is discourse related to other utterance such as in the discourse “John is a rock but Sam is a pebble”. Specifically, they argue that inferring the *Contrast* discourse relation would help us to work out the metaphorical meaning of (3)). A similar point is made by Hobbs [9]:

(4) John is an elephant.

Which Hobbs argue can only be interpreted if we add extra information such that the example now consists of:

(5) Mary is graceful but John is an elephant.

Hobbs also infers *Contrast* in order to work out the meaning of “John being an elephant” as oppose to “Mary being graceful”. We claim that in some cases, the inference of some rhetorical relation does not provide all the information we need to interpret the metaphor:

(6) Mary is a fox and John is an elephant.

We can infer a *Coordination* discourse relation (we follow Gómez Txurruka on this point [8]) to account for the conjunction of the two segments. However, it seems that inferring Coordination would not be enough to address the fact that the information conveyed by (6) may be related to attributes of Mary (e.g., being cunning) and John (possessing a good memory).

Discourse-based approaches to metaphor such as [9] and [2] do not account for map-transcending entities, but they usually assume that there is some straightforward correspondence between the concepts in the source and target domains. Moreover, it seems that in some cases the inference of discourse relations is not enough to interpret some utterances. At the same time, a computational account of metaphor should address the occurrence of metaphor in discourse.

## 4 VNMA in ATT-Meta

Previous work [14] has shown evidence that there are metaphorical aspects (relations between events such as *causation* and event properties such as *rate* and *duration*) that, subject to being called, invariantly map from source to target whatever metaphorical view is being used. We refer to these type of mappings as VNMA. The VNMA are a central component of the ATT-Meta approach and AI System to metaphor interpretation previously presented by our group [5].

ATT-Meta [5] is an AI System and approach to metaphor interpretation that, apart from providing functionalities such as uncertainty and conflict handling, introduces two features central to the interpretation of metaphorical utterances such as (1) and (2): Instead of attempting the creation of new mappings to extend an existing metaphorical view, ATT-Meta employs query-driven reasoning within the terms of the source domain using various sources of information including *world* and *linguistic knowledge*. The nature of source domain reasoning in metaphor interpretation has not previously been adequately investigated, although a few authors have addressed it to a limited extent [9, 12, 13].

By means of VNMA and source domain reasoning it is possible to reach an interpretation of (3) without necessarily needing a rhetorical relation such as Contrast to guide the reasoning. Thus, linguistic knowledge and source domain reasoning about ‘pebbles’ may establish that they are small, and a very frequent association of unimportant entities with “small size” allows the defeasible inference that something is low, inferior, limited in worth (see Wordnet or any other lexical database). Using a Value-Judgment VNMA to express that “Levels of goodness, importance, etc., assigned by the understander in the source domain map identically to levels of goodness, etc.”, we can convey the meaning that Sam is limited in worth (worthless). Of course, the interpretation of (3) will vary if we change the discourse context.

Following this, and subject to the appropriate contextual query to be provided by the discourse, size-related features might be transferred in our approach by a Physical Size VNMA; in an appropriate context (6) could also be used to convey that John has a good memory and that Mary is cunning. In this case, forgetfulness could be seen a tendency to perform a mental act of a certain type and non-forgetfulness could be handled by a Negation VNMA, Mental states VNMA and a Event-Shape VNMA (for tendencies).

It may well be possible that studying the interaction between VNMA and discourse relations may allow us to naturally extend the study of metaphor to discourse. For example, in cases such as (6) both VNMA and rhetorical relations would be needed in order to give a full account of its interpretation. The interaction between VNMA and rhetorical relations is particularly clear when we consider cases of temporal metaphor (see Glasbey *et al.* [7] for details on temporal metaphor and discourse structure).

## 5 Source Domain Reasoning and VNMA's

(1) "McEnroe starved Connors to death."

Assuming a commonsensical view of the world and if (1) is being used metaphorically to describe the result of a tennis match, a plausible target interpretation would be that McEnroe defeated Connors by performing some actions to deprive him of his usual playing style. In the ATT-Meta approach, source domain inferencing produces a proposition to which we may apply a mapping to transfer that information. Thus, and assuming a commonsensical view of the world, a source domain meaning would be that McEnroe *starved* Connors to death in a biological sense. The source domain reasoning can then conclude that McEnroe *caused* Connors's death by *depriving* or disabling him. Leaving some details aside, the partial logical form (in the source domain) of the metaphorical utterance (1) may be represented as follows (without taking into account temporal issues):

(i)  $\exists x, y, e (McEnroe(x) \wedge Connors(y) \wedge starve - to - death(e, x, y))$

This says that there is an event  $e$  of  $x$  starving  $y$  to death (we use the notion of event á la Hobbs [9] to describe situations, processes, states, etc.). It may be suggested that if we were trying to map the partial expression (i), its correspondent proposition in the target could be expressed by this formula:

(ii)  $\exists x, y, e (McEnroe(x) \wedge Connors(y) \wedge defeat(e, x, y))$

According to this, the event of  $x$  defeating  $y$  in the target would correspond to the event of  $x$  starving  $y$  to death in the source. However, by saying "McEnroe starved Connors to death" instead of simply "McEnroe killed Connors" the speaker is not merely intending to convey that McEnroe defeated Connors, but rather something related to the manner in which Connors was defeated. Following this, *starving* may be decomposed into the cause  $e_1$  and its effect, namely, "being deprived of food":

(iii)  $\exists x, y, z, e_1, e_2, e_3 (McEnroe(x) \wedge Connors(y) \wedge food(z) \wedge starve(e_1, x, y) \wedge death(e_2, y) \wedge deprived(e_3, y, z) \wedge cause(e_1, e_3))$

Note that by factoring out "starving to death" in this way we not only distinguish the cause from the effect but doing so allows us to establish a relation between "death" in the source to "defeat" in the target using the known mapping in DEFEAT AS DEATH (and possibly "starving" to "McEnroe's playing" although we will not press this issue here).

Now, by means of lexical information regarding "starving", it can be inferred that McEnroe deprived Connors of a necessity (see, e.g., Wordnet), namely, of the food required for his normal functioning (the NECESSITIES AS FOOD metaphorical view would provide mappings to transfer food to the type of shots that Connors *needs* to play his normal game). In other

words, Connors is defeated by the particular means of depriving him of a necessity (food) which means that being deprived causes Connors's defeat. This fits well with the interpretation of (1) where McEnroe's playing deprived Connors of his usual game. Moreover, linguistic knowledge also provides the fact that starving someone to death is a gradual, slow process. The result of source domain inferencing may be represented as follows:

(iv)  $\exists x, y, z, e_1, e_2, e_3 (McEnroe(x) \wedge Connors(y) \wedge food(z) \wedge starve(e_1, x, y) \wedge death(e_2, y) \wedge deprived(e_3, y, z) \wedge cause(e_1, e_3) \wedge cause(e_3, e_2) \wedge rate(e_1, slow))$

'Slow' refers to a commonsensical source domain concept related to the progress rate of *starving*. Now, the existing mapping DEFEAT AS DEATH can be applied to derive, outside the source domain, that McEnroe defeated Connors, but no correspondences are available to account for the fact that McEnroe *caused* the defeat of Connors by depriving him of his normal play. Furthermore, the same problem arises when trying to map the slow progress *rate* of a process like starving.

In the ATT-Meta approach to metaphor interpretation, the mappings of *caused* and *rate* discussed above are accomplished by the type of invariant mappings that we specify as VNMA's (the Causation and Rate VNMA's, respectively; see [14] for an informal but detailed description of a number of VNMA's). VNMA's account for the mapping of aspects of the source domain that do not belong to a specific metaphorical view but that often carry an important informational contribution (or even the main one) of the metaphorical utterance. These source domain aspects can be captured as relationships and properties (causation, rate, etc.) between two events or entities that, subject to being called, identically transfer from source to target.

Summarizing, the following processes, amongst others, are involved in the understanding of map-transcending utterances: 1) Construction of source domain meaning of the utterance. 2) Source-domain reasoning using the direct meaning constructed in 1) with world and linguistic knowledge about the source domain. 3) Transfers by application of specific mappings in metaphorical views and often invariant mappings specified as VNMA's.

## 6 Description of VNMA's

By using VNMA's and source domain inference, we do not need to extend the mappings in the metaphorical view to include information about "depriving of a necessity", "food" or "causing Connors's death". VNMA's transfer those properties or relations between mappers that are *view-neutral*. Moreover, VNMA's are *parasitic* on the metaphorical views in the sense that they depend on some mappings to be established for the VNMA's to be triggered. That is why VNMA's are merely "adjuncts". VNMA's can also be seen as pragmatic principles that guide the understanding of metaphor by transferring aspects of the source domain that remain invariant.



In example (1), there are two VNMA's involved in the transfer of the causation and the "slowness", namely, the Causation and Rate VNMA's which are described below. Additionally, we also discuss a VNMA related to the temporal order of events (others are described in [4, 14]).

### 6.1 Causation/Ability

The idea is that there are relationships and properties (causation, (dis)enablement, etc.) between two events or entities that identically transfer from source to target. We use the  $\mapsto$  symbol to express that this mapping is a default.

**Causation/Ability VNMA:** "Causation, prevention, helping, ability, (dis)enablement and easiness/difficulty relationships or properties of events between events or other entities in the source domain, map to those relationships between their mappees (if they have any) in the target." The invariant mapping involved in the interpretation of (1) could be represented as follows:

$$\text{Causation: } \forall e_1, e_2 (\text{cause}(e_1, e_2)_{\text{source}} \mapsto \text{cause}(e_1, e_2)_{\text{target}})$$

As an additional note, the specific mapping of each event or state variable does not depend on the VNMA but on the metaphorical view in play. For example, if we consider the contemporary situation in which McEnroe and Connors are tennis pundits on TV, we may need a metaphorical view such as ARGUMENT AS WAR to interpret the utterance "McEnroe starved Connors to death". In other words, VNMA's do not themselves establish the mappees between source and target.

### 6.2 Rate

**Rate:** "Qualitative rate of progress of an event in the source domain maps identically to qualitative rate of progress of its mappee. E.g., if an event progresses slowly (in the context of the everyday commonsensical world), then its mappee progresses slowly (in the target context)".

Consider the following utterance:

- (7) My car gulps gasoline.

Briefly, the metaphorical view involved is MACHINES AS CREATURES, that maps biological activity to mechanical activity. Source domain reasoning may be performed along the following lines: It can be inferred that gasoline helps the car to be alive, therefore, it helps the car to be biologically active. The Causation/Ability VNMA (which deals with helping) combined with the above metaphorical view provide the target domain contribution that gasoline helps the car to run. Given that we can assume that an act of gulping is normally moderately fast the use of the Rate VNMA allows us to conclude that the car's use of gasoline is moderately fast. The logical form of this VNMA is could be expressed as follows:

$$\text{Rate: } \forall e, r (\text{rate}(e, r)_{\text{source}} \mapsto \text{rate}(e, r)_{\text{target}})$$

If the rate an event  $e$  in the source is  $r$ , then the rate maps to the mappee event in the target, that is, it also has rate  $r$ ;  $r$  refers to the qualitative rate of progress or duration of an specific event  $e$ .

### 6.3 Time-Order

**Time-Order:** "The time order of events in a source domain is the same as that of their mappee events, if any".

Time-order is quite useful for map-transcending examples such as

- (8) McEnroe stopped hustling Connors.

We might infer in the source domain that McEnroe was once hustling Connors which would be transferred by the Time-Order VNMA. For the formalization of this VNMA, we say that if event  $e_1$  precedes event  $e_2$  in the source, then the mappee events in the target exhibit the same ordering.

$$\text{Time-Order: } \forall e_1, e_2 (\text{precede}(e_1, e_2)_{\text{source}} \mapsto \text{precede}(e_1, e_2)_{\text{target}})$$

## 7 Metaphor in a Semantic Framework

Embedding the VNMA's in a semantic framework for metaphor interpretation is useful as a first step towards their implementation as default rules in the ATT-Meta system, but it is also interesting in its own right to show the contribution that the ATT-Meta approach can make towards a semantics of metaphor. In the somewhat simplified discussion on the source domain reasoning and VNMA's employed in the interpretation of (1), we have not stressed the fact that actually the source domain reasoning performed by the ATT-Meta system is query-driven. Although in previous sections we used various sources of contextual information to license certain source domain inferences, we have only considered isolated metaphorical utterances, and metaphor understanding has been illustrated as a process of forward reasoning from the direct meaning of utterances (in the source domain) and then the application of various metaphorical mappings to the result of source domain reasoning to arrive at the informational contributions in the target. Moreover, other possible inferences that could be drawn were ignored without specifying any principles or criteria whereby the reasoning could be guided towards the particular informational contributions discussed. The notion of discourse-query-directed reasoning provides such a guidance. When analyzing previous examples, we assume that the surrounding discourse context supplies queries that guide source domain reasoning in broadly the reverse order to that in which we described them in section 5. Other authors such as Hobbs [9] and Asher and Lascarides [2] also acknowledge the importance of context-derived reasoning queries play an important role in the interpretation of metaphorical utterances.

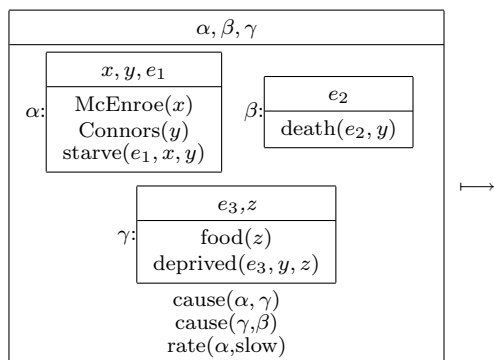
We are not claiming that query-directed reasoning may be the only type of reasoning involved in the

processing of metaphor, but it seems to be particularly important in the processing of connected discourse. Although the ATT-Meta system at present works with single-sentence utterances (albeit with the aid of discourse-query-directed reasoning), an aim for future versions is to extend it to the processing of *discourse*, and the semantic framework will need to allow for this.

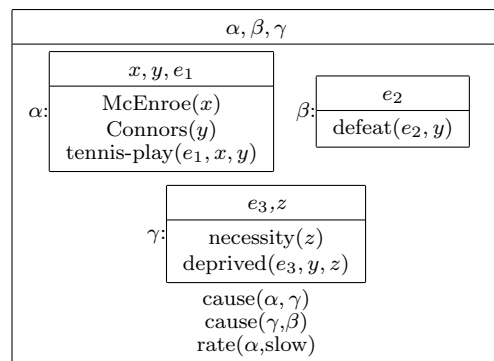
We have been using various sources of contextual knowledge that interact in the processing of the utterance: a) View-specific mappings provided by the relevant metaphorical views (DEFEAT AS DEATH and NECESSITIES AS FOOD); b) Linguistic and contextual information necessary for source domain reasoning; c) Relations and properties between events such as *causation* and *rate* that are inferred in the source; d) VNMA that transfer event relations and properties from source to target; and finally, e) Rhetorical relations that take into account the structure of discourse. In our view, a suitable approach to metaphor in discourse should include at least these five components.

## 7.1 Semantics for the ATT-Meta approach

Metaphor is a highly contextual phenomenon, and one of the most interesting semantic approaches that model context are dynamic semantics such as Segmented Discourse Representation Theory (SDRT) [3]. Specifically, we adapt the semantic representation procedure of SDRT to build Segmented Discourse Representation Structures (SDRSs) consisting of the result of source domain reasoning. The conclusion of source domain inference can in turn be mapped to the target by using various view-specific mappings and VNMA. In other words, we can see the source SDRS as the input for what the ATT-Meta system does when interpreting metaphor – it will reason with it, producing an output of inferred target facts which we may also represent by means of an SDRS. The result of reasoning in the source domain to interpret (1) would now look as follows:



where  $\alpha$  and  $\beta$  are labels for DRSs representing events and  $\mapsto$  mappings (VNMA and central mappings) needed in the interpretation of the metaphorical utterance. Importantly, the VNMA would pick upon aspects such as causation and rate from the source to transfer them to the target producing an output which could also be represented as a SDRS:



Note that this formal representation integrates the systematicity of mapping invariantly certain aspects of metaphorical utterances by formulating them as relations and properties of events that can be represented as relations and properties of DRSs. For this purpose we will need to modify the construction rules of SDRSs to be able to infer properties and relations involving individuals ( $x, y, \dots$ ) and not only DRSs' labels such as  $\alpha$  and  $\beta$ . In addition to this, we need to capture the interaction of the various sources of information used (linguistic knowledge, world knowledge, etc.) to infer causation and rate in the source domain. Thus, we partially adopt SDRT formal framework to represent ATT-Meta's source domain reasoning, event relations, event properties and VNMA with the purpose of developing a semantic account of metaphor interpretation.

## 7.2 Discourse Contexts

Source domain reasoning partially relies on inferences provided by the discourse context and linguistic and world knowledge. In the ATT-Meta system, world knowledge roughly corresponds to source domain knowledge. On the one hand, we have been using our commonsensical knowledge about McEnroe and Connors to interpret example (1) as metaphorically describing a tennis match. On the other hand, linguistic knowledge is used to *pretend* that the direct meaning of the metaphorical utterance is true, which allows us to derive *causation* and *rate*. Thus, we assume that the understander possesses some world knowledge that provides information about “starving someone to death”:

- If  $e_3$  where  $y$  is deprived and  $e_1$  where  $x$  starves  $y$  are connected, then by default,  $e_1$  causes  $e_3$ .
- If  $e_2$  where  $y$  dies and  $e_3$  where  $y$  is deprived are connected, then by default,  $e_3$  causes  $e_2$ .
- If  $e_1$  where  $x$  starves  $y$ , then by default, the rate of progress of  $e_1$  is *slow*.

Furthermore, common sense about causation tells us that “if  $e_1$  causes  $e_3$  then  $e_3$  does not occur before  $e_1$ ”. Following this, the knowledge needed to interpret example (7) needs to include the that the drinking rate is fast:

If  $e$  where  $x$  gulps, then by default,  $x$  in  $e$  drinks moderately fast.

SDRT specifies where in the preceding discourse structure the proposition introduced by the current sentence can attach with a discourse relation. In order to do that, it is necessary to provide a set of rules for the understander to infer which discourse relation should be used to do attachment. We adopt a similar notation to represent discourse update (see [3] for details on the discourse update function) so that defeasible knowledge about causation, rate, temporal order, etc., allows the inference of source domain event relations and properties.

Let us suppose that in a context (source domain)  $\omega$  we want to attach some event denoted by  $\beta$  to  $\alpha$ , such that  $\langle \omega, \alpha, \beta \rangle$ . This update function can be read as “the representation  $\omega$  of a text so far is to be updated with the representation  $\beta$  of an event via a discourse relation with  $\alpha$ ” [3]. Let  $\rightsquigarrow$  represent a defeasible connective as a conditional, and let  $ev(\alpha)$  stand for “the event described in  $\alpha$ ”; although  $ev(\alpha)$  is quite similar to the notion of main eventuality  $me$  defined by Asher and Lascarides [3], we do not commit to other assumptions of their theory.

Thus, some of the source domain knowledge about causation in (1) discussed above could now be represented as follows:

$$\langle \omega, \alpha, \beta \rangle dies(connors, ev(\beta)) \wedge starves(mcenroe, connors, ev(\alpha)) \rightsquigarrow cause(ev(\alpha), ev(\beta))$$

We can then infer in the source a *causation* relation between  $\alpha$  and  $\beta$  if the event represented in  $\alpha$  normally causes  $\beta$ :

$$\mathbf{Causation:} \langle \omega, \alpha, \beta \rangle \wedge (cause(ev(\alpha), ev(\beta))) \rightsquigarrow causation(\alpha, \beta)$$

Note that ‘cause’ refers to the epistemic notion of one event causing another, whereas ‘causation’ refers to an inferred semantic relation between segments of discourse or, in other words, between semantic representation of events by means of DRSs. In order to include properties (and not only relations) in this framework, we assume a conceptualist point of view and consider that properties such as rate or value-judgement denote *concepts* (fast, slow, good, bad) which may correspond to the absolute rate in a commonsensical view of the world. Its representation in our semantic framework could be defined by adding an extra clause to the definition of DRS-formulae:

- If  $P$  is a property symbol and  $\alpha$  and  $r$  are an episode label and a property label respectively, then  $P(\alpha, r)$  is an DRS-formula (see [3] for the complete definitions of DRS-formulae and SDRS construction).

Thus, a rule encoding contextual knowledge to infer rate in the source would look as follows (note that when considering event properties we only need to consider one DRS  $\alpha$  in our rules, even though a discourse usually consists of one or more DRSs):

$$\langle \omega, \alpha \rangle gulps(car, gasoline, ev(\alpha)) \rightsquigarrow fast(ev(\alpha))$$

Supported by this rule we can then infer an event property in the source for its subsequent transfer to target via the Rate VNMA (when the Rate VNMA is instantiated):

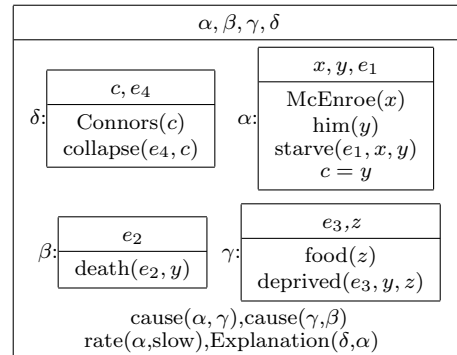
$$\mathbf{Rate:} \langle \omega, \alpha \rangle (fast(ev(\alpha)) \rightsquigarrow rate(\alpha, fast))$$

### 7.3 VNMA and Rhetorical Relations

We are now ready to extend the use the VNMA introduced in section 6 and the above points about source domain inferencing and contextual knowledge to offer SDRT-based semantic representations, based on the ATT-Meta approach to metaphor, for discourse examples. For simplicity of exposition, we leave out any details not directly relevant to the discussion on VNMA. Consider the following variation of (1):

- (1b) Connors collapsed as McEnroe starved him to death.

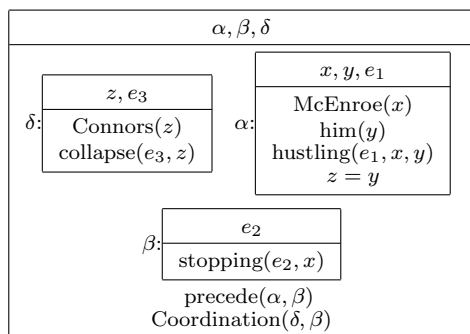
(1b) suggests that the cause or explanation of the collapsing of Connors is the “starving of Connors” which precedes Connors’s collapsing. Without going into many details (see [3]) the two main clauses of the discourse are linked by the Explanation rhetorical relation. This means that in order to interpret fully (1b) we need to take into account both the metaphorical aspects and its structure. Leaving aside the specific metaphorical meaning of ‘collapse’ and pro-nominal issues, the result of source domain reasoning for (1b) could be represented as follows:



Thus, inferring Explanation allows us to conclude that being starved to death *explains* Connors collapsing. The following example will allow us to show how our approach deals with coordination and temporal precedence discussed in examples (6) and (8):

- (1c) Connors collapsed and McEnroe stopped hustling him.

In terms of discourse structure, we follow Txurruka’s approach [8]: ‘and’ marks a Coordination relation between the conjuncts, blocking any other plausible interpretation of (1c) such as Result (the second conjunct will be the result of the first one). If we also consider the metaphorical analysis offered while discussing example (8), the result is the following semantic structure representing the conclusion of source domain reasoning:



Summarizing, the semantic framework outlined in this section consists of: (i) DRSs and SDRSs consisting of events, individuals, states, etc. They can be thought of as situations or as representation structures as in dynamic semantics. A context consists of one or more DRSs, DRSs relations and properties; (ii) Event relations and properties such as *causation*, *rate*, *time-order*, etc inferred in the source domain for the systematic transfer of certain type of information conveyed by metaphorical utterances. The transfer of this type of information via VNMA is a contribution of the ATT-Meta approach to metaphor interpretation [4, 14]; (iii) Rhetorical relations to address the structure of discourse and provide a more complete analysis of metaphor occurring in discourse.

## 8 Concluding Remarks

This paper investigates the formalization and semantic representation of the ATT-Meta approach to metaphor interpretation. The ATT-Meta approach is backed up by a powerful implementation that performs sophisticated reasoning to interpret metaphorical utterances. We have focused on description and formalization of several VNMA, mappings for the systematic transference of invariant aspects from source to target. We have shown how a dynamic semantic approach can be adapted for these purposes to offer an unified semantic representation of ATT-Meta's view of metaphor interpretation.

Map-transcending entities pose a problem for several analogy-based approaches to metaphor interpretation, both from a computational and a theoretical point of view. With respect to the computational approaches, theories of metaphor interpretation based on analogy [6, 10] usually require a conceptual similarity between the source and the target domains. Map-transcending entities need to be mapped by extending on the fly the metaphorical views with new correspondences. We have argued that this strategy is both computationally expensive and in some cases, plainly impossible.

Formal semantic approaches [3] do not account for metaphorical utterances including map-transcending entities. Other works [9, 12, 13] have addressed source domain reasoning to a limited extent, but its role in metaphor interpretation has not previously been adequately investigated. Moreover, map-transcending entities pose a problem for analogy-based approaches to metaphor interpretation [6], which usually require a conceptual similarity between the source and the target domains.

## References

- [1] R. Agerri, J. Barnden, M. Lee, and A. Wallington. On the formalization of invariant mappings for metaphor interpretation. In *Proceedings of ACL 2007 Demo and Poster Sessions*, pages 100–113, Prague, June 2007. Association for Computational Linguistics.
- [2] N. Asher and A. Lascarides. The semantics and pragmatics of metaphor. In P. Bouillon and F. Busa, editors, *The Language of Word Meaning*, pages 262–289. Cambridge University Press, 2001.
- [3] N. Asher and A. Lascarides. *Logics of Conversation*. Cambridge University Press, 2003.
- [4] J. Barnden, S. Glasbey, M. Lee, and A. Wallington. Domain-transcending mappings in a system for metaphorical reasoning. In *Companion to the 10th EAACL*, pages 57–61, 2003.
- [5] J. Barnden and M. Lee. An artificial intelligence approach to metaphor understanding. *Theoria et Historia Scientiarum*, 6(1):399–412, 2002.
- [6] B. Falkenhainer, K. Forbus, and D. Gentner. The structure-mapping engine: algorithm and examples. *Artificial Intelligence*, 41(1):1–63, 1989.
- [7] S. Glasbey, J. Barnden, M. Lee, and A. Wallington. Temporal metaphors in discourse. Technical Report CRSP-04-03, School of Computer Science, University of Birmingham, October 2004. <ftp://ftp.cs.bham.ac.uk/pub/tech-reports/2004/CSRP-04-03.ps.gz>.
- [8] I. Gomez-Txurruka. The natural language conjunction ‘and’. *Linguistics and Philosophy*, 26(3):255–285, 2003.
- [9] J. Hobbs. An approach to the structure of discourse. [www.isi.edu/hobbs/discourse-references/discourse-references.html](http://www.isi.edu/hobbs/discourse-references/discourse-references.html), 1996. Probably not to be published.
- [10] K. Holyoak and P. Thagard. Analogical mapping by constraint satisfaction. *Cognitive Science*, 13(3):295–355, 1989.
- [11] G. Lakoff. The contemporary theory of metaphor. In A. Ortony, editor, *Metaphor and Thought*, 2nd edition. Cambridge University Press, Cambridge (MA), 1993.
- [12] J. Martin. *A computational model of metaphor interpretation*. Academic Press, New York, 1990.
- [13] S. Narayanan. *KARMA: Knowledge-based action representations for metaphor and aspect*. PhD thesis, Computer Science Division, EECS Department, University of California, Berkeley, August 1997.
- [14] A. Wallington, J. Barnden, S. Glasbey, and M. Lee. Metaphorical reasoning with an economical set of mappings. *Delta*, 22(1), 2006.

# Hierarchical Agglomerative Clustering of English-Bulgarian Parallel Corpora\*

Rayner Alfred, Dimitar Kazakov, Mark Bartlett  
Computer Science Department  
University of York, York, UK  
{ralfred,kazakov,bartlett}@cs.york.ac.uk

Elena Paskaleva  
Bulgarian Academy of Sciences  
Sofia, Bulgaria  
hellen@lml.bas.bg

## Abstract

Multilingual corpora are becoming an essential resource for work in multilingual natural language processing. In this article, we report on our work on applying hierarchical agglomerative clustering (HAC) to a large corpus of documents where each appears both in Bulgarian and English. We cluster these documents for each language and compare the results both with respect to the shape of the tree and the content of clusters produced. Further, we study the effects of reducing the set of terms used for clustering. On the data available, the results of clustering one language resemble the other, provided the number of clusters required is relatively small. Reducing the number of terms used appears a viable strategy for English, but is not acceptable for Bulgarian. These results can be used to design information retrieval (IR) strategies where NLP tools are not available for the language of interest, but the documents in question have been translated to a language for which such tools exist.

## Keywords

Multilingual NLP, evaluation, corpus-based language processing, bilingual parallel clustering

## 1. Introduction

Effective and efficient document clustering algorithms play an important role in providing intuitive navigation and browsing mechanisms by categorizing large amounts of information into a small number of meaningful clusters. In particular, clustering algorithms that build illustrative and meaningful hierarchies out of large document collections are ideal tools for their interactive visualization and exploration, as they provide data-views that are consistent, predictable and contain multiple levels of granularity. There has been a lot of research in clustering text documents. However, there are few experiments that examine the impacts of clustering bilingual parallel corpora, possibly due to the problem of the availability of large corpora in translation, i.e. parallel corpora. Fortunately, we have obtained a large collection of over 20,000 pairs of English-Bulgarian documents that form our bilingual parallel corpus. Compared to a clustering algorithm based on a single language, applying clustering to the same documents in two languages can be attractive for several reasons. Firstly, clustering in one language can be used as a source of annotation to verify the clusters produced for the other language. Secondly,

combining results for the two languages may help to eliminate some language-specific bias, e.g., related to the use of homonyms, resulting in classes of better quality. Finally, the alignment between pairs of clustered documents can be used to extract words from each language and can further be used for other applications, such as cross-linguistic information retrieval (CLIR) [5].

The aim of the experiments presented in this paper is to investigate the effect of applying a clustering technique to parallel multilingual texts. Specifically, the aim is to introduce the tools necessary for this task and display a set of experimental results and issues, which have become apparent. In this paper, we provide the comparison results of clustering parallel corpora of English-Bulgarian texts in three main areas: English-Bulgarian cluster mappings, English-Bulgarian tree structures and the extracted most representative terms for English-Bulgarian clusters. Additionally, the effect of term reduction on the cluster mappings is examined.

Chapter 2 covers some of the background about the vector space model representation of documents and the hierarchical agglomerative clustering method. Chapter 3 explains the experimental design set-up and the experimental results are outlined in Chapter 4. Chapter 5 concludes this paper by suggesting what can be done to improve the hierarchical agglomerative clustering of bilingual parallel corpora of English-Bulgarian.

## 2. Background

### 2.1 Vector Space Model Representation

We use the vector space model [2], where a document is represented as a vector in  $n$ -dimensional space ( $n$  = number of different words). Here, documents are categorized by the words they contain and their frequency. Before obtaining the weights for all the terms extracted from these documents, stemming and stopword removal is performed. Stopword removal eliminates unwanted terms and thus reduces the number of dimensions in the term-space.

$$\text{tf-idf} = \text{tf}(t,d) \cdot \text{idf}(t) \quad (1)$$

$$\text{idf}(t) = \log \left( \frac{|D|}{\text{df}(t)} \right) \quad (2)$$

$$\text{sim}(d_i, d_j) = \frac{(d_i \cdot d_j)}{\|d_i\| \|d_j\|} \quad (3)$$

\* The work was fully funded by a grant provided within the project BIS-21++ at the Institute for Parallel Processing, Bulgarian Academy of Sciences. BIS-21++ is a project funded by the European Commission in FP6 INCO via contract no.: INCO-CT-2005-016639.

$$\text{Precision } P(C,L) = \frac{|C \cap L|}{|C|}, \quad C \in C_{ALL}, L \in L_{ALL} \quad (4)$$

$$\text{Purity} = \sum_{C \in C_{ALL}} \frac{|C|}{|D|} \cdot P(C,L) \quad (5)$$

$$\text{Precision (EBM)} = \frac{|C(E) \cap C(B)|}{|C(E)|} \quad (6)$$

$$\text{Precision (BEM)} = \frac{|C(B) \cap C(E)|}{|C(B)|} \quad (7)$$

Weights are assigned to indicate the importance of a word in characterizing a document as distinct from the rest of the corpus. Each document is viewed as a vector whose dimensions correspond to words or terms extracted from the document. The component magnitudes of the vector are the tf-idf weights of the terms. In this model, tf-idf (1) is the product of term frequency  $tf(t,d)$ , which is the number of times term  $t$  occurs in document  $d$ , and the inverse document frequency, equation (2), where  $|D|$  is the number of documents in the complete collection and  $df(t)$  is the number of documents in which term  $t$  occurs at least once.

## 2.2 Hierarchical Agglomerative Clustering

In this work, we concentrate on hierarchical agglomerative clustering (HAC). Unlike partitional clustering algorithms that build a hierarchical solution from top to bottom, repeatedly splitting existing clusters, HAC builds the solution by initially assigning each document to its own cluster and then repeatedly selecting and merging pairs of clusters, to obtain a single all-inclusive cluster, generating the cluster tree from leaves to root [3]. The main parameters in HAC algorithms are the metric used to compute the similarity of documents and the method used to determine the pair of clusters to be merged at each step.

**Table 1. Statistics of Document News and Features**

Category (Num Docs)	Language	Total Words	Avg. Words	Different Terms
News briefs (1835)	English	279,758	152	8,456
	Bulgarian	288,784	157	15,396
Features (2172)	English	936,795	431	16,866
	Bulgarian	934,955	430	30,309

The cosine distance, equation (3), is used to compute the similarity between two documents  $d_i$  and  $d_j$ . The two clusters to merge at each step are found using the average link method. In this scheme, the two clusters to merge are those with the greatest average similarity between the documents in one cluster and those in the other.

Given a set of documents  $D$ , one can measure how consistent the results of clustering for each of the languages to which these documents are translated in the

following way. The clusters produced for one language are used as ‘gold standard’, a source of annotation assigning each document in the set  $D$  a cluster label  $L$  from the list  $L_{ALL}$  of all clusters for that language. Clustering in the other language is then carried out and *purity*<sup>1</sup> [6], equation (5), used to compare each of the resulting clusters  $C \in C_{ALL}$  to its closest match among all clusters  $L_{ALL}$ .

## 3. Experimental Design

In this experiment, there are two categories of parallel corpora (News Briefs and Features) in two different languages, English and Bulgarian. In both corpora, each English document  $E$  corresponds to a Bulgarian document  $B$  with the same content, see Table 1. It is worth noting that the Bulgarian texts have a higher number of terms after stemming and stopword removal. The process of stemming English corpora is relatively simple due to the low inflectional variability of English. However, for morphologically richer languages, such as Bulgarian, where the impact of stemming is potentially greater, the process of building an accurate algorithm becomes a more challenging task [1]. In this experiment, the Bulgarian texts are stemmed using the BulStem algorithm [1] and the English documents are stemmed using a simple affix removal algorithm. Figure 1 illustrates the experimental design set up. The documents in each language are clustered separately according to their categories (News Briefs or Features) using HAC. The output of each run consists of three elements: a list of terms characterizing the cluster, the cluster members, and the cluster tree for each set of documents. The next section contains a detailed comparison of the results for the two languages based on these issues.

## 4. Experimental Results

### 4.1 Mapping of English-Bulgarian cluster memberships

In a first experiment, every cluster in English is paired with the Bulgarian cluster with which it shares the most documents. The same is repeated in the direction of Bulgarian to English mapping. Two precision values of these pairs are then calculated, the precision of the English-Bulgarian mapping (EBM) and that of the Bulgarian-English mapping (BEM). Figures 2–5 show the precisions for the EBM and BEM for the cluster pairings obtained with varying numbers of clusters,  $k$  ( $k = 10, 20, 40$ ) for each of the two domains, News Briefs and Features. The X axis label indicates the ID of the cluster whose nearest match in the other language is sought, while the Y axis indicates the precision of the

<sup>1</sup> *Precision* is the probability of a document in cluster  $C$  being labelled  $L$ . *Purity* is the percent of correctly clustered documents.

best match found. For example, in Figure 2, EN cluster 7 is best matched with BG cluster 6 with the EBM mapping precision equal to 58.7% and BEM precision equal to 76.1%.

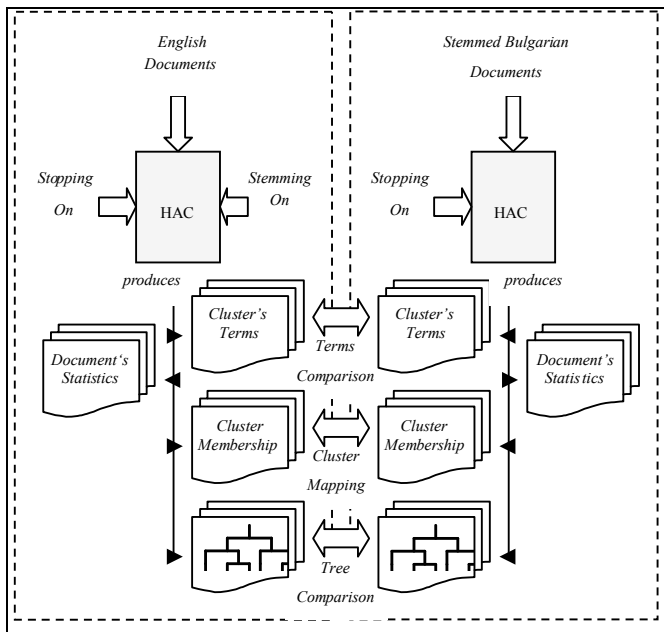


Figure 1. Experimental set up for parallel clustering task

A final point of interest is the extent to which the mapping EBM matches BEM. Table 3 shows that alignment between the two sets of clusters is 100% when  $k = 10$  for both domains, News Briefs and Features. However, as the number of clusters increases, there are more clusters that are unaligned between the mappings. This is probably due to the fact that Bulgarian documents have a greater number of distinct terms. As the Bulgarian language has more word forms to describe English phrases, this may affect the computation of weights for the terms during the clustering process.

It is also possible to study the purity of the mappings. Table 2 indicates the purity of the English-Bulgarian document mapping for various values of  $k$ . This measure has only been based on the proportion of clusters that have been aligned, so it is possible to have a case with high purity, but a relatively low number of aligned pairs.

Table 2. Degree of Purity for Cluster Mapping for English-Bulgarian Documents

Category	k=5	k=10	k=15	k=20	k=40
News briefs	0.82	0.63	0.67	0.65	0.59
Features	N/A	0.77	N/A	0.61	0.54

Table 3. Percentage Cluster Alignment

Category	k = 10	k = 20	k = 40
News briefs	100.0%	85.0%	82.5%
Features	100.0%	90.0%	80.0%

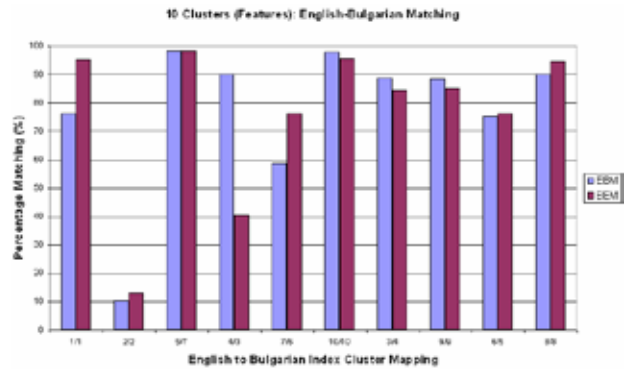


Figure 2. Ten clusters, Features corpus.

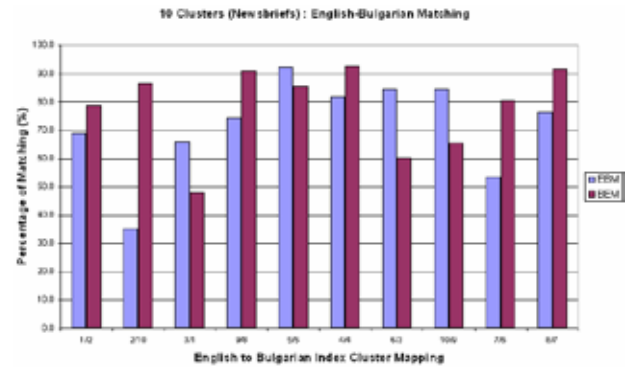


Figure 3. Ten clusters, News Briefs corpus.

## 4.2 Comparison of HAC Tree Structures

The cluster trees obtained for each language are reduced to a predefined number of clusters (10, 20 or 40) and then the best match is found for each of those clusters in both directions (EBM, BEM). Here we would only pair a Bulgarian cluster  $C_{BG}$  with an English cluster  $C_{EN}$  if they are each other's best match, that is,  $C_{BG} \xrightarrow{BEM} C_{EN}$  and  $C_{EN} \xrightarrow{EBM} C_{BG}$ .

The pair of cluster trees obtained for each are compared by first aligning the clusters produced from both sets of documents and then plotting the corresponding tree for each language. Figure 8 and Figure 10 illustrate that when  $k = 10$ , all clusters can be paired, and the tree structures for both the English and Bulgarian documents

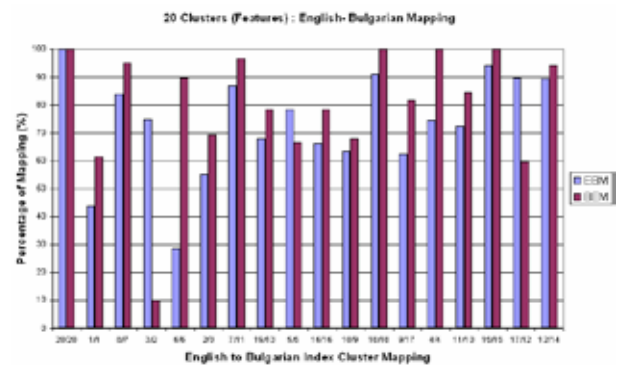


Figure 4. Twenty clusters, Features corpus.

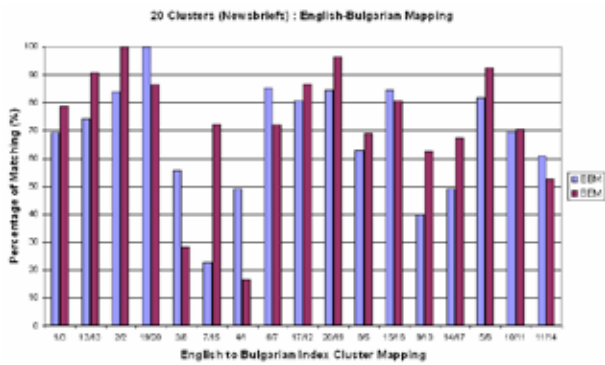


Figure 5. Twenty clusters, News Briefs corpus.

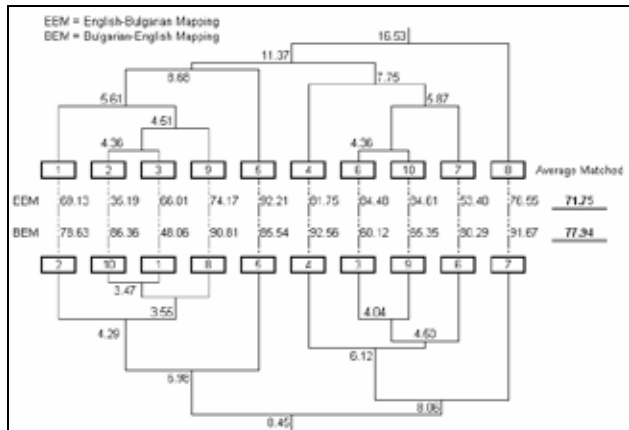


Figure 6. Ten clusters, News Briefs corpus.

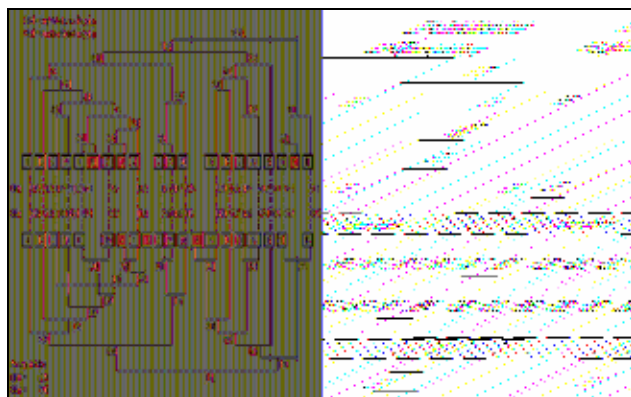


Figure 7. Twenty clusters, News Briefs corpus.

are identical (although *distances* between clusters may vary). However, when  $k = 20$ , there are unpaired clusters in both trees, and after the matched pairs are aligned, it is clear that the two trees are different. We hypothesise that this may be a result of the higher number of stems produced by the Bulgarian stemmer, which demotes the importance of terms that would correspond to a single stem in English.

### 4.3 Comparison of Terms Extracted from English and Bulgarian Clusters

The ten most representative terms that describe the matching English and Bulgarian clusters have a similar meaning as illustrated in Tables 4 and 5. The only notable exception is listed in column 2 of Table 4, where all top Bulgarian terms are related to the topic of bird flu, whereas the English terms are split between this topic and the one of Olympic games. This difference disappears when the number of clusters is increased to 20 (and a consistent bird flu  $19_{EN}/20_{BG}$  pair of clusters is formed).

### 4.4 Term Reduction

Having seen in the previous experiment that the most representative words for each cluster are similar for each language, an interesting question is whether clustering using only these words improves the overall accuracy of alignment between the clusters in the two languages. The intuition behind this is that, as the words characterizing each cluster are so similar, removing most of the other words from consideration may be more akin to filtering noise from the documents than to losing information.

The clustering is rerun as before, but with only a subset of terms used for the clustering. That is to say, before the tf-idf weights for each document are calculated, the documents are filtered to remove all but  $n$  of the terms from them. These  $n$  terms are determined by first obtaining 10 clusters for each language, and then extracting the top 10 (resp. 50) terms which best characterise each cluster, with the total number of terms equals to at most  $10 \times 10 = 100$  (resp.  $10 \times 50 = 500$ ).

The results of comparing clusters in English and Bulgarian are shown in Table 6. These clearly indicate that as the number of terms used in either language falls, the number of aligned pairs of clusters also decreases. While term reduction in either language decreases the matching between the clusters, the effect is fairly minimal for English and far more pronounced for Bulgarian. In order to seek to explain this difference between the languages, it is possible to repeat the process of aligning and calculating purity, but using pairs of clusters from the same language, based on datasets with different levels of term reduction. The results of this are summarised in Table 7.

This table demonstrates that, for both languages, as the number of terms considered decreases, the clusters formed deviate further and further from those for the unreduced documents. While the deviation for English is quite low (and may indeed be related to the noise reduction sought), for Bulgarian reducing the number of terms radically alters the clusters formed. As with earlier experiments, the high morphological variability of Bulgarian compared to English may again be the cause of the results observed.



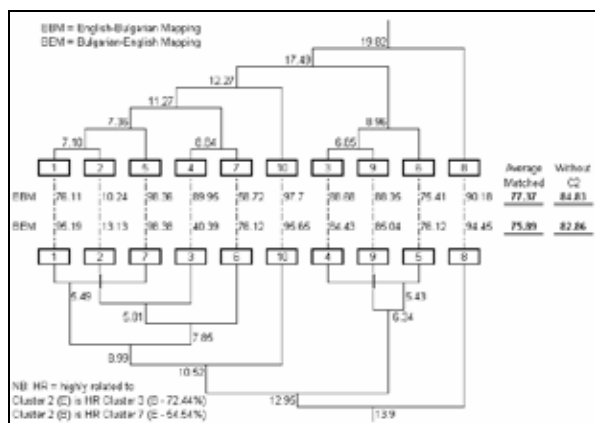


Figure 8. Ten clusters, Features corpus.

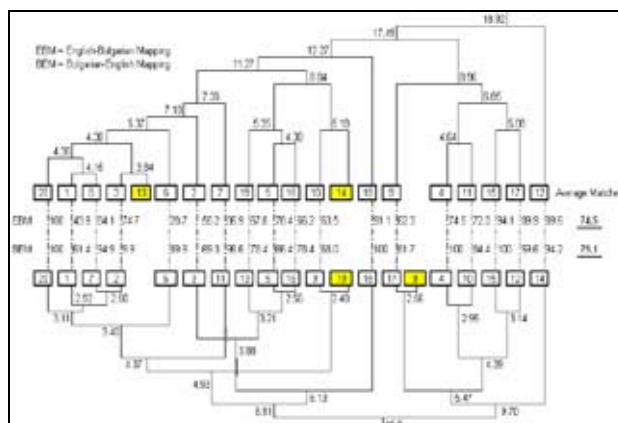


Figure 9. Twenty clusters, Features corpus.

Table 4. Top ten terms for pairs of English and Bulgarian clusters (k = 10, all paired)

1	2	3	4	5	6	7	8	9	10
macedonia	olymp	eu	kosovo	turkei	tribunal	serbia	bih	bulgarian	croatia
macedonian	bird	albania	provinc	turkish	crime	serbian	rs	bulgaria	croatian
tv	flu	albanian	statu	erdogan	war	montenegro	ashdown	mediapool	gotovina
a1	game	romania	unmik	eu	milosev	mladic	novin	sofia	hina
skopj	test	minist	serb	ankara	trial	belgrad	nezavisn	btv	zagreb
vesnik	medal	countri	pristina	cypru	court	tanjug	repres	iraq	list
utrinski	greek	cent	albanian	cypriot	prosecutor	b92	pb	bta	sanad
makfax	athen	europelan	belgrad	hagu	minist	zoran	high	parvanov	vecernji
crvenkovski	greec	nato	jessen	agenc	bosnian	kostunica	republika	minist	ant
mia	bronz	bih	petersen	greek	serb		srpska	trud	hrt
2	10	1	4	5	3	6	7	8	9
македони	грип	алба	косов	турци	трибунал	гора	рс	българск	хърват
македонск	птичи	ес	провинци	турск	престъпл	сърбия	бих	българи	хърватск
A1	птици	парти	статут	ердоган	милошевич	Черна	ашдаун	ирак	готовин
цървенковск	вирус	румъни	причин	ес	оон	младич	представител	софия	кина
скопие	H5N1	нато	юнмик	анкар	военни	сърбия-Черн	сръбск	бтв	лист
тв	лебед	минист	косовск	кипър	сръбск	белград	независн	медиапул	загреб
бучковск	птичия	други	йесен-петерсен	анадолск	обвин	сръбск	новин	първанов	санадер
утринск	случаи	правителств	оон	агенци	г	b92	пбс	бнт	ес
макфакс	мъртв	новин	сръбск	кипърск	хага	танюг	републи	бта	месич
трайковск	шам	македони	белград		понте	ес	върхов	минист	вечер

Table 5. Top ten terms for pairs of English and Bulgarian clusters (k = 20, 17 are paired, only the first 9 are listed here)

1	2	3	4	5	6	7	8	9
macedonia	olymp	albanian	cent	kosovo	turkei	eu	tribun	serbia
macedonian	game	albania	gt	provinc	turkish	romania	crime	serbian
tv	medal	tirana	lt	statu	eu	romanian	war	montenegro
a1	greek	osc	bih	unmik	ankara	rompr	milosev	b92
skopj	athen	elec	bank	serb	erdogan	minist	trial	tanjug
vesnik	greec	moisiu	deficit	pristina	acces	wednesdai	court	djindjic
utrinski	bronz	ata	govern	albanian	istanbul	croatia	prosecutor	parti
makfax	won	tuesdai	imf	belgrad	membership	europelan	hagu	zoran
crvenkovski	men	a1fr	undp	jessen	talk	countri	bosnian	belgrad
mia	stadium	countri	world	petersen	ntv	acces	serb	minist
3	2	8	1	6	7	15	5	10
македони	олимпийск	алба	сръбск	косов	турци	румъни	трибунал	гора
македонск	медал	нато	млн	провинци	турск	румънск	престъпл	Черна
A1	атин	македони	правителств	статут	ес	ромпрес	военни	сърбия
цървенковск	олимпиад	ес	бежан	причин	анкар	ес	оон	сърбия-Черн
скопие	игрит	албанск	други	юнмик	ердоган	търичану	обвин	сръбск
тв	гърци	тиран	новин	косовск	преговор	попеску	г	белград
бучковск	слечел	минист	%	йесен-петерсен	членств	найн	караджич	b92
утринск	игри	комиси	евро	оон	кюрдск	о' клок	понте	референдум
макфакс	бронзов	европейск	бих	сръбск	нтв	калин	дел	таджич
трайковск	категори	ек	представител	белград	гюл	настас	хага	танюг

**Table 6. Number of aligned clusters and their purity for reduced term clustering (k = 10)**

		Bulgarian Terms		
		All	500	100
English Terms	All	10 (74.9%)	4 (54.2%)	3 (53.0%)
	500	9 (72.9%)	4 (46.0%)	3 (51.5%)
	100	9 (70.3%)	4 (60.1%)	2 (75.5%)

**Table 7. Number of aligned clusters and their purity for reduced term datasets against the unreduced dataset (k=10)**

	All	500	100
English	10 (100%)	10 (80.1%)	9 (74.2%)
Bulgarian	10 (100%)	4 (53.0%)	3 (53.0%)

## 5. Conclusions and Future Work

This paper has presented the idea of using hierarchical agglomerative clustering on a bilingual parallel corpus. The aim has been to illustrate this technique and provide mathematical measures which can be utilised to quantify the similarity between the clusters in each language. In the paper, we have clustered a bilingual English-Bulgarian corpus. The differences of all the clusters were compared, based on the tree structures. We can conclude that with a smaller number of clusters,  $k$ , all of the clusters from English texts can be mapped into the clusters of Bulgarian texts, with higher degree of purity. In contrast, with a larger number of clusters, fewer clusters from English texts can be mapped into the clusters of Bulgarian texts, and the degree of purity is very low. In addition, the tree structures for both the English and Bulgarian texts are similar when  $k$  is reasonable small (and identical for  $k \leq 10$ ).

A common factor of all the aspects of parallel clustering studied was the importance that may be attached to the higher degree of inflection in Bulgarian. From the very beginning, the significantly lower degree of compression that resulted from stemming Bulgarian was noted. This implies that there were a larger number of Bulgarian words which expressed the same meaning, but which were not identified as such. It is likely that this is one of the factors responsible for decreasing the alignment between the clusters for larger values of  $k$ .

To summarise, here we compared the results of clustering of documents in each of two languages with quite different morphological properties: English, which has a very modest range of inflections, as opposed to Bulgarian with its wealth of verbal, adjectival and nominal word forms. (This difference was additionally emphasised by the fact that the Bulgarian stemmer used produced results which was not entirely consistent in its choice between removing the inflectional or derivational ending.) The clusters produced and the underlying tree structures were compared, and the top 10 most representative terms for each language and cluster listed. As most of the top

terms seemed to represent the same concepts in the two languages, the possibility of restricting the number of terms used to a much smaller than the original set was considered as a way of making the results more robust with respect to differences between languages and speeding up clustering. The results show a slight decline in performance (a drop of up to 10% in the clusters paired and 4.6% lower cluster purity) when reducing the list of English terms, and a catastrophic decline when this is done for Bulgarian in the cases of 100 and 500 terms studied. Knowing how well a cluster tree in one language approximates the one for the same documents in another language could provide guidance for the development of IR approaches where a multilingual corpus of documents is available, but one has access to natural language processing tools only for one of them. In addition, we have shown that when that language is English, one can reduce the number of terms used without a great loss in performance. This could help reduce the search space and achieve a speed up when the term weights used by a clustering algorithm are fine-tuned by machine learning (e.g. a genetic algorithm) to obtain a tree of clusters in one language that more closely matches the tree for the other language, a novel approach we introduce in [10].

## 6. References

- [1] P. Nakov, BulStem: Design and Evaluation of Inflectional Stemmer for Bulgarian. In Proceedings of Workshop on Balkan Language Resources and Tools, Thessaloniki, Greece, November, 2003.
- [2] G. Salton and J. Michael, McGill, Introduction to Modern Information Retrieval, McGraw-Hill Inc., New York, NY, 1986.
- [3] Y Zhao and G Karypis. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10(2):141-168, 2005.
- [4] A. Hotho, S. Staab, and G. Stumme. Text clustering based on background knowledge. Technical Report No. 425, 2003.
- [5] S. Dumais, T. Landauer, and M. Littman, Automatic cross-linguistic information retrieval using latent semantic indexing. In SIGIR '96 – Workshop on Cross-Linguistic Information Retrieval, pp. 16–23, 1996.
- [6] P. Pantel and D. Lin. Document clustering with committees. In *Proc. Of SIGIR'02, Tampere, Finland*, 2002.
- [7] C.J. van Rijsbergen. *Information Retrieval*. Second edition. London: Butterworths, 1979.
- [8] T. de Simone and D. Kazakov. Using WordNet Similarity and Antonymy Relations to Aid Document Retrieval. RANLP 2005, September 2005, Borovets, Bulgaria.
- [9] J. Sedding and D. Kazakov. WordNet-based Text Document Clustering. In *Proc. of the 3<sup>rd</sup> ROMAND workshop*, pp.104-113, Geneva, 2004.
- [10] R. Alfred and D. Kazakov. Aggregating Multiple Instances in Relational Databases Using Semi-Supervised Genetic Algorithm-based Clustering. In the *Proc. of MDAI 2007*, Kitakyushu, Japan, August 2007.

# Obtaining coarse-grained classes of subcategorization patterns for Spanish

Laura Alonso i Alemany  
InCo FaMAF  
UdelaR UNC  
Uruguay Argentina  
alemany@famaf.unc.edu.ar

Irene Castellón Masalles Nevena Tinkova Tincheva  
Departament de Lingüística General  
Facultat de Filologia, UB  
Espanya  
icastellon@ub.edu

## Abstract

In this paper we introduce a method for automatically assigning a subcategorization frame to each verb in a grammar for deep parsing of Spanish. Our final objective is to learn a classifier to assign subcategorization frames to previously unseen verbs for which this information is not available in a hand-made lexicon. To do that, we first need to establish classes of equivalence of verbs according to their subcategorization frames. In this paper we describe how we apply clustering techniques to obtain coarse-grained subcategorization classes from an annotated corpus of Spanish and propose a methodology to evaluate them for the application of assigning subcategorization to previously unseen verbs.

## 1 Introduction

In this paper we introduce a method for automatically assigning subcategorization frames to previously unseen verbs of Spanish, as an aid to automated deep parsing. It is commonly believed that this kind of information can significantly improve the performance of automatic parsers.

Our approach consists in extrapolating the behaviour of known verbs to unknown ones. To do that, we first characterize the behaviour of the verbs annotated in the SENSEM [6] corpus. Then, we apply clustering techniques to generalize the behaviour of these verbs, obtaining coarse-grained classes. These classes group together verbs with similar syntactic behaviour, that is, they represent distinct verbal subcategorizations. Each annotated example in the SENSEM corpus is assigned to one of these classes. From these tagged examples, we learn a classifier that can assign an unseen example to one of the coarse-grained classes obtained from the corpus.

Our final objective is to apply this classifier to previously unseen verbs. In this paper we focus in the first step, inducing subcategorization classes and evaluating them. [1] presents some experiments on applying these classes to automatically annotated examples.

The rest of the paper is organized as follows. In the following Section we describe the annotated corpus we learn from and how examples are transformed to represent subcategorization patterns, and the way we have processed it to generalize the learning data. Then, in Section 2 we present our method to create

coarse-grained equivalence classes of verbs, and the procedures to evaluate them. In Section 3 we describe some of the solutions that we obtained, and justify their adequacy with qualitative linguistic analysis. Finally, in Section 4 we make a quick overview of related work and in Section 5 we draw some conclusions and sketch our future work.

### 1.1 The annotated corpus

Our departure point is SENSEM [6], an annotated corpus of Spanish consisting of 25,000 naturally occurring clauses that are tagged with a verbal sense, and where sentence constituents have been annotated with their morphosyntactic category, syntactic function and semantic role. The most frequent 250 verbs of Spanish are represented, and 1161 senses are distinguished. Each sense in SENSEM has been associated to a subcategorization frame obtained as a synthesis of the structures found in the examples of the corpus.

From that corpus, we characterize verbal senses by the arguments they occur with in annotated examples, regardless of the order they occur with. Each verbal sense is characterized as a vector, whose dimensions are possible realizations of arguments in a given example. The value of each vector in each dimension is the number of times that sense has occurred with that particular realization. We assume that these realizations are an adequate representation of the subcategorization frame of verbs. See Figure 1 for an illustration. The space of dimensions consists of every realization found in annotated corpus. Different transformations of the corpus are carried out, thus configuring different spaces, as explained in the following Section.

### 1.2 Transformations of examples

We do not work with the examples directly, but we perform a compactation of categories [5], in order to reduce the search space and data sparseness.

Then, we consider different subsets of the information available for each example: category of constituents only, category and syntactic function, and finally we also characterize examples with the whole of the available information: category, function and semantic role. Moreover, we also reduce the attribute space by considering only realizations that occur more than 5 or 10 times in the corpus. These different configurations significantly change the size of the attribute

	Dir.Obj.:NP & Subj.:NP	Prep.Obj.:PP & Subj.:NP	Subj.:NP	Dir.Obj.:NP	Prep.Obj.:PP
<i>aclarar_6</i>	26	0	2	2	0
<i>acceder_2</i>	0	70	0	0	5

**Fig. 1:** Illustration of how verbal senses can be characterized in terms of its contexts of occurrence, with a subset of the patterns of realization that are actually found in the corpus.

	realizations		
	all	> 5	> 10
category	240	98	69
category + function	785	213	130
category + function + role	2854	44	317

**Table 1:** Reduction of the attribute space by using different subsets of the information associated to examples and by discarding unfrequent realizations.

space, as can be seen in Table 1, but they also change the detail by which examples are described. Reducing the level of detail is beneficial for those attribute spaces that suffer from data sparseness, as is the case when examples are characterized by category, function and semantic role. However, for cases where examples are poorly characterized, reducing the number of attributes may produce a significant information loss.

Moreover, we have to take into account that some of the information we are using to characterize manually annotated examples will not be available for unseen examples, like for example argumentality, semantic role. To our knowledge, no freely available parser can provide this kind of information reliably for Spanish. However, to induce equivalence classes, we resort to some of the information that is available in the manually annotated corpus, hoping that classes will be better motivated.

## 2 Obtaining equivalence classes

Then, we apply clustering techniques to obtain classes of verbal senses that are similar according to their realizations in the corpus, that is, verbal senses that have similar subcategorization behaviours. We use some of the clustering algorithms provided by Weka [17]. More specifically, we have tried Simple KMeans [10] and Expectation-Maximization clustering (EM) [8].

EM is specially suited for our purposes because the method can find the optimal number of classes for a given dataset, so that the number of classes is not provided by the researcher as an additional bias. For comparison, we also provide some runs with Simple KMeans, but evaluation will show EM is superior.

EM is specially suited for our purposes because the method can find an optimal number of classes for a given dataset, so that the number of classes is not provided by the researcher as an additional bias. In order to find the optimal clustering, the EM method assumes the cluster points follow certain probability distribution, and so it groups points in clusters that are optimal based on that assumption. Since we use Weka, we are assuming a Gaussian distribution, but we did not check whether the data actually follow that distribution. However, compared with Simple KMeans,

<i>hallar_3</i>	<i>encontrar_3</i>	<i>lie_1</i>
<i>acceder_2</i>	<i>entrar_2</i>	<i>go_in_2</i>
<i>crear_1</i>	<i>construir_1</i>	<i>produce_2</i>
<i>valer_1</i>	<i>costar_1</i>	<i>cost_1</i>
<i>contener_1</i>	<i>constituir_1</i>	<i>contain_2</i>

**Table 2:** Verb senses with highly similar subcategorization patterns, which are expected to be assigned to the same cluster in good clustering solutions.

EM results are linguistically more adequate.

As with all unsupervised techniques, evaluation is an unclear issue. Since we have not implemented this method in a final application, we cannot use the kind of indirect evaluation obtained from the impact in application’s performance. However, we have envisaged some methods to help evaluate the adequacy of different clustering solutions.

### 2.1 Qualitative evaluation

In the first place, a manual, qualitative evaluation of clustering solutions was carried out. We studied the **population** of clusters, and clustering solutions that presented classes with only one verb were dispreferred. We also found **pairs of highly similar verb senses**, shown in Table 2, and checked whether they were assigned to the same cluster or to different clusters. Finally, we also inspected the **global content of clusters**, and determined whether the majority of verbs in each cluster actually shared similar subcategorization behaviour (for example, if they were all transitives, ditransitives, etc.).

### 2.2 Quantitative evaluation

As for objective metrics, we developed two quantitative methods for the intrinsic evaluation of clustering solutions. The metric *Overlap* ( $O$ ) measures the amount of subcategorization patterns that are shared by different clusters, weighted by the relative frequency of each pattern in each cluster:

$$O_{A,B} = \frac{\sum_{p \in (P_A \cap P_B)} F_A(p) + F_B(p)}{\sum_{p \in (P_A \cup P_B)} F_A(p) + \sum_{p \in (P_B \cup P_A)} F_B(p)} \quad (1)$$

where

$A, B$  are clusters

$P_A$  is the set of patterns  $p$  in  $A$

$F_A(p)$  is the frequency of occurrence of pattern  $p$  in  $A$

We assume that low overlap between classes indicates that the classes contain verbal senses with different syntactic behaviours, while a higher overlap indicates that verbs in different classes share an important part of their syntactic behaviour, which is not intended

in our case. As can be expected, overlap is conditioned by the number of classes: the more classes, the higher the chances that overlap is low.

In many cases, different verbal senses are distinguished by different subcategorization frames. That is why we provide a measure of how different senses are distributed in clusters, **distribution of senses** ( $SD$ ), calculated as follows:

$$SD = \frac{1}{\#V} \sum_{v \in V} \frac{\#C(v)}{\#S(v)} \quad (2)$$

where

$V$  is the set of verb lemmas  $v$

$S(v)$  is the set of senses of  $v$

$C(v)$  is the set of clusters where at least one sense of  $v$  is found

This indicator must be considered with some caution, since there are some verbal senses that share the same subcategorization frames. In any case, it is useful to complement the overall perspective of the distribution of senses across clusters.

Finally, we considered **classifier accuracy**, that is, the accuracy that automatic classifier could achieve to classify unseen instances in its most adequate cluster. So, we first obtained a clustering solution, then tagged each example in the training corpus with its corresponding cluster, and finally performed ten-fold cross validation of classifiers, which were trained on 90% of the corpus and then evaluated on the 10% that was left, and this procedure was repeated 10 times with the 10 possible different partitions of the corpus. This measure gives us a good idea of the adequacy of a given clustering solution for automatic analysis, and it doesn't present any additional effort, since there is no need to develop an additional evaluation corpus. Classifiers were also trained and evaluated with Weka.

### 3 Evaluation of clustering solutions

In what follows we describe different clustering solutions obtained, using the evaluation methods described in the previous section. Then, in the following section we describe the solution that we found optimal up to this point of experimentation, that is, the solution using as attributes realizations of constituents characterized by category and syntactic function that occur more than 10 times in the corpus.

In general, solutions with the KMeans method provided worse results than solutions with EM, most of all regarding the *population of clusters*, producing many singleton classes. This caused significantly worse overlap indices, since solutions had less "real" classes than their EM counterparts. However, even if a smaller number of real classes was obtained, similar verbs were clustered in different classes more often than in EM solutions. That is why we discarded KMeans and focused in solutions obtained with EM.

If only **morphosyntactic categories** are used to characterize arguments in the examples, and only realizations that occur more than 5 or 10 times are taken into account, EM clustering provides solutions

where the population is well distributed in medium-sized classes. There are very few differences between the solution with realizations that occur more than 5 times and with realizations occurring more than 10.

As can be seen in Figure 3, there is a light degradation of the performance of all classifiers when less attributes are used, which leads us to believe that it is counterproductive to reduce the number of attributes when little attributes are available.

It is difficult to obtain linguistically sound generalizations of the behaviour of the verbs in these classes, because of the high ambiguity of the realizations described by morphosyntactic category only, so these solutions were not considered for further analysis.

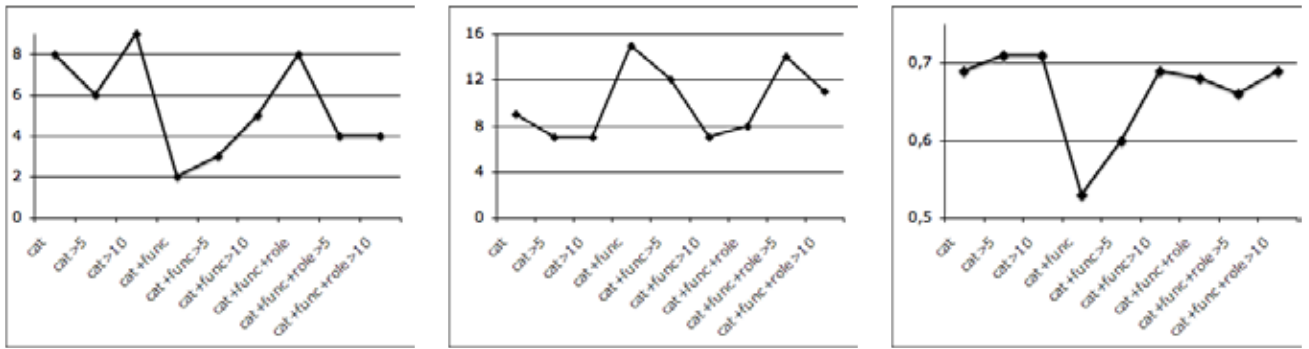
With examples characterized both by the **morphosyntactic category and syntactic function** of arguments, considering all realizations, EM provides an optimum of 2 classes, which is far too coarse-grained for the purpose of enriching a lexicon. Some of the additional measures give very good results for this solution (similar pairs of verbs clustered together, Figure 2, performance of classifiers, Figure 3) precisely because only two classes are distinguished, so in this case these measures lose their significance.

When considering only realizations that occur more than 5 times, a solution in 3 classes is obtained, and a solution with 5 classes is obtained when considering only realizations that occur more than 10 times. As will be seen in Section 3.1, the solution with realizations occurring more than 10 times provides linguistically sound classes and groups together many pairs of similar verbs with respect to the relatively high number of classes distinguished, so this will be the solution chosen for further analysis and development.

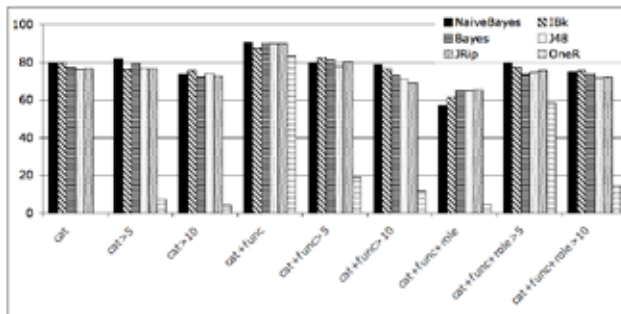
With examples characterized by their **morphosyntactic category, syntactic function and semantic role** of arguments, solutions that take into account realizations occurring more than 5 or 10 times are far better than those using all realizations. It can be seen in Figure 3 that automate classifiers perform better for solutions with realizations that occur more than 5 or 10 times, probably because they suffer less from data sparseness. Also the number and population of clusters is more understandable for these solutions, and pairs of similar verbs are grouped together more often (see Figure 2).

In these solutions we find four classes. The biggest one is populated by verbs with virtually any pattern of constituents but with a clear predominance of intransitive diatheses, explained because of the ellision of some aof the arguments in the actual realizations in corpus, together with purely intransitive verbs. A second class is populated by strongly transitive verbs, with few intrasitive diatheses, and the two smallest classes are populated by verbs with a very marked semantic roles (*origin*, *goal*), also with few intransitives.

These classes were not considered for further analysis because the predominant phenomena (role of intransitive diatheses, transitives, etc.) had already been found in solutions with category and syntactic function only, which is precisely the information that will be available in automatic analysis, so solutions with role were momentarily left aside.



**Fig. 2:** Some objective metrics for comparing clustering solutions: Number of clusters, number of similar verb pairs in the same cluster and distinguishability of senses.



**Fig. 3:** Objective metrics for comparing clustering solutions: classifier accuracy.

### 3.1 Analysis of an interesting clustering solution

We chose for further analysis the clustering solution with the EM algorithm provided the most adequate results for our purposes. Five classes of verb senses are distinguished, according to their subcategorization patterns:

1. the biggest class, populated with 477 verb senses that alternate between **transitive and intransitive** realizations, and some cases of prepositional realizations.
2. a class with 163 senses with predominantly **prepositional and intransitive** realizations. Intransitive realizations can be explained by the omission of the prepositional argument.
3. a class with 103 senses where realizations alternate between **ditransitives, transitives and intransitives**. Realizations with less arguments can mostly be explained by the omission of one or two of the arguments.
4. a class with 68 senses, populated by senses very similar to those in 3.
5. the smallest class, with 63 senses that occur with mostly **prepositional** arguments that alternate with intransitives and some attributes.

It can be seen that these classes contain heterogeneous verbal senses. Therefore, we performed some

further clustering within each of these classes to obtain finer-grained distinctions, as described in [5]. We found that at the level of subclasses, it is possible to associate clusters with classical subcategorization frames like *NounPhrase Verb (NounPhrase)* and the like. Therefore, the use of hierarchical techniques seems promising to obtain the granularity of subcategorization information we are looking for. The optimal way to do that is by applying a hierarchical clustering algorithm, as [16] and [9], but in this first approach we just performed some further EM clustering within each of the classes, in order to inspect their population better. Hierarchical clustering is left for future work.

## 4 Related Work

It is commonly assumed that subcategorization frames can significantly improve the performance of automatic syntactic analyzers of natural language. However, the manual construction of lexica with subcategorization information is very costly. That's why there have been several approaches to acquiring such information automatically. A good review of previous work can be found in [15]. Most of the work in subcategorization acquisition has been done for English. Only a few works can be found for other languages, particularly for Spanish we know of [7, 9]. Here we highlight the main differences of our work with respect to some well-established previous work.

In this work we focus in finding equivalence classes, working upon subcategorization patterns that have already been established in the SENSEM corpus. A big difference is found in the information provided by the subcategorization patterns of verbs, which is also dependent on the corpus subcategorizations are learnt from. In some cases the corpus is analyzed automatically [14] or not annotated at all [3], in many other cases subcategorizations are acquired from a manually annotated corpus [12, 4]. Different kinds of annotation make it possible to distinguish verbal senses [11] or else it is necessary to work at the level of verb lemma [3, 4], leaving ambiguous verbs as such. Since SENSEM provides information about verbal senses, our unit is not the verbal lemma, but the verbal sense.

When working with examples from corpus, it is necessary to discriminate which constituent patterns are determined of the verb's subcategorization behaviour,

and which are not verb-dependent, that is, which constituents are *arguments* and which are *adjuncts*, respectively. The SENSEM corpus provides information about constituents that are arguments in each example, so adjuncts can be discarded to model examples.

With respect to the method for establishing equivalence classes, different approaches have been taken. [2] uses a confidence interval for indicative cues to classify between two classes of verbs, [13] use decision trees and [16] and [9] use a hierarchical clustering algorithm. In this work we use unsupervised clustering using the EM algorithm for clustering. However, as will be seen in the analysis, it seems more adequate to employ a hierarchical clustering algorithm, which we will do in future work.

## 5 Conclusions and Future Work

We have presented a procedure to obtain coarse-grained subcategorization classes to assign a subcategorization frame to each verb in a grammar for deep parsing of Spanish. These classes allow to extrapolate the behaviour of known verbs to unknown verbs, thus dramatically increasing the coverage of this kind of information in a grammar.

We have used the information provided in an annotated corpus to characterize verbs, then applied clustering techniques to find coarse-grained classes that are linguistically well motivated and can be automatically recognized with a small error rate. We have developed various methods for evaluating diverse clustering solutions, both qualitatively and quantitatively.

One important line of future work is the use of hierarchical clustering techniques to obtain subcategorization classes at a level of granularity that is more useful for grammatical description. Also as future work, we will use these classes and the classifier learned from the corpus to assign a subcategorization class to previously unseen verbs. We will have to deal with the problem of verb sense disambiguation, and assess how much sense disambiguation contributes to determining the adequate subcategorization frame, and viceversa.

## 6 Acknowledgements

This research has been partially funded by project KNOW (TIN2006-1549-C03-02) from the Spanish Ministry of Education and Science, a Beatriu de Pinós Postdoctoral Fellowship granted by the Generalitat de Catalunya to Laura Alonso and by a Postgraduate Scholarship FI-IQUC also granted by the Generalitat de Catalunya to Nevena Tinkova, with file number 2004FI-IQUC1/00084.

## References

- [1] L. Alonso Alemany, I. Castellón, and N. Tinkova Tincheva. Inducción de clases de comportamiento verbal a partir del corpus SENSEM. In *XXIII Congreso Anual de la Sociedad Española para el Procesamiento del Lenguaje Natural, XXIII SEPLN*, 2007.
- [2] M. R. Brent. Automatic acquisition of subcategorization frames from untagged text. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 209–214, Morristown, NJ, USA, 1991. Association for Computational Linguistics.
- [3] M. R. Brent. From grammar to lexicon: Unsupervised learning of lexical syntax. *Computational Linguistics*, 19:243–262, 1993.
- [4] T. Briscoe and J. Carroll. Automatic extraction of subcategorization from corpora. In *Proceedings of the 35th annual meeting on Association for Computational Linguistics*, 1997.
- [5] I. Castellón, L. Alonso Alemany, and N. Tinkova Tincheva. A procedure to automatically enrich verbal lexica with subcategorization frames. In *Proceedings of the Argentine Symposium on Artificial Intelligence, ASAI'07*, 2007.
- [6] I. Castellón, A. Fernández-Montraveta, G. Vázquez, L. Alonso, and J. Capilla. The SENSEM corpus: a corpus annotated at the syntactic and semantic level. In *5th International Conference on Language Resources and Evaluation (LREC 2006)*, 2006.
- [7] G. Chrupala. Acquiring verb subcategorization from spanish corpora. Master's thesis, Universitat de Barcelona, 2003.
- [8] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39, 1977.
- [9] E. Esteve Ferrer. Towards a semantic classification of spanish verbs based on subcategorisation information. In *ACL'04*, 2004.
- [10] J. A. Hartigan and M. A. Wong. Algorithm as136: a k-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.
- [11] A. Korhonen. Assigning verbs to semantic classes via wordnet. In *Proceedings of the COLING Workshop on Building and Using Semantic Networks*, Taipei, 2003.
- [12] A. Korhonen and J. Preiss. Improving subcategorization acquisition using word sense disambiguation. In *Proceedings of ACL*, pages 48–55, 2003.
- [13] P. Merlo and S. Stevenson. Automatic verb classification based on statistical distributions of argument structure. *Computational Linguistics*, 27(3):373–408, 2001.
- [14] A. Sarkar and D. Zeman. Automatic extraction of subcategorization frames for czech. In *COLING'2000*, 2000.
- [15] S. Schulte im Walde. The Induction of Verb Frames and Verb Classes from Corpora. In A. Lüdeling and M. Kytö, editors, *Corpus Linguistics. An International Handbook.*, chapter 61. Mouton de Gruyter. To appear.
- [16] S. Schulte im Walde. Clustering verbs semantically according to their alternation behaviour. In *COLING'00*, pages 747–753, 2000.
- [17] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

# Practical application of one-pass Viterbi algorithm in tokenization and part-of-speech tagging

Miguel A. Molinero Álvarez  
Universidad de La Coruña  
Facultad de Informática  
Campus de Elviña S/N  
15071 - A Coruña, Spain  
*mmolinero@udc.es*

Juan Otero Pombo  
Universidad de Vigo  
E.S. de Ingeniería Informática  
Campus das Lagoas S/N  
32004 - Ourense, Spain  
*jop@uvigo.es*

Fco. Mario Barcala Rodríguez  
Centro Ramón Piñeiro  
Ctra. Santiago-Noia Km. 3. A Barcia  
15896 - Santiago de Compostela, Spain  
*fbarcala@cirp.es*

Jorge Graña Gil  
Universidad de La Coruña  
Facultad de Informática  
Campus de Elviña S/N  
15071 - A Coruña, Spain  
*grana@udc.es*

## Abstract

Sentence word segmentation and Part-Of-Speech (POS) tagging are common pre-processing tasks for many Natural Language Processing (NLP) applications. This paper presents a practical application for POS tagging and segmentation disambiguation using an extension of the one-pass Viterbi algorithm called Viterbi-N. We introduce the internals of the developed system, which is based on lattices and a stochastic model built using second order Hidden Markov Models (HMMs). Also, we present the results of an evaluation process and the analysis of the error cases. The results achieved suggest that the Viterbi-N algorithm applied on lattices allows POS tagging and segmentation disambiguation to be accomplished in a common process. Although the tests were done for the Galician language, the solution proposed could be easily exported to other languages.

## Keywords

Part-Of-Speech tagging, tokenization, segmentation disambiguation, Hidden Markov Models, lattices.

## 1 Introduction

Current Part-Of-Speech (POS) taggers assume that their input is already correctly tokenized. This means that every token in the input is an individual linguistic component suitable for being tagged with a single POS tag. The tokenization task tends to be relatively simple, since in most cases each word corresponds to one linguistic token. However, there are cases where this segmentation can be more complex. On one hand, there are contractions and verbal forms with enclitic pronouns, where the same word contains information

about two or more linguistic components which have to be split into individual tokens. On the other, there are idioms, where several words act together as one linguistic component, and must be joined to form a unique compound token.

Segmentation ambiguities arise when one or more words can be segmented into linguistic tokens in more than one way. This kind of phenomenon is quite common in languages with a rich morphology, such as Spanish or Galician. To deal with such ambiguities, several works [8] [9] use artificial tags to be assigned to compound tokens or to tokens which are part of only one linguistic reality. However, they postpone the solution of these segmentation tasks to later phases of Natural Language Processing (NLP), which in most cases are not documented.

Our approach lies in using the one-pass Viterbi algorithm extension [6] over second order Hidden Markov Models (HMMs) to carry out the segmentation just at the moment of assigning POS tags. Segmentation ambiguities are detected by a morphological preprocessor using lexicons and provided as input to the algorithm.

This way, POS tagging and segmentation disambiguation are accomplished in one unique process using a lattice structure. Lattices will allow us to represent every possible segmentation and to manage all the computations needed for the classic Viterbi algorithm at the same time, as we will explain later.

## 2 Segmentation Issues

As we have indicated earlier, many POS tagging environments simply ignore segmentation issues, leaving them to be solved in later steps. For example, a common approach is to use agglutinations of tags<sup>1</sup> which are assigned to contractions and enclitic

<sup>1</sup> To simplify, in this work, we use Adj for adjective, Adv for adverb, C for conjunction, Det for determiner, P for



forms. A contraction formed by a preposition and a determiner could be tagged with a compound tag like *P+Det*, instead of being split into one token tagged with *P* and another one tagged with *Det*. Given that many NLP applications need to know the linguistic information of each word component, when using this approach the contraction will need to be processed in a later step in order to extract its linguistic information. Moreover, it causes an unnecessary growth of the tagset, with its negative consequences (sparse data, larger training corpus needed, etc.) [4].

In comparison, we detect tokenization ambiguities just before the POS tagging phase with a morphological preprocessor [7]. This is done using external lexicons and some segmentation rules for verbal forms with enclitic pronouns. If a word can make sense with different segmentations, the morphological preprocessor provides every alternative to the POS tagger. Then, the POS tagger will choose the best one.



Fig. 1: Ambiguous segmentations of ‘polo’ and ‘sin embargo’ for Galician and Spanish languages respectively.

Contractions, verbal forms with enclitic pronouns, idioms and proper nouns are the categories which are able to generate segmentation ambiguities. For example, as we can see in figure 1, the Galician word ‘polo’ could be treated as a noun (chicken), as a contraction of the preposition ‘por’ and the determiner ‘o’ (by the) or even as a verbal form ‘pos’ with the enclitic pronoun ‘o’ (put it). On the other hand, a sequence of words like the Spanish expression ‘sin embargo’ could be joined together and tagged as a conjunction (however) or it could be tagged individually as a preposition and a noun (without seizure).

Once a sentence has been preprocessed and segmentation ambiguities detected, a tagging model is used to assign the correct POS tag to each of the tokens. The model is built as a second order Hidden Markov Model (HMM) and its parameters are estimated from a training corpus using linear interpolation of uni-, bi- and trigrams as our smoothing technique [5].

### 3 Lattices

In the context of POS tagging with HMMs, the classic version of the Viterbi algorithm is applied on trellises [3], where the first row contains the words of the sentence to be tagged, and the candidate tags appear in columns below the words. However,

preposition, Pro for pronoun, N for noun, V for verb, Id for idiom and Q for punctuation mark.

this structure does not allow the representation of ambiguous segmentations.

A practical solution lies in using lattices to represent sentences. Figure 2 shows a Galician language sentence which contains several types of ambiguous segmentations: ‘*Non poden verse a causa de certo individuo*’ (they cannot meet each other because of a certain person). The gaps between the words are enumerated and an arc can span one or more words. Such an arc is labelled with the words spanned and their corresponding POS tag. For example, gap 3 marks the beginning of an ambiguous segmentation for the word ‘verse’. It could be segmented into verb ‘ver’ (to meet) and reflexive enclitic pronoun ‘se’, or as verb ‘verse’ (it may deal with). In gap 5, the idiom ‘a causa de’ (because of) could be also segmented into several different tokens and the same in gap 7 for ‘de certo’ (certainly).

Although there are 40 possible paths in this sentence, only the one formed by the arcs drawn in the upper part of the lattice shows the correct segmentation. Each arc represents a token, so the correct segmentation is seven tokens long, while the longest possible segmentation of this sentence is nine tokens long.

Therefore, lattices will allow us to represent all the information about ambiguous segmentations. Now we will see how an extension of the Viterbi algorithm can use them to tag sentences without repeating computations for each path.

### 4 Viterbi-N: the one-pass Viterbi algorithm with normalization

The Viterbi algorithm [10] is a dynamic programming algorithm for finding the most likely sequence of hidden states (called the Viterbi path) that explains a sequence of observations for a given stochastic model. In the context of POS tagging, we are looking for the most likely sequence of tags that explains a sequence of words in a sentence. In order to do so, a trellis is built from the sentence to be tagged. For each state (tag) in that trellis the cumulative probability for all paths reaching that state must be computed but, given that such paths in trellises have the same length, it is only necessary to store the cumulative probability of the best one. At the end, the most likely sequence of tags for the sentence is obtained by comparing cumulative probabilities of final states and going backwards.

On the contrary, the Viterbi-N algorithm is applied on lattices [6], so it is possible to reach one state coming from paths of different length. Thus, for each state in the lattice, it will be necessary to store as many cumulative probabilities as there are different lengths of path reaching that state. Therefore, let  $\Delta_{t,t',l(q)}$  be an accumulator which collect the maximum probability of state  $q$  covering words from position  $t$  to  $t'$ , and with length  $l$ ,  $l$  being the number of states from first state to state  $t'$ .

Only accumulators with the same length would be directly comparable, because of the different number of factors involved in their computation.

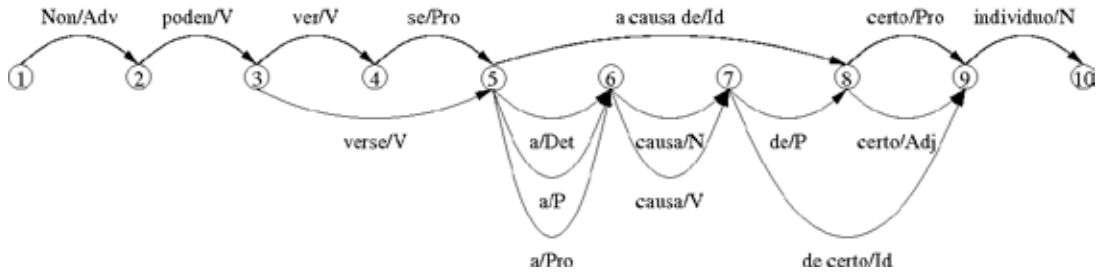


Fig. 2: Ambiguous segmentations represented on a lattice.

As accumulators are computed by products of probabilities, longer segmentations are penalized by the higher number of factors, making them less likely than shorter ones. In practice, this means that alternatives which imply the joining of words would be chosen more often than others which imply segmentation into several words. To solve this, a normalization step must be accomplished in order to compare segmentation paths of different lengths. Moreover, it must be noted that the algorithm works using only one lattice and performing only one pass of the Viterbi algorithm.

Figure 3 shows how the algorithm is applied on the Galician language sentence ‘*Non poden verse.*’ (they cannot meet each other.). Lattices can be implemented as graphs in which each node is a probability accumulator associated to one linguistic token and one POS tag. In the figure we can see the accumulators needed to tag and disambiguate this sentence. Such accumulators are written with the format  $\Delta(t, t', l, q)$ , where  $t$  and  $t'$  are the instants where the current token starts and ends,  $l$  is the number of tokens from the beginning of the sentence until the current token and  $q$  is the associated POS tag. As there are two possible segmentation paths reaching the last token of the sentence, it has two accumulators, with lengths 5 and 4. The algorithm will normalize both accumulators by their lengths and choose the best one. Then, the sequence of tags compounding the best segmentation path can be obtained by going backwards in the lattice.

The equations of the classic Viterbi algorithm can be adapted to process lattices [2]. Assuming the use of logarithmic probabilities to avoid problems of precision with factors less than 1, we replace products by sums and adapt the Viterbi-N algorithm’s equations as follows:

Let’s use  $\delta_{i,j}(q)$  to denote the probability of the derivation emitted by state  $q$  having a terminal yield that spans positions  $i$  to  $j$ .

- Initialization:

$$\Delta_{0,t,1}(q) = P(q/q_s) + \delta_{0,t}(q)$$

- Recursion:

$$\Delta_{t,t',l}(q) = \max_{(t'',t',q') \in \text{Lattice}} \Delta_{t'',t',l-1}(q') + P(q/q') + \delta_{t,t'}(q) \quad (1)$$

for  $1 \leq t < T$

- Termination:

$$\max_{Q \in \mathcal{Q}^*} P(Q, \text{Lattice}) = \max_l \frac{\max_{(t,T,q) \in \text{Lattice}} \Delta_{t,T,l}(q) + P(q_e/q)}{l}$$

Additionally, it is also necessary to keep track of the elements in the lattice that maximized each  $\Delta_{t,t',l}(q)$ . When reaching time  $T$ , we get the length of the best path in the lattice:

$$L = \arg \max_l \frac{\max_{(t,T,q) \in \text{Lattice}} \Delta_{t,T,l}(q) + P(q_e/q)}{l}$$

Next, we get the best last element of all paths of length  $L$  in the lattice:

$$(t_1^m, T, q_1^m) = \arg \max_{(t,T,q) \in \text{Lattice}} \Delta_{t,T,L}(q) + P(q_e/q)$$

Setting  $t_0^m = T$ , we collect the arguments  $(t'', t, q')$  Lattice that maximized equation (1) by going backwards in time:

$$(t_{i+1}^m, t_i^m, q_{i+1}^m) = \arg \max_{(t'',t_i^m,q') \in \text{Lattice}} \Delta_{t'',t_i^m,L-i}(q') + P(q_i^m/q') + \delta_{t_i^m,t_{i-1}^m}(q_i^m)$$

for  $i = 1$ , until we reach  $t_k^m = 0$ . Now,  $q_1^m \dots q_k^m$  is the best sequence of phrase hypothesis (read backwards).

To sum up, the normalized probabilities calculated by the Viterbi-N are directly compared and the highest one is chosen to build the best segmentation path for current sentence.

## 5 Defining alternatives

The input for the algorithm is based on the input format of classic taggers [3]. That is, one word per line, optionally followed by its candidate POS tags. However, this classic representation does not allow the inclusion of segmentation alternatives.

We have decided to use XML-like tags for the definition of such alternatives. An alternative structure starts with a line containing only the tag `<alternatives>`. Then, each segmentation alternative starts with a line containing only the tag `<alternative>` and ends with the tag `</alternative>`. Between those tags, the segmentation alternatives are presented using the classic format. Finally, the alternative structure ends with

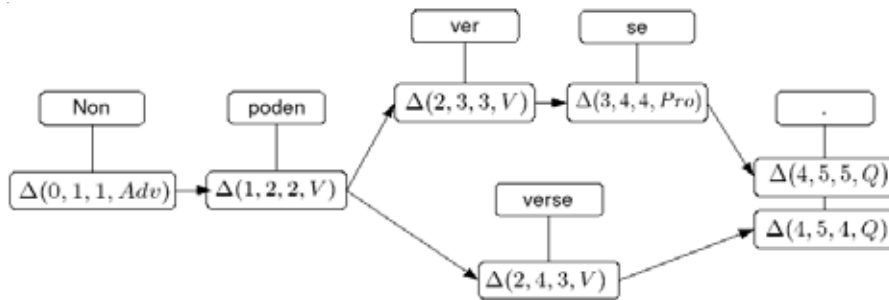


Fig. 3: Viterbi-N algorithm applied on a lattice

the tag `<\alternatives>`. For example, the alternative structure for the galician word ‘*polo*’ (see figure 1) would be as follows:

```

<alternatives>
  <alternative>
    polo      N
  </alternative>
  <alternative>
    por       P
    o         Det
  </alternative>
  <alternative>
    pos       V
    o         Pro
  </alternative>
</alternatives>

```

Alternative structures may appear at any place inside a sentence. Their construction is a task that should be accomplished by the previously mentioned morphological preprocessor<sup>2</sup>. It should build every alternative and assign candidate tags in each branch. It must be noted that some branches may already have already the correct POS tags (e.g. contractions have usually unique tags when they are segmented), providing valuable information that can be used to choose the correct alternative.

## 6 Evaluation

We have performed three experiments using Galician language texts obtained from the “Reference Corpus from Present-day Galician Language” project [1] to test the accuracy of our approach. We have implemented the Viterbi-N algorithm over a lattice structure, and fed it with the input described in section 5. The main goal of these tests is to establish both how accurate the segmentation disambiguation process is, and how dependent it is from the trained model.

We worked with a manually tagged corpus, containing 115754 words and organized in 3920 sentences. In this corpus, our morphological preprocessor detects 1967 sentences with at least

<sup>2</sup> Details about how the preprocessor accomplishes this lie outside the scope of this work [7].

one ambiguous segmentation. The whole number of segmentation ambiguities in the corpus is 3037.

Our first experiment (E1), only to figure out the possibilities of our system, consisted in tagging ambiguous sentences from the training corpus. A high degree of accuracy would be expected in this experiment, since there are no unknown words in the text. We performed this experiment on a set of 434 sentences randomly extracted from the training corpus, with the only requisite of containing at least one ambiguous segmentation. This set contained 702 cases of ambiguous segmentations.

For the second experiment (E2), we randomly extracted 185 sentences, again containing at least one ambiguous segmentation. These sentences are formed by 6073 words, and were used as a testing corpus. The remaining 109681 words were used as a training corpus.

As a high number of segmentation ambiguities remained undetected in experiment E2, we decided to carry out a third experiment avoiding this problem. Thus, in the third experiment (E3), we again tagged the extracted testing corpus, but with an improved version of the morphological preprocessor, which is able to detect new ambiguous segmentations, not detected in experiment E2.

Although the size of the testing corpus could seem a little small, we have chosen such a size for three reasons. First, the Galician language is a less-resourced language, so the amount of tagged text available was small. Second, it is difficult to align manual and automatic tagged text to compare results when alternative segmentation options are given. Therefore, with a small corpus errors could be easily detected and checked. Third, we wanted to make a detailed study of the error cases in order to determine where they come from, and how to avoid them.

Table 1 shows the experimental results. The first column shows the number of ambiguous segmentations detected by the preprocessor. The second column shows the number of segmentations where the correct segmentation was chosen. The third shows the number of ambiguous segmentations not detected by the preprocessor. The next column shows the percentage accuracy of the segmentation disambiguation taking cases of the third column as errors, and the last one shows the accuracy of the segmentation disambiguation process when the cases of the third

	CASES	GOOD CHOICE	NO OPTION GIVEN	TOTAL ACCURACY	REAL ACCURACY
E1	702	662	8	94.30%	95.39%
E2	309	241	41	77.99%	89.92%
E3	309	255	5	82.52%	83.88%

**Table 1:** Test results for experiments E1, E2 and E3.

column are not treated as errors.

As expected, experiment E1 produced very good results. Only 8 cases of ambiguous segmentation were not detected by the morphological preprocessor. We cannot consider these cases as real errors, since no alternatives are given to the algorithm and they could be detected just by upgrading the lexicons used by the morphological preprocessor. The real accuracy achieved in this experiment is over 95%.

Experiment E2 is a more natural one, because unknown words appear in the testing corpus. As can be seen, there is a high number of ambiguous segmentations not given by the preprocessor. This fact has a simple explanation: idioms which are in the corpus but not included in the morphological preprocessor lexicons, unknown enclitic forms, etc. A human linguist is able to detect them, but our preprocessor simply does not have the necessary information to do so. Once again, if we do not take these cases as errors, the accuracy is 89.92%. This accuracy descends to 77.99% if we treat them as errors.

For experiment E3, we added to the lexicons of the morphological preprocessor many of the unknown cases of experiment E2. In fact, all but those that do not meet the usual criteria for inclusion in a lexicon (Latin or foreign idioms, etc.). Now, we have to keep in mind that these new added cases are not in the trained model, so some branches of an alternative segmentation could be an unknown word. In these conditions, which could be considered as the worst case for our system, we achieved 83.88% accuracy. We judge this value as a real approximation to the overall accuracy of the system in segmentation disambiguation and we adopt it as a baseline for future developments.

Although the results obtained were not outstanding, we believe it is a very promising technique. We must note that the training corpus used is very small for the size of the tagset<sup>3</sup> and at the moment we have no more corpora available. In fact the training corpus is still under development and the one used here has a lack of coherence. So we think most errors come from the training corpus and not from the technique itself. Unfortunately, we have no other approaches to compare with, or we do not know any other work which explains and tests the segmentation disambiguation for Western European languages.

Concerning the pure POS tagging results, they are subordinate to the success of the tokenization task. Taking each segmentation error as one POS tagging error, we achieved 87.14% accuracy in experiment E3. We have checked that this poor result comes once again from the poverty of the training corpus.

<sup>3</sup> The tagset used has near 300 different tags. It can be consulted in <http://corpus.cirp.es/xiada/etiquestario.html>

## 6.1 Error analysis

In a detailed analysis of the errors, we became aware of some interesting points. First, we have detected two different kinds of error, which we could classify as soft and hard:

- Soft errors are those from idioms. Such errors arise when several words are not joined into an idiom, but are correctly tagged individually, or when they are joined into an idiom when they should not be. These kinds of error choose segmentations that commonly make sense with the rest of the sentence. In some cases it is not even clear for linguists when some idioms should be built, so the information of the model is limited for this purpose.
- Hard errors are those from contractions, enclitic forms, etc. If the correct segmentation is not chosen in such cases, the error is very hard, since it could even start a cascade error for the rest of the sentence. As a result of this kind of error, the tagged sentence makes no sense and it could be considered a whole tagging error. For example, in the Galician sentence ‘*o polo comeu millo*’ (the chicken ate corn), if *polo* is segmented as a contraction, we will have ‘*o por o comeu millo*’ (the by the ate corn), a completely meaningless sentence.

Table 2 shows the rates of soft and hard errors detected in experiment E3. As can be seen, we achieved 63.56% accuracy for idioms. Further analysis of the training corpus revealed that it was very poor in idioms. Linguists who tagged it, usually chose not to join several words to make an idiom, even when it was possible. Therefore, the training corpus had very little information about idioms.

However, we achieved outstanding results for the rest of segmentations. It is worth noting that every segmentation ambiguity was detected for such categories, and only two among 175 cases were wrongly solved, giving us 98.85% accuracy.

Moreover, we realized that most soft errors come from the fact that some idioms contain very common words. This means that the alternative branches where words are not joined have a very high probability according to the training model. For example, the preposition ‘*a*’ (to, at, on) is one of the most common words in the model. It has a very high occurrence probability and it is also very common to find it inside idioms.

The real problem is that idioms themselves appear little in the training corpus. So the trained model will give more weight to the segmented branch over the joined branch when the word ‘*a*’ appears in the idiom.

	CASES	GOOD CHOICE	NO OPTION GIVEN	TOTAL ACCURACY	REAL ACCURACY
Soft errors	134	82	5	61.19%	63.56%
Hard errors	175	173	0	98.85%	98.85%

**Table 2:** Test results for experiment E3 classified by kind of errors.

This happens with several very common words as ‘*que*’ (that, which, than), ‘*de*’ (of, from), etc. A possible solution would be to upgrade the size of the training corpus.

However, almost these errors could still be solved with morphosyntactic information, leading us to think that it is possible to upgrade the accuracy of the system with some rules. This approach would be a less expensive solution than increasing the training corpus. Such rules may act in the lattice structure itself, pruning segmentation branches that prove impossible for the current context. From our point of view, a small set of rules could greatly improve the accuracy of the system for idioms and bring it near to 100% for other categories. In this case we would have a hybrid system with a very high degree of accuracy in the tokenization task.

## 7 Conclusions and future work

The tokenization task is usually simplified, leaving segmentation ambiguities to be solved in later steps of the NLP applications. In our case, we chose to accomplish segmentation tasks in the POS tagging phase, making it more complex, but the benefits will affect all successive applications.

In this paper, we have presented a practical application of the Viterbi-N algorithm for segmentation disambiguation and POS tagging. Segmentation ambiguities arise when one or several words can be segmented into linguistic tokens in more than one way. These are the cases of some contractions, verbal forms with enclitic pronouns, idioms, etc. The underlying idea for this combination of tasks is that POS categories provide a lot of information that can be used when choosing the correct alternative for an ambiguous segmentation. In the end, we have developed a POS tagger able not only to decide the tag to be assigned to every token, but also to choose the best sequence of tokens from a set of possible segmentation paths as well. Since the approach is purely stochastic, the technique could be easily exported to other languages.

Another advantage of the approach used, is that segmentation disambiguation could be considered a costless add-on for the POS tagging environment. If the training corpus is built carrying out the segmentations, they will be included in the learned model automatically.

The developed system was tested in the context of the Galician language, which has a very rich morphology, that is, the worst scenario for our system, and quite good results were achieved in the segmentation disambiguation task. We believe that they will be improved when the training corpus will be mature enough.

Indeed, another way to improve results is to use

rules based on linguistic information which could prune some erroneous segmentation candidates. This would be particularly useful when the training corpus is of small size or low quality.

## Acknowledgments

This paper arises out of a project developed at Centro Ramón Piñeiro para a Investigación en Humanidades and is partially supported by Ministerio de Educación y Ciencia (TIN2004-07246-C03-1), Xunta de Galicia (PGIDIT05PXIC30501PN) and “Galician Network for Language Processing and Information Retrieval” 2006-2009.

## References

- [1] <http://corpus.cirp.es/corgaxml>. *Reference Corpus from Present-day Galician Language (CORGA)*.
- [2] T. Brants. Cascaded markov models. *Proceedings of the Ninth Conference of the European chapter of the Association for Computational Linguistics (EACL-99)*, 1999.
- [3] T. Brants. Tnt – a statistical part-of-speech tagger. *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP 2000)*, 2000.
- [4] D. Elworthy. Tagset design and inflected languages. *Proceedings of EACL SIGDAT workshop From Texts to Tags: Issues in Multilingual Language Analysis*, 1995.
- [5] J. Graña. *Técnicas de Análisis Sintáctico Robusto para la Etiquetación del Lenguaje Natural*. Doctoral thesis, Universidad de La Coruña, Spain, 2000.
- [6] J. Graña, M. Alonso, and M. Vilares. A common solution for tokenization and part-of speech tagging: One-pass viterbi algorithm vs. iterative approaches. *Petr Sojka, Ivan Kopecek and Karel Pala (eds.), Text, Speech and Dialogue, volume 2448 of Lecture Notes in Artificial Intelligence, pp. 3-10, Springer-Verlag, 2002*.
- [7] J. Graña, F. M. Barcala, and J. Vilares. Formal methods of tokenization for part-of-speech tagging. *Computational Linguistics and Intelligent Text Processing, volume 2276 of Lecture Notes in Computer Science, pp. 240-249, Springer-Verlag, 2002*.
- [8] J. L. A. Moreno, A. Álvarez Lugrís, and X. G. Guinovart. Aplicación do etiquetario morfosintáctico do sli ó corpus de traducións tectra. *Viceversa: Revista Galega de Traducción, 7/8, pp. 189-212, 2003*.
- [9] M. Vilares Ferro, A. Valderruten Vidal, J. Graña Gil, and M. Alonso Pardo. Une approche formelle pour la génération d’analyseurs de langages naturels. In P. Blache, editor, *Actes de la Seconde Conférence Annuelle sur le Traitement Automatique du Langage Naturel*, Marseille, France, June 1995.
- [10] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Information Theory, vol. IT-13, 1967*.

# Designing a Valence Dictionary for Romanian

Ana-Maria Barbu  
Institute of Linguistics  
Calea 13 Septembrie nr.13, Bucharest  
anabarbu@unibuc.ro

Emil Ionescu  
University of Bucharest  
Str. Edgar Quinet nr. 3-5, Bucharest  
emilionescu@unibuc.ro

## Abstract

This paper presents the first step in building a Romanian Valence Dictionary for NLP purposes. Since we can not use Romanian work as an appropriate starting point, the first step was to define the criteria used for describing the valence information and the entry structure.

## Keywords

Valence dictionary, subcategorisation frames, verbs, arguments, Romanian.

## 1. Introduction

This paper aims at presenting some important theoretical and methodological lines for designing a valence dictionary for Romanian (hereafter **RVD** ‘Romanian Valence Dictionary’)<sup>1</sup>. Valences are sets of elements required by a predicate. Syntactically, they are *subcategorisation frames* and their elements are called *complements*, while semantically, they are represented by *argument structures* with *arguments*. Predicates are usually expressed by verbs, but can also be nouns, adjectives etc. RVD includes only verbs; one or more valences are described for each item. We tried to show what information can be relevant to describe such verbal valences.

Valence dictionaries are useful in many and important domains of NLP. We mention here only some of them: deep-parsing used in Machine-Translation and Question-Answering systems; shallow parsing used, for instance, in Information Retrieval or Extraction; and part-of-speech disambiguation used for corpus annotation and speech recognition. Actually, we intend to use RVD especially in Word Sense Disambiguation (WSD), which is mainly a semantic-oriented task, where the relationship between meanings and argument structures can be fully exploited.

Such valence lexica are either paper-based (for German, Polish, Slovak, Bulgarian, Russian) or in electronic format (for English, German, Japanese, Bulgarian, French and Dutch, Czech, Polish, Russian, Armenian, Turkish, Arabic, Chinese) (for a presentation of these projects see [17]). Creating valence dictionaries implies different procedures from one project to another: some are created entirely manually. This is the case of the valency lexicon of Czech verbs VALLEX [18] based on Functional Generative

Description [16], the Polish syntactico-semantic lexicon [11] and the Bulgarian valence dictionary in electronic format [1] which uses Head-driven Phrase Structure Grammar (HPSG) [9], [10] to represent grammatical knowledge/information.

Attempts to extract verbal subcategorisation frames from corpora using machine learning techniques have been done recently: [7], [14], [8] most of them using syntactically parsed corpora.

We have chosen to build RVD manually to ensure the necessary quality of such an important resource. First of all we had to establish criteria concerning the information encoded in valence descriptions.

Section 2 presents the previous Romanian works on this topic, namely two printed dictionaries of so-called verbal constructions or verb syntax. For comparison, an example of a RVD entry is given in section 3. The information structure of RVD entries is largely explained in the main section of the paper (4), together with the criteria applied. The fifth section sketches our future lines of research.

## 2. Romanian paper-based dictionaries

No electronic valence dictionary for Romanian has been designed yet. There are only two paper-based dictionaries with valence descriptions for verbs:

1. *Verbul românesc. Dicționar sintactic*, (‘Romanian Verb. Syntactic Dictionary’) by Ionescu and Steriu [3]. It contains 1088 verbs.

2. *Dicționar de construcții verbale român-francez-italian-englez*, (‘Dictionary of Verbal Constructions Romanian-French-Italian-English’) by Drăghicescu [2]. It comprises about 500 verbs. Each entry presents different structures in which the verb can occur, with examples in Romanian, French, Italian and English.

To illustrate how an entry in these dictionaries looks like, we stopped at the verb A DISTRUGE ‘to destroy’.

Ionescu and Steriu (1999) provide the following information for the verb A DISTRUGE (to destroy).

(1) A. Ceva Mama a distrus *vița-de-vie*. (My mother destroyed the grape.)  
          cuiva ceva Grindina i-a distrus *grădina*. (Hail destroyed his garden.)  
          pe cineva Vestea a distrus-o *pe Maria*. (The news destroyed Mary.)

<sup>1</sup> The research reported in this paper has been supported by the National University Research Counsel of Romania (CNCSIS), grant no. 1156/A.

de (căt-re)                    Recolta a fost distrusă *de secetă*.  
(The crop was destroyed by the drought.)

B. *a-și distruge*  
ceva                    Și-a distrus toate *manuscrisele*. (He  
destroyed all his manuscripts.)

C. *a se distruge*                    În accident mașina *s-a* distrus  
complet. (The car was destroyed completely in the accident.)  
cuiva                    I *s-a* distrus casa. (His house was destroyed.)

Nevertheless, Drăghicescu (2002) is much closer to what a valence dictionary should be. The entry of the verb A DISTRUGE has the following content (we left out the examples in French and Italian).

(2) I. **vb. tr.**

a. *a ~ ceva*

**subiect [± animat]**

R. Grindina a distrus toată recolta.

E. Hail has destroyed the entire crop.

b. *a-și ~ ceva*

**(cu dat. posesiv)**

R. Paul și-a distrus toate manuscrisele.

E. Paul destroyed all his manuscripts.

II. **vb. refl.**

a. *a se ~ + circ. (cauză)*

**subiect [-animat]**

R. Multe drumuri s-au distrus din cauza inundațiilor.

(sin. *a se strica*)

E. Many roads were destroyed due to floods.

b. *a se ~ (cu sens pasiv)*

R. Gândacii de bucătărie se distrug cu produse speciale.

(sin. *a stârpi*)

E. Cockroaches are killed with special products.

Some critics to this manner of representing the valence information are at stake here.

- In both examples, the verb's arguments are expressed by words such as *ceva* (something.ACC), *cuiva* (to someone.DAT), *cuiva* (someone.ACC). Thus, the morpho-syntactic information is not explicitly indicated, making such dictionaries improper for a MRD (Machine-Readable Dictionary) use.
- The description is complicated with general structures like passive or impersonal forms –(1).C and (2).II–, and with the so-called possessive dative –(1).B and (2).I.b– which are not specific for the verb under discussion.
- Very few semantic restrictions are presented, though sometimes they are left implicit.
- In (2), a distinction of meanings is marked by synonyms (sin.), but the due verbal constructions are not consistently represented. For instance, the valence in (2).II.b is also valid for the sense in (2).II.a.
- It is not clear if the complement of cause in (2).II.a is a real complement or an adjunct, or if it is obligatory or optional.

### 3. An example of RVD verbal valences

Before building RVD in a machine-readable form, we described each entry in a meta-language, accessible to its authors and users. We stooped at three of the eight valences

of the verb *a trăi* 'to live', in an abbreviated example, to be further explained and referred in 4..

(3) *a trăi* 'to live'

Argument structures:

1. NP[*nom*, +animate]

Senses:

- to live: *Victima trăiește*. 'The victim is alive'.
2. NP[*nom*, +animate]  
NP[*ac*, +period]  
(AdvP[manner] or PP[*la/cu*, -])
- Meanings:
- to spend: *Ion își trăiește tinerețea (intens / la maxim)*. 'John lives his youth (intensively / to the maximum)'.
  - to feel intensively: *Spectatorii au trăit momentul (cu entuziasm)*. 'The public lived the moment (enthusiastically)'.

3. NP[*nom*, +animate]

PP[*pentru*, +goal]

Meanings:

- to devote his/her life: *Femeia trăiește pentru răzbunarea soțului*. 'The woman lives for avenging her husband'.

A RDV entry ends, if needed, with a list of multi-word expressions in which the entry verb occurs. For such expressions no argument structure is given.

The following section is dedicated to the information that should be captured in a valence description and the criteria used in designing the RVD.

## 4. Criteria for designing the RVD

First of all, we need a criterion for distinguishing arguments from adjuncts. We adopted the one in [4, p.75] which states that "A participant role is a (semantic) argument of a verb [...] if its presence is required of all situations described by that verb and if it is required of the denotation of only a restricted set of verbs". In other words, an argument is *obligatory* and *specific* for a verb (or for a restricted class of verbs to which that verb belongs). This criterion will be further expanded in section 4.3, but it was mentioned here as an argument definition. Next, different aspects of a verbal valence are presented.

### 4.1 Valence Restrictions

#### 4.1.1 Morpho-syntactic restrictions

Valence information in a MRD has to be completely explicit and to cover all linguistic levels relevant for the verb's valences. Therefore, morphological, syntactical, semantic and lexical information should be described.

We chose to express complements in terms of syntactic phrases: noun phrases (NP), prepositional phrases (PP), adverbial phrases (AdvP), adjectival phrases (AP) and verbal phrases (VP) (instead of sentences). Complements must have restrictions or properties, which are

conventionally represented inside right brackets ([...]), as one can see in example (3) (section 3).

NPs are morphologically characterized by the grammatical cases of their heads. In Romanian, cases assigned by a verb, valid for all its inflected forms, can be nominative (*nom*), accusative (*acc*) or dative (*dat*). Therefore, NPs are represented like this: NP[*nom*], NP[*dat*], NP[*acc*].

PPs are lexically described in some situations. Some verbs require certain prepositions or series of prepositions. For instance, the verb *a recurge* ‘to resort’ always occurs with a PP introduced by the preposition *la* ‘to’: *a recurge la forță* ‘to resort to force’. On the other hand, a verb like *a scoate* ‘to take out’ requires prepositions indicating ‘the source’: *de la, de pe, din* ‘of, from’: *Ion a scos mobila din casă* ‘John took out the furniture from the house’. Finally, there are verbs whose PP complements should be introduced, for instance, by any location preposition; such a verb is *a pune*, ‘to put’: *a pune cartea pe / lângă / sub ... masă* ‘to put the book on / near / under ... the table’. As a convention of description, lexicalized prepositions are written in italics, while general types (such as location, manner, time) are written normally. So for example, for the verbs mentioned above (i.e. *a recurge*, *a scoate* and *a pune*) the PP complements are described PP[*la*], PP[*de la, de pe, din*] and PP[location], respectively. It is worth mentioning that every argumental PP has to have such a restriction; that is, there is no verb that requires just a PP, of any kind.

The same situation holds for AdvP, as well. Most AdvP complements should express manner or location, making these complements described as in AdvP[manner] or AdvP[location].

VP complements can contain a finite or non-finite verb, such as an infinitive, for instance the complement of the verb *a putea*: *eu pot merge* ‘I can go’, or a participle, for verbs such as *a trebui* ‘must’: *trebuie știut* ‘one must know’, *a merita* ‘it is worth’: *merită menționat* ‘it is worth mentioning’, etc. Finite verbal complements are in fact sentences. We avoided the term sentence because not all complements must have their subjects expressed in the same sentence; this is the case of raising and control verbs. Moreover, subjects themselves are complements in VPs. Relevant to VP complements is the verb mood, which can be subjunctive or not. Some VP complements are introduced by the subjunctive marker *să*, while others are introduced by the complementizer *că* which allows any finite mood, except the subjunctive. Verbal complements are described with expressions such as VP[*să*] or VP[*că*].

#### 4.1.2 Restrictions on lemma

Przepiórkowski [11] mentions that although valence dictionaries are supposed to provide information about lexemes (or lemmas), this does not hold for all forms of a given lexeme. He offers the example of Polish where direct

objects in accusative change their case in genitive for gerundial forms in *-nie/-cie* and in the scope of verbal negation (roughly speaking). In Romanian, there is not a similar situation, but the third person of singular when is used with an impersonal sense can have different valences from the other senses and the inflected forms. For instance, *a merita* has different valences for the sense ‘to deserve’ and for the impersonal sense ‘it is worth’. One can say *eu merit<sub>1.sg.</sub> un premiu* ‘I deserve a prize’, but not *merită<sub>3.sg.</sub> un premiu* ‘it is worth a prize’. The NP complement holds only for the first sense.

Another significant restriction on lemma is the use of negation. Certain verbs can actualize a certain meaning with a special valence structure only if it is used in a negative form. So, for example, the verb *a căuta* ‘to search’ has the uniquely determined meaning ‘to pay attention to’: *Nu căuta că sunt mic* ‘Do not pay attention to my height’, if it is negated and subcategorizes for a VP[*că*].

These facts made us include such morphological restrictions on lemmas in valence descriptions, if necessary.

#### 4.1.3 Semantic restrictions

Besides the morpho-syntactic characterization of complements, some semantic restrictions must be also indicated. These restrictions characterize either all verb meanings or only some of them. For instance, the verb *a bea* ‘to drink’ should have a subject marked with the +animate restriction for all its senses. On the contrary, the verb *a merge* ‘to go’, which is, in general, a motion verb, does not imply motion anymore if its subject is a road: *această autostradă merge la București* ‘this highway goes to Bucharest’. For this sense, the verb *a merge* is assigned a subject with the semantic restriction +road.

Meanings can be distinguished through semantic restrictions. Therefore, these restrictions correlated with a semantic ontology are very important for WSD.

We did not use any pre-defined inventory of semantic restrictions initially. We named the necessary restrictions *ad hoc* and we will refine and unify them after the dictionary is completed.

## 4.2 Valences and meanings

Our valence dictionary is conceived for WSD. For this purpose, different valence structures of a verb are put in correspondence with its different meanings. In our description, each valence structure of a verb is assigned a group of meanings which share that particular valence structure. We consider that a valence structure is common to a group of meanings if all valence restrictions, including the semantic ones, are valid for all meanings in group.

Verb meanings are taken from a medium-sized Romanian Explanatory Dictionary [13]. For each meaning described in our valence dictionary, a synonym and an example are provided (see example (3)).



### 4.3 Obligatory and optional complements

Saying which element is an obligatory complement of a verb, namely the element which co-occurs with the verb in all contexts (for a certain meaning) is less difficult than saying which element is an optional complement. Optional complements have to be distinguished from adjuncts, due to the fact that both of them co-occur with the verb randomly. This distinction is an important, but controversial issue and we will not tackle it here (see, Pustejovsky's [12]). Some criteria to justify the registration of an element among the valences of a verb, even if it has an optional status, are a must.

First of all, the *specificity* criterion mentioned in [4, p.75] claims that if an element is specific to a verb or a restricted set of verbs, it is an argument and it has to be included among valences of that verb. Instead, if an element can co-occur with any verb, it is an adjunct [16].

Location adjuncts are very frequent. However, motion verbs presuppose a starting point and a target point expressed in valence descriptions by two PPs[location]. Since either of these points can be omitted in contexts (for different reasons), the corresponding PPs should be marked as optionally – here, they are conventionally placed inside round brackets (see example (3).2).

Another criterion used for distinguishing complements is whether an element imposes the usage of a certain preposition or certain semantic restriction. For example, the verb *a scoate* can have the meaning 'to publish', a case in which it has two obligatory complements: a subject (NP[*nom*, +person]) and a direct object (NP[*acc*, +product]), but also an optional one described by PP[*la*, +company], which expresses which company made the product, such as in: *Ion a scos o carte la editura Polirom* 'John has published a book at Polirom Publishing House'. A simple location adjunct should allow any location preposition, not only the preposition *la* (which in fact can not be substituted in this context). Besides, the semantic restriction +company does not characterize a general location adjunct.

### 4.4 Valence alternations

Valence alternations are changes the subcategorisation frame of a verb can undergo. In other words, arguments of a verb can be syntactically expressed in different manners. These changes can also trigger semantic differences, but this is not compulsory. These changes create a problem of whether all should be registered in the lexicon or not. As it was mentioned in [6], registering all changes as different valences of the same verb could be a substantial source of inconsistency during annotation and could cause redundancy in the lexicon. We claim that the solution to this problem can be found in the type of alternations that can be regular or specific, as one can see below.

#### 4.4.1 Regular syntactic phenomena

The most common alternations are due to the different verbal voices; besides an active voice, Romanian has a passive voice and an impersonal one. Passive and impersonal constructions follow a regular pattern and their corresponding subcategorisation frames can be simply obtained by applying transformation rules (see [6]).

Another quite frequent alternation is the so-called *possessive dative* construction, which has been presented in section 2.1. This phenomenon characterizes any transitive verb (eventually restricted by an animate subject). Therefore, a subcategorisation frame corresponding to the dative possessive construction can be also obtained with a transformation rule. There is another Romanian phenomenon similar to the *possessive dative*, named *object duplication*: the verb gets a pronominal clitic duplication of its direct or indirect object in some precise situations.: *Ion citește cartea* 'John reads the book' → *Cartea o citește Ion* 'The book<sub>acc</sub> it<sub>clitic,acc</sub> reads John<sub>nom</sub>'. Again, this is a too regular syntactic phenomenon for assigning two different subcategorisation frames to the verb *a citi* 'to read' in the lexicon. Actually, it is controversial whether these verbal cliticization phenomena are a matter of valence alternation.

Any verb can also undergo a valence alternation regarding the change of a noun phrase into a free relative clause. For instance, one can say 'John loves me' or 'Who knows me loves me'. In general, any complement can be expressed by a corresponding VP complement, and this fact should not determine the multiplication of subcategorisation frames of every verb. However, the opposite does not hold. For instance, the verb *a convinge* 'to persuade' always requires a VP complement (VP[*să*] in our notation), which can never be replaced by an NP: *Ion o convinge pe Maria să rămână* 'John persuades Mary to stay'. Situations of this kind have to be included in valence entries.

#### 4.4.2 Alternations on classes of verbs

Apart from syntactically regular valence alternations, there are also alternations determined (at least in part) by the verb content. These are alternations are described, for instance, Beth Levin [5].

Levin's approach is a guide and a model for our own description of valence alternations in Romanian. We decided to use the format of Levin's description, in which verbs are grouped into semantic classes. Thus, according to this format, a given verb belonging to a given class could exhibit a number of given valence alternations. For instance, the causative verb *a amuza* 'to amuse' may undergo an inchoative alternation (which is not allowed in English): *Copiii se amuză* 'Children amuse themselves', and a direct object deletion: *Clovnii amuză* 'Clowns amuse'. All this information must accompany the verb valence description in the lexicon. We depart from Levin's format in that we do not group all verbs with common

valence alternations under the same category – for instance there is no class gathering the verb *a amuza* and, say, *a enerva* ‘to annoy’ even if they share valence alternations.

#### 4.4.3 Morpho-syntactic alternations

There is another type of alternations, which we called morpho-syntactic alternations. This is the case of arguments which can be expressed by different types of phrases. For instance, in (3) (section 3), the argument structure 2 contains an optional complement described like this: (AdvP[manner] or PP[*la/cu*, -]). The corresponding argument can be expressed either by an AdvP or by a PP, with the above mentioned characteristics. Even if the alternation AdvP / PP is quite frequent in Romanian, it can not be captured by transformation rules because it is not deterministic. That is why we have to specify the cases in which such an alternation works. Actually, morpho-syntactic alternations can be of many types.

## 5. Conclusions and further work

The paper gives an all encompassing perspective on the problems which appear when designing a valence dictionary.

The meta-language we adopted for describing valence entries is also accessible guidelines for experts who build the lexicon and a friendly typeset for a paper-based dictionary. This can be enriched as one goes along the project. For instance, enrichments could refer to the representation of the semantic roles and to the problem of raising and control verbs, whose importance was highlighted in [11]. So far, we have left these aspects aside, because, in our opinion, they rather bear on text understanding than WSD. Despite the fact that semantic roles do not lack in the valence descriptions of other languages, we decided to pay more attention to semantic restrictions which we found much more relevant for NLP tasks. Note further that the inventory of semantic roles is pretty controversial and its applying can be sometimes confusing for different human experts. Of course, we do not claim that this problem should be completely ignored. We have just postponed it for a later phase.

A future stage of the project will be to get RVD in a machine readable format. We intend to automatically transfer our meta-language representation into XML format. In so doing, we plan to take advantage of the facilities offered by CLaRK System [15].

## 6. References

- [1] E. Balabanova and K. Ivanova. Creating a Machine-readable Version of Bulgarian Valence Dictionary, in *Proceedings of the First Workshop on Treebanks and Linguistic Theories*, Sozopol, Bulgaria, p. 1-12, 2002.
- [2] J. Drăghicescu (coord.). *Dicționar de Construcții Verbale Român-Francez-Italian-Englez*, Editura Universitaria, Craiova, 2002.
- [3] A. Ionescu and M. Steriu, Maria. *Verbul Românesc. Dicționar Sintactic*, Editura Universității din București, 1999.
- [4] J.-P. Koenig, G. Mauner and B. Bienvenue. “Arguments for adjuncts”, in *Cognition* 89, Elsevier, p. 67-103, 2003.
- [5] B. C. Levin. *English Verb Classes and Alternations: A Preliminary Investigation*, University of Chicago Press, Chicago, IL, 1993.
- [6] M. Lopatková, Z. Žabokrtský and K. Skwarska. Valency Lexicon of Czech Verbs: Alternation-Based Model. In *Proceedings of the 5<sup>th</sup> International Conference on Language Resources and Evaluation*, 24-26 May, Genoa, Italy, 2006.
- [7] C. D. Manning. Automatic Acquisition of a large subcategorisation dictionary from corpora in *Proceedings of the 31<sup>st</sup> ACL*, p. 235-242, 1993.
- [8] M. Maragoudakis, K. Keramidis, N. Fakotakis, G. Gokkinakis. Learning Automatic Acquisition of Subcategorization Frames using Bayesian Inference and Support Vector Machines. ICDM '01, The 2001 IEEE International Conference on Data Mining, San Jose, p. 623-625, 2001.
- [9] C. Pollard and I. A. Sag. *Information-based Syntax and Semantics*, CSLI, Stanford, California, 1987.
- [10] C. Pollard and I. A. Sag. *Head-driven Phrase Structure Grammar*, Chicago University Press/CSLI Publications, Chicago, IL, 1994.
- [11] A. Przepiórkowski, Towards the Design of a Syntactico-Semantic Lexicon for Polish, in *Proceeding of New Trends in Intelligent Information Processing and Web Mining*, Zakopane, Springer Verlag, 2004.
- [12] J. Pustejovsky. *The Generative Lexicon*, The MIT Press, Cambridge, Massachusetts, London, England, 2001.
- [13] *Romanian Explanatory Dictionary*, Univers Enciclopedic, Bucharest, 2005.
- [14] A. Sarkar and D. Zeman. Automatic Extraction of Subcategorization Frames for Czech, in *Proceedings of The 18<sup>th</sup> International Conference on Computational Linguistics: COLING2000*, Saarbruecken, Germany, July 31 – August 4, p. 691-697, 2000.
- [15] K. Simov, A. Simov, M. Kouylekov, K. Ivanova. CLaRK System: Construction of Treebanks. In *Proceedings of the First Workshop on Treebanks and Linguistic Theories*, Sozopol, Bulgaria, p. 183-198, 2002.
- [16] P. Sgall, E. Hajičová and J. Panevová. *The Meaning of the Sentence and Its Semantic and Pragmatic Aspects*. Academia/Reidel Publishing Company, Prague Czech Republic/Dordrecht, Netherlands, 1986.
- [17] Z. Žabokrtský. *Valency Lexicon of Czech Verbs*. PhD Thesis, Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University, Prague, 2005.
- [18] Z. Žabokrtský and M. Lopatková. Valency Frames of Czech Verbs in VALLEX 1.0, in *Proceedings of the Workshop of the HLT/NAACL Conference*, May 6, p.70-77, 2004.

# Harvesting Ontologies from Open Domain Corpora: a Dynamic Approach

R. Basili(\*), A. Gliozzo (†), M. Pennacchiotti (‡)

(\*), DISP - University of Roma, *Tor Vergata*  
Via del Politecnico, 1 - 00133 Roma (Italy)  
*basili@info.uniroma2.it*

(†) Fondazione Bruno Kessler  
Povo, Trento (Italy)  
*gliozzo@itc.it*

(‡) Computational Linguistics, Saarland University  
Saarbrücken, Germany.  
*pennacchiotti@coli.uni-sb.de*

## Abstract

In this work we present a robust approach for dynamically harvesting domain knowledge from open domain corpora and lexical resources. It relies on the notion of Semantic Domains and provides a fully unsupervised method for terminology extraction and ontology learning. It makes use of an algorithm based on Conceptual Density to extract useful relations from WordNet. The method is efficient, accurate and widely applicable, as the reported experiments show, opening the way for effective applications in retrieval tasks and ontology engineering.

## Keywords

Lexical Acquisition, Ontology Learning, Word Sense Disambiguation

## 1 Introduction

Ontology learning from text is a popular field of research in Natural Language Processing (NLP). The increasing amount of textual information at our disposal needs to be properly identified, structured and formalized to make it accessible and usable in applications. Much work has focused on the harvesting phase of ontology learning. Researchers have successfully induced terminologies, word similarity lists [13], generic and domain relations [20, 17], facts [6], entailments [22] and other resources.

However, these resources must be structured in a richer semantic network in order to be used in inference and applications. So far, this issue has been solved by linking the harvested resources into existing ontologies or structured lexical repositories like WordNet [7], as in [16, 21].

Yet, applications often require domain specific knowledge but this means that adapting the existing general purpose resources, such as WordNet, is required. In general, this task is not trivial, as large scale resources are ambiguous (i.e. terms may refer to multiple concepts in an ontology, even if only some of them are actually relevant for the domain) and not balanced (i.e. some portions of WordNet are much more densely populated than others [1]). These problems are typically addressed by performing the following tasks.

**Lexical ambiguity resolution** : disambiguate terms by linking them to the correct sense(s) for the specific domain.

**Ontology pruning** : prune the ontology and induce only the sub-portion which is relevant for the given domain. This can be intended as a side effect of ambiguity resolution.

**Ontology Population** : extend an existing ontology with novel instances, concepts and relations found into domain specific corpora.

Most of these domain-oriented approaches (e.g. [23]) require domain specific corpora and are typically semi-supervised, as they need manual intervention to alleviate the errors due to the typically low precision achieved by automatic techniques. This constraint prevents the use of such techniques into open domain scenarios in applications in which the domain of interest is specified at run-time (such as Information Retrieval (IR) and Question Answering).

In this paper, we propose a solution to the above issue, by focusing on the problem of on-line domain adaptation of large scale lexical ontologies. The requirement for such an application is to implement an adaptation process which is:

- performed at run time;
- tuned by using only the user information need;
- fully automatized, and therefore accurate enough for the application in which it is located.

In contrast to classical approaches, we propose a novel unsupervised technique to induce *on-the-fly* domain specific knowledge from *open domain corpora*, starting from a simple user query formulated in a IR style.

Our algorithm is inspired by the notion of *Semantic Domains* and is based on the combined exploitation of two very well known techniques in NLP: Latent Semantic Analysis (LSA) [5] and Conceptual Density (CD) [1]. The main idea is to first apply LSA to extract a domain terminology from a large open domain corpus, as an answer to the user query. Then, the algorithm leverages CD to project the inferred terms into WordNet to identify domain

specific sub-regions in it, that can be regarded as lexicalized core ontologies for the domain of interest. The overall approach allows to achieve the goals of *lexical ambiguity resolution* and *ontology pruning*, and offers an online solution to the problem of domain adaptation of lexical resources discussed in [18, 24]. An example of the output of our system for the query MUSIC is illustrated in Figure 1.

In our setting, the use of LSA guarantees a major advantage. Unlike classical methods to estimate term similarity (e.g. [25, 12]) which are based on contextual similarity [4], LSA relies on a domain restriction hypothesis [10] stating that two terms are similar, and therefore are very likely to be semantically related, when they belong to the same domain, i.e. when they co-occur in the same texts. LSA detects as similar terms not those having the same ontological type (e.g. the most similar terms to *doctor* will be concepts belonging to the type *PERSON*) but those referring to the same domain, as needed in ontology learning (for example, in the medical domain we need both *doctors*, and *hospital*).

In the rest of the paper we will show evidences supporting the following contributions of this work: (i) the induction process is triggered by a simple IR-like query, providing to the user/application the required domain ontology *on the fly*; (ii) unlike previous approaches, our method does not need domain corpora, (iii) the method guarantees high precision both in the lexical ambiguity resolution and in the ontology induction phases.

We will also show that the main contribution of our method is a very accurate Word Sense Disambiguation (WSD) algorithm, largely outperforming a most frequent baseline and achieving performance close to human agreement. The paper is organized as follows. In Section 2 we introduce the concept of Semantic Domain as a theoretical framework motivating our work and we describe the terminology extraction step, required to provide an input to the CD algorithm producing the final domain ontology (Section 3). Section 4 concerns evaluation issues, while Section 5 concludes the paper.



Fig. 1: Core ontology extracted from WordNet for the “music” domain

## 2 Terminology Extraction in the Domain Space

The theoretical foundation underlying this work is the concept of *Semantic Domain*, introduced for WSD purposes [14] and further exploited in different tasks, such as Text

Categorization and Relation Extraction [8]. Semantic Domains are common areas of human discussion, such as Economics, Politics and Law. Three properties of Semantic Domains are relevant for our task. First, they are characterized by high lexical coherence [14]. This allows us to automatically induce specific terminologies from open domain corpora. Secondly, the ambiguity of terms in specific domains decreases drastically, motivating our lexical ambiguity resolution process. For example, the (potentially ambiguous) word *virus* is fully disambiguated by the domain context in which it is located (it is a *software agent* in the COMPUTER SCIENCE domain and a *infectious agent* in the MEDICINE domain). Third, as shown in [8], semantic relations tend to be established mainly among domain specific terms.

Semantic Domains are described by Domain Models (DM) [9], by defining a set of term clusters, each representing a Semantic Domain, i.e. a set of terms having similar topics (see Figure 2). DMs can be acquired from texts by exploiting term clustering algorithms. For our experiments we adopted a clustering strategy based on LSA, following the methodology described in [9].

To this aim, we first identify candidate terms in the open domain document collection by imposing simple regular expressions on the output of a Part of Speech tagger (e.g.  $((Adj|Noun)+|((Adj|Noun)*(NounPrep)?)(Adj|Noun)^*)Noun$ ), as described in [11]. The obtained term by document matrix is then decomposed by means of Singular Value Decomposition (SVD) [5] in a lower dimensional domain matrix  $\mathbf{D}$ . The  $i^{th}$  row of  $\mathbf{D}$  represents the Domain Vector (DV) for the term  $t_i$   $V$ , where  $V = \{t_1, t_2, \dots, t_k\}$  is the vocabulary of the corpus (i.e., the terminology). DVs represent the domain relevance of both terms and documents with respect to any domain.  $\mathbf{D}$  is then used to estimate the similarity in a Domain Space (i.e. a  $k'$  dimensional space in which both documents and terms are associated to DVs) by using the cosine operator on the DVs.

When a query  $Q$  is formulated (e.g. MUSIC), our algorithm retrieves the ranked list  $dom(Q) = (t_1, t_2, \dots, t_{k_1})$  of domain specific terms such that  $sim(t_i, Q) > \theta$  where  $sim(Q, t)$  is the cosine between the DVs corresponding to  $Q$  and  $t$ , capturing domain proximity, and  $\theta_t$  is the *domain specificity* threshold.

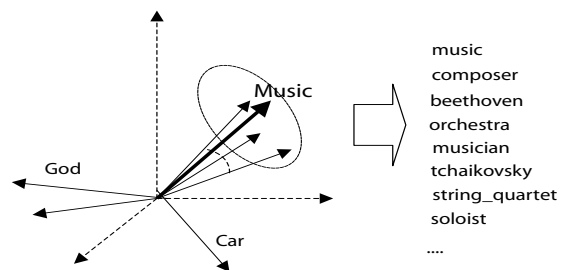


Fig. 2: Semantic Domain generated by the query MUSIC

The process is illustrated in Figure 2. The output of the Terminology Extraction step is then a ranked list of domain specific candidate terms and an associate ranked list of domain specific documents.

### 3 Inducing a core ontology via Conceptual Density

Once a semantic domain has been identified as an unstructured set of domain specific terms, our algorithm induces a core ontology from WordNet, by selecting the maximally dense sub-regions including them. This step involves a WSD process, as only the domain specific synsets associated to the terms extracted in the previous step have to be selected. To induce the core ontology from the terminology, we developed an algorithm, based on CD, that adapts the Dynamic Domain Sense Tagging algorithm proposed in [2]. The goal of our algorithm is twofold:

1. *Lexical ambiguity resolution.* Selecting the domain specific senses of ambiguous domain specific words.
2. *Ontology induction/pruning.* Selecting the best generalizations of the domain specific concepts associated to the word senses.

The algorithm achieves these goals applying a variant of the notion of CD proposed in [3] In the literature, the classical notion of CD has been applied in “local” context of words to be disambiguated, represented as word sets. The main problem of this approach is that small contexts, typically composed by few words appearing in the same sentence, do not allow generalization over the WordNet structure, being them typically spread in the graph, and then not well connected. For example the words *surgeon* and *hospital* lie in different WordNet hierarchies, preventing us from finding the common generalization necessary for disambiguation via CD.

To solve the problem, we apply the CD definition given in [3], integrating it with Domain Information, as in [2]. The context is here intended as the domain terminology  $dom(Q)$  inferred from the previous step. The terminology provides the evidence needed to start the generalization process (e.g. in the medical domain we expect to find much more words related to *surgeon*, such as *oncologist* and *dentist*, both related by the common hyperonym *doctor*).

The hypothesis is that when all the paradigmatic relations among terms in  $dom(Q)$  are imposed, the CD algorithm is able to select the proper sub-region of WordNet containing the suitable domain specific concepts, discarding most of irrelevant senses associated to the extracted terminology. The outcome of the process is thus the subset of senses or their generalizations able to explain  $dom(Q)$  according to WordNet. The result is a “view” of the original WordNet, as the core domain ontology for  $Q$  (Figure 1).

Specifically, terms  $t \in dom(Q)$  can be generalized through their senses  $\sigma_t$  in the WordNet hierarchy. The likelihood of a sense  $\sigma_t$  is proportional to the number of other terms  $t' \in dom(Q)$  that have common generalizations with  $t$  along the paths activated by their hyperonyms  $\alpha$  in the hierarchy. A measure of the suitability of the synsets  $\alpha$  for the terms in  $dom(Q)$  is thus the *information density* of the subtrees rooted at  $\alpha$ . The higher is the number of nodes under  $\alpha$  that generalizes some nouns  $t \in dom(Q)$ , the better is the interpretation  $\alpha$  for  $dom(Q)$ . The CD of a synset  $\alpha$  given a query  $Q$ ,  $cd^Q(\alpha)$ , models the former notion and provides a measure for the latter.

**Ontology Induction.** The target core ontology is the set of

synsets  $G(Q)$  that represents the *best paradigmatic interpretation* of the domain lexicon  $dom(Q)$ . This can be efficiently computed by the *greedy* search algorithm described in [3] that outputs the minimal set  $G(Q)$  of synsets that are *the maximally dense generalizations of at least two terms in  $dom(Q)$* . Terms  $t \in dom(Q)$  that do not have a generalization are not represented in  $G(Q)$ <sup>1</sup>.

As any  $\alpha \in G(Q)$  is a WordNet synset, by completing  $G(Q)$  with the topmost nodes we obtain a subset of WordNet that can be intended as a full domain-specific ontology for the triggering domain  $Q$ . An excerpt of the core domain ontology, for  $Q = \{music\}$  is shown in Figure 1 where terms are leaves (*green nodes*), *yellow nodes* are their common hyperonyms  $\alpha \in G(Q)$  and *red nodes* are the topmost nodes.

The core ontology, triggered by the short specification of a domain of interest given in  $Q$ , is thus the comprehensive explanation of all the paradigmatic relations between terms of the same domain.

**Lexical ambiguity resolution.** The semantic disambiguation of a target term  $t \in dom(Q)$  depends on the subset of generalizations  $\alpha \in G(Q)$  concerning some of its senses  $\sigma_t$ . Let  $G_t(Q)$  be such a subset, i.e.

$$G_t(Q) = \{\alpha \in G(Q) \mid \exists \sigma_t \text{ such that } \sigma_t \prec \alpha\} \quad (1)$$

where  $\prec$  denotes the transitive closure of the hyponymy relation in WordNet. The set  $\sigma(t, Q)$  of inferred domain specific sense  $\sigma_t$  for  $t$  is given by:

$$\sigma(t, Q) = \{\sigma_t \mid \sigma_t \prec \bar{\alpha}\} \quad (2)$$

where  $\bar{\alpha} = \operatorname{argmax}_{\alpha \in G_t(Q)} cd^Q(\alpha)$ . Also, multiple senses may be assigned to a term. The CD score associated to each inferred domain sense  $\sigma_i \in \sigma(t, Q)$  (i.e.  $cd^Q(\bar{\alpha}_i)$ ) is then mapped to the probability  $P(\sigma_i | t, Q)$ , which accounts for how reliable the sense is for the term  $t$  in the given domain, by normalizing them so that their sum over all senses of  $t$  is equal to 1.

## 4 Evaluation

Our evaluation aims at assessing the ability of our model in: (1) determining a suitable terminological lexicons; (2) extracting a proper ontological description of the target domain. We then focus on measuring the precision of the terminology extraction step in proposing correct candidates (Subsection 4.1), and on the accuracy and coverage of the induced core ontology (Subsection 4.2).

### 4.1 Terminology Extraction

#### 4.1.1 Experimental Settings

We evaluated terminology extraction in 5 different domains: MUSIC, CHEMISTRY, COMPUTER.SCIENCE, SPORT and CINEMA. We described them by simple queries made by their single names (e.g. SPORT is described by the query “*Sport?*”). As open domain corpus, we adopted the British National Corpus (BNC). In a preprocessing step, we split texts into 40 sentence segments, regarded as different documents, amounting to about 130,000 documents.

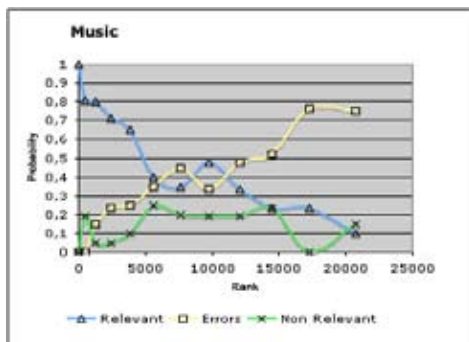
<sup>1</sup> A Web version of the greedy CD-based algorithm is available at <http://ai-nlp.info.uniroma2.it/Estimator/cd.htm>.

Each document is PoS-tagged and terms are identified by regular expressions as in [11]. Terms occurring in less than 4 documents are filtered out so that a source vocabulary of about 450,000 different terms is obtained. We run the SVD process on the resulting 450,000 x 130,000 term by document matrix, and we induce a DM from it, by considering a cut to the first 100 dimensions<sup>2</sup>.

For each domain, we use the similarity function *sim* (Section 2) to rank the candidate terms thus obtaining a ranked list of the overall dictionary. To carry out the evaluation we extract a sample of candidate terms in different positions in the list. Specifically, we divide the list in 11 rank levels, and extract 20 random terms from each of the level. The samples are then submitted (neglecting the ordering) to two domain experts. Each term is judged as *Relevant* or *Not Relevant* for the query domain or *Errors* for ill formed expressions (e.g. *olive\_neighbour*), unmeaningful (e.g. *aunty\_yakky\_da*) or non-terms (e.g. *good\_music*). For each rank level, the percentage of each label over the 20 candidates is computed. Results for the domain MUSIC are reported in Figure 3<sup>3</sup>.

#### 4.1.2 Results

As far as recall is concerned, systems for terminology extraction are hard to evaluate [19]. This problem is even more relevant in an open domain scenario, where it is not possible to have a comprehensive picture of the domain knowledge actually contained in texts. Thus we focused only on evaluating precision.



**Fig. 3:** Evaluation of the Terminology Extraction algorithm for the MUSIC domain

Results in Figure 3 show that Domain similarity is highly correlated to the precision of the terminology extraction step, providing an effective selection criterion. Setting the domain similarity threshold to 0.8, the algorithm retrieves about 2500 terms, among which 80% are relevant for the domain. When the domain is less represented in the corpus the number of terms retrieved with the same threshold is sensibly lower (e.g. in the domain chemistry the algorithm retrieves about 20 terms), but the accuracy is basically preserved. Therefore domain similarity provides a meaningful selection criterion to retrieve domain specific terminology, ensuring very accurate results without requiring further domain specific parameter settings. We also compared our term extractor to a baseline heuristic, consisting on ranking the same terms with respect to their *frequency* in the top 1,000 domain specific documents for each query, obtained

according to their similarity with respect to the initial query (as described in [5]). The precision of the two systems is measured against the labeling of the domain experts of the best ranked 100 terms proposed by each system. Results for all the domains are reported in Table 1. Our algorithm largely outperforms the baseline on all domains.

Domain	TE	Baseline
<b>Chemistry</b>	<b>0.85</b>	0.58
<b>Cinema</b>	<b>0.93</b>	0.34
<b>Computer</b>	<b>0.92</b>	0.46
<b>Music</b>	<b>0.93</b>	0.46
<b>Sport</b>	<b>0.95</b>	0.48

**Table 1:** Precision of our term extractor (TE) and the baseline system, on the top ranked 100 terms for each domain.

The lower performance obtained on the CHEMISTRY domain are due to the inclusion in the LSA space of some documents/terms relevant for the more general academic domain, which in the BNC slightly overlaps with chemistry. While these are only preliminary results, they show that a LSA based algorithm for ranking terms offers a high degree of precision and can be effectively adopted to perform on-line terminology extraction.

## 4.2 Inducing Domain Specific Core Ontologies

The goal of the ontology pruning step is to identify coherent sub-portions of WordNet as useful models for a domain: the hypothesis is that these contain most of the selected terms and their generalizations. The CD algorithm presented in Section 3 achieves both goals. In this section we evaluate the ontology pruning step according to two factors: the ability of identifying only correct senses for the terms (Subsection 4.2.2); the “*capacity*” of the core ontologies, i.e. their ability to be populated by novel concepts and/or instances (Subsection 4.2.3).

### 4.2.1 Experimental Settings

The induction of the core ontology in each area of interest is based on Wordnet (version 2.0). We focused on the noun hierarchy, which is organized on 41 taxonomies describing the hyponymy relation. Due to its huge dimension, pruning WordNet is not an easy task. Out of the 115,524 synsets in WordNet, a core ontology is expected to contain only hundreds of concepts, making the retrieval problem very hard. Given the quality of the terminology extraction process we used as seed the list of domain specific terms for each domain. For each domain we selected all the lemmata in WordNet comprises within the top ranked 1,000 terms for each domain (set *r* in Section 3) to initialize the CD algorithm. The result is the best (i.e. most conceptually dense) Wordnet substructure. An example is in Figure 1 and 4. Each term that appears in the ontology is also disambiguated, as the CD provides very low scores (close to 0) for all irrelevant senses, which are then discarded in the ontology generation phase.

### 4.2.2 Identifying domain specific senses

In a first analysis we focused on unambiguous terms, as their corresponding synsets are necessarily domain specific

<sup>2</sup> SVD is applied through LIBSVD (<http://tedlab.mit.edu/~dr/SVDLIBC/>)

<sup>3</sup> Results on other domains do not significantly differ from those reported for Music and will be not reported because of space limitation.



senses. The percentage of monosemous words varies sensibly among the different domains, ranging from 48% in MUSIC to 84% in CHEMISTRY. Figure 3 suggests that less than 20 % of entries within the first 1,000 candidates are not relevant for the ontology. An analysis of the first 200 monosemous terms in the candidate list has been carried out for all domains revealed that about 95% of terms are correct. In such cases the accuracy of the method is higher, as monosemous terms included in Wordnet, are clearly less affected by errors.



**Fig. 4:** Core ontology extracted from WordNet for the CHEMISTRY domain

The real issue is here to validate the senses proposed for ambiguous domain specific terms. This can be regarded as an unsupervised disambiguation task, as we did not use any training data. In contrast to the common WSD settings (where WSD is evaluated as the selection of the correct sense for words in a textual context), we need to measure the ability of selecting *domain specific senses*. In the literature this problem has been also referred as *predominant sense identification* for specific domains, e.g. [15]. Unlike these approaches, our algorithm does not require domain specific collections nor the use of any complex preprocessing tool (e.g. a dependency parser).

To evaluate the disambiguation accuracy, we selected from the top 200 terms in the ranked list of each domain all the ambiguous terms contained in WordNet. We then asked two lexicographers to mark their senses with respect to the query: domain vs. non-domain specific senses are thus labeled. For example, the lemma *percussion* has four senses (i.e. “the act of playing a percussion instrument”, *detonation*, *rhythm\_section* and *pleximetry*), but only the first and the third have been judged relevant for the domain MUSIC. Table 2 shows some statistics about the annotated resource produced as a gold standard. For each domain, the number of ambiguous cases analyzed and the relative polisemy (according to Wordnet 2.0) is reported in the first two columns. The last two columns report two different inter-annotator agreement measures. *AgrF* represents the “full” agreement, estimated by counting all senses in which the annotators agreed (either positives or negatives) and by dividing it by the number of all possible senses. This figure provides an upper bound for the *accuracy* of the system. Since we are mostly interested in defining an upper bound for the F1, we computed a second agreement score. As precision and recall are measured on the positive senses only, the last column (*AgrP*) reports the agreement on positive

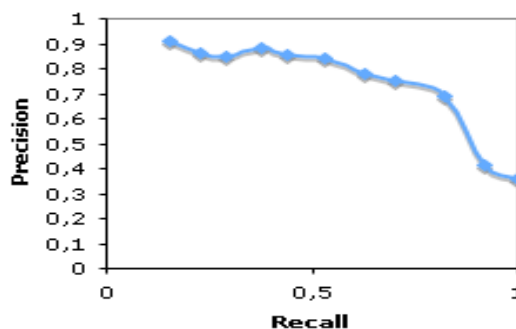
examples, computed over those cases in which *at least one annotator* provided a positive labeling.

Domain	Amb	Pol	AgrF	AgrP
<b>Music</b>	35	3.9	0.91	0.83
<b>Sport</b>	21	5.6	0.92	0.76
<b>Computer</b>	16	4.8	0.97	0.89
<b>Chemistry</b>	9	3.7	0.74	0.53
<b>Cinema</b>	4	5.3	0.95	0.85
<b>Total</b>	95	4.0	0.91	0.78

**Table 2:** Domain Specific Gold Standards for Sense disambiguation

The output of the CD algorithm is an estimation of the probability, for each sense, to be relevant for the domain expressed by the query. We can obtain a flexible binary classifier imposing a threshold  $\tau > 0$  on the output sense probabilities: a sense is accepted *iff* its probability is above  $\tau$ . Figure 5 shows the micro F1, averaged over all domains, obtained by the classifier parameterized with different values of  $\tau$ , (i.e. from 0, all accepted, to 1, none accepted).

The best F1 value (i.e. 0.75) is obtained by selecting all those senses whose probability is above 0.1. The system is also very precise, at cost of some points of recall: precision is over 0.8 at recall 0.56, and over 0.9 at recall 0.2. This trade-off is interesting as in ontology learning more precise results are often preferable.



**Fig. 5:** Precision and recall for different probability thresholds obtained by the WSD algorithm.

Table 3 summarizes the individual F1 scores over positive examples, in all domains, obtained with the optimal settings of the classification threshold, i.e.  $\tau = 0.1$ <sup>4</sup>. Two different baselines are reported: random and most frequent sense selection. The model outperforms both baselines. Notice how the performance is close to the upper bound provided by the agreement *AgrP* on positive examples of Table 2. As the CD algorithm is fully unsupervised, the improvement on the first sense heuristic is a very good result.

Dom	Prec	Rec	F1	rnd	MF
<b>Mus</b>	0.85	0.88	<b>0.87</b>	0.27	0.38
<b>Spo</b>	0.54	0.71	0.61	0.22	<b>0.67</b>
<b>Com</b>	0.58	0.82	<b>0.68</b>	0.23	0.18
<b>Chem</b>	0.64	0.875	<b>0.74</b>	0.32	0.29
<b>Cine</b>	0.56	0.71	0.63	0.22	<b>0.72</b>
<b>Micro</b>	0.69	0.82	<b>0.75</b>	0.25	0.40

**Table 3:** WSD performances

<sup>4</sup> Although this setting is derived from the test set itself, it is worthwhile to remark that the same optimal value is preserved over all domains.

### 4.2.3 Capacity

A final evaluation has been carried out to measure the capability of the core ontologies to host novel concepts and/or instances retrieved in the terminology extraction phase (i.e. their *capacity*). We gave to domain experts the lists of the top ranked 100 terms not included in WordNet for the MUSIC and CHEMISTRY domains. Then, they were asked to judge whether it was possible to attach the terms not in WordNet either to a *High Level* concept in the ontology (i.e. the topmost nodes, such as *entity* or *person*) or to a *domain specific* concept (i.e. the leaves in the ontology). Terms that could not be attached to any node of the core ontology have been marked as *Null*. Results are reported in Table 4. As the class of *Null* terms is also including errors from the terminology acquisition step, we can conclude that most of the terms are covered by the acquired domain ontology and can then be further exploited to populate domain specific nodes.

	NULL	HIGH	DOMAIN
MUSIC	22%	31%	47%
CHEMISTRY	46%	7%	47%

**Table 4:** *Capacity evaluation. Percentage of terms not in Wordnet covered by the automatically extracted core ontologies*

## 5 Conclusions and Future Work

In this paper we proposed a robust and widely applicable approach for dynamically harvesting domain knowledge from general corpora and lexical resources. The method exploits the notion of Domain Space and an  $n$ -ary semantic similarity measure over Wordnet for terminology extraction and ontology acquisition. Both processes are very accurate, fully unsupervised and efficient. The disambiguation power of the entire chain is very good, largely outperforming traditional effective baselines. The good impact over complex tasks such as term disambiguation and projection of suitable hyponymy/hyperonymy relations in Wordnet opens a number of potential applications. From a methodological point of view, we plan to extend the acquisition process targeting novel relations among concepts implicitly embodied in the original corpus. Also, we plan to develop automatic methods to further populate the core ontology with novel terms retrieved in the terminology extraction phase. The *on-the-fly* derivation of ontological descriptions for the specific domain of interest can be very attractive in Web applications (e.g. querying or navigation scenarios) and every process dealing with complex (e.g. distributed on-line) meaning negotiation problems. A tool for the automatic compilation of the induced ontology into standard knowledge representation formalisms for the semantic WEB, like OWL, is currently under development, as a general Web service to be easily integrated into an Ontology Engineering framework.

## Acknowledgments

All authors are grateful to Marco Cammisa for his technical contribution to the experiments. Alfio Gliozzo was supported by the FIRB-israel co-founded project N.RBIN045PXH.

## References

- [1] E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In *Proceedings of COLING-96*, Copenhagen, Denmark, 1996.
- [2] R. Basili, M. Cammisa, and A. Gliozzo. Integrating domain and paradigmatic similarity for unsupervised sense tagging. In *In Proceedings of ECAI06*, 2006.
- [3] R. Basili, M. Cammisa, and F. Zanzotto. A semantic similarity measure for unsupervised semantic disambiguation. In *Proceedings of LREC-04*, Lisbon, Portugal, 2004.
- [4] I. Dagan. *Contextual Word Similarity*, chapter 19, pages 459–476. Merce Dekker Inc, Handbook of Natural Language Processing, 2000.
- [5] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 1990.
- [6] O. Etzioni, M. Cafarella, D. Downey, A.-M. A.M. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–143, 2005.
- [7] C. Fellbaum. *WordNet. An Electronic Lexical Database*. MIT Press, 1998.
- [8] A. Gliozzo. The god model. In *Proceedings of EACL-2006*, Trento, 2006.
- [9] A. Gliozzo, C. Giuliano, and C. Strapparava. Domain kernels for word sense disambiguation. In *Proceedings of ACL-2005*, 2005.
- [10] A. Gliozzo, M. Pennacchiotti, and P. Pantel. The domain restriction hypothesis: Relating term similarity and semantic consistency. In *In proceedings of NAACL-HLT-06*, 2006.
- [11] J. S. Justeson and S. M. Katz. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1:9–27, 1995.
- [12] D. Lin. Automatic retrieval and clustering of similar words. In *COLING-ACL*, pages 768–774, 1998.
- [13] D. Lin and P. Pantel. DIRT-discovery of inference rules from text. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD-01)*, San Francisco, CA, 2001.
- [14] B. Magnini, C. Strapparava, G. Pezzulo, and A. Gliozzo. The role of domain information in word sense disambiguation. *Natural Language Engineering*, 8(4):359–373, 2002.
- [15] D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. Finding predominant senses in untagged text. In *Proceedings of ACL-04*, pages 280–287, Barcelona, Spain, 2004.
- [16] P. Pantel. Inducing ontological co-occurrence vectors. In *Proceedings ACL-2005*, Ann Arbor, Michigan, June 2005.
- [17] P. Pantel and M. Pennacchiotti. Espresso: A bootstrapping algorithm for automatically harvesting semantic relations. In *Proceedings of COLING/ACL-06*, 2006.
- [18] B. S. Paul Buitelaar. Ranking and selecting synsets by domain relevance. In *Proceedings on NAACL-2001 Workshop on WordNet and Other Lexical Resources Applications, Extensions and Customizations*, Pittsburgh, USA., 2001.
- [19] M. Paziienza, M. Pennacchiotti, and F. Zanzotto. Terminology extraction: an analysis of linguistic and statistical approaches. In S. Sirmakessis, editor, *Knowledge Mining*, volume 185. Springer Verlag, 2005.
- [20] D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. In *Proceedings of ACL-02*, 2002.
- [21] R. Snow, D. Jurafsky, and A. Ng. Semantic taxonomy induction from heterogeneous evidence. In *Proceedings of the ACL/COLING-06*, pages 801–808, Sydney, Australia, 2006.
- [22] I. Szpektor, H. Tanev, I. Dagan, and B. Coppola. Scaling web-based acquisition of entailment relations. In *Proceedings of EMNLP-2004*, Barcellona, Spain, 2004.
- [23] P. Velardi, R. Navigli, A. Cucchiarelli, and F. Neri. *Ontology Learning from Text: Methods, Evaluation and Applications*, chapter Evaluation of OntoLearn, a Methodology for Automatic Learning of Domain Ontologies. IOS Press, 2005.
- [24] P. Vossen. Extending, trimming and fusing wordnet for technical documents. In *Proceedings on NAACL-2001 Workshop on WordNet and Other Lexical Resources Applications, Extensions and Customization*, Pittsburgh, USA, 2001.
- [25] D. Widdows. *Geometry and Meaning*. CSLI Publications, 2004.



# A Lightweight on-the-fly Capitalization System for Automatic Speech Recognition

Fernando Batista<sup>1,2</sup>, Nuno Mamede<sup>1,3</sup>, Diamantino Caseiro<sup>1,3</sup>, Isabel Trancoso<sup>1,3</sup>

<sup>1</sup>*L<sup>2</sup>F* – Spoken Language Systems Laboratory - INESC ID Lisboa

R. Alves Redol, 9, 1000-029 Lisboa, Portugal

<sup>2</sup>ISCTE – Instituto de Ciências do Trabalho e da Empresa, Portugal

<sup>3</sup>IST – Instituto Superior Técnico - Technical University of Lisbon, Portugal

{fmmb, njm, dcaseiro, imt}@l2f.inesc-id.pt

## Abstract

This paper describes a lightweight method for capitalizing speech transcriptions. Several resources were used, including a lexicon, newspaper written corpora and speech transcriptions. Different approaches were tested both generative and discriminative: finite state transducers, automatically built from Language Models; and maximum entropy models. Evaluation results are presented both for written newspaper corpora and speech transcriptions of broadcast news corpora.

## Keywords

Rich transcription, capitalization, truecasing, maximum entropy, language models, weighted finite state transducers

## 1 Introduction

Enormous quantities of digital and video data are daily produced by media organizations, such as radio and TV stations. Automatic speech recognition systems can now be applied to such sources of information in order to enrich it with alternate information for applications, such as: indexing, cataloging, subtitling, translation and multimedia content production. The Automatic Speech Recognition (ASR) output consists of raw text, often in lower-case format. Even if useful for many applications, such as indexing and cataloging, the ASR output benefits from capitalization information for other tasks, such as subtitling and multimedia content production. In general, enriching the speech output aims to improve legibility, enhancing information for future human and machine processing. Besides capitalization, enriching speech recognition covers other activities, such as insertion of punctuation marks and detection and filtering of disfluencies, not addressed in this paper.

This paper describes a method for capitalization of automatic speech recognition transcriptions, using a reduced set of data, which can be integrated, for example, on an on-the-fly system for subtitling. The different data sources used for our experiments are described in section 2. Section 3 defines the performance measures used for evaluation. Section 4 de-

Corpus	Duration	Tokens	
train	61h	467k	81%
development	8h	64k	11%
test	6h	46k	8%

Table 1: Different parts of the SR corpus

scribes the different methodologies employed. The results achieved for capitalization are presented in section 5. The paper ends with some final comments and ideas for future work.

## 2 Data sources

The ultimate goal of this work is to perform automatic capitalization on the output of an ASR system. We will start by using written newspaper corpora for training and testing a set of methods and finally we will apply these methods on speech transcriptions. By doing so, we expect to analyze the performance degradation when moving from written corpora to speech transcriptions, and combine the available data sources in order to provide richer training sets, thus enhancing final results. Some small lexicons are also experimented in order to overcome the problem of using small data sets for training. The following subsections provide details about each one of the used data sources.

### 2.1 Speech Recognition Corpus

The Speech Recognition (SR) is an European Portuguese broadcast news corpus, collected in the scope of the ALERT international project<sup>1</sup>. The training data of the SR corpus was recorded during October and November 2000, the development data was recorded during December, and the evaluation data was recorded during January 2001<sup>2</sup>. Table 1 shows details about the corpus data sets.

The manual orthographic transcription of this corpus includes information such as punctuation marks, capital letters and special marks for proper nouns, acronyms and abbreviations. Each file in the corpus is divided into segments with information about the

<sup>1</sup> <https://www.l2f.inesc-id.pt/wiki/index.php/ALERT>

<sup>2</sup> [https://www.l2f.inesc-id.pt/wiki/index.php/ALERT\\_Corpus](https://www.l2f.inesc-id.pt/wiki/index.php/ALERT_Corpus)



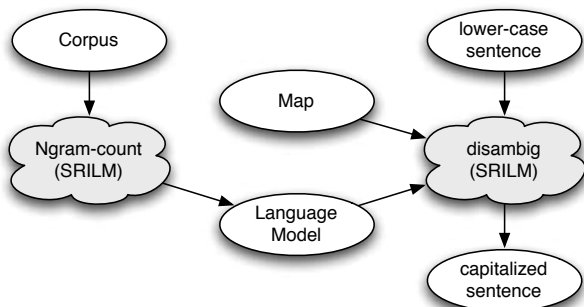


Figure 2: Using only the SRILM toolkit

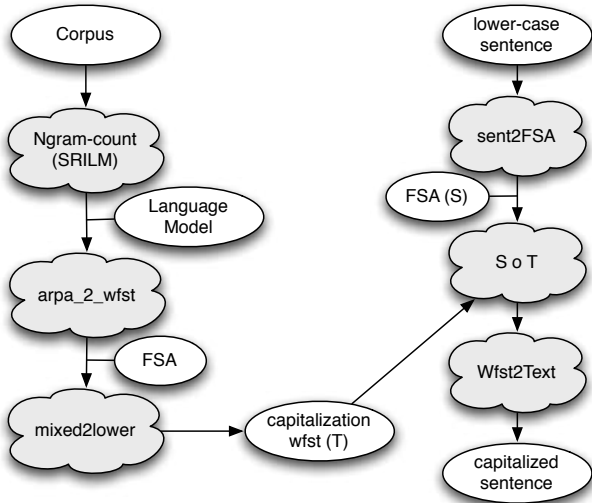


Figure 3: Using a WFST to perform capitalization

the SRILM toolkit, and can be used to perform capitalization directly from the language model. Figure 2 illustrates the process, where each cloud represents a process and an ellipse represents data. The *Map* corresponds to a file with all alternate forms of writing each word in the vocabulary. This is the most straightforward method, producing fast results, often used by the scientific community for this kind of task. It was part of the baseline suggested in the IWSLT2006 workshop competition<sup>3</sup>.

Another method, based on Weighted Finite State Transducers (WFST), is illustrated in figure 3. The SRILM toolkit is firstly used to produce an LM from the corpus and then the LM is converted into a finite state automaton (FSA), which can be viewed as a WFST having the input equal to the output. The transducer *T*, used for performing capitalization, results from the previous transducer where each input word was converted to its lower-case representation. The input of the resultant transducer can be represented by a lower-case vocabulary, while the output contains all graphical forms. The right side of figure 3 shows the process of capitalizing a sentence. The input sentence is firstly converted into an FSA (*S*) and then the operation *bestpath(S o T)* produces the desired result, in

<sup>3</sup> [http://www.slt.atr.jp/IWSLT2006/downloads/case+punc\\_tool\\_using\\_SRILM.instructions.txt](http://www.slt.atr.jp/IWSLT2006/downloads/case+punc_tool_using_SRILM.instructions.txt)

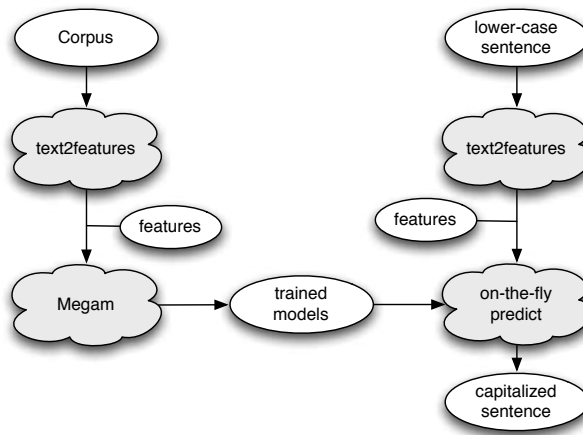


Figure 4: The maximum entropy approach

the form of another automaton.

Both methods use the *ngram-count* tool for creating the LM from the training data. As a consequence of that, experiments performed in the same conditions by the two methods share the same language model.

## 4.2 Maximum entropy

The discriminative modeling approach here applied is based on Maximum Entropy (ME) models. The MegaM tool - Maximum Entropy Model Optimization Package [2] is used for training, and the *on-the-fly* predicting tool uses previously trained models for performing the capitalization task. Figure 4 illustrates the overall process. The first step consists of training the models using a set of predefined features and the next step consists of using that information in order to predict the results. The MegaM tool includes an option for predicting results from previously trained models, but unfortunately it was not prepared to deal with a stream of data and produces results only after completely reading the input. The *on-the-fly* predicting tool overcomes this problem while using previously trained models in the original format.

The ME modeling approach allows easy combination of several sources of information, such as word information and POS tagging information. Nevertheless, the experiments here described only use features capturing word information, sometimes combined as bigrams and trigrams. The delay between the input and the output constitutes a problem for a module required to work on an *on-the-fly* system. Besides the computational time delay, an important aspect to be taken into consideration is the number of words on the right of the current word required to make a decision. For the results presented here, the feature set was chosen in order to avoid a right context greater than one.

## 5 Results

We assume that the first word of each sentence will always be capitalized in other processing step, for example along with the punctuation, since its correct graphical form mostly depends on its position in the

LM options	LM size
unigrams	7.3Mb
bigrams	27Mb
trigrams	78Mb

Table 4: Different LM sizes

LM options	Prec	Recall	F	SER
unigrams	91%	74%	82%	0.333
bigrams	<b>94%</b>	<b>84%</b>	<b>89%</b>	<b>0.212</b>
3-gram	93%	79%	85%	0.271
3-gram, interpol.	93%	80%	86%	0.266

Table 5: SRILM Toolkit results over RecPUB corpus

sentence. These words are excluded from training and evaluation, seeing that evaluation results may be influenced when taking such words into account [3].

The next subsections will show results achieved with both the generative and discriminative approaches: We will start by presenting some results obtained with the SRILM toolkit and the WFST, applied to both written newspaper corpora and speech transcriptions. Then some experiments, using maximum entropy with a limited quantity of data, will be described. Results achieved using only the most common graphical form are included in all experiments, which is a popular baseline for similar work [1, 3].

## 5.1 The generative approach

The first set of experiments were performed on written newspaper corpora, using RecPUB both for training and testing. As we use a vocabulary, all words outside vocabulary were marked “unknown” and punctuation marks were also removed from the corpus. The content of the corpus became closer to a speech transcription, but without recognition errors or disfluencies. A large size written corpora often contains a number of orthographic errors and less common words which, used in bigrams and trigrams, originates large quantities of ineffective data. Because of that, bigrams and trigrams occurring less than 5 times were not considered for LM training. Table 4 shows the size of each LM depending on the building options: unigrams, bigrams, and trigrams.

The first capitalization results for written newspaper corpus are presented in table 5. Both training and evaluation were performed with the RecPUB corpus, using the SRILM toolkit. Results achieved by unigrams show that, using only the current word, an SER of 33% can be achieved. The use of bigrams conducts to the best result, increasing both precision and recall, and showing that word co-occurrence is an im-

LM options	Prec	Recall	F	SER
unigrams	91%	77%	83%	0.307
bigrams	94%	88%	91%	0.176
3-gram	95%	89%	92%	<b>0.155</b>
3-gram, interpol.	95%	89%	92%	<b>0.154</b>

Table 6: WFST results over RecPUB corpus

LM options	Prec	Recall	F	SER
unigrams	81%	76%	78%	0.418
bigrams	78%	85%	81%	<b>0.388</b>
3-gram	79%	81%	80%	0.409
3-gram, interpol.	80%	81%	81%	0.390

Table 7: Results of SRILM method on the SR corpus

LM options	Prec	Recall	F	SER
unigrams	81%	77%	79%	0.422
bigrams	79%	86%	82%	<b>0.368</b>
3-gram	78%	87%	82%	0.380
3-gram, interpol.	78%	86%	82%	0.382

Table 8: Results of WFST method on the SR corpus

portant aspect to be taken into consideration for a capitalization task. The `disambig` tool has produced poor results for trigrams, which can be related to an increase of the search space when moving to a trigram language model. These results provide a baseline for the following experiments.

The second experiment was performed using WFSTs on the same corpus. Moreover, the capitalization transducers were produced from the same LM used in the previous experiment. Results from this experiment are shown on table 6. This method produces better results independently of the option for building the LM. The increase in the precision and recall values is correlated with the usage of higher order ngrams, and trigram models achieves the best results. The biggest difference, in terms of SER, occurs when moving from unigrams to bigrams, given that trigram models only add about 1% to precision and recall values.

The following experiments use the previous LM models, built for written newspaper data, in order to capitalize broadcast news speech transcriptions. Tables 7 and 8 shows the results of these experiments, using both the SRILM toolkit and WFST methods, over the SR corpus evaluation data. Results show the expected decrease of performance when going from written newspaper corpora to speech transcriptions. Notice however that the training was performed in the written newspaper corpora, which do not share the same properties as the speech transcription. The best results were achieved using bigrams for both methods, revealing a significant difference between written corpora and speech transcriptions.

Other experiments on capitalization were also performed for BN speech transcriptions, using only the SR data for training. The best result in terms of SER was 0.434, corresponding to a precision of 82% and recall of 72%. This result is no better than the worse result achieved using the written newspaper corpora for training, even so this was an expected result given the small training data size.

The WFST method consistently produces better results than using the `disambig` tool. Nevertheless, the current implementation of the WFST method implies loading, composing and searching a large non-deterministic transducer, thus being the most computationally expensive method proposed.

Exp	Corpora features	Lexicons	Prec	Rec	F	SER
1	$w_i$		85%	65%		0.466
2	$w_i (w_{i-1}, w_i) (w_i, w_{i+1})$		84%	67%		0.455
3	$w_i (w_{i-1}, w_i) (w_i, w_{i+1}) (w_{i-2}, w_{i-1}, w_i)(w_{i-1}, w_i, w_{i+1})$		84%	67%		0.458
4		PubLEX	80%	73%	76%	0.453
5	$w_i (w_{i-1}, w_i) (w_i, w_{i+1})$	LEX	84%	68%	75%	0.446
6	$w_i (w_{i-1}, w_i) (w_i, w_{i+1})$	PubLEX	85%	73%	79%	<b>0.391</b>
7	$w_i (w_{i-1}, w_i) (w_i, w_{i+1})$	LEX, PubLEX	85%	73%	79%	<b>0.391</b>

Table 9: Results of maxent over the BN speech transcriptions (SR corpus)

## 5.2 The discriminative approach

The Maximum Entropy approach requires that all information be expressed in terms of features, according to a previously defined feature set. The resultant data size may be several times the original one, making it difficult to use large corpora, such as the RecPUB corpus, for training purposes. The SR corpus training material (467k words) is clearly insufficient for covering the 57k vocabulary. In order to mitigate this problem we also used the two lexicons, previously described in section 2. By using this approach we expect to achieve gains while introducing small data resources.

Table 9 shows results for the most relevant experiments, combining different feature sets and information sources. For each one of the experiments, the table describes all the features used for capturing knowledge from SR corpus, where:  $w_i$  is the word at position  $i$  of the corpus,  $(w_i, w_j)$  is the bigram containing words  $w_i$  and  $w_j$  and  $(w_i, w_j, w_k)$  is the trigram containing words  $w_i, w_j$  and  $w_k$ .

The first 3 experiments were conducted using only the speech transcription data for training, without any additional resource. Experiment 1 establishes a baseline for what can be achieved using only the most common way of writing a given word, taking the SR corpus training data as reference. For this experiment, if no training data was available for a given word, it was kept lower-case. Experiments 2 and 3 show that adding bigrams and trigrams do not produce large changes, even so, bigram models is a good compromise between size and performance. These three experiments show that the SR corpus is far from sufficient for training.

Experiment 4 shows that by using only the most common way of writing a word, taking RecPUB data as reference, produces better results than using SR corpus alone. This experiment also shows that the ME approach produces lower results than previous generative approaches. The first line of each one of the tables 7 and 8 corresponds to the same task performed either with SRILM toolkit or the WFST, and the SER is about 3.3% better than current results. This is due to the representation of the information used in both approaches: the generative approaches considers the two words from bigram  $(w_i, w_j)$  independently, while the ME approach consider the bigram as a whole.

Experiment 5 shows the contribution of a small lexicon resource (LEX). The best result is achieved by combining the speech transcriptions from the SR corpus and the PubLEX lexicon, as shown in experiment 6. Experiment 7 also shows that LEX resource does

not add much information when using PubLEX.

The SER achieved using bigrams with the maximum entropy is only 2% worse than best results achieved using a generative approach, however this method allows a much faster way of performing capitalization directly from an input stream, given that the correct graphical form of a given word is calculated by means of a weighted sum of values, given by the word's correspondent features.

## 6 Concluding remarks

This paper addresses the problem of producing the capitalization information for texts without that information, such as the output of an ASR system. Three different methods were described and results were presented both for manual transcriptions of speech and written newspaper corpora. One of the methods, described as lightweight, combines different data resources for training and uses a straightforward procedure for predicting results. The performance achieved using this method is almost as good as using our best approach, while using a smaller number of resources. It has been integrated on an on-the-fly subtitling module for broadcast news, deployed at the Portuguese national television broadcaster.

Results for recovering capitalization both from written unpunctuated newspaper corpora and from broadcast news transcription were presented. Concerning the written newspaper corpus, we conclude that bigram and trigram information significantly contributes to enhance results, despite that trigram information only contributes with about 1% to precision and recall values. The used BN speech transcription corpus is too small and does not cover much of the vocabulary. Results show that using trigrams do not significantly improve results achieved by bigram when dealing with speech transcriptions. Lexica contribute to enhance the results when dealing with small size training data.

## 7 Future work

For now only three ways of writing a word were explored: lower-case, all-upper, first-capitalized, not covering mixed-case words such as RTPi and SuSE. We expect to address these cases in a near future, perhaps using a small lexicon.

Experiments concerning speech transcriptions and achieved results were produced using a manual BN

speech transcription. We plan to define a strategy for performing evaluation directly on automatic speech transcriptions, either performing a previous alignment with the manual transcriptions, or performing a human evaluation.

The problem of dealing with a dynamic vocabulary remains unaddressed in our experiments. Other features, such as word prefix and suffix, number of vowels and consonants shall also be explored. We also plan to introduce information coming from a part-of-speech tagger, in our ME models, already shown to improve results [5].

In the scope of the national TECNOVOZ<sup>4</sup> project, large amounts of broadcast news hand-annotated transcriptions, are now being daily produced. In the near future we plan to have much more training material, which will hopefully provide more accurate results.

## 8 Acknowledgments

This work was partially funded by the FCT projects LECTRA (POSC/PLP/58697/2004), NLE-GRID (POSC/PLP/60663/2004) and WFST (POSI/PLP/47175/2002). INESC-ID Lisboa had support from the POSI Program of the “Quadro Comunitário de Apoio III”.

## References

- [1] C. Chelba and A. Acero. Adaptation of maximum entropy capitalizer: Little data can help a lot. *EMNLP '04*, 2004.
- [2] H. Daumé III. Notes on CG and LM-BFGS optimization of logistic regression. Implementation available at <http://hal3.name/megam/>, August 2004.
- [3] J.-H. Kim and P. C. Woodland. Automatic capitalisation generation for speech input. *Computer Speech & Language*, 18(1):67–90, 2004.
- [4] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel. Performance measures for information extraction. In *Proceedings of DARPA Broadcast News Workshop*, Herndon, VA, February 1999.
- [5] A. Mikheev. Periods, capitalized words, etc. *Computational Linguistics*, 28(3):289–318, 2002.
- [6] A. Stolcke. SRILM - An extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*, volume 2, pages 901–904, Denver, CO, 2002.
- [7] A. Stolcke and E. Shriberg. Automatic linguistic segmentation of conversational speech. In *Proc. ICSLP '96*, volume 2, pages 1005–1008, Philadelphia, PA, 1996.

---

<sup>4</sup> <http://www.tecnovoz.com.pt/>

# Confidence Measures for Stochastic Parsing

José-Miguel Benedí, Joan-Andreu Sánchez and Alberto Sanchis  
Instituto Tecnológico de Informática  
Universidad Politécnica de Valencia  
Camino de Vera s/n, Valencia 46022 (Spain)  
*jbenedi,jandreu,josanna@dsic.upv.es*

## Abstract

In this work the use of confidence measures for detecting errors of stochastic parsing is explored. The confidence measures are based on posterior probabilities computed over a list of the  $n$ -best parse trees. Several confidence measures are proposed and a naive Bayes model is also considered. The proposed confidence measures are tested with the Charniak parser and the Penn Treebank corpus.

## Keywords

Parsing, Confidence measures, Naive Bayes

## 1 Introduction

Syntactic parsing is the process of recognizing an input sentence and assigning to it a syntactic structure. Syntactic parsing [11] is an important problem related to Natural Language Processing (NLP) and thus plays an important role in problems like Semantic Analysis, Question Answering, RNA Modeling [16], Language Modeling [5], and Machine Translation [23, 4], among others. In stochastic syntactic parsing, a syntactic structure (parse tree) is obtained according to some criterion by using a stochastic model and a parsing algorithm. This paper focuses on the use of Stochastic Context-Free Grammars (SCFGs) as stochastic models and context-free parse trees as syntactic structures. SCFGs is a powerful formalism that has been widely used for stochastic syntactic parsing [1, 20, 2, 15, 6]. A large variety of parsing algorithms can be found in the Computer Science and Computational Linguistic literature. Some of the current syntactic parsing algorithms are based on the classical CYK [9] and Earley [7] algorithms. However, in recent years, other syntactic parsers have been considered for real tasks of NLP [2, 15, 10, 6].

Given the difficulty and the importance of parsing in all of these applications, there exists an increasing necessity to detect the erroneous syntactic structures.

Confidence measures have been extensively used in Speech Recognition [22, 18], in Spoken Dialogue System [19] and Statistical Machine Translation [21]. A confidence measure can be defined as the probability of a part (typically word-level) of the output sentence being correctly recognized. Confidence measures have been used for different purposes. They have mainly been used for detecting recognition errors and for improving the recognition accuracy.

In this paper, the use of confidence measures is proposed to detect errors in a statistical syntactic parsing. Therefore, given the output hypothesis provided by a stochastic syntactic parser, the main goal will be to estimate a confidence measure for each syntactic substructure in the output hypothesis, in order to detect those syntactic substructures that are likely to be incorrectly parsed.

The estimation of confidence measures can be seen as a classical pattern recognition problem. A feature vector is obtained for each hypothesized syntactic substructure in order to classify it as either correct or incorrect. Thus, the basic problems will be to find appropriate pattern features and to design an accurate classifier.

N-best lists have been used for different purposes in confidence estimation for Speech Recognition [22] and Machine Translation [21]. The N-best list is a collection of the  $N$  most probable sentences sorted according to their probabilities. They have been used both to directly estimate the confidence measure and to compute predictor features. In this work, N-best parse trees for both purposes are used.

For combining the predictor features, a *smoothed naive Bayes* classification model has been adopted. This model has been successfully used for confidence estimation in Speech Recognition [18, 17]. The model itself is a combination of *tag-dependent* (specific) and *tag-independent* (generalized) naive Bayes models. This classification model provides a sound framework to profitably combine the predictor features.

The paper is organized as follows. A brief review of stochastic syntactic parsing is given in Section 2; Section 3.1 presents the confidence measures and the predictor features used in this work; Section 3.2 describes the naive Bayes classification model; and, finally, Section 4 presents the experimental setup, evaluation metrics and the experimental results.

## 2 Parsing

Stochastic parsing aims to find the most probable parse tree of a given input using a grammatical model and a parsing algorithm.

Some of the current syntactic parsing algorithms [6] are based on the classical CYK [9] and Earley [7] algorithms. The CYK and Earley algorithms are based on dynamic programming scheme. An important problem related with these algorithm is their cubic time complexity.

In recent years, other syntactic parsing algorithms have been considered for real tasks of NLP. In these algorithms other search strategies different from the dynamic programming scheme are used to compute the most probable parsing. In [3], an agenda-based chart parser is described in which the items are chosen from an agenda according to a *figure of merit*. The number of items that are processed before obtaining the most probable parsing is notably less than the number of items that are obtained with an exhaustive search.

In [2], a maximum-entropy-inspired parser is presented. First, a parser that uses a chart together with an agenda is used to generate candidate possible parses. The figure of merit that is used to choose the item from the agenda is defined by using a lexicalized SCFG [3]. Second, a probabilistic model that is based on the maximum entropy principle is used to evaluate the candidates parse trees introduced in the agenda. The parse tree obtained in this way is not guaranteed to be the exact most probable parse tree according to the SCFG. The experiments reported in [3] shown a very good performance on the Penn treebank corpus. Recent improvements on this parsing algorithm achieves about 92% f-measure by using a semi-supervised learning [13].

In [15], a beam-search strategy is used and therefore the optimality of the solution is not guaranteed.

In [10], an A\* algorithm is presented to compute the exact most probable parsing of a string. In this parser the search is driven by a function that guarantees that the best parse string is not lost. In that work, several bounds are proposed for the A\* search, and experimental results are reported for delexicalized strings of the Penn treebank corpus.

The output of all parsers that have been previously described is a parse tree in which each span is labeled with a tag. Obviously, the parsing process introduces errors both in the size of the spans and in the tags. The following section describes the estimation of confidence measures to detect automatically erroneous tags.

### 3 Confidence Measures

Confidence measure estimation can be seen as a two-class pattern recognition problem in which each hypothesized tag is transformed into a vector of *features* and then classified as either correct or incorrect. The basic problem then is to decide which predictor *features* (pattern) and classification model should be used.

#### 3.1 Predictor Features

A *predictor feature* can be defined as a numeric value which is computed automatically for a tag and which helps detecting errors.

A set of *predictor features* based on N-best lists has been selected in order to classify each tag as either correct or incorrect. These features are based on posterior probabilities and they have been successfully applied for confidence estimation in machine translation [21].

Let us assume that, given an input sentence  $s = s_1 \cdots s_{|s|}$ , the parser produces the most probable parse

tree  $\hat{t}$ . The syntactic structure  $\hat{t}$  is composed by substructures that are habitually referred as edges [3]. Let  $t_{ij}^x$  be an edge which represents a syntactic tag  $x$  that covers the substring between positions  $i$  and  $j$ . Let  $\mathcal{L}_N$  be the  $N$ -best parse trees generated by the parser.

For the computation of the features for an edge  $t_{ij}^x$  of  $\hat{t}$ , a subset  $\mathcal{S}_M$  of  $M$  parse trees ( $0 \leq M \leq N$ ) is extracted from  $\mathcal{L}_N$  based on two different criteria:

- *Levenshtein position*:  $\mathcal{S}_M$  is composed of those  $M$  parse trees containing an edge that is aligned with edge  $t_{ij}^x$  by means of an edit distance.
- *Target position*:  $\mathcal{S}_M$  is composed of those  $M$  sentences containing an edge that is aligned with edge  $t_{ij}^x$  in exactly the same position.

Different features can be calculated for each  $t_{ij}^x$  of  $\hat{t}$  as:

$$\mathcal{F}(t_{ij}^x) = \frac{1}{R} \sum_{\hat{t} \in \mathcal{S}_M} W(\hat{t}) \quad (1)$$

Depending on how  $W(\hat{t})$  and  $R$  are defined and computed, two features can be defined:

- *based on tree parse probabilities*:  $W(\hat{t})$  is the posterior probability of  $\hat{t}$ , and  $R$  is computed by summing up the probabilities over all parse trees in the  $N$ -best parse tree list.
- *based on relative frequencies*:  $W(\hat{t})$  is 1 and  $R$  is  $N$ .

Therefore, given an edge  $t_{ij}^x$  of  $\hat{t}$ , 4 different features are computed by using the  $N$ -best parse tree list  $\mathcal{L}_N$ . Table 1 shows the feature acronyms used in the experiments described in Section 4.

	Levenshtein position	Target position
Probabilities	ProbLev	ProbTarget
Frequencies	FreqLev	FreqTarget

Table 1: Four predictor features used in this work.

#### 3.2 Naive Bayes model

We have adopted a *smoothed naive Bayes* classification model for obtaining the confidence measures. This model has been successfully used for speech confidence estimation [18].

The class variable is denoted by  $c$ ;  $c = 0$  for correct and  $c = 1$  for incorrect. Given an edge  $e$  and a  $D$ -dimensional vector of features  $\mathbf{x}$ , the class posteriors can be calculated via the Bayes' rule as

$$P(c|\mathbf{x}, e) = \frac{P(c|e) P(\mathbf{x}|c, e)}{\sum_{c'} P(c'|e) P(\mathbf{x}|c', e)} \quad (2)$$

For simplicity, the model includes the naive Bayes assumption that the features are mutually independent given a class-edge pair,

$$P(\mathbf{x}|c, e) = \prod_{d=1}^D P(x_d|c, e) \quad (3)$$



Therefore, the basic problem is to estimate  $P(c|e)$  for each edge and  $P(x|c, e)$  for each class-edge pair. Given  $T$  training samples  $\{(\mathbf{x}_n, c_n, e_n)\}_{n=1}^T$  obtained from all the edges of the most probable parse trees associated to the sentences of a training corpus, the unknown probabilities can be estimated using the conventional frequencies:

$$P(c|e) = \frac{N(c, e)}{N(e)} \quad (4)$$

$$P(x_d|c, e) = \frac{N(x_d, c, e)}{N(c, e)} \quad (5)$$

where the  $N(\cdot)$  are suitably defined event counts; i.e., the events are  $(c, e)$  pairs in (4) and  $(x_d, c, e)$  triplets in (5).

In practice, some features may have continuous rather than discrete domains. In that case, the use of Eq. 5 requires the discretization of continuous features. This is performed by dividing the feature domain into a fixed number of evenly-spaced bins of fixed size (usually around 20). The minimum, maximum and bin size are set by visual inspection of the histograms of the features of the examples from the correct and incorrect classes. Given this information, the naive Bayes implementation includes a function that maps the continuous feature value  $x_d$  to the corresponding discrete bin number.

Unfortunately, these frequencies often underestimate the true probabilities involving rare edges and the incorrect class. To circumvent this problem, the model is smoothed using the *absolute discounting* smoothing technique imported from statistical language modelling [14]. The idea is to discount a small constant  $b \in (0, 1)$  to every positive count and then distribute the gained probability mass among the null counts (unseen events). A detailed explanation of the smoothed model can be found in [18].

Once the parameters of the model are estimated, in the test phase, a edge is classified as incorrect if the confidence estimation  $P(c = 1|x, e)$  is greater than a certain threshold  $\tau$ .

## 4 Experiments

### 4.1 Evaluation Metrics

Given a certain parser, it produces a set  $N_c$  of edges that are labeled as correct and  $N_i$  edges that are labeled as incorrect. Then, after confidence classification is performed for a certain classification threshold  $\tau$ , the result achieved is a set  $N_f(\tau)$  ( $0 \leq N_f(\tau) \leq N_c$ ) of edges labeled as correct which are classified as incorrect (false rejection), and  $N_t(\tau)$  ( $0 \leq N_t(\tau) \leq N_i$ ) labeled as incorrect which are classified as incorrect (true rejection).

Based on the false rejection  $N_f(\tau)$  and the true rejection  $N_t(\tau)$ , two measures are of interest for the evaluation of confidence estimation:

1. The *False Rejection Rate* (FRR), defined as:

$$R_f(\tau) = \frac{N_f(\tau)}{N_c} \quad (6)$$

2. The *True Rejection Rate* (TRR), defined as:

$$R_t(\tau) = \frac{N_t(\tau)}{N_i} \quad (7)$$

The trade-off between  $R_f$  and  $R_t$  values depends on the decision threshold  $\tau$ . A *Receiver Operating Characteristic* (ROC) [8] curve represents  $R_f$  against  $R_t$  for different values of  $\tau \in [0, 1]$ .

The area under a ROC curve divided by the area of a worst-case diagonal ROC curve, provides an adequate overall estimation of the classification accuracy. We denote this area ratio as AROC. The AROC value is in the range of 1.0 to 2.0, with 1.0 corresponding to a random classification of correct and incorrect edges, and 2.0 would indicate that all edges can be correctly classified.

Another different criterion is the *Confidence Error Rate* (CER). This metric is defined as the number of classification errors divided by the total number of classified syntactic edges. Thus, the CER value also depends on the decision threshold  $\tau$ . CER can be computed as:

$$CER(\tau) = \frac{N_f(\tau) + (N_i - N_t(\tau))}{N_c + N_i} \quad (8)$$

A baseline CER is obtained assuming that all syntactic edges are classified as correct. Then, the baseline CER is computed as:

$$CER_{baseline} = \frac{N_i}{N_c + N_i} \quad (9)$$

### 4.2 Experimental setup

The experiment of this section were carried out with the Penn Treebank corpus [12] and the Charniak parser<sup>1</sup> [2]. Such as it is described in [2], this parser was trained on sections 2-21 of the Penn Treebank corpus.

For the experiments regarding confidence measures, section 24 was used for training the naive Bayes model, section 22 was used as development and section 23 was used for test. No length restriction was imposed on the training, development and test corpora. For each sentence of the training, development and test corpus, we computed at most the 5,000 best parses. For the test corpus, 7.6% of the sentences had less than 5,000 parses available, the average number of  $N$ -best parses per sentence was 4,719 and the typical deviation was 1,043.

### 4.3 Automatic labeling of edges

In order to evaluate the performance of the automatic confidence estimation and to estimate the parameters of the classification model described in Section 3, it is necessary to label as correct or incorrect each edge of the most probable parse tree of all sentences of corpora.

Given that a manually annotated parse tree is available for each sentence of the Penn Treebank corpus,

<sup>1</sup> ftp://ftp.cs.brown.edu/pub/nlparser/

the automatic labeling of edges was based on an edition distance process between the manually annotated reference edges and the edges of the most probable parse sentences. The same edition distance is computed between the most probable parse sentences and the  $N$ -best parse trees in order to compute the predictor features.

#### 4.4 Results

The unknown probabilities of the *smoothed naive Bayes* model, described in Section 3.2, were estimated using the training corpus. Different smooth parameters of the model were optimized using the development corpus. Also, the development corpus was used to find the best classification threshold  $\tau$  i.e., that with minimum  $CER(\tau)$ .

The performance of each single feature has been evaluated in two different manners. On the one hand, the features have been used directly as confidence measures. On the other hand, the smoothed naive Bayes model based on one-dimensional feature vectors is used. Additionally, to further exploit the usefulness of the features, the naive Bayes model was employed to explore the performance of different feature combinations. All the possible combinations using the 4 predictor features (summarized in Table 1) were evaluated.

The test corpus was classified using the best classification threshold  $\tau$  for which minimum CER was achieved over the development corpus. This implies that for the test corpus only CER values can be computed since ROC curve plots the performance for all possible classification thresholds.

Tables 2 and 3 show the best results achieved using only the *feature* as confidence measure or the one-dimensional *Naive Bayes* model over the development and the test corpus, respectively. Also the result of the best feature combination is showed.

Feature	Technique			
	Feature		Naive Bayes	
	AROC	CER	AROC	CER
FreqLev	1.61	7.59	1.65	7.60
ProbLev	1.61	7.59	1.65	7.60
FreqTarget	1.40	8.01	1.56	7.95
ProbTarget	1.40	8.01	1.56	7.95
FreqLev+FreqTarget	-	-	1.66	7.58
Baseline	-	8.01	-	8.01

**Table 2:** CER and AROC values in the development corpus.

Feature	Technique	
	Feature	Naive Bayes
FreqLev	7.23	7.38
ProbLev	7.25	7.39
FreqTarget	7.61	7.56
ProbTarget	7.61	7.56
FreqLev+FreqTarget	-	7.43
Baseline	7.61	

**Table 3:** CER values in the test corpus.

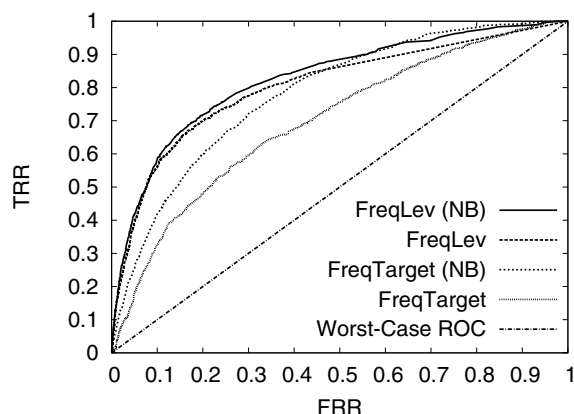
Consistently for both corpus the features computed

based on edition distance (FreqLev and ProbLev) achieved the best classification accuracy. The features based on target positions perform worst. Similar performance is obtained by using frequencies or probabilities.

In general, the naive Bayes model produced overall better performance than using directly the features as confidence measures. This can be appreciated by AROC values over the development corpus (note that this value ranges from 1.0 to 2.0). Figure 1 shows the comparative ROC curves between *FreqLev* and *FreqTarget* features. Note that the nearer ROC curve lies close to the upper left corner of the graph (and away from the diagonal), better is the performance of the confidence measure. The diagonal worst-case ROC curve is also plotted for better appreciate the confidence measure performance.

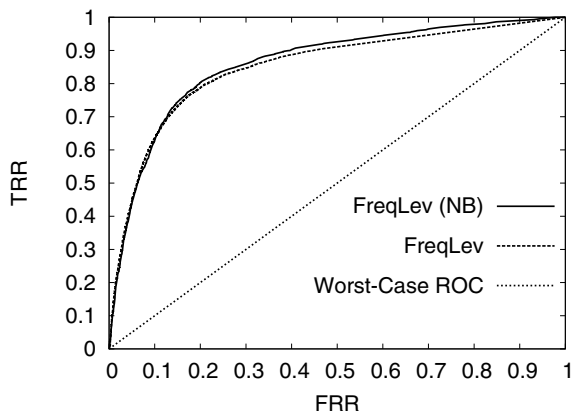
However, the CER values were very similar between both techniques even, for the test corpus, the best CER values were obtained using the features directly. The combination of features did not improve the single performance. This was surprising since it has been proved that (naive Bayes) feature combination produces better accuracy than the use of single features [18]. The use of very similar predictor features seemed to be the reason of this effect.

The ROC curve of the best feature *FreqLev* computed over the test corpus is showed in Figure 2. It can be observed that the naive Bayes model slightly improved the direct use of the feature as confidence measure.



**Fig. 1:** ROC curves on the development corpus for the single features. NB corresponds to the Naive Bayes model performance.

In order to analyze the classification accuracy for each sort of tags separately, they were divided in two separate classes: syntactical and lexical tags. Tables 4 and 5 summarize the classification results achieved for syntactical and lexical tags, respectively. The tags are sorted by frequency of occurrence. The parser produced more errors in the syntactical tags. The baseline CER was approximately three times upper than in the lexical tags. In fact, the baseline CER for the lexical tags was very low (4.3%). This causes to be very difficult to reduce the CER baseline for this class. More effort seems to be necessary to apply for syntactical class



**Fig. 2:** ROC curves on the test corpus for the best single feature. NB corresponds to the Naive Bayes model performance.

since near of 70% of the parser errors belong to this kind of tags. An important feature is that the highest number of parser errors for this class corresponded to the tags with lower frequency of occurrence. Specifically for tags with a frequency lower than 2% the CER baseline is 22.8%. For this kind of tags a significant reduction on the CER baseline was achieved. Also acceptable classification accuracy was obtained for the three most frequent syntactical tags.

Edge	Rel. Fr.	Basel.	CER	Rel.Red.	AROC
NP	18.2	10.7	9.9	7.5	1.64
VP	8.9	10.7	9.9	7.5	1.44
PP	5.7	15.7	13.9	11.5	1.61
S	5.5	9.7	9.4	3.1	1.60
Other	5.5	22.8	19.8	13.2	1.42
All	43.8	12.8	11.8	7.8	1.58

**Table 4:** Results for the syntactic edges on the development corpus (feature FreqLev). Tag of edge, Relative frequency of occurrence, CER Baseline, CER, Relative Reduction [%] of CER Baseline, and AROC values are showed for each edge.

## 5 Conclusions

In this work, we have explored and proposed the use of confidence measures for parsing. We have adopted a general layout which have been successfully used for confidence estimation in speech recognition and machine translation. Predictor features based on N-best parse trees have produced improvements on the baseline system performance. The syntactical edges appeared as the most important edges for confidence estimation since it represented the 70% of the parsing errors. For future work, new features should be explored in order to profitably combine them in a solid framework. Better accuracy classification is expected by using the (naive Bayes) combination of the features.

Edge	Rel. Fr.	Basel.	CER	Rel.Red.	AROC
NN	7.5	4.3	4.1	3.7	1.33
IN	6.1	1.8	1.6	10.9	1.15
NNP	5.5	2.9	2.4	19.1	1.74
DT	5.0	0.7	0.7	0.0	1.27
NNS	3.8	4.4	3.8	14.8	1.40
JJ	3.6	12.7	10.2	20.0	1.62
,	2.9	1.1	1.1	0.0	0.99
AUX	2.1	0.0	0.0	0.0	2.00
.	2.1	2.3	2.3	0.0	1.00
RB	2.0	12.5	6.4	48.8	1.81
CD	2.0	4.0	4.0	0.0	0.92
CC	1.5	0.8	0.4	50.0	1.43
TO	1.4	2.7	2.7	0.0	0.93
VB	1.3	7.9	7.4	5.9	1.42
VBD	1.3	4.8	4.8	0.0	1.08
VBN	1.1	8.5	7.9	6.6	1.24
Other	7.0	6.2	6.1	1.6	1.34
All	56.2	4.3	4.3	0.0	1.47

**Table 5:** Results for the lexical edges on the development corpus (feature FreqLev). Tag of edge, Relative frequency of occurrence, CER Baseline, CER, Relative Reduction [%] of CER Baseline, and AROC values are showed for each edge.

## Acknowledgment

This work has been partially supported by the *Universidad Politécnic de Valencia* with the ILETA project and by the EC (FEDER) and the Spanish MEC under grant TIN2006-15694-CO2-01.

## References

- [1] J. Baker. Trainable grammars for speech recognition. In Klatt and Wolf, editors, *Speech Communications for the 97th Meeting of the Acoustical Society of America*, pages 31–35. Acoustical Society of America, June 1979.
- [2] E. Charniak. A maximum-entropy-inspired parser. In *Proc. of NAACL-2000*, pages 132–139, 2000.
- [3] E. Charniak, S. Goldwater, and M. Johnson. Edge-based best-first chart parsing. In *Sixth Workshop on Very Large Corpora*, pages 127–133, 1998.
- [4] E. Charniak, K. Knight, and K. Yamada. Syntax-based language models for statistical machine translation. In *Proc. of MT Summit IX*, New Orleans, USA, September 2003.
- [5] C. Chelba and F. Jelinek. Structured language modeling. *Computer Speech and Language*, 14:283–332, 2000.
- [6] M. Collins. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637, 2003.
- [7] J. Earley. An efficient context-free parsing algorithm. *Communications of the ACM*, 8(6):451–455, 1970.
- [8] J. P. Egan. Signal detection theory and roc analysis. *Academic Press.*, 1975.
- [9] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.
- [10] D. Klein and C. D. Manning. A\* parsing: Fast exact viterbi parse selection. In *Proceedings of HLT-NAACL 03*, 2003.
- [11] M. Lease, E. Charniak, M. Johnson, and D. McClosky. A look at parsing and its applications. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*, 16–20 July 2006.
- [12] M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

- [13] D. McClosky, E. Charniak, and M. Johnson. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL'06)*, pages 337–344, Sydney, Australia, July 2006. Association for Computational Linguistics.
- [14] H. Ney, S. Martin, and F. Wessel. Statistical language modeling using leaving-one-out. *Young and Bloothoft, editors, Corpus Based Methods in Language and Speech Processing*, pages 174–207, 1997.
- [15] B. Roark. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249–276, 2001.
- [16] I. Salvador and J. Benedí. Rna modeling by combining stochastic context-free grammars and n-gram models. *International Journal of Pattern Recognition and Artificial Intelligence*, 16(3):309–315, 2002.
- [17] A. Sanchis. *Estimación y aplicación de medidas de confianza en reconocimiento automático del habla*. PhD thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, May 2004.
- [18] A. Sanchis, A. Juan, and E. Vidal. Improving utterance verification using a smoothed naive bayes model. In *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 592–595, 2003.
- [19] R. S. Segundo, B. Pellom, K. Hacioglu, W. Ward, and J. Pardo. Confidence measures for spoken dialogue systems. In *Proc. of ICASSP*, 2001.
- [20] A. Stolcke. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–200, 1995.
- [21] N. Ueffing and H. Ney. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33(1):9–40, 2007.
- [22] F. Wessel, R. Schlüter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Trans. on Speech and Audio Processing*, 3(9):288–298, 2001.
- [23] K. Yamada and K. Knight. A Decoder for Syntax-based Statistical MT. In *Meeting of the Association for Computational Linguistics*, 2002.

# Unsupervised Part-Of-Speech Tagging Supporting Supervised Methods

Chris Biemann  
University of Leipzig, NLP Dept.  
Johannisgasse 26  
04103 Leipzig, Germany  
biem@informatik.uni-leipzig.de

Claudio Giuliano  
ITC-IRST  
Via Sommarive, 18  
I-38050 Povo (Trento), Italy  
giuliano@itc.it

Alfio Gliozzo  
ITC-IRST  
Via Sommarive, 18  
I-38050 Povo (Trento), Italy  
gliozzo@itc.it

## Abstract

This paper investigates the utility of an unsupervised part-of-speech (PoS) system in a task oriented way. We use PoS labels as features for different supervised NLP tasks: Word Sense Disambiguation, Named Entity Recognition and Chunking. Further we explore, how much supervised tagging can gain from unsupervised tagging. A comparative evaluation between variants of systems using standard PoS, unsupervised PoS and no PoS at all reveals that supervised tagging gains substantially from unsupervised tagging. Further, unsupervised PoS tagging behaves similarly to supervised PoS in Word Sense Disambiguation and Named Entity Recognition, while only chunking benefits more from supervised PoS. Overall results indicate that unsupervised PoS tagging is useful for many applications and a veritable low-cost alternative, if none or very little PoS training data is available for the target language or domain.

## Keywords

Unsupervised PoS Tagging, Named Entity Recognition, Word Sense Disambiguation, Chunking

## 1. Introduction

Even if, in principle, supervised approaches reach the best performance in many NLP tasks, in practice it is not always easy to make them work in applicative settings. In fact, supervised systems require to be trained on a large amount of manually provided annotations. In most of the cases this scenario is quite unpractical, if not infeasible. In the NLP literature the problem of providing large amounts of manually annotated data is known as the knowledge acquisition bottleneck. A promising direction to tackle this problem is to provide unlabeled data together with labeled texts, which is called semi-supervised learning.

The underlying idea behind our approach is that syntactic similarity of words is an inherent property of corpora, and it can be exploited to help a supervised classifier to build a better categorization hypothesis, even if the amount of labeled training data provided for learning is very low.

Previous work on distributional clustering for word class induction was mostly not evaluated in an application-based way. [4] and [7] state that their clustering examples look plausible. [17], [5] and [8] evaluate their tagging by comparing it to predefined tagsets. Notable exceptions to this are [20], where distributional clustering supports a

supervised PoS tagger (see Section 3.1), and the incorporation of an unsupervised tagger into a NER system in [9] (see Section 4.3).

This is, to our knowledge, the first comprehensive study on the utility of distributional word classes for a variety of NLP tasks. As the same unsupervised tagger is used for all tasks tested, we show the robustness of the system across tasks and languages.

In this work, the unsupervised PoS tagger as described in [2] is evaluated by testing performance of applications equipped with this tagger. Section 2 is devoted to a short description of the tagger; Section 3 lays out the systems the tagger has been incorporated into. In Section 4, evaluation results examine the competitiveness of the unsupervised tagger, Section 5 concludes.

## 2. Unsupervised PoS tagging

Unlike in standard (supervised) PoS tagging, the unsupervised variant relies neither on a set of predefined categories, nor on any labeled text. As a PoS tagger is not an application of its own right, but serves as a preprocessing step for systems building upon it, the names and the number of syntactic categories is very often not important.

The basic procedure behind our unsupervised PoS tagging is as follows: (i) (soft) clusters of contextually similar words are identified, each class is assumed being a different PoS, and (ii) words belonging to more than one class are disambiguated by considering the context in which they are located. The clustering methodology at the basis of the first step is motivated by the fact that words belonging to the same syntactic classes can be substituted in the same context producing grammatical sentences as well, leading us to adopt contextual similarity features for clustering.

For a detailed description of the unsupervised PoS tagger system, we refer to [2]. Increased lexicon size up to some 50,000 words is the main difference between this and other approaches (cf. Section 1.1), that typically operate with 5,000 clustered words. The tagsets obtained with this method are usually more fine-grained than standard tagsets and reflect syntactic as well as semantic similarity.

In [2], the tagger output was directly evaluated against supervised taggers for English, German and Finnish via information-theoretic measures. While it is possible to

relatively compare the performance of different components of a system or different systems along this scale, it does only give a poor impression on the utility of the unsupervised tagger's output. Therefore, an application-based evaluation is undertaken here.

Corpus	BNC	CLEF	Wortschatz
Language	English	Dutch	German
Size (Tokens)	100M	70M	755M
Nr. of Tags	344	418	511
Lexicon Size	25706	21863	74398

**Table 2: Three corpora used for the induction of tagger models. BNC = British National Corpus, for CLEF see [14], Wortschatz is described in [15]**

To induce tagger models, three different corpora are used in our experiments. Table 2 lists some corpus characteristics as well as quantitative data of the respective tagger model.

### 3. Supervised NLP Systems

In this section, the systems that are used for evaluation are described: a simple Viterbi trigram tagger as used in [2], the supervised WSD system of [10], and the simple NER and chunking systems we set up.

In the design of all of these systems, the task is perceived as a machine learning exercise: the PoS tagger component provides some of the features that are used to learn a function that assigns a label to unseen examples, characterized by the same set of features as the examples used for training.

The systems were chosen to cover a wide range of machine learning paradigms: Markov chains in the PoS tagging system, kernel methods in the WSD system and Conditional Random Fields (CRFs, see [11]) for NER and chunking.

#### 3.1 PoS Tagger

The tagger employed in [2] is a very simple trigram tagger that does not use parameter re-estimation or smoothing techniques. It was designed to be trained from large amounts of unlabeled data, arguing that increasing training data will lead to better results than increasing model complexity, cf. [1]. For training, the frequency of tag trigrams and the number of times each word occurs with each tag are counted and directly transformed into (transition) probabilities by normalization.

The sequence of tags for a chunk of text is found by maximizing the probability of the joint occurrence of tokens  $T=(t_i)$  and categories/tags  $C=(c_i)$  for a sequence of length  $n$ :

$$P_{plain}(T, C) = \prod_{i=1}^n P(c_i | c_{i-1}, c_{i-2}) P(c_i | t_i).$$

In the unsupervised case, the transition probabilities  $P(c_i | c_{i-1}, c_{i-2})$  are only estimated from trigrams where all three tags are present. In the supervised case, tags are provided for all tokens in the training corpus. The probability  $P(c_i | t_i)$ <sup>1</sup> is obtained from the tagger's lexicon and equals 1 if  $t_i$  is not contained.

For the incorporation of unsupervised tags, another factor  $P(c_i | u_i)$  is introduced that accounts for the fraction of times the supervised tag  $c_i$  was found together with the unsupervised tag  $u_i$  in the training text, which has been tagged with the unsupervised tagger before:

$$P_{unsu}(T, C) = \prod_{i=1}^n P(c_i | c_{i-1}, c_{i-2}) P(c_i | t_i) P(c_i | u_i).$$

Notice that only the unsupervised tag at the same position influences the goal category in this simple extension. Using surrounding unsupervised tags would be possible, but was not carried out. More elaborate strategies, like morphological components as in [3] or the utilization of a more up-to-date tagger model, are not considered here. The objective is to examine the influence of unsupervised tags, not to construct a state of the art PoS tagger.

A somewhat related strategy is described in [20], where a hierarchical clustering of words was used for reducing the error rate of a decision-tree-based tagger up to 43%, achieving 87% accuracy on a fine-grained tagset. However, the improvements were reached by manually adding rules that made use of the cluster IDs yielded by a word clustering method and this approach therefore caused extra work as opposed to narrowing down the acquisition bottleneck.

#### 3.2 Word Sense Disambiguation (WSD)

For performing WSD, we used a state of the art supervised WSD methodology based on a combination of syntagmatic and domain kernels [10] in a Support Vector Machine classification framework.

Kernel WSD basically takes two different aspects of similarity into account: domain aspects, mainly related to the topic (i.e. the global context) of the texts in which the word occurs, and syntagmatic aspects, concerning the lexical-syntactic pattern in the local contexts. Domain aspects are captured by the *domain kernel*, while syntagmatic aspects are taken into account by the *syntagmatic kernel*.

For our experiments, we substitute the sequences of PoS required by the syntagmatic kernel by using

<sup>1</sup> Although [6] report that using  $P(t_i | c_i)$  instead leads to superior results in the supervised setting, we use the 'direct' lexicon probability, which does not require smoothing and re-estimation. For the purely unsupervised setting, this does not affect results negatively, as a much larger training corpus levels out the effects measured in [6].

unsupervised PoSs, comparing the results obtained with different combinations.

### 3.3 Named Entity Recognition and Chunking

For performing chunking and NER, we perceived these applications as a tagging task. For both tasks, we train the MALLET tagger<sup>2</sup>.

The tagger operates on a different set of features for our two tasks. In the NER system, the following features are accessible, time-shifted by -2, -1, 0, 1, 2: a) Word itself, b) PoS-tag, c) Orthographic predicates and d) Character bigram and trigram predicates.

In the case of chunking, features are only time-shifted by -1, 0, 1 and consist only of: a) Word itself and b) PoS-tag.

Per system, three experiments were carried out, using standard PoS features, unsupervised PoS features and no PoS features.

## 4. Evaluation

The systems are tested in a standard way on annotated resources. For supervised PoS tagging, we evaluate on the German NEGRA corpus [18]. The English lexical sample task (fine-grained scoring) of Senseval-3 [12] is chosen for WSD. For NER, the Dutch dataset of CoNLL-2002 [16] is employed, and the evaluation set for English chunking is the CoNLL-2000 dataset [19]. The supervised PoS tags for WSD, NER and chunking were provided in the respective datasets.

Supervised PoS tagging is measured in accuracy, which is obtained through dividing the number of correctly classified instances by the total number of instances. For NER and chunking, results are reported in terms of the F1<sup>3</sup> measure. WSD performance is measured using the scorer provided by Senseval-3. All evaluation results are compared in a pair wise fashion using the approximate randomization procedure of [13] as significance test.

### 4.1 Unsupervised PoS for supervised PoS

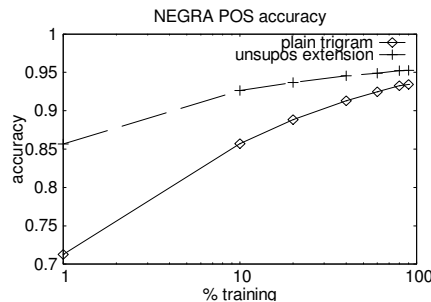
To evaluate the influence of unsupervised tags on a supervised tagger, training sets of varying sizes were selected randomly from the 20,000 sentences of NEGRA corpus, the remainder was used for evaluation. We compare the performance of the plain Viterbi tagger with the performance of the tagger using unsupervised tags (cf. formulae in section 3.1), which were obtained by tagging the NEGRA corpus with a tagger model induced on the Wortschatz corpus, which is 2,000 times larger. Results are reported in tagging accuracy, averaged over three different

<sup>2</sup> <http://mallet.cs.umass.edu>

<sup>3</sup>  $F1 = \frac{2PR}{P+R}$  with  $P = \frac{\#correct}{\#classified}$ ,  $R = \frac{\#correct}{\#total}$

splits per training size each. Figure 1 shows the learning curve.

Results indicate that supervised tagging can clearly benefit from unsupervised tags: already at 20% training with unsupervised tags, the performance on 90% training without the unsupervised extension is surpassed. At 90% training, error rate reduction is 27.8%, indicating that the unsupervised tagger grasps very well the linguistically motivated syntactic categories and provides a valuable feature to either reduce the size of the required annotated training corpus or to improve overall accuracy. Despite its simplicity, the unsupervised extension does not fall too short of the performance of [3], where an accuracy of 0.967 at 90% training on the same corpus is reported.



%	1	10	20	40	60	80	90
plain	0.713	0.857	0.888	0.913	0.925	0.933	0.934
unsu.	<b>0.857</b>	<b>0.926</b>	<b>0.937</b>	<b>0.946</b>	<b>0.949</b>	<b>0.952</b>	<b>0.953</b>

Figure 1: Learning curve for supervised PoS tagging with and without using unsupervised PoS tags (accuracy)

### 4.2 Unsupervised PoS for WSD

The modularity of the kernel approach makes it possible to easily compare systems with different configurations by testing various kernel combinations. To examine the influence of PoS tags, two comparative experiments were undertaken.

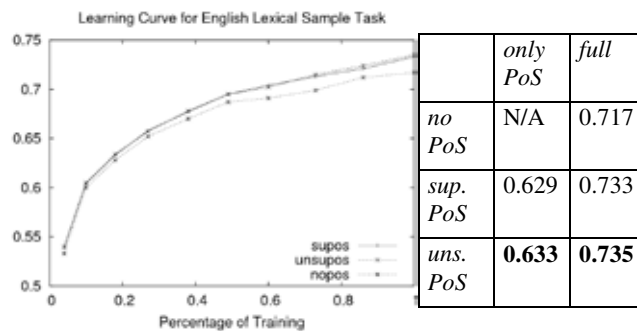


Figure 2: Comparative evaluation on Senseval scores for WSD and learning curve. No differences are significant at  $p < 0.1$

The first experiment uses only the PoS kernel, i.e. the PoS labels are the only feature visible to the learning and classification algorithm. In a second experiment, the full system of [10] is tested against replacing the original PoS kernel with the unsupervised PoS kernel and omitting the

PoS kernel completely. Figure 2 summarizes the results in terms of accuracy.

Results show that PoS information generally contributes to a small extent to WSD accuracy in the full system. Using the unsupervised PoS tagger results in a slight performance increase, improving over the state of the art results in this task, that have been previously achieved by [10]. However, the learning curve suggests that it does not matter whether to use supervised or unsupervised tagging.

From this, we conclude that supervised tagging can safely be exchanged in kernel WSD with the unsupervised variant. Replacing the only preprocessing step that is dependent on manual resources in the system of [10], state of the art supervised WSD is proven to not being dependent on any linguistic preprocessing at all.

### 4.3 NER Evaluation

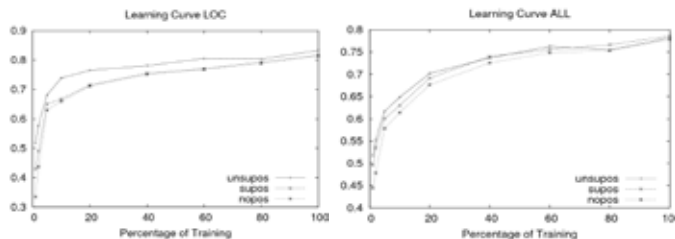
To evaluate the performance on NER, we employ the methodology as proposed by the providers of the CoNLL-2002 dataset. We provide no PoS information, supervised PoS information and unsupervised PoS information to the system and measure the difference in performance in terms of F1. Table 3 summarizes the results for this experiment for selected categories using the full train set for training and evaluating on the test data.

**Table 3: Comparative evaluation of NER on the Dutch CoNLL-2002 dataset in terms of F1. All differences are not significant with  $p < 0.1$**

Category	PER	ORG	LOC	MISC	ALL
no PoS	0.8084	<b>0.7445</b>	0.8151	0.7462	0.7781
su. PoS	<b>0.8154</b>	0.7418	0.8156	<b>0.7660</b>	<b>0.7857</b>
un. PoS	0.8083	0.7357	<b>0.8326</b>	0.7527	0.7817

The figures in table 3 indicate that PoS information is hardly contributing anything to the system's performance, be it supervised or unsupervised. This indicates that the training set is large enough to compensate for the lack of generalization when using no PoS tags, in line with e.g. [1]. The situation changes when taking a closer look on the learning curve, produced by using train set fractions of differing size. Figure 3 shows the learning curves for the categories *LOCATION* and the micro average F1 evaluated over all the categories (ALL).

On the *LOCATION* category, unsupervised PoS tags provide a high generalization power for a small number of training samples. This is due to the fact that the induced tagset treats locations as a different tag; the tagger's lexicon plays the role of a gazetteer in this case, comprising 765 lexicon entries for the location tag. On the combination of ALL categories, this effect is smaller, yet the incorporation of PoS information outperforms the system without PoS for small percentages of training.



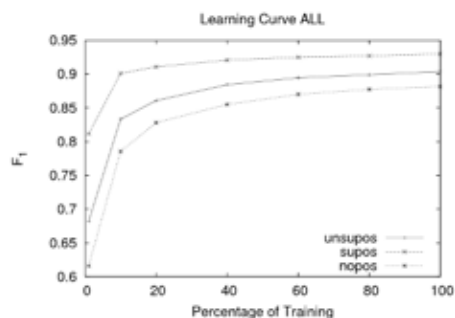
**Figure 3: Learning curves in NER task in F1 for category LOC and combined category**

This disagrees with the findings of [9], where features produced by distributional clustering were used in a boosting algorithm. Freitag reports improved performance on *PERSON* and *ORGANISATION*, but not on *LOCATION*, as compared to not using a tagger at all. In [9], however, a different training corpus for PoS induction and English NER data was used.

Experiments on NER reveal that PoS information is not making a difference, as long as the training set is large enough. For small training sets, usage of unsupervised PoS features result in higher performance than supervised or no PoS, which can be attributed to its more fine-grained tagset.

### 4.4 Chunking Evaluation

For testing performance of our simple chunking system, we used different portions of the training set as given in the CoNLL-2000 data and evaluated on the provided test set. Performance is reported in Figure 4.



**Figure 4: Learning curve for the chunking task in terms of F1. Performance at 100% training is 0.882 (no PoS), 0.904 (unsupervised PoS) and 0.930 (supervised PoS), respectively**

As PoS is the only feature that is used here apart from the word tokens themselves, and chunking reflects syntactic structure, it is not surprising that providing this feature to the system results in increased performance: both kinds of PoS significantly outperform not using PoS ( $p < 0.01$ ).

In contrast to the previous systems tested, using the supervised PoS labels resulted in a significantly better chunking ( $p < 0.01$ ) than using the unsupervised labels. This can be attributed to the fact that both supervised tagging and chunking aim at reproducing the same perception of syntax, which does not necessarily fit the distributionally acquired classes of an unsupervised system. Anyhow, the use of unsupervised PoS provide very useful information to



the chunking learning process, demonstrated by the fact that the use of unsupervised PoS improves significantly the baseline provided by the system trained without PoS.

Despite the low number of features, the chunking system using supervised tags compares well with the best system in the CoNLL-2000 evaluation (F1=0.9348).

## 5. Conclusion

To summarize our results, we have shown that employing unsupervised PoS tags as features are useful in many NLP tasks. Improvements over the pure word level could be observed in all systems tested. We demonstrated that especially if few training data or no supervised PoS tagger is available, using this low-cost alternative leads to significantly better performance and should be used beyond doubt. In addition, unsupervised PoS tagging can be used to improve supervised PoS tagging, especially as far as the learning curve is concerned.

Comparing the two kinds of PoS tags tested, we observed that the performances achieved by the final systems are comparable in all tasks but chunking. In addition, we reported a slight improvement on WSD.

Another conclusion is that, in general, the more training data is provided, the lower the gain of using PoS tagging in supervised NLP, either if PoS tags are supervised or not. Even if this result is in itself not very interesting from our particular point of view, being in line with learnability theory, it confirms our basic motivation of adopting unsupervised PoS tagging for minority languages and, in general, for all those linguistic processing systems working with very limited manually tagged resources but huge unlabeled datasets. This situation is very common in Information Retrieval systems, and in all applications dealing with highly specialized domains (e.g. bioinformatics). In the future we plan to apply our technology to a Multilingual Knowledge Extraction scenario working on web scale corpora.

## Acknowledgements

Alfio Gliozzo was supported by the FIRB-israel co-founded project N.RBIN045PXH. Claudio Giuliano was supported by the X-Media project (<http://www.x-media-project.org>), sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-026978.

## 6. References.

- [1] M. Banko and E. Brill. 2001. Scaling to Very Very Large Corpora for Natural Language Disambiguation. In Proceedings of ACL-01, pp. 26-33, Toulouse, France
- [2] C. Biemann. 2006. Unsupervised Part-of-Speech Tagging Employing Efficient Graph Clustering. In Proceedings of the COLING/ACL-06 Student Research Workshop, Sydney, Australia
- [3] T. Brants. 2000. TnT - a statistical part-of-speech tagger. In Proceedings of ANLP-2000, Seattle, USA
- [4] P. F. Brown, V. J. Della Pietra, P. V. DeSouza, J. C. Lai and R. L. Mercer. 1992. Class-Based n-gram Models of Natural Language. Computational Linguistics 18(4), pp. 467-479
- [5] A. Clark. 2003. Combining Distributional and Morphological Information for Part of Speech Induction, In Proceedings of EACL-03, Budapest, Hungary
- [6] E. Charniak, C. Hendrickson, N. Jacobson and M. Perkowski. 1993. Equations for part-of-speech tagging. In Proceedings of the 11<sup>th</sup> Natl. Conference on AI, pp. 784-789, Menlo Park
- [7] S. Finch and N. Chater. 1992. Bootstrapping Syntactic Categories Using Statistical Methods. In Proc. 1st SHOE Workshop. Tilburg, The Netherlands
- [8] D. Freitag. 2004a. Toward unsupervised whole-corpus tagging. In Proceedings of COLING-04, pp. 357-363, Geneva, Switzerland
- [9] D. Freitag. 2004b. Trained named entity recognition using distributional clusters. In Proceedings of EMNLP 2004, pp. 262-269, Barcelona, Spain
- [10] A. M. Gliozzo, C. Giuliano and C. Strapparava. 2005. Domain Kernels for Word Sense Disambiguation. In Proceedings of ACL-05, pp. 403-410, Ann Arbor, Michigan, USA
- [11] J. Lafferty, A. K. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of ICML-01, pages 282-289
- [12] R. Mihalcea, T. Chklovsky and A. Kilgarrieff. 2004. The Senseval-3 English lexical sample task. In Proceedings of Senseval-3, Barcelona, Spain.
- [13] E. W. Noreen. 1989. Computer-Intensive Methods for testing Hypothesis. John Wiley & Sons, New York
- [14] C. Peters. 2006. Working notes for the CLEF 2006 Workshop. Alicante, Spain
- [15] U. Quasthoff, M. Richter and C. Biemann: Corpus Portal for Search in Monolingual Corpora. In Proceedings of LREC 2006, pp. 1799-1802, Genova, Italy
- [16] D. Roth and A. van den Bosch. Editors. 2002. Proceedings of the Sixth CoNLL Workshop, Taipei, Taiwan
- [17] H. Schütze. 1995. Distributional part-of-speech tagging. In Proceedings of EACL 7, pp. 141-148
- [18] W. Skut, B. Krenn, T. Brants and H. Uszkoreit. 1997. An Annotation Scheme for Free Word Order Languages. In Proceedings of the ANLP-97. Washington, DC, USA
- [19] E. F. Tjong Kim Sang and S. Buchholz. 2000. Introduction to the CoNLL-2000 Shared Task: Chunking. In Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal
- [20] A. Ushioda. 1996. Hierarchical clustering of words and applications to NLP tasks. In Proceedings of the Fourth Workshop on Very Large Corpora, pp. 28-41, Somerset, NJ, USA

# Generating models for temporal representations

Patrick Blackburn  
INRIA Lorraine  
615 rue du Jardin Botanique  
54602 Villers ls Nancy Cedex, France  
*Patrick.Blackburn@loria.fr*

Sébastien Hinderer  
INRIA Lorraine  
615 rue du Jardin Botanique  
54602 Villers ls Nancy Cedex, France  
*Sebastien.Hinderer@loria.fr*

## Abstract

We discuss the use of model building for temporal representations. We chose Polish to illustrate our discussion because it has an interesting aspectual system, but the points we wish to make are not language specific. Rather, our goal is to develop theoretical and computational tools for temporal model building tasks in computational semantics. To this end, we present a first-order theory of time and events which is rich enough to capture interesting semantic distinctions, and an algorithm which takes minimal models for first-order theories and systematically attempts to “perturb” their temporal component to provide non-minimal, but semantically significant, models.

## Keywords

model building, first-order logic, higher-order logic, computational semantics, events, tense, aspect

## 1 Introduction

In this paper we discuss the use of model building for temporal representations. We chose Polish to illustrate the main points because (in common with other Slavic languages) it has an interesting aspectual system, but the main ideas are not language specific. Rather, our goal is to provide theoretical and computational tools for temporal model building tasks. To this end, we present a first-order theory of time and events which is rich enough to capture interesting semantic distinctions, and an algorithm which takes minimal models for first-order theories and systematically attempts to “perturb” their temporal component to provide non-minimal, but semantically significant, models.

The work has been implemented in a modified version of the Curt architecture. This architecture was developed by Blackburn and Bos [2] to illustrate the interplay of logical techniques useful in computational semantics. Roughly speaking, the Curt architecture consists of a representation component (which implements key ideas of Montague semantics [10]) and an inference component. In this paper we have used a modified version of the representation component (based on an external tool called *Nessie* written by Sébastien Hinderer) which enables us to specify temporal representations using a higher-order logic called *TY<sub>4</sub>*. However, although we shall briefly discuss how we build our temporal representations, the main focus of the paper

is on the other half of the Curt architecture, namely the inference component.

Inference is often thought of simply as theorem proving. However one of the main points made in [2] is that a wider perspective is needed: theorem proving should be systematically coupled with model building and the Curt architecture does this. Model building takes a logical representation of a sentence and attempts to build a model for it; to put it informally, it attempts to return a simple picture of the world in which that formula is true. This has a number of uses. For example, as is emphasized in [2], model building provides a useful positive test for consistency; if a model for a sentence can be built, then that sentence is consistent (this can be useful to know, as it enables us to prevent a theorem prover fruitlessly searching for a proof of inconsistency). Moreover, in subsequent papers, Johan Bos and his co-workers have demonstrated that model building can be a practical tool in various applications (see for example [6, 5, 4]).

The work described here attempts to develop a Curt style architecture rich enough to handle natural language temporal phenomena. So far we have concentrated on the semantic problems raised by tense and aspect. We have developed a first-order theory of time and events, which draws on ideas from both [9] and [3]. Although these theories were developed for English, we believe the underlying ideas are more general, and to lend support to this claim we shall work here with Polish.

As we shall see, however, more than a theory of time and events is required. Model builders typically build the smallest models possible, but such models may not be suitable for all tense and aspectual combinations, which often underspecify the temporal profile of the situations of interest. We thus provide an algorithm which takes as input a first-order theory, a first-order formula, and a model for the theory and formula, and systematically attempts to “perturb” the temporal part of the model to find non-minimal but semantically relevant models.

## 2 Modelling tense and aspect

In this section, we shall discuss the logical modeling of tense and aspect, drawing on some simple examples from Polish, and informally introduce a temporal ontology of time and events which will let us express temporal and aspectual distinctions in a precise way. The formal definition of a theory over this temporal

ontology (which draws on ideas from [3] and [9]) will be given in Section 4.

Consider the following four Polish sentences:

1. Piotr pospaceruje
2. Piotr pokochal Aline
3. Piotr napisal list  
and  
Piotr popisal list

The first sentence refers to a walking event and adopts a perfective point of view: it insists on the fact that the mentioned action will be terminated at some point in the future. The second sentence mentions an eventuality of loving and also adopts a perfective point of view. However, the reading of this sentence differs from the previous one. The first sentence insisted on the *termination* of the event, whereas the second one insists on its *beginning*. In other words, the second sentence has an *inchoative* reading. This is because the verb “kocha” from which “pokochac” is derived is a *state verb*, and perfective state verbs have inchoative readings in Polish. So the second sentence means that at some point in the past Piotr started to love Alina.

The last two sentences, which are also perfective, both refer to the termination of a writing event which is located in the past. The difference between these two sentences concerns the way the writing event terminated. In the “napisac” variant, an idea of successful termination is conveyed: that is, at some point the writing stopped, because the letter was finished. In the “popisal” variant, the writing also stopped but the conveyed idea is that the writing event was interrupted before its normal termination, which implies that the letter could not be finished. To distinguish between a “normal” termination and a termination due to an unexpected, premature interruption, we talk about *culminations*. An event *culminates* when it terminates and has also been completed, or fully accomplished. Thus the event of writing reported by the sentence “Piotr napisal list” culminates, whereas the one in “Piotr popisal list” does not.

Note that in our two first examples, it makes no sense to talk about the culmination of the walking or loving eventualities; neither walking events nor states of loving have natural culminations in the way that writing events do. More generally, different types of events have different properties, and verbs can be classified according to the properties of the event they refer to. Such a classification has been proposed for Polish verbs by Młynarczyk [8], and we follow this classification in our work. The classification proposes five verb classes, including the three just mentioned: a class for processes (“to walk” belongs to this class), a class of state verbs and gradual transitions (a member of which is “to love”) and a class for culminations (“to write” belongs to this class). Processes are non-instantaneous events which have no particular properties; it is possible to look at them either as ongoing (imperfective), or as finished (perfective). State verbs are also non-instantaneous. Their imperfective use corresponds to a vision of the state as holding, whereas (as was already mentioned) their perfective use has an inchoative reading. Culminations have an imperfective variant and

two perfective ones: one for events that have culminated, another for event that have not culminated.

Now, our aim is to translate simple Polish sentences like those just discussed into logical formulas that encode their meaning. More precisely, we are interested in obtaining logical formulas that give an account of the sentence’s temporal and aspectual properties suitable for theorem proving and model building purposes. This means we should choose a logic that makes it easy to distinguish various kinds of entities (for example, ordinary individuals and events) and that lends itself naturally to semantic construction. To achieve these goals we will use a higher-order typed logic called  $TY_4$ . This logic belongs to the  $TY_n$  family of logics. This family of logics has long been advocated by Muskens (see, for example, [11]) as an appropriate logical setting for natural language semantics. The four basic vocabulary types we shall build the formulas of this logic over (in addition to the type of truth-values which is always included in  $TY_n$  theories) are:

**entity** : for individuals and objects;

**time** : for moments of time;

**event** : for the events introduced by verbs;

**kind** : to classify events into kinds.

The first type (entity) will certainly be familiar to the reader used to Montague-style semantic construction. The second type, time, is clearly needed to give an account of notions like past, present and future. The abstract entities known as events (introduced by [7]) are a convenient object one can use to talk about actions introduced by verbs. Each verb introduces an event, which is then used to record additional information about the action the verb describes. For example, if the verb “to eat” introduces an event  $e$ , then the fact that the entity doing the eating is  $x$  will be encoded as  $agent(e, x)$ , the fact that the eaten entity is  $y$  will be encoded as  $patient(e, y)$ , and so on. Event-based representations for the verbs make it easy to attach additional information, for example information contributed by verb modifiers; for each modifier, one simply introduces a binary predicate whose first argument is the event of interest and whose second argument is the piece of information to be attached to this event. Finally, every event has a kind, and we assume that each verb picks out a distinct kind of event.

The logic we work with makes use of the following binary predicates relating events and times:

- $inception(e, t)$  means that the event  $e$  starts to take place at the moment  $t$ ;
- $conc(e, t)$  means that the event  $e$  ends at the moment  $t$ ;
- $induration(e, t)$  means that the event  $e$  is going on at the moment  $t$ ;
- $ek(e, k)$  means that the event  $e$  is of kind  $k$ .

In addition, it has the following binary relation which relates times:

- $lt(t, t')$  means that time  $t$  is before time  $t'$ .

Furthermore, it has the following binary relation between events:

- $culm(e, e')$  means that event  $e'$  is the culmination of event  $e$ .

This relation plays a key role in analysing the semantics of verbs like “napisal/popisal”.

There are also number of other unary relations involving events (such as  $culminated(e)$ ), and a temporal constant NOW to represent the time of utterance. The way these items are inter-related will be formally spelt out in Section 4.

### 3 Computing semantic representations

Before turning to the formal specification of the theory of time and events, we shall briefly outline the process that allows us to automatically translate Polish sentences into a logical formula over the vocabulary introduced in the previous section. This process is done in three steps: parsing, computing a semantic representation in higher-order logic, and translating this representation to plain first-order logic. The translation process uses a modified version of the Curt architecture.

#### 3.1 Parsing

The parsing is done using a Prolog DCG. It parses the text given as input and produces a syntax tree reflecting its structure. The leaves of this tree can be labelled either by a word and its syntactic category, or by an operator encoding a verb’s temporal and aspectual meaning.

For example, here is the parse tree produced for the sentence “Piotr pospaceruje” (Piotr will have walked):

```
binary(s,
  unary(np, leaf(piotr, pn)),
  binary(vp, leaf(pastiv, op),
    leaf(pospacerowac, iv))
)
```

The first and third leaves refer to lexical entries, whereas the second carries an operator. This operator indicates that the verb carried by the following leaf is in the past.

#### 3.2 Computing higher-order logic representations

This step is performed by an external tool that has been especially developed to compute semantic representations from a parse tree. The tool is called *Nessie*, and it takes as input a parse tree similar to the one just presented and a lexicon specifying the semantic representation for each word; it was designed to handle the  $TY_n$  family of logics. Thus for present purposes we simply declare to *Nessie* the four basic vocabulary types we have selected (namely entity, time, event, and kind) and *Nessie* is then equipped to handle the higher-order language they give rise to. The

simply typed lambda-calculus lies at the heart of the  $TY_n$  family of logics, and *Nessie* handles such tasks as type-checking and  $\beta$ -reduction. In other words, the work *Nessie* does is very much inspired by Muskens’s adaptation of Montague’s original approach to natural language semantics.

The output of this second step of processing is, generally speaking, a typed lambda-term. In our temporal representations, once *Nessie* has  $\beta$ -reduced the term, there will be neither applications nor abstractions present in the final formula. In other words, the semantic formula provided by this second step is close to a genuine first-order formula, the only difference being that the variables occurring in the term are typed.

To continue with our example, *Nessie* would compute the following representation for the sentence:

$$\exists t : time. \exists e : event. (lt(now, t) \wedge ek(e, spacerowac) \wedge agent(e, piotr) \wedge conc(e, t)).$$

#### 3.3 From higher-order to first-order representations

In logical semantics there are important trade-offs between higher-order and first-order logics. As Montague, Muskens and others have demonstrated, higher-order logics are a natural medium for specifying semantic theories: their expressivity allows semantic representations for all syntactic categories to be given (and entailment relations between them to be stated). Moreover, the fact that they incorporate the simply typed lambda calculus gives a uniform and simple approach to semantic construction.

But higher-order approaches have a drawback. They are inherently more complex than first-order approaches. Because of this, relatively few automated reasoning tools exist for higher-order logics, and those that do are not particularly efficient. But all is not lost. As formal semanticists have long known, in natural language semantics, the higher-order constructs typically produce representations which are very close to first-order ones. So, if we could translate the  $TY_n$  expressions output by *Nessie* into first-order logic, we could have the best of both worlds.

At first glance, it could seem that the only thing to do to convert a higher-order formula (like the one shown above) into a first-order one is to remove the types. In fact, things are slightly more complex than this, as the following example should make clear. Consider the formula:  $\Phi = \forall X : \tau P(x)$ , where  $\tau$  is a type. If we throw types away too quickly, we get as candidate for a first-order translation of  $\Phi$ :  $\Phi' = \forall X P(X)$ . But  $\Phi$  and  $\Phi'$  don’t have the same meaning: the former formula states that the predicate  $P$  holds for every object of type  $\tau$ , whereas the latter claims that  $P$  holds for every object, no matter what its type is.

A semantically correct translation can however be obtained, with the help of a unary predicate that characterizes the object of type  $\tau$ . With the help of such a predicate (which will be written  $\tau'$ ), it becomes possible to propose a semantically correct translation of  $\Phi$  in first-order logic:  $\Phi'' = \forall X (\tau'(X) \rightarrow P(X))$ . To obtain a complete specification of a translation function translating higher-order formulas into first-order formulas, a similar trick should be used for the

existential quantifier:  $\exists X : \tau P(X)$  is translated to  $\exists X(\tau'(X) \wedge P(X))$ . The translation of other formulas is straightforward.

The complete translation mechanism has been implemented in *Nessie* which can on demand produce either higher-order or first-order semantic representations. Thus, here is the final first-order representation we get for our initial sentence:

$$\begin{aligned} \exists t(\text{TIME}(t) \wedge \exists e(\text{EVENT}(e) \\ \wedge \text{LT}(\text{NOW}, t) \wedge \text{EK}(e, \text{SPACEROWAC}) \\ \wedge \text{AGENT}(e, \text{PIOTR}) \wedge \text{CONC}(e, t))). \end{aligned}$$

## 4 A first-order theory of time and events

We are interested in computationally modeling tense and aspectual distinctions. In particular, we want to derive logical representations useful for model building purposes. But we have not yet achieved this goal. Although *Nessie* can output first-order representations, simply giving such representations to a first-order model builder won't give us what we want, for as yet we have said nothing about how the various symbols we are using are interrelated. For example, the previous representation talks about an event taking place in the future, as the  $\text{LT}(\text{NOW}, t)$  conjunct makes clear. A model for such a representation should of course reflect this. But nothing in the representation itself prevents the model builder from identifying  $t$  with  $\text{NOW}$ , or from building a model where both  $\text{now} < t$  and  $t < \text{now}$  hold, as we have said nothing about the properties of  $\text{NOW}$  or  $\text{LT}$  or how they are related. And this is only the tip of the iceberg. It is relatively clear what properties  $\text{LT}$  should have (for example, it should be transitive) but many other constraints (notably on the way times and events are interrelated) need to be expressed too. In short: to automatically compute models for a semantic representation, we need to work with respect to a theory of time and events, and the purpose of this section is to sketch the theory we use.

In essence, the theory we need should take into account some basic typing facts (for example that two objects of different types can not be identified, and that predicates impose typing constraints over their arguments), structural properties of time (such as the transitivity of  $\text{LT}$ ), and, most importantly of all, the way times and events are inter-related. The following sections give first-order axioms which formalise the required constraints. We won't give all the axioms (for example, we omit all axioms covering events for verb classes not discussed here) but we have given enough to convey a flavour of what is required to carry out model building for tense and aspectual information.

### 4.1 Type definitions

The following axioms state that the set of elements of the models should be partitioned by the four types we use: event, kind, time and entity. The following two axioms are typical:

$$\text{not\_event\_entity} : \forall A \neg(\text{EVENT}(A) \wedge \text{ENTITY}(A))$$

$$\text{not\_entity\_time} : \forall A \neg(\text{ENTITY}(A) \wedge \text{TIME}(A))$$

There is also an axiom stating that every object should belong to at least one type.

### 4.2 Typing constraints

Another family of axioms reflects the typing constraints imposed by the predicates over their arguments. For example, the binary predicate  $\text{AGENT}$  requires that its first argument is an event and that its second argument is an entity. The following is a sample of such axioms:

$$\begin{aligned} \text{now\_type:} \\ \text{TIME}(\text{now}) \end{aligned}$$

$$\begin{aligned} \text{lt\_type:} \\ \forall A \forall B (\text{LT}(A, B) \rightarrow (\text{TIME}(A) \wedge \text{TIME}(B))) \end{aligned}$$

$$\begin{aligned} \text{agent\_type:} \\ \forall A \forall B (\text{AGENT}(A, B) \rightarrow (\text{EVENT}(A) \wedge \text{ENTITY}(B))) \end{aligned}$$

$$\begin{aligned} \text{conc\_type:} \\ \forall A \forall B (\text{CONC}(A, B) \rightarrow (\text{EVENT}(A) \wedge \text{TIME}(B))) \end{aligned}$$

$$\begin{aligned} \text{inception\_type:} \\ \forall A \forall B (\text{INCEPTION}(A, B) \rightarrow (\text{EVENT}(A) \wedge \text{TIME}(B))) \end{aligned}$$

$$\begin{aligned} \text{ek\_type:} \\ \forall A \forall B (\text{EK}(A, B) \rightarrow (\text{EVENT}(A) \wedge \text{KIND}(B))) \end{aligned}$$

### 4.3 Structure of time

The previous two groups of axioms were essentially organisational: they laid out the basic constraints individuating types and imposed restrictions and requirements on the relations the various types of entity could enter into. We are now ready to turn to more substantial axioms, that is, axioms that impose structure on our ontology. The simplest such axioms are those regulating the temporal part of the ontology. The following requirements are standard (see for example [1]):

$$\begin{aligned} \text{lt\_irreflexive:} \\ \forall A \neg \text{LT}(A, A) \end{aligned}$$

$$\begin{aligned} \text{lt\_transitive:} \\ \forall A \forall B \forall C ((\text{LT}(A, B) \wedge \text{LT}(B, C)) \rightarrow \text{LT}(A, C)) \end{aligned}$$

$$\begin{aligned} \text{lt\_total:} \\ \forall A \forall B ((\text{TIME}(A) \wedge \text{TIME}(B)) \rightarrow (\text{LT}(A, B) \vee (\text{EQ}(A, B) \vee \text{LT}(B, A)))) \end{aligned}$$

Other axioms could be imposed (such as the requirement that every point has a successor, or that the structure of time is dense) but for present purposes we won't make use of such options. Instead we will turn to the heart of our formalisation, namely its treatment of events and the way they interact with time. This part draws on and generalises ideas presented in [3] and [9].

## 4.4 Structure of events

This group of axioms is itself divided into three parts, namely general axioms regulating the relationship between times and events, axioms for instantaneous events, and axioms for culminations (actually, in the full version of the theory there are axioms constraining the events required for other verb classes, but we omit them here).

### 4.4.1 Relating times and events

The following is a sample of the axioms we use to regulate the interplay between the structure of time and the structure of events. As a rough mental picture, it may be useful to think of events as hanging from the temporal structure (a bit like balloons hanging by string from a long stick). The following axioms (which have been abstracted from [3]) then ensure that the two kinds of entity are properly coordinated:

agent\_unique:  
 $\forall A \forall B \forall C ((\text{AGENT}(A,B) \wedge \text{AGENT}(A,C)) \rightarrow \text{EQ}(B,C))$

event\_has\_inception:  
 $\forall A (\text{EVENT}(A) \rightarrow \exists B \text{INCEPTION}(A,B))$

inception\_unique:  
 $\forall A \forall B \forall C ((\text{INCEPTION}(A,B) \wedge \text{INCEPTION}(A,C)) \rightarrow \text{EQ}(B,C))$

event\_has\_conc:  
 $\forall A (\text{EVENT}(A) \rightarrow \exists B \text{CONC}(A,B))$

conc\_unique:  
 $\forall A \forall B \forall C ((\text{CONC}(A,B) \wedge \text{CONC}(A,C)) \rightarrow \text{EQ}(B,C))$

inception\_not\_after\_conc:  
 $\forall A \forall B \forall C ((\text{INCEPTION}(A,B) \wedge \text{CONC}(A,C)) \rightarrow \neg \text{LT}(C,B))$

duration\_before\_conc:  
 $\forall A \forall B \forall C ((\text{INDURATION}(A,B) \wedge \text{CONC}(A,C)) \rightarrow \text{LT}(B,C))$

not\_inception\_and\_induration:  
 $\forall A \forall B \neg (\text{INCEPTION}(A,B) \wedge \text{INDURATION}(A,B))$

not\_induration\_and\_conc:  
 $\forall A \forall B \neg (\text{INDURATION}(A,B) \wedge \text{CONC}(A,B))$

### 4.4.2 Instantaneous events

Our account of the semantics of culmination (which is essential for some Polish verbs) makes use of the notion of instantaneous events. There are a number of plausible ways of axiomatising this notion. For model building purposes, we work with the following axioms:

instantaneous\_definition.1:  
 $\forall A (\text{INSTANTANEOUS}(A) \rightarrow \exists B (\text{INCEPTION}(A,B) \wedge \text{CONC}(A,B)))$

instantaneous\_definition.2:  
 $\forall A \forall B (\text{EVENT}(A) \rightarrow ((\text{INCEPTION}(A,B) \wedge \text{CONC}(A,B))$

$\rightarrow \text{INSTANTANEOUS}(A)))$

Note that the second axiom is the converse of the first.

### 4.4.3 Culminations

We turn to the semantics of culmination. In essence, this part of our theory formalises key ideas from Moens and Steedman [9]. That is, we view eventualities such as writing a book as a relation between *two* events. The first event is the lead-up, or preparatory process, for example the act of writing. The second event (which we view as instantaneous) is the event of culmination, in the case the event of finishing the book. Sometimes the culmination is not achieved, and Moens and Steedman use evocative terminology to describe what goes on in this case: they talk of the eventuality being “stripped” of its culmination. To use their terminology, Polish lexicalises the distinction between stripped (for example “popisal”) and unstripped (for example “napisal”) eventualities. The following axioms capture these ideas in a form suitable for model building:

culm\_unique:  
 $\forall A \forall B \forall C ((\text{CULM}(A,B) \wedge \text{CULM}(A,C)) \rightarrow \text{EQ}(B,C))$

culm\_injective:  
 $\forall A \forall B \forall C ((\text{CULM}(A,C) \wedge \text{CULM}(B,C)) \rightarrow \text{EQ}(A,B))$

culm\_no\_fixpoint:  
 $\forall A \neg \text{CULM}(A,A)$

culm\_antisymmetric:  
 $\forall A \forall B (\text{CULM}(A,B) \rightarrow \neg \text{CULM}(B,A))$

culm\_preserves\_agent:  
 $\forall A \forall B \forall C ((\text{CULM}(A,B) \wedge \text{AGENT}(A,C)) \rightarrow \text{AGENT}(B,C))$

culm\_preserves\_patient:  
 $\forall A \forall B \forall C ((\text{CULM}(A,B) \wedge \text{PATIENT}(A,C)) \rightarrow \text{PATIENT}(B,C))$

culm\_preserves\_kind:  
 $\forall A \forall B \forall C ((\text{CULM}(A,B) \wedge \text{EK}(A,C)) \rightarrow \text{EK}(B,C))$

culm\_inception:  
 $\forall A \forall B \forall C ((\text{CULM}(A,B) \wedge \text{CONC}(A,C)) \rightarrow \text{INCEPTION}(B,C))$

culm\_imp\_instantaneous:  
 $\forall A \forall B (\text{CULM}(A,B) \rightarrow \text{INSTANTANEOUS}(B))$

culminated\_definition:  
 $\forall A (\text{CULMINATED}(A) \rightarrow \exists B (\text{EVENT}(B) \wedge \text{CULM}(A,B)))$

culminated\_imp\_not\_instantaneous:  
 $\forall A (\text{CULMINATED}(A) \rightarrow \neg \text{INSTANTANEOUS}(A))$

## 4.5 A first model

With the help of the previously given axioms, a model builder will generate far more reasonable models than the one mentioned at the beginning of this section. As an example, here is the model produced by the Paradox model builder for the sentence “Piotr pospaceruje” (Piotr will have walked):

```
D=[d1,d2,d3,d4,d5]
f(0, spacerowac, d2)
f(0, piotr, d1)
f(0, now, d5)      f(2, inception, [(d3,d5)])
f(1, entity, [d1]) f(2, ek, [(d3,d2)])
f(1, event, [d3])  f(2, lt, [(d5,d4)])
f(1, kind, [d2])   f(2, agent, [(d3,d1)])
f(1, process, [d2]) f(2, conc, [(d3,d4)])
f(1, time, [d4,d5])
f(1, instantaneous, [])
```

Roughly speaking, this model describes a situation where Piotr starts to walk right now and finishes its walk at some point in the future.

## 5 Building non-minimal models

Although the situation described in the model we just built is realistic, it is not the only realistic situation the sentence describes. It is compatible with the semantics of Polish perfective verbs in the present tense that Piotr has already walked for a long time, or that his walk has not started yet but will start later in the future. That is, this particular combination of tense and aspectual information underspecify the temporal profile of the situations of interest.

However model builders typically will *not* find these other models. Why not? Because they are not *minimal*. Model builder attempt to find the smallest model they can, and in the above example it has identified d5 with both now and with the inception of event d3. This gives rise to a perfectly legitimate model — but the strategy of identifying points when possible rules out the other two semantic options just mentioned. The other model are non-minimal because they do *not* identify the time of utterance with the inception time. And one of these models may well turn out to be the one required for processing subsequent sentences.

So we need to do more, and this section presents an algorithm which returns a list of *all* the realistic situations, as far as tense and aspect are concerned. The input of this algorithm is a model similar to the one shown in the previous section. The output models can be seen as perturbations of the initial one. The construction procedure takes place in two steps. First, a generation step produces a list of possible models. Second, a selection step is used to filter out those models that actually satisfy both the initial semantic representation and the axioms. The second step essentially uses first-order model checking as described in [2], so we focus here on the generation step.

Our initial input are a sentence  $S$ , its representation  $R$  as a first-order formula, and a theory  $T$  of time and events (such as the one given in the previous section). The formula  $R$  is supposed closed and consistent with

$T$ . Thus, there is a model  $M_0$  of  $T$  in which  $R$  is satisfiable. Our purpose is, starting from  $M_0$ , to build the set  $\mathcal{M}_f$  of all non-isomorphic “minimal perturbations” of models of  $T$  in which  $R$  is satisfiable.

First, we build a set  $\mathcal{M}_i$  of candidate models. All the generated models can be seen as perturbations of the initial model  $M_0$ . The part of  $M_0$  that is not related to time and events will be the same for all the produced models. The variations from model to model only affect the points denoting moments in time and relations those points belong to. To put it more precisely, the constant part of the final models (which will be called the *core* in the rest of this paper), is obtained by removing the time-related information from  $M_0$ . For instance, if  $M_0$  is the model given previously, then its core is:

```
D=[d1,d2,d3]
f(0, piotr, d1)      f(1, entity, [d1])
f(0, spacerowac, d2) f(1, event, [d3])
f(2, agent, [(d3,d1)]) f(1, kind, [d2])
f(2, ek, [(d3,d2)])  f(1, instantaneous, [])
f(1, process, [d2])
```

From the core model, we build another intermediate model, where all the significant moments in time are represented by distinct points. By significant moment, we mean those moments where something happens. We start by adding a point which interprets the constant NOW. Then, we go through the events present in the core model, and for every event  $e$  we proceed as follows:

1. If  $e$  is instantaneous, one point  $d_k$  is added, and the pair  $(e, d_k)$  is added to the INCEPTION and CONC binary relations;
2. If  $e$  is not instantaneous, we examine the relations INCEPTION,  $\text{induration}$  and CONC of the model  $M_0$ . For each of these binary relations  $R$  in which  $e$  is involved, we add a new point  $d_i$  and extend the relation  $R$  of the currently built model with the pair  $(e, d_i)$ .

Applying this algorithm to the core seen previously yields the following intermediate model:

```
D=[d1,d2,d3,d4,d5,d6]
f(0, piotr, d1)      f(1, entity, [d1])
f(0, spacerowac, d2) f(1, event, [d3])
f(0, now, d4)        f(1, instantaneous, [])
f(2, ek, [(d3,d2)])  f(1, kind, [d2])
f(2, conc, [(d3,d6)]) f(1, process, [d2])
f(2, agent, [(d3,d1)]) f(1, time, [d4,d5,d6])
f(2, inception, [(d3,d5)])
```

The model obtained after this extension step is quasi-complete. The only missing part is the LT relation specifying how the moments just introduced are ordered. What we do is that we generate all the possible orders (called successions) and, for each succession, we build the associated model.

The number of possible successions grows exponentially with the considered number of moments: 2 moments  $x$  and  $y$  give 3 possible successions ( $x < y$ ,

$x = y, y < x$ ), 3 moments give 13 successions, 4 moments give 75 successions.

Before a succession is used to complete a model, it is simplified. The simplification consists in replacing all the elements that denote the same moment in time by one single element. For example, the succession  $d_i = d_j$  would be replaced by a single element  $d_k$ , and a mapping would be generated to rename both  $d_i$  and  $d_j$  to  $d_k$ . This substitution must of course be applied to the intermediate model so that the merges are taken into account correctly.

What we get as result of the succession simplification process is a list of moments in time, and a substitution to be applied to the intermediate model. The order of the elements in the list encodes their chronological order. The final model corresponding to one given succession is hence obtained from the intermediate model by performing the two following steps:

1. Apply the substitution provided by the succession's simplification;
2. If  $x_1, \dots, x_n$  is the list of moments returned by the succession's simplification, every pair  $(x_i, x_j)$  such that  $1 < i < j < n$  is added to the LT relation. This ensures that the properties of LT such as its transitivity and irreflexivity will hold in the new model.

This marks the end of the first (generation) step we mentioned before. Since the intermediate model we presented before makes use of 3 moments in time, we obtain 13 possible successions, hence 13 possible models. This 13 models are tested (using a first-order model checker) to see which really satisfy both the semantic representation and the theory  $T$ . Finally, three models are kept. The first is the initial model  $M_0$ . The second looks like this:

```
D=[d1,d2,d3,d4,d5,d6]
f(0, piotr, d1)      f(1, entity, [d1])
f(0, spacerowac, d2) f(1, event, [d3])
f(0, now, d4)        f(1, instantaneous, [])
f(2, ek, [(d3,d2)]) f(1, kind, [d2])
f(2, agent, [(d3,d1)]) f(1, process, [d2])
f(2, conc, [(d3,d6)]) f(1, time, [d4,d5,d6])
f(2, inception, [(d3,d5)])
f(2, lt, [(d5,d4), (d5,d6), (d4,d6)])
```

As required, this corresponds to a situation where the walking event starts in the past. The third model differs from the second only in the information on the temporal ordering, which looks like this:

```
f(2, lt, [(d4,d5), (d4,d6), (d5,d6)])
```

In this model the walking event starts in the future.

The algorithm also finds the possible models for the other example sentences we talked about in section 2. For the sentence "Piotr pokochal Aline", the system provides the three distinct models. On the other hand "Piotr napisał list" and "Piotr popisał list" only have one model each. The external model builder finds this model, and our algorithm correctly concludes that the model cannot be perturbed.

## 6 Conclusion

In this paper we have discussed a logic-based approach to modeling temporal information, and in particular, information about tense and aspect. Our approach has been general and generic. On the representational side, we have used a tool called *Nessie* which allows us to specify temporal (and other ontologies) within the generous expressive limits provided by  $TY_n$ . On the inference side we have provided a first-order theory which, although inspired by work on English, seems general enough to provide analyses of tense and aspect in other languages. Finally, we have provided an algorithm which allows us to perturb the temporal component of models in the hope of finding non-minimal but semantically significant variants. This algorithm is not dependent on the axiomatic choices made in this paper; in fact (as we have discovered) is a very useful tool when one is investigating the effects that varying the underlying theory can have.

Much remains to be done. For a start, the work reported here does not consider many other important temporal phenomena, such as dates, temporal prepositions, and temporal adverbs. Furthermore, it is not integrated with a theory of discourse structure; incorporating the ideas reported here into a Discourse Representation Theory (DRT) based approach would be a natural path to investigate. We plan to turn to such extensions shortly.

## References

- [1] J. Bentham. *The Logic of Time*. Kluwer Academic Publishers, Dordrecht, second edition, 1991.
- [2] P. Blackburn and J. Bos. *Representation and Inference for Natural Language. A First Course in Computational Semantics*. CSLI, 2005.
- [3] P. Blackburn, C. Gardent, and M. de Rijke. Back and forth through time and events. In *Proceedings of the Ninth Amsterdam Colloquium*, pages 161–175, 1993.
- [4] J. Bos and K. Markert. Recognising textual entailment with robust logical inference. In J. Q. et al, editor, *MLCW 2005*, volume LNAI 3944, pages 404–426, 2006.
- [5] J. Bos and T. Oka. Meaningful conversation with mobile robots. *Advanced Robotics*, 21(2):209–232, 2007.
- [6] J. Bos and T. Oka. A spoken language interface with a mobile robot. *Artificial Life and Robotics*, 11(1):42–47, 2007.
- [7] D. Davidson. The logical form of action sentences. In N. Rescher, editor, *The Logic of Decision and Action*. University of Pittsburgh Press, 1976.
- [8] A. Młynarczyk. *Aspectual Pairing in Polish*. PhD thesis, University of Utrecht, 2004. LOT Dissertation Series 87.
- [9] M. Moens and J. Steedman. Temporal ontology and temporal reference. *Computational Linguistics*, 14:15–28, 1988.
- [10] R. Montague. *Formal Philosophy. Selected Papers of Richard Montague. Edited and with an Introduction by Richmond H. Thomason*. Yale University Press, New Haven, 1974.
- [11] R. Muskens. *Meaning and Partiality*. Studies in Logic, Language and Information. CSLI Publications, 1996.



# Comparison of Word-based and Letter-based Text Classification

Victoria Bobicev  
Technical University of Moldova  
Studentilor, 7, Chisinau, Moldova  
victoria\_bobicev@rol.md

## Abstract

In this paper the comparison of two PPM (Prediction by Partial Matching) methods for automatic content-based text classification is described: on the basis of letters and on the basis of words.

The investigation was driven by the idea that words and especially word combinations are more relevant features for many text classification tasks than letters and letter combinations. The results of the experiments proved applicability of PPM models for content-based text classification, although PPM model on the basis of words did not perform better than model on the basis of letters.

## 1. Introduction

Text or document classification is the assignment of documents to predefined categories on the base of their content.

In this paper the application of word-based PPM (Prediction by Partial Matching) model for automatic content-based text classification is explored. Although the application of PPM model to the document classification is not new, all the PPM models used for text classification were character-based and used sequences of two or more letters as features [20]. On the other hand, typical approaches to text classification use words as features for feature vector creation.

The main idea investigated in the paper is that words and especially word combinations are more relevant features for many text classification tasks. It is known that key-words for a document in most cases are not just a single word but combination of two or three words. That is why word-based PPM model was created and used for text classification.

## 2. Related Works

A wide variety of learning approaches to text categorisation have been used, including Bayesian classification [6], decision trees [15], cluster classification [12], k-NN algorithms [5] and neural nets [17]. Lately the most wide spread classification techniques are based on the SVM (support vector machine) [11].

Several approaches that apply compression models to text classification have been presented recently [2], [7], [21]. The underlying idea of using compression methods for text classification was their ability to create the language model adapted to particular texts. It was supposed that this model captures individual features of the text being modelled. Theoretical background to this approach was given in [20].

## 3. PPM Compression

PPM (prediction by partial matching) is an adaptive finite-context method for compression. It is based on probabilities of the upcoming symbol in dependence of several previous symbols. Firstly this algorithm was presented in [3], [4]. Lately the algorithm was modified and an optimized PPMC (Prediction by Partial Matching, escape method C) algorithm was described in [16]. PPM has set the performance standard for lossless compression of text throughout the past decade. The PPM technique blends character context models of varying length to arrive at a final overall probability distribution for predicting upcoming characters in the text.

For example, the probability of character '*m*' in context of the word '*algorithm*' is calculated as a sum of conditional probabilities in dependence of different length context up to the limited maximal length:

$$P_{PPM}('m') = \lambda_5 \cdot P('m' | 'orith') + \lambda_4 \cdot P('m' | 'rith') + \lambda_3 \cdot P('m' | 'ith') + \lambda_2 \cdot P('m' | 'th') + \lambda_1 \cdot P('m' | 'h') + \lambda_0 \cdot P('m') + \lambda_{-1} \cdot P('esc'),$$

where  $\lambda_i$  ( $i = 1 \dots 5$ ) is normalization factor;  
5 - maximal length of the context;

$P('esc')$  - 'escape' probability, the probability of the character that have never been encountered so far.

## 4. Classification Using PPM Models

Most of compression models are character-based. They treat the text as a string of characters. This method has several potential advantages. For example, it avoids the problem of defining word boundaries; it deals with different types of documents in a uniform way. It can work with text in any language and it can be applied to diverse types of classification.

In [14] the simplest way of compression-based categorization called 'off-the-shelf algorithm' is used for authorship attribution. The main idea of this method is as follows. Anonymous text is attached to texts which characterize classes, and then it is compressed. A model, providing the best compression of document, is considered as having the same class with it.

The other approach is direct measuring of text entropy using a certain text model. PPM is appropriate in this case, because text modelling and its statistic encoding are two different stages in this method. In [13] was shown that results of this method were very similar to the results of the 'off-the-shelf algorithm'.

In [21] several compression schemes were used for source based text categorization. The result was not as satisfactory as the author desired. Furthermore, the word-based PPM model tested in the paper performed worse than the letter-based. The author considered that it happened due to the small training set. Performing a great number of different experiments of compression-based categorization, author concluded that more work needs to be done to evaluate the technique.

In [7] extensive experiments on the use of compression models for categorization were performed. They reported some encouraging results; however they found that compression-based methods did not compete with the published state of the art in use of machine learning for text categorization. Authors considered that the results in this area should be evaluated more thoroughly.

In [2] the letter-based PPM models were used for spam detecting. In this task there existed two classes only: spam and legitimate email (ham). The created models were applied to TREC<sup>1</sup> spam filtering task and exhibited strong performance in the official evaluation, indicating that data-compression models are well suited to the spam filtering problem.

## 5. Word-based Models

A number of word-based text compression schemes have already been proposed. In [9], four word-based compression algorithms were implemented in order to take advantage of longer-range correlations between words and thus achieve better compression. The performance of these algorithms was consistently better than UNIX *compress* program.

In [18] the adaptive word-based PPM bigram model was used to improve text compression. This model created the shorter code in comparison with letter-based model, because the code was created for the whole word at once, so less number of bits was used to code each letter. Besides, it provided faster compression than character-based models because fewer symbols were being processed.

Results with these models have shown that the word-based approach generally performs better when applied to compression.

## 6. Word-based PPM Model Classification

Usually, PPM based classification methods use character-based models. However, if texts are classified by the contents, they are better characterized by words and word combinations than by fragments consisting of five letters. We believe that words are more indicative text features for content-based text classification. That's why we decided to use a model based on words for PPM text classification.

As proposed in [19], minimum cross-entropy as a text classifier was used in the experiments. The modelling part of PPM compression algorithm was used to estimate the entropy of text. The entropy provides a measure of

how well the probabilities were estimated; the lower entropy is, the better probabilities are estimated.

Cross-entropy is the entropy calculated for a text if the probabilities of its symbols have been estimated on another text:

$$H^m_d = -\sum_{i=1}^n p^m(x_i) \log p^m(x_i)$$

were

$H^m_d$  – text  $d$  entropy obtained using model  $m$ ;  
 $p^m(x_i)$  – probability of symbol  $x_i$  using model  $m$

for all symbols in the text  $d$  ( $i = 1 \dots n$ );

$m$  – a statistic model created on the base of another text.

Usually, the cross-entropy is greater than the entropy, because probabilities of symbols in diverse texts are different. The cross-entropy can be used as a measure for document similarity; the lower cross-entropy for two texts is, the more similar they are. Hence, if several statistic models had been created using documents that belong to different classes and cross-entropies are calculated for an unknown text on the base of each model, the lowest value of cross-entropy will indicate the class of the unknown text. In this way cross-entropy is used for text classification.

Thus, two steps were realized: (1) creation of PPM models for every class of documents; (2) estimation of entropy for unknown document using models for each class of documents. The unknown document considered to be of the same class with the model providing the lowest value of entropy.

For the experiments Several Perl scripts were created: scripts that produce letter-based and word-based PPM models, scripts for cross-entropy calculation, for class assignment and for F-measure determination.

In order to evaluate word-based PPM classification method a number of experiments were performed. The aim of the experiments was twofold:

- to evaluate quality of PPM-based document classification
- to compare letter-based and word-based PPM classification.

## 7. Experiments

Classification algorithms were evaluated on three corpora. Firstly, the corpus of articles from the Romanian electronic newspaper «Evenimentul zilei» (Event of The Day)<sup>2</sup> was used in the experiments. Secondly, experiments were carried out with clinical free text collected from the Cincinnati Children's Hospital Medical Centre's Department of Radiology and provided for training and testing by Computational Medicine Centre in Medical NLP Challenge 2007<sup>3</sup>. Finally, the algorithms

<sup>2</sup> kindly provided by Constantin Orasan (<http://pers-www.wlv.ac.uk/~in6093/>)

<sup>3</sup> <http://www.computationalmedicine.org/challenge/index.php>

<sup>1</sup> [http://trec.nist.gov/pubs/trec14/t14\\_proceedings.html](http://trec.nist.gov/pubs/trec14/t14_proceedings.html)

were evaluated on Reuters-21578<sup>4</sup> corpus as a standard benchmark for the text categorization tasks.

In text classification, effectiveness is always measured by a combination of *precision*, the percentage of documents classified into  $c_i$  that indeed belong to  $c_i$ , and *recall*, the percentage of documents belonging to  $c_i$  that are indeed classified into  $c_i$ . When effectiveness is computed for several categories, the results for individual categories can be averaged in several ways; one may opt for *microaveraging* (categories count proportionally to the number of their positive test examples) or for *macroaveraging* (all categories count the same).

The macroaveraged form of the balanced F-measure [10] was used in the experiments. The balanced F-measure is the harmonic mean of precision (P) and recall (R), written as:

$$F = 2PR / P + R,$$

$$\text{where } P = A / A + B \text{ and } R = A / A + C$$

A represents the number of true positives (i.e. the number of documents classified into  $c_i$  that indeed belong to  $c_i$ ), B represents the number of false positives (i.e. the number of documents classified into  $c_i$  that do not belong to  $c_i$ ), C represents the number of false negatives (the number of documents not classified into  $c_i$  that indeed belong to  $c_i$ ).

### 7.1.Experiments on Romanian Newspaper

The first experiment was carried on using corpus of 2 464 articles from the Romanian electronic newspaper «Evenimentul zilei» (Event of The Day). This was the easiest corpus for the evaluation. All the articles in this newspaper belonged to one of the 7 categories: editorial; money, business; politics; investigations; quotidian; in the world; sport.

Each category was considered a class of documents in the classification task. Each document belongs to exactly one class. Documents were of medium size about 2000 words, sufficient for classification. For testing 10 test documents were taken from each category (70 documents in total).

Firstly, the word-based method was evaluated. For the model creation figures, punctuation marks and others non-alphabetic symbols were eliminated, all letters were converted in lowercase. The PPM compression method with order 1(one word in context) and escape method C [1] was used for text modelling. Seven models were created, each of them reflecting features of a certain class. The entropies of test documents were calculated using the created models. Having the entropy calculated on the base of seven models, we attributed the document to the category for which its entropy was minimal.

In the Table 1 the classification result is presented. Columns show seven models accordingly to the categories, rows refer to test files of the given category. Figures in the table cells show number of test files classified to the category of the column.

Documents of only one category were classified wrongly: quotidian. It is obvious that the errors in classification

were influenced by the category. The category ‘quotidian’ is not a well-defined class of documents; it contains topical articles. Accordingly to the errors in classification, in most cases those were articles about finances and investments.

**Table 1. Test documents classification (bigram model).**

categories	Total number of test documents	money, business	quotidian	editorial	in the world	investigations	politics	sport
Money, business	10	10						
quotidian	10	1	5			4		
editorial	10			10				
in the world	10				10			
investigations	10					10		
politics	10						10	
sport	10							10

The next experiment with word-based PPMC method with order 2(two word in context) did not showed much improvement, classifying 4 documents from ‘quotidian’ to ‘investigation’ and one to ‘money, business’. The same set of documents was used for word-based PPMC method with order 0(no words in context). 12 documents were misclassified for zero-context method. Because of the low efficiency of order 0 PPM method it was not be used in the following experiments.

The experiment with letter-based PPMC method showed the same results as word-based with order 2.

Finally, three methods were cross-validated on five different test sets each containing 70 documents. The results are the following:

- for word-based PPM method with order 1: F=0.95;
- for word-based PPM method with order 2: F=0.948;
- for letter-based PPM method with order 5: F=0.97.

In spite of our expectations, letter-based method yielded slightly better results for the first corpus.

### 7.2.Experiments on Medical Free Texts

Second step of PPM classification evaluation was testing it on medical free texts. Data for the corpus was collected from the Cincinnati Children’s Hospital Medical Centre and consist of sampling of all outpatient chest x-ray and renal procedures with ICD-9-CM codes assigned. The collection is rather challenging for text classification systems as the documents are quite small and multi-labelled. An example of the text is given on Figure 1.

CLINICAL HISTORY: Cough, congestion, fever.  
 IMPRESSION: Increased markings with subtle patchy disease right upper lobe. Atelectasis versus pneumonia.

Figure 1. Example of medical free text.

A training set with 978 documents was provided for the experiments. Each document was labelled by one or more ICD-9-CM labels. 45 ICD-9-CM labels (e.g 780.6) are used in this dataset, these labels form 94 distinct

<sup>4</sup> <http://www.daviddlewis.com/resources/testcollections/reuters21578/>

combinations (e.g. the combination 780.6, 786.2). 33 of these combinations have only one training example, 27 of them have two examples. Keeping in mind the size of those examples (15-20 words) one can imagine the difficulty of the task.

In this experiment the problem of multiple-classifying appeared. Unlike the previous experiment in this case the decision about the number of labels for each document should be made. Entropies of all test documents for one category was normalized (each of them was divided by their mean), and document was attributed to the categories for which its entropy was lower than the mean. For some documents the number of categories attributed was too high, up to ten or even fifteen categories. For these documents only three categories with minimal entropy was selected. Three types of PPM method were tested: word-based with order 1, word-based with order 2 and letter-based with order 5. And again the results were quite similar:

- for word-based PPM method with order 1:  
P=0.33 R=0.45 F=0.38;
- for word-based PPM method with order 2:  
P=0.33 R=0.45 F=0.38;
- for letter-based PPM method with order 5:  
P=0.36 R=0.42 F=0.39.

Both word-based methods had the same results because the length of the documents. They were too small for two-word context method training. Letter-based model performed better but not considerably. The result in general is not high but considering the difficulty of the corpus it could be accepted as satisfactory.

### 7.3. Experiments on Reuters

The last set of experiments was performed on Reuters-21578 corpus. The Reuters-21578 test collection has been a standard benchmark for the text categorization task throughout the last years.

In order to be able to compare results with other methods standard Modified Apte ("ModApte") split was used in the experiments. Following the methodology used in [8] three subsets of the collection were used for testing: the set of the 15 categories with the highest number of positive training examples (R15); the set of the 96 categories with at least three positive examples (R96); the set of the 105 categories with at least two positive examples (R105).

For the first experiment with 15 categories, documents with only one label were selected from the whole test set. Thus, for this group of test documents only one category with minimal cross-entropy was selected. In the Table 2 only f-measure is shown for this task.

The method of multi-labelling was the same as in experiments with medical texts. It should be mentioned that the problem of selecting more than one category was not solved properly. All the attempts to add more than one label to the documents drastically affected precision and decreased F-measure. Actually, about 3/4 of documents in test set were labelled with only one topic and only about 2% of documents had more than three topics assigned. If

at least one topic for each document is assigned correctly, the result is satisfactory anyway.

Two PPM methods were compared: word-based with order 1 and letter-based with order 5. The results are presented in Table 2.

Table 2. Comparison of two classification methods on three subsets of Reuters21578

subset	Word-based method			Letter-based method		
	P	R	F	P	R	F
R(15)			0.88			0.91
R(96)	0.61	0.68	0.64	0.72	0.57	0.64
R(105)	0.77	0.62	0.68	0.78	0.63	0.69

The same results are presented on the diagram in the Figure 2.

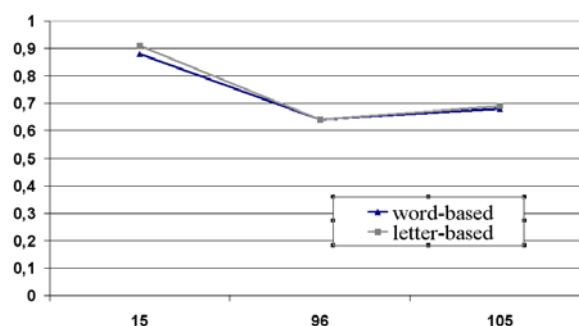


Figure 2. Comparison of two classification methods on three subsets of Reuters21578

The obtained diagram is quite similar with those presented in [8]. Moreover, the figures are similar to figures obtained by other classification methods. As for the comparison of the word-based and letter-based models, the difference is quite small. Again, our idea that word-based method performed better, was not confirmed by the experiments.

## 8. Conclusion and discussion

In the paper a comparative experimental study of two PPM-based text classification methods is presented. The experiments were carried out on a variety of experimental contexts, including three corpora and three subsets of Reuters-21578. The results of the experiments show that PPM-based text compression efficiency is comparable with other well-performed approaches presented in [8]. F-measure obtained for PPM is very close to the one obtained for SVM in [8]. However, there is not exactly the same set of documents used for training and testing and it cannot be asserted that PPM method performed better or not.

On the other hand, comparison of two PPM methods showed that word-based method is not better than letter-based, though the difference is quite small. The possible explanation for this is the quality of texts. In general, texts are noisy and contain errors of different types. For example, in Reuters the common error is word merging, that, obviously, affected word-based method. Letter-based methods avoid these problems and in general better capture the characteristics of the text. The possible preprocessing for words as stemming or lemmatization

might be done in order to improve the word-based model, but it does not solve the problem of unknown words and words with errors.

It should be mention that letter-based model is more compact and faster.

Thus, the experiments proved applicability of PPM models for content-based text classification, although PPM model on the basis of words did not perform better than model on the basis of letters.

## 9. References

- [1] Bell, T.C., Cleary, J.G. and Witten, I.H. Text compression. Prentice Hall, Englewood Cliffs, NJ. 1990.
- [2] Andrej Bratko and Bogdan Filipic. Spam Filtering Using Compression Models Technical Report IJS-DP 9227. Department of Intelligent Systems, Jozef Stefan Institute, Ljubljana, Slovenia. December, 2005
- [3] Cleary J.G. and Witten I.H. 1984a. A comparison of enumerative and adaptive codes. IEEE Trans. Inf. Theory, IT-30, 2 (Mar.), 306-315.
- [4] Cleary J.G. and Witten I.H. 1984b. Data compression using adaptive coding and partial string matching. IEEE Trans. Commun. COM-32, 4 (Apr.),396-402.
- [5] C. D'Amato, D. Malerba, F. Esposito & M. Monopoli (2003). Extending the K-Nearest Neighbour classification algorithm to symbolic objects. Atti del Convegno Intermedio della Società Italiana di Statistica "Analisi Statistica Multivariata per le scienze economico-sociali, le scienze naturali e la tecnologia". Napoli. Italia
- [6] S. Dumais, J. Platt, D. Heckermann, and M. Sahami. Inductive learning algorithms and representations for text categorization. In Proc. Intl. Conf. on Info. and Knowledge Management, pages 148-155, 1998.
- [7] Eibe Frank, Chang Chui and Ian H. Witten. Text categorisation using compression models, Proceedings of DCC-00, IEEE Data Compression Conference. 2000.
- [8] Franca Debole and Fabrizio Sebastiani An analysis of the relative hardness of Reuters-21578 subsets. Journal of the American Society for Information Science and Technology, 56(6):584-596, 2005.
- [9] R. Nigel Horspool and Gordon V. Cormack. Constructing Word-Based Text Compression Algorithms. Proceedings of Data Compression Conference (DCC'92), Snowbird, UT, March 1992, pp. 62-71.
- [10] Peter Jackson, Isabelle Moulinier. Natural Language Processing for Online Applications: Text Retrieval, Extraction, and Categorization. John Benjamins Publishing Co. 2002.
- [11] Thorsten Joachim Learning to Classify Text using Support Vector Mashine. Methods, Theory, and Algorithms. Kluwer Academic Publishers, May 2002.
- [12] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: good, bad and spectral. In Proc. 41th IEEE Symp. on Foundations of Comp. Science, 2000.
- [13] Khmelev D. V., Teahan W. J. Verification of text collections for text categorization and natural language processing: Tech. Rep. AIIA 03.1: School of Informatics, University of Wales, Bangor, 2003.
- [14] Kukushkina O., Polikarpov A., Khmelev D. Using Letters and Grammatical Statistics for Authorship Attribution . Problems of Information Transmission. 2001. Vol. 37, no. 2. pp. 172-184.
- [15] D. D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. In Proc. Annual Symposium on Document Analysis and Information Retrieval, pages 37-50, 1994.
- [16] Moffat, A. 1990. Implementing the PPM data compression scheme. IEEE Transaction on Communications, 38(11): 1917-1921.
- [17] H. T. Ng, W.B. Goh, and K.L. Low. Feature selection, perceptron learning, and a usability: case study for text categorization. In Proc. ACM SIGIR, pages 67-73, 1997.
- [18] William John Teahan 1998. Modelling English text. PhD thesis, University of Waikato, 1998.
- [19] Teahan, W. J. Text classification and segmentation using minimum cross-entropy. In Proceeding of RIAO-00, 6th International Conference "Recherche d'Information Assistee par Ordinateur", Paris, FR. 2000.
- [20] W. J. Teahan and D. J. Harper. Using compression based language models for text categorization. In J. Callan, B. Croft and J. Lafferty, editors, *Workshop on Language Modeling and Information Retrieval*, pages 83-88. ARDA, Carnegie Mellon University, 2001.
- [21] Nitin Thaper Using Compression For Source Based Classification of Text. Bachelor of Technology (Computer Science and Engineering), Indian Institute of Technology, Delhi, India. 1996.

# A Cosine Maximization-Minimization approach for User-Oriented Multi-Document Update Summarization

Florian Boudin<sup>‡</sup> and Juan-Manuel Torres-Moreno<sup>‡,‡</sup>

<sup>‡</sup>Laboratoire Informatique d'Avignon  
339 chemin des Meinajaries, BP1228,  
84911 Avignon Cedex 9, France.

{florian.boudin,juan-manuel.torres}@univ-avignon.fr

<sup>‡</sup> École Polytechnique de Montréal - Département de génie informatique  
CP 6079 Succ. Centre Ville H3C 3A7  
Montréal (Québec), Canada.

## Abstract

This paper presents a User-Oriented Multi-Document Update Summarization system based on a maximization-minimization approach. Our system relies on two main concepts. The first one is the cross summaries sentence redundancy removal which tempt to limit the redundancy of information between the update summary and the previous ones. The second concept is the newness of information detection in a cluster of documents. We try to adapt the clustering technique of bag of words extraction to a topic enrichment method that extend the topic with unique information. In the DUC 2007 update evaluation, our system obtained very good results in both automatic and human evaluations.

## Keywords

User-Oriented Multi-Document Summarization, Question Focused Summarization, Update Summarization, Statistical approach, Detection of Newness, DUC evaluation, Cross summaries redundancy removal

## 1 Introduction

The seventh edition of the Document Understanding Conference<sup>1</sup> (DUC) has introduced a pilot task in counterpart to the question-focused multi-document summarization main task. Named update task, its goal is to produce short update summaries of newswire articles under the assumption that the user has already read a set of earlier articles. This is the first time, as far as we know, that an update summarization task is evaluated. We have chosen to base our system's approach on two main concepts: cross summaries sentence redundancy removal and newness of information detection using a bag of words extraction method for topic enrichment. The rest of the paper is organized as follows. Section 2 describes the previous works and section 3 the update task of DUC 2007. The section 4

introduces the two main ideas of our approach quote above. The section 5 gives an overview of the experiments and section 6 the performance of the system at the DUC 2007. Section 7 concludes this paper and examines possible further work.

## 2 Background and related work

Interest in multi-document summarization of newswire started with the on-line publishing and the constant growth of internet. Introduced by Luhn [5] and Rath et al. [12] in the 50s-60s with single-document summarizers (SDS), research on automatic summarization can be qualified as a long tradition. However, the first automatic Multi-Document Summarizers (MDS) were developed only in the mid 90s [9]. Lately, DUC 2007 conference introduced the over-the-time update MDS evaluation. Most of work in automatic summarization apply statistical techniques to linguistic units such as terms, sentences, etc. to select, evaluate, order and assemble them according to their relevance to produces summaries [6]. In general, summaries are constructed by extracting the most relevant sentences of documents. Automatic summarization systems can be divided in two categories: single document summarizers and more complex multidocument summarizers. Multi-document systems can be viewed as a fusion of the SDS systems outputs by using additionnal information about the document set as a whole, as well as individual documents [1]. MDS perform the same task as SDS but increase the probability of information redundancy and contradictions. Previous works comparing the redundancy techniques [10] have shown that using a simple *zero knowledge* vector based cosine similarity [15] for measuring sentence similarities make no difference in performance with more complex representation, such as Latent Semantic Indexing [2]. Contrary to redundancy removal, precious little researchers have focused on time-based summarization. A natural way to go about time-based summarization is to extract the temporal tags [7] (dates, elapsed times, temporal expressions, ...) or to automatically construct the timeline from the documents [14]. For the last technique, the well known  $\chi^2$  measure [8] is used to extract

<sup>1</sup> Document Understanding Conferences are conducted since 2000 by the National Institute of Standards and Technology (NIST), <http://www-nlpir.nist.gov>

unusual words and phrases from documents. Our approach is based on the same principle of term extraction but differs from these in several ways. Our system relies on the simple idea that the most important unique terms of a cluster are suitable for representing the unique and unseen information.

### 3 Description of the DUC 2007 pilot task

The DUC 2007 update task goal is to produce short (~100 words) multi-document update summaries of newswire articles under the assumption that the user has already read a set of earlier articles. The purpose of each update summary will be to inform the reader of new information about a particular topic. Given a DUC topic and its 3 document clusters: A, B and C, the task is to create from the documents three brief, fluent summaries that contribute to satisfying the information need expressed in the topic statement.

1. A summary of documents in cluster A.
2. An update summary of documents in B, under the assumption that the reader has already read documents in A.
3. An update summary of documents in C, under the assumption that the reader has already read documents in A and B.

Within a topic, the document clusters must be processed in chronological order; i.e., we cannot look at documents in cluster B or C when generating the summary for cluster A, and we cannot look at the documents in cluster C when generating the summary for cluster B. However, the documents within a cluster can be processed in any order.

### 4 A Cosine Maximization-Minimization approach

This paper proposes a statistical method based on a maximization-minimization of cosine similarity measures between sentence vectors. The main motivation behind this approach is to find a way to quantify the newness of information contained in an document cluster assuming a given topic and a set of already "known" document clusters but at the same time minimize the possible redundant information. The main advantage of this approach is that *zero knowledge* is required and that makes the system fully adjustable to any language. The following subsections formally define the measures formulas and the method to apply them to the update summarization task.

#### 4.1 Back to basics: a simple User-Oriented MDS

We have first started by implementing a *baseline* system for which the task is to produce topic focused summaries from document clusters. Standard pre-processings are applied to the corpora, sentences are

filtered (words which do not carry meaning are removed such as functional words or common words) and stemmed using the Porter algorithm [11]. An  $N$ -dimensional termspace  $\Xi$ , where  $N$  is the number of unique terms found in the corpus, is constructed. Sentences of a document are represented in  $\Xi$  by a vector. Similarity measures between sentences are calculated by using the angle cosine. The smaller the angle, the greater is the similarity. The system scores each sentence of a document by calculating the cosine similarity angle measure [13] (defined in formula 1 and illustrated by figure 1 with the  $\theta_t$ ) between the topic vector and the sentence vector using the well known  $tf \times idf$  measures as weights.  $tf$  is the term frequency in the document and  $idf$  is the inverse document frequency.  $idf$  values are calculated on the whole DUC 2007 corpus (main and update task).

$$\cos(\vec{s}, \vec{t}) = \frac{\vec{s} \cdot \vec{t}}{\sqrt{\|\vec{s}\|^2 + \|\vec{t}\|^2}} \quad (1)$$

In our case,  $\vec{s}$  is the vectorial representation of the candidate sentence and  $\vec{t}$  of the topic.

#### 4.2 Redundancy removal techniques

Sentences coming from multiple documents and assembled together to generate a summary theoretically engender redundancy problems for classified document cluster. In practice, sentences of a cluster are all scored by calculating an angle regarding a particular topic, accordingly all high scored sentences are syntactically related. We have to deal with two different redundancy problems in our update MDS system, the within summary syntactical sentence redundancy and the cross summaries redundancy. The first one refers to the detection of duplicate sentences within a summary. We choose to measure the sentence similarity between the sentences already contained in the summary and the candidate sentences and remove them if this similarity is greater than a threshold  $\tau_o$ , empirically fixed. The second problem is more specific to the task, candidate summaries are generated assuming that other summaries have previously been produced. Therefore they have to contain different information about the same topic and inform the reader of new facts. Formally,  $n_p$  early summaries are represented as a set of vectors  $\Pi = \{\vec{p}_1, \vec{p}_2, \dots, \vec{p}_{n_p}\}$  in the termspace  $\Xi$ . Our sentence scoring method (formula 2) calculates a ratio between two angles: the sentence  $\vec{s}$  with the topic  $\vec{t}$  and the sentence with the all previous  $n_p$  summaries. The smaller value  $\eta(\vec{s}, \vec{t})$  and the higher value  $\phi(\vec{s}, \Pi)$  produces the greater score  $R(\bullet)$ :

$$R(\vec{s}, \vec{t}, \Pi) = \frac{\eta(\vec{s}, \vec{t})}{\phi(\vec{s}, \Pi) + 1} \quad (2)$$

where:  $\eta(\vec{s}, \vec{t}) = \cos(\vec{s}, \vec{t})$

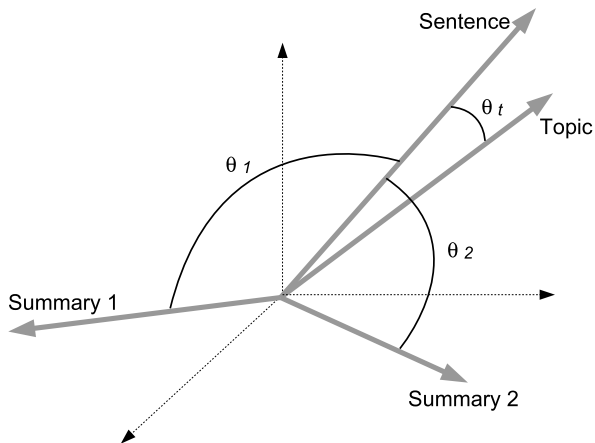
$$\phi(\vec{s}, \Pi) = \sqrt{\sum_{i=1}^{n_p} \cos(\vec{s}, \vec{p}_i)^2}$$

$$0 \leq \eta(\bullet); \phi(\bullet) \leq 1$$

Therefore:

$$\max R(s) \Rightarrow \begin{cases} \max \eta(\bullet) \\ \min \phi(\bullet) \end{cases} \quad (3)$$

The highest scored sentence  $\vec{s}$  is the most relevant assuming the topic  $\vec{t}$  (i.e.  $\eta \rightarrow 1$ ) and, simultaneously, the most different assuming the previous summaries  $\Pi$  (i.e.  $\phi \rightarrow 0$ ).



**Fig. 1:** The case of two previous summaries Cosine Maximization-Minimization illustration example: for each sentence, minimize the angle  $\theta_t$  and maximize the angles  $\theta_1$  and  $\theta_2$

### 4.3 Newness of information

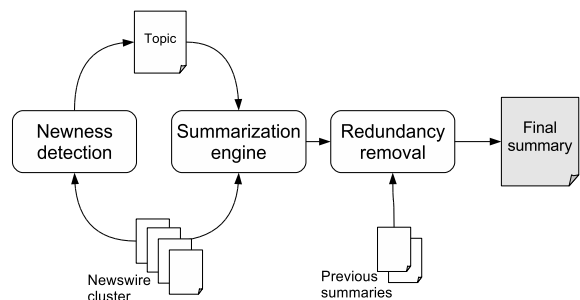
The detection of the newness of information is a critical point in the update summarization process. Indeed, how to detect, quantify and "blend" unseen information into an existing MDS system are challenging questions that we try to answer with our approach. In the same way that several previous works in document clustering use a list of high  $tf \times idf$  terms as topic descriptors, we have chosen to represent the most important information of a document cluster  $X$  by a bag of words  $B_X$  of the  $n_t$  highest  $tf \times idf$  words. The newness of information of a document cluster  $A$  in relation to already processed clusters is the difference of its bag of words  $B_A$  and the intersection of  $B_A$  with all the previous cluster's bags of words (see formula 3). The system uses the terms of  $B_X$  to enrich the topic  $t$  of the cluster  $X$ , the topic is extended by a small part of the unique information contained in the cluster. Selected sentences are not only focused on the topic but also on the unique information of the cluster.

$$B_X = B_X \setminus \bigcup_{i=1}^{n_p} B_i \quad (4)$$

### 4.4 Summary construction

The final summary is constructed by arranging the most high scored sentences until the limit size of 100 words is reached. As a consequence the last sentence

have a very high probability to be truncated. We propose a last sentence selection method to improve the summary's reading quality by looking at the next sentence. This method is applied only if the remaining word number is greater than 5 otherwise we just produce a non-optimal size summary. The next last sentence is preferred to the last if its length is almost 33% shorter and to avoid noise if its score is greater than a threshold  $\tau_o = 0.15$ . In all cases the last summary sentence is truncated of 3 words maximum. We try to protect the sentence grammaticality by removing only stop-words and very high frequency words. A set of about fifty re-writing patterns and a dictionary based name redundancy removal system have been specially created for the DUC update task. The figure 2 shows a global overview of the main architecture of our system.



**Fig. 2:** General architecture of the update summarization system.

## 5 Experiments

The method described in the previous section has been implemented and evaluated. The following subsections present some details of the different parameter settings experiments.

### 5.1 Experimental Settings

We used for our experimentations the DUC 2007 update task data set, the task is described in section 3. The corpus is composed of 10 topics, with 25 documents per topic. For each topic, the documents will be ordered chronologically and then partitioned into 3 sets: A, B and C, where the time stamps on all the documents in each set are ordered such that  $\text{time}(A) < \text{time}(B) < \text{time}(C)$ . There is approximately 10 documents in set A, 8 in set B, and 7 in set C. Tuning the system parameters requires to find a way of automatically evaluate the quality of the produced summaries and producing reliable and stable scores. All existing automated evaluation methods work by comparing the systems output summary to one of more reference summaries (ideally, produced by humans). The ROUGE [4] and Basic Elements [3] automated performance measures are considered relevant and will be used for our experiments.



### 5.1.1 Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

ROUGE [4] is a word  $n$ -gram recall between a candidate summary and a set of reference summaries. In our experiments the two ROUGE-2 and ROUGE-SU4 measures will be computed. ROUGE-2 measure which is based on bigram of words is defined in equation 5.  $Count_{match}$  stands for the maximum number of bigrams co-occurring in a candidate summary and a set of reference summaries  $R_S$ . The ROUGE-2 is a recall-related measure because of the denominator of the equation is the total sum of the number of bigrams occurring in the reference summaries.

$$ROUGE-2 = \frac{\sum_{s \in R_S} \sum_{bigram \in s} Count_{match}}{\sum_{s \in R_S} \sum_{bigram \in s} Count} \quad (5)$$

The ROUGE-SU4 measure is also a word bigram recall but extended to take into account the unigrams and allowing for arbitrary gaps of maximum length 4. For example the sentence "why using text summarization" has  $Count(4, 2) = 6$  skip-bigrams which are: "why using", "why text", "why summarization", "using text", "using summarization", "text summarization". We calculated the count of skip-bigrams with an arbitrary gap  $\gamma$  and we defined it in equation 6.

$$Count(k, n) = C \binom{n}{k} - \sum_0^{k-\gamma} (k-\gamma); \gamma \geq 1, k > \gamma \quad (6)$$

where  $n$  is the  $n$ -gram length and  $k$  the sentence length in words.

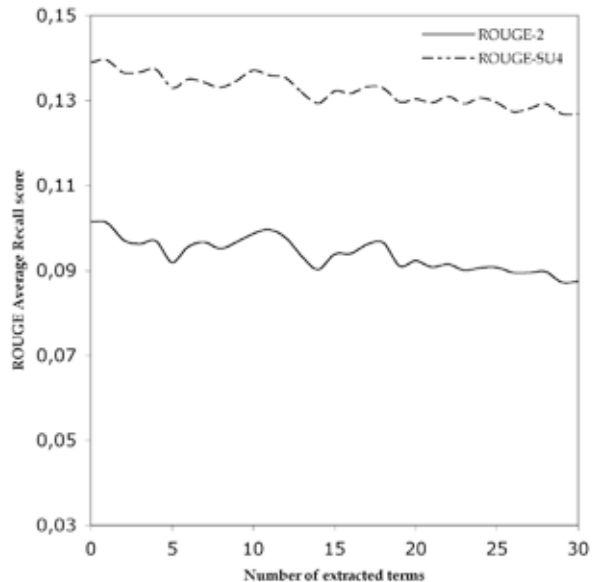
### 5.1.2 Basic Elements (BE)

Basic Elements [3] is a specific evaluation method using very small units of content, called Basic Elements, that address some of the shortcomings of  $n$ -grams. The problem of the ROUGE evaluation is that multi-word units (such as "United Mexican States") are not treated as single items, thereby skewing the scoring, and that relatively unimportant words (such as "from") count the same as relatively more important ones. The Basic Elements evaluation attempt to solve these problems by using a syntactic parser to extract just the valid minimal semantic units, called BEs.

## 5.2 Newness of information

One of the major difficulties is to evaluate and optimize the quantity of "newness" terms extracted from the clusters. If too much terms are extracted the produced summaries will be away from the point considering the topic. Otherwise, if too few terms are extracted, summaries readability will decrease due to the high information redundancy. We can observe in figure 3 that the topic enrichment always decreases automatic evaluation scores. This is due to the "noise" introduced by the newness of information terms extracted. Our experiments have also shown that the

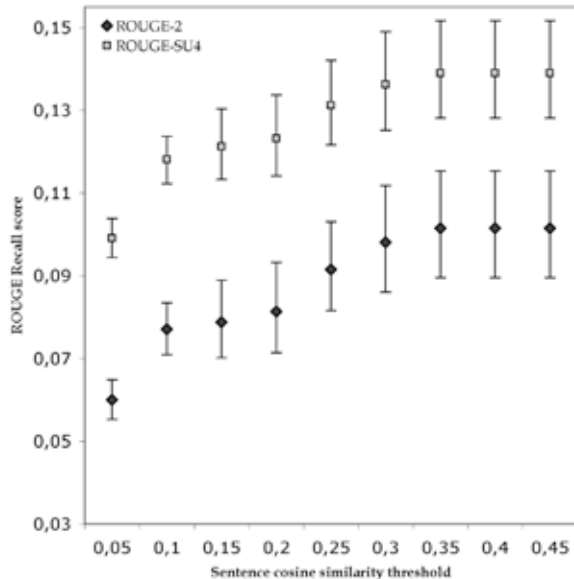
newness of information enrichment considerably enhances the readability and the intrinsic quality of the produced summaries. The information containing in the summaries is more heterogeneously spread, syntactical redundancy decrease and so readability and general quality enhance.



**Fig. 3:** ROUGE average recall scores in comparison to the number of extracted terms for the topic enrichment.

## 5.3 Within summary redundancy

We have implemented two similarity measures to deal with the within summary sentence redundancy problem. These measures are calculated between a candidate sentence and the sentences that are already considered as summary's sentences. The first one is a normalized symmetrical word overlapping measure whereas the second is a classic cosine similarity measure. A candidate sentence is accepted in the final summary only if its similarity scores with the other summary sentences are lower than a threshold  $\tau_o$ . Previous works [10] have shown that the classic cosine similarity measure (see equation 1) is the most performant measure for redundancy removal task. The two measures are binded by the fact that they use the terms as units of comparison so we decide to use only the classic cosine similarity. The sentence acceptance threshold has been tuned empirically using the ROUGE automatic evaluation as reference measure, ROUGE scores are increasing until the threshold is reaching  $\tau_o = 0.4$  (see figure 4). In other words, the deletion of sentence with lower cosine score that 0.4 remove information from the candidate summary and a sentence is considered as increasing the summary redundancy if at least one of its cosine scores with the other sentences is greater than 0.4.



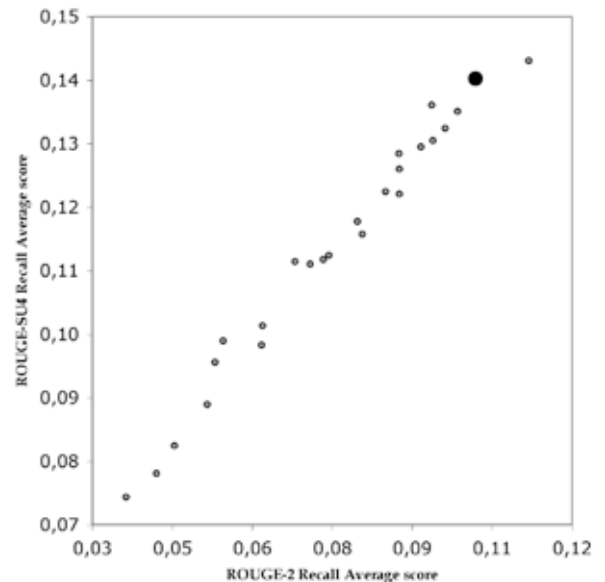
**Fig. 4:** *ROUGE average recall scores versus the redundancy similarity measure threshold  $\tau_o$ .*

#### 5.4 Experiments on DUC 2007 data

The above sections delineate the tuning techniques using the DUC 2007 corpus as reference and so how we found the optimal parameter combination by comparing our system automatic evaluation scores. This section will evaluate our system performance in the optimal parameter combination with the 24 participants of DUC 2007 update task (in which we participate with a non-optimal version of this system, the system's id is 47). An example of our system output for the topic D0726F is shown in the appendixes section. We observe in the figure 5 that our system is the second best system for the ROUGE automatic evaluation, this is a very good performance in view of the fact that the applied post-processings achieve poor performance and that they are not designed especially for the task but are more generic ones. An important margin of progress in improving these main post-processings appears. Sentence rewriting process in the specific kind of document used in the DUC conferences is not yet developed but we are currently investigating sentence reduction techniques.

## 6 The system at DUC 2007

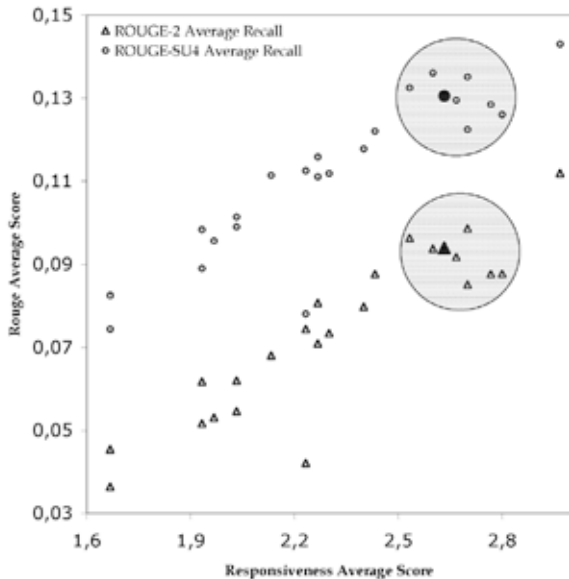
This section present the results obtained by our system at the DUC 2007 update evaluation. No training corpus was, at the time of submission, available and there was, as far as we known, no equivalent corpora for training systems. Only manual evaluation of the output summaries was possible, this explain why the parameters used for the system submission are not the optimal ones. The following parameters have been used for the final evaluation: Bag of words size: 15, Redundancy threshold:  $\tau_o = 0.4$ , minimal sentence length: 5. Among the 24 participants, our system ranks 4<sup>th</sup> in both ROUGE-2 and Basic Elements eval-



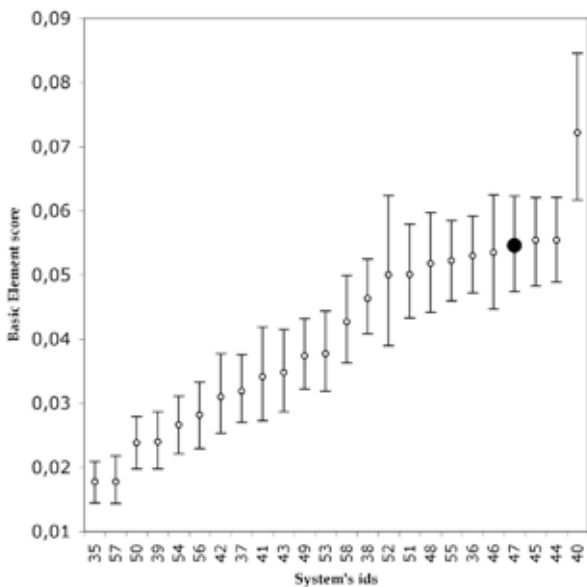
**Fig. 5:** *ROUGE-2 versus ROUGE-SU4 scores for the 24 participants of DUC 2007 update evaluation (our system is the dark circle).*

uation, the 5<sup>th</sup> in ROUGE-SU4 evaluation and the 7<sup>th</sup> in overall responsiveness. The figure 6 shows the correlation between the average ROUGE scores (ROUGE-2 and ROUGE-SU4) of the systems and their average responsiveness scores. ROUGE-2 and ROUGE-SU4 scores were computed by running ROUGE-1.5.5 with stemming but no removal of stopwords. The input file implemented jackknifing so that scores of systems and humans could be compared. The content responsiveness evaluation assesses how well each summary responds to the topic. The content responsiveness score is an integer between 1 (very poor) and 5 (very good) and is based on the amount of information in the summary that helps to satisfy the information need expressed in the topic narrative. The average responsiveness score obtained by our system was 2.633, which is above the mean (2.32 with standard deviation of 0.35). Our system is contained in the group of the top 8 well balanced systems (It must be noticed that the value of the scores range in a small interval), the mean responsiveness score (ranked only 7<sup>th</sup>) is due to the poor rewriting sentence post-processing (only less than fifty general rewriting regular expressions).

The figure 7 illustrates another automatic measure, the previously described Basic Elements (BE) evaluation measure. Basic Elements (BE) scores were computed by first using the tools in BE-1.1 to extract BE-F from each sentence-segmented. The BE-F were then matched by running ROUGE-1.5.5 with stemming, using the Head-Modifier (HM) matching criterion. For average BE our system scored 0.05458, which is above the mean (0.04093 with standard deviation of 0.0139) and ranked 4<sup>th</sup> out of 24 systems. We observe in the figure 8 that the average automatic scores are better for the last summary (cluster C) and most of all that the standard deviations extensively decrease (see table 1). The stability of our system enhance with the quan-



**Fig. 6:** *ROUGE* versus responsiveness scores for the 24 participants of the DUC 2007 update evaluation. Our system is the dark circle (*ROUGE-2*) and the dark triangle (*ROUGE-SU4*).



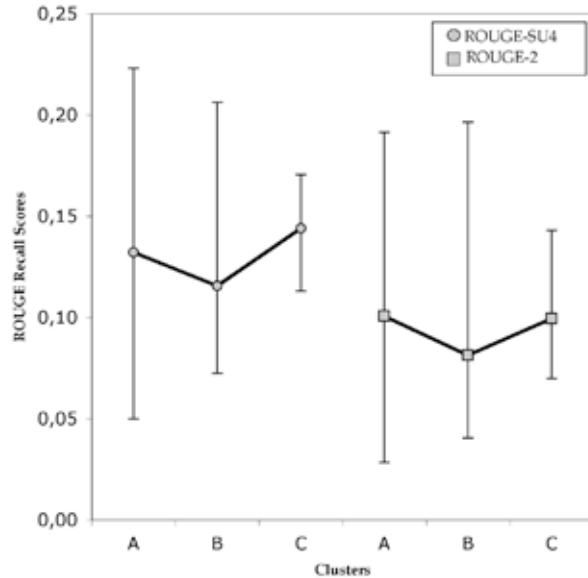
**Fig. 7:** *Basic Elements (BE)* scores of the 24 participants of the DUC 2007 update task. Our system id is 47 (marked in the figure by the dark circle).

tity of previous time documents, the light fall with the cluster B summaries may be due to the non-optimal enrichment done without enough previous extracted terms.

After analysing all the figures, one system clearly stand out from the crowd (this system id is the 40), this system ranks first in all the automatic and manual evaluations. Our system definitely is, in term of performance, in the pack leading group. We would like to say, in a word, that our system runs very fast,

Cluster	A	B	C
ROUGE-2	0,08170	0,08080	0,03670
ROUGE-SU4	0,08657	0,06826	0,02878

**Table 1:** *ROUGE* scores standard deviations of our system for each document cluster used.



**Fig. 8:** *ROUGE* recall scores (average and maximum - minimum deviations) for each document clusters (A~10, B~8 and C~7 articles).

it only take  $\approx 1$  minute to compute the whole DUC 2007 update corpus on a 1.67Ghz G4 with 1.5Gb of RAM running MAC OS X 10.4.9.

## 7 Discussion and applications

We have presented a cosine maximization - minimization technique for producing user-oriented update summaries. This summarization system achieves efficient performances in the Document Understanding Conference 2007 evaluation regarding to other participants: 4th in *ROUGE-2* average recall and *Basic Elements* average recall, 5th in *ROUGE-SU4* average recall and 7th in responsiveness in relation to 24 participants. The results of our experiments point out several research questions and directions for future work. The detection of the newness of information in the document clusters introduces too much "noise" in the summaries, considering only the most relevant sentences for the term extraction have to enhance the responsiveness. We are currently working on a more precise similarity maximization in the redundancy removal process by changing the granularity (using the sentence instead of the whole summary). Applications to a domain of speciality, the Organic Chemistry, is currently in development<sup>2</sup> with a Chemistry textbook question-answering system. This system will allow

<sup>2</sup> In collaboration with the University of Namur, (Belgium).

users to spare time by reading only new facts and skip all already known informations.

## Acknowledgment

This work was partially supported by the *Laboratoire de chimie organique de synthèse*, FUNDP (*Facultés Universitaires Notre-Dame de la Paix*), Namur, Belgium.

## References

- [1] F. Boudin and J. Torres-Moreno. NEO-CORTEX: A Performant User-Oriented Multi-Document Summarization System. In *Computational Linguistics and Intelligent Text Processing*, pages 551–562. CICLing, 2007.
- [2] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.
- [3] E. Hovy, C. Lin, L. Zhou, and J. Fukumoto. Automated Summarization Evaluation with Basic Elements. *Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC)*, 2006.
- [4] C. Lin. Rouge: A Package for Automatic Evaluation of Summaries. *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, pages 25–26, 2004.
- [5] H. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.
- [6] I. Mani and M. Maybury. *Advances in Automatic Text Summarization*. MIT Press, 1999.
- [7] I. Mani and G. Wilson. Robust temporal processing of news. *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 69–76, 2000.
- [8] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.
- [9] K. McKeown and D. Radev. Generating summaries of multiple news articles. *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 74–82, 1995.
- [10] E. Newman, W. Doran, N. Stokes, J. Carthy, and J. Dunnion. Comparing redundancy removal techniques for multi-document summarisation. *Proceedings of STAIRS*, pages 223–228, 2004.
- [11] M. Porter. An algorithm for suffix stripping, 1980.
- [12] G. Rath, A. Resnick, and T. Savage. The formation of abstracts by the selection of sentences. *American Documentation*, 12(2):139–143, 1961.
- [13] G. Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.
- [14] R. Swan and J. Allan. Automatic generation of overview timelines. *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56, 2000.
- [15] C. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann Newton, MA, USA, 1979.

## Appendix

This is an example of our system output for the topic D0726F of the DUC 2007 task. The title is "Al Gore's 2000 Presidential campaign" and the narrative part is "Give the highlights of Al Gore's 2000 Presidential campaign from the time he decided to run for president until the votes were counted."

### UPDATE DOCSUBSET="D0726F-A"

Vice President Al Gore's 2000 campaign has appointed a campaign pro with local Washington connections as its political director. Al Gore, criticized for not having enough women in his inner circle, has hired a veteran female strategist to be his deputy campaign manager for his 2000 presidential bid. Al Gore will take his first formal step toward running for president in 2000 by notifying the Federal Election Commission that he has formed a campaign organization, aides to the vice president said. Al Gore took his presidential campaign to a living room that helped launch Carter and Clinton into the White House.

### UPDATE DOCSUBSET="D0726F-B"

Patrick Kennedy, D-R.I., endorsed Vice President Al Gore for the Democratic presidential nomination in 2000. Al Gore named a veteran of the Clinton-Gore presidential campaigns to be his campaign press secretary. Bradley retired from the Senate in 1996, briefly mulled an independent run for president, then spent time lecturing at Stanford University in California before deciding to challenge Gore for the Democratic presidential nomination. Klain was criticized by some Gore allies after President Clinton called a reporter for The New York Times and said Gore needed to loosen up on the campaign trail. Bill Bradley of New Jersey, Gore's sole competitor.

### UPDATE DOCSUBSET="D0726F-C"

After hearing that Stamford-native Lieberman had been chosen as Al Gore's running mate, Marsha Greenberg decided to knit him a gift. Vice President Al Gore, who continues to reshuffle his struggling presidential campaign, has selected Donna Brazile to be his new campaign manager, officials said. Al Gore declared "a new day" in his presidential bid with a symbolic homecoming and the opening of a new campaign headquarters far from the constant political intrigue and daily odds-making of Washington. Coelho, Brazile and Carter Eskew, the media consultant hired to help develop Gore's campaign message, are already working out of the Nashville office.

# Re-engineering free texts to obtain XML documents: a discourse based approach

Amanda Bouffier      Thierry Poibeau  
Laboratoire d'Informatique de Paris-Nord  
Université Paris 13 and CNRS UMR 7030  
99, avenue Jean-Baptiste Clément  
F-93430 Villetaneuse

firstname.lastname@lipn.univ-paris13.fr

## Abstract

This paper describes a system aiming at semi-automatically fill an XML template with free texts from the clinical domain (Practice Guidelines). The XML template requires semantic information not explicitly encoded in the text (pairs of conditions and recommendations). The system tries to compute the exact scope of conditions over text sequences expressing recommendations (actions to be done). It has been applied to the analysis of several French Practice Guidelines, showing good performance.

**Keywords:** Clinical Practice Guidelines, XML, document re-engineering, discourse processing, blackboard architecture

## 1 Introduction

For a collection of textual documents, migrating to XML-based structured documents means re-engineering the whole database. Moreover, it requires analyzing the full set of textual documents so that they can fit with strict templates, as required either by XML schemas or DTD (document type definition). Most of the time, XML schemas model semantic blocs of information that are not explicitly marked in the original text.

This issue renewed the interest for the recognition and management of discourse structures, especially for technical domains. In this study, we show how technical documents belonging to a certain domain (namely, clinical Practice Guidelines) can be semi-automatically structured using NLP techniques. Practice Guidelines describe best practices with the aim of guiding decisions and criteria in specific areas of healthcare, as defined by an authoritative examination of current evidence (evidence-based medicine, see Wikipedia or Brownson *et al.*, 2003).

The Guideline Elements Model (GEM) is an XML-based guideline document model that can store and organize the heterogeneous information contained in Practice Guidelines (Schiffman, 2000). It is intended to facilitate translation of natural language guideline documents into a format that can be processed by computers. The main element is called `knowledge component` and contains most of the useful information, especially sequences of conditions and recommendations. Our aim is thus to format these documents, manually written without any pre-

cise model, according to the GEM DTD.

The organization of the paper is as follows: first, we present the task and some previous approaches (section 2). We then describe the task (section 3) and the different processing steps (section 4) along with the implementation (section 5). We finish with the presentation of some results (section 6), before the conclusion (section 7).

## 2 Discourse analysis for the re-structuration of Practice Guidelines

Clinical practices have considerably evolved these last years towards standardization and effectiveness. A major improvement is the development of Practice Guidelines throughout the world (Brownson *et al.*, 2003). However, even if they are widely distributed to the medical staff, it has been found that some simple clinical Practice Guidelines are not routinely followed to the extent they might be<sup>1</sup>.

These documents are not easy to be accessed by the doctor when he is consulting. Moreover, it can be difficult for the doctor to find relevant pieces of information from these guides, even if they are not very long documents. To overcome these problems, national health agencies try to promote the diffusion of guidelines on electronic devices, so that these recommendations could be checked by the doctor directly from his computer.

### 2.1 Previous work

Several attempts have been made already to improve the translation from the text to the formal model (e.g. Séroussi *et al.*, 2001). GEM Cutter (<http://gem.med.yale.edu/>) is a tool intended to aid people fill the GEM DTD from texts. However, this software is only an interface allowing the end-user to perform the task through a time-consuming cut-and-paste process. The overall process described in Shiffman *et al.* (2004) is also largely manual, even if it is an attempt to automate and regularize the translation process. The main problem for the automation of the translation process is to identify that a list of recommendations expressed over several sentences is under the scope of a specific condition (a specific pathology or a specific kind of patients). However, previous ap-

---

<sup>1</sup> See (Kolata, 2004). This newspaper article is a good example of the huge social impact of this research area.

proaches have been based on the analysis of isolated sentences and do not compute the exact scope of conditional sequences (Georg and Jaulent, 2005): this part of the work still has to be done by hand. It is thus a strong limitation of previous attempts to automate the process.

## 2.2 Discourse theories

Clinical Practice Guidelines express series of conditions and recommendations. These are clearly made of a set of hierarchical discourse items. This kind of discourse structure has been investigated by a series of theories, among others RST (Rhetorical Structure Theory, Mann and Thompson, 1988) or SDRT (Segmented Discourse Representation Theory, Asher and Lascarides, 2003). These theories assume that discourse is made of “segments” and that these segments are linked together by “discourse relations”, which explain the global coherence of a text. Discourse relations form a hierarchy. These theories however differ when considering the number and the type of relations they include. They are widely used for generation systems but are harder to implement for analysis (see however, Marcu 2000).

The limits of these theories are well-known: No theory gives a precise and definitive list of relations. The analysis of a text cannot be a completely deterministic process since variation may exist among different readers. Moreover, the relation between two segments can be explicit (e.g. via a linguistic cue) or implicit (e.g. juxtaposition of sentences). It can also be marked through a combination of different cues. Consequently, and this is an important issue in our perspective, the recognition of discourse relation is not completely formalized.

Finally, text structure is another feature that plays an important role for the recognition and typing of discourse relations (“the text-forming component in the linguistic system” from Halliday and Hasan, 1976:23). It is not described as such in most frameworks (like RST) but recent experiments have shown the fundamental role of text structuring (section, paragraphs, lists ...) and formatting (bold, italic, ...) for discourse (Power *et al.*, 2003, Virbel and Luc, 2001). The structure of Practice Guidelines is typically based on these features. We thus have to deal with heterogeneous knowledge sources to analyze the structure of conditions and actions.

## 3 Our approach

The main objective of the software is to go from a textual document to a GEM based document. We focus on conditions and recommendations since these elements are of paramount importance for the task and are especially difficult to deal with since they require to accurately determine the scope of conditions. Several actions may depend from one condition. A condition can introduce various actions and sub-conditions, from which depend other actions. The task consists then in the recognition of a collection of nested pairs of conditions and actions, which scope is difficult to automatically determine.

## 4 Processing steps

Segmenting a guideline to fill an XML template is a complex process involving several steps. We detail the most important steps, mainly the way we compute the scope of conditional sequences, and we just give some hints about the pre-processing stages, so that the reader understand what is the basis of the analysis.

### 4.1 Experiment material

This study is made through the analysis of 18 French Practice Guidelines published by French national health agency (ANAES, *Agence Nationale d'Accréditation et d'Evaluation en Santé* and AFSSAPS, *Agence Française de Sécurité Sanitaire des Produits de Santé*) between 2000 and 2005. These practice guidelines deal with different pathologies (diabetes, high blood pressure, asthma etc.) as well as with clinical examination processes (digestive endoscopy). This corpus is thus homogeneous, and is about 250 pages long (150.000+ words). Most of these Practice Guidelines are publicly available at: <http://www.anaes.fr> or <http://affsaps.sante.fr>. The same kind of documents exist in English and other languages. The GEM DTD is language independent.

### 4.2 Overview

Text genres can be automatically identified since they follow norms, that is to say regularities in the way of writing. This is especially important for the analysis of the scope of conditional segments.

As for quantifiers, a conditional segment may have a *scope* that extends over several basic segments (a *frame*). It has been shown by several authors (e.g. Charolles, 2005) working on different types of texts that *by default* introducers detached from the sentence have most of the time a scope beyond the current sentence whereas introducers integrated to a sentence (not at the beginning of a sentence) have a scope that is limited to the current sentence.

The segmentation is however *non-monotonic*: it can be revised if some linguistic cues suggest another more accurate segmentation (violation of the norm). This correction of the initial segmentation can be inspired by the *shift and reduce* strategy from Marcu (1999). Some “cohesion cues” suggest extending the default segmentation (*shift*) while some others suggest limiting the scope of the conditional sequence (*reduce*) – see section 4.5.

### 4.3 Preprocessing (cue identification)

The preprocessing stage corresponds to the analysis of relevant linguistic cues. These cues are of different natures: they can be related to the material text structure or to its content; they are based on morphology, syntax or semantics. We chose to mainly focus on task-independent knowledge so that the method is portable, as far as possible (we took inspiration from Halliday and Hasan, 1976 and Marcu, 2000).

*Material structure cues.* This step includes the recognition of titles, section, enumerations and paragraphs.

*Morpho-syntactic cues.* Recommendations are not expressed in the same way as conditions from a morpho-syntactic point of view.

*Anaphoric cues.* A basic and local analysis of anaphoric elements is performed. We especially focused on expressions such as *dans ce cas*, *dans les N cas (précédents)* which are very frequent in clinical documents. The recognition of such expressions is based on a limited set of possible nouns that occurred in context, together with specific constraints (use of demonstrative pronouns, etc).

*Conjunctive cues (discourse connectors).* Conditions are mainly expressed through conjunctive cues. The following forms are especially interesting: forms prototypically expressing conditions (*si, en cas de, dans le cas où... if, in case of...*) and temporal frames (*lorsque, au moment où, avant de... when, before...*)

*Lexical cues.* Recommendations are mainly expressed through lexical cues. We have observed forms prototypically expressing recommendations (*recommander, prescrire, ... recommend, prescribe*), obligations (*devoir, ...*) or options (*pouvoir, ...*). Most of these forms are highly ambiguous but can be automatically acquired from an annotated corpus.

#### 4.4 Basic segmentation

A *basic segment* corresponds to a text sequences expressing either a condition or a recommendation. It is most of the time to a sentence, or a proposition inside a sentence.

Relevant features described in the previous section may be highly ambiguous. Conditional segments are most of the times tagged according to a list of specific introducers. The recognition of action segments is more difficult: it is rarely done according to a single feature, but most of the time according to a bundle of relevant features acquired from a representative corpus. For example, if a text sequence contains an *injunctive* verb, with an infinitive form at the beginning of a sentence, then the whole sequence is typed as *action*. The relevant sets of co-occurring features have been automatically derived from a set of annotated Practice Guidelines, using the chi-square test to compute distribution independence.

After this step, the text is segmented into basic typed segments expressing either a recommendation or a condition (the rest of the text is left untagged).

#### 4.5 Computing the scope of conditions

We propose a strategy in two steps: first, the default segmentation is done and then a revision process is able to correct the main errors caused by the default segmentation (the norm).

##### Default segmentation

As was said in the overview, we developed a strategy which makes use of the notion of default. By default:

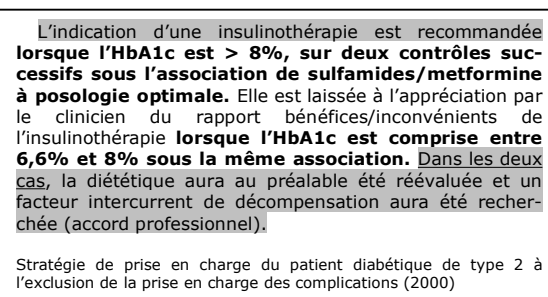
1. Scope of a heading goes up to the next heading;
2. Scope of an enumeration's header covers all the items of the enumeration ;
3. If a conditional sequence is detached (at the beginning of a paragraph or of a sentence), its scope is the whole paragraph;
4. If the conditional sequence is integrated inside a sentence, its scope is equal to the current sentence.

Cases 3 and 4 cover 50 to 80% of all the cases, in function of the practice guidelines taken into account. However, this default segmentation not monotonic: it is revised and modified when a feature bundle of linguistic cues forms a continuation mark within the text or, on the other hand, when the default segmentation seems to contradict some cohesion cue.

##### Revising the default segmentation

There are two cases which require revising the default segmentation: 1) when a cohesion mark indicates that the scope is larger than the default unit; 2) when a rupture mark indicates that the scope is smaller. We only have room for two examples, which, we hope, give a good idea of the kind of process undergoing.

1) Anaphoric relations are strong cues of text coherence: they most of the time indicate the continuation of a frame after the end of its default boundaries (*shift*).



**Figure 1.** The last sentence introduced by *dans les deux cas* is under the scope of the conditions introduced by *lorsque*.

On Figure 1, the expression *dans les deux cas* (*in the two cases...*) is an anaphoric mark referring to the two previous utterances. The scope of the conditional segment introduced by *lorsque* (that would normally be limited to the sentence they appear in) is thus extended accordingly. The identification of this extension requires the co-occurrence of complex features (an anaphoric noun referring to two basic *recommendation* segments) that are dynamically identified thanks to the blackboard architecture (see section 5).

2) Other discursive cues are strong indicators that a frame must be closed before its default boundaries (*reduce*). These cues may indicate some contrastive, corrective or adversative information (*cependant, en revanche...*). Justifications cues (*en effet, en fait, ...*) also

pertain to this class since a justification is not part of the `action` element of the GEM DTD. On Figure 2, the linguistic cue *en effet* (*in effect*) closes the frame introduced by *Chez les patients ayant initialement...(<1g/l)* since this sequence should fill the `explanation` element of the GEM DTD and is not an `action` element.

**Chez les patients ayant initialement une concentration très élevée de LDL-cholestérol, et notamment chez les patients à haut risque dont la cible thérapeutique est basse (<1g/l), le prescripteur doit garder à l'esprit que la prescription de statine à fortes doses ou en association nécessite une prise en compte au cas par cas du rapport bénéfice/risque et ne doit jamais être systématique. En effet, les fortes doses de statines et les bithérapies n'ont pas fait l'objet à ce jour d'une évaluation suffisante dans ces situations.**

(Prise en charge thérapeutique du patient dyslipidémique, 2005, p4)

**Figure 2.** The last sentence contains a justification cue (*en effet*) which limits the scope of the condition in the preceding sentence.

## 5 Implementation

Discourse processing requires a lot of relevant information ranging from lexical cues to complex co-occurrence of different features. We choose to implement these through a classical blackboard architecture (Englemore and Morgan, 1988). The advantages of such an architecture are easy to grasp for our problem: each linguistic phenomenon is treated through an independent agent; inference rules are also coded through specific agents and a facilitator rules the overall process. Basic linguistic information is collected through a set of modules called “linguistic experts”. Each module is specialized in a specific phenomenon (text structure recognition, port-of-speech tagging, term spotting, etc.). Another series of experts then combine the initial disseminated knowledge brought by the linguist experts to recognize basic segments (section 4.4) and help compute scopes and frames (section 4.5). These experts form the “inference engine”, able to analyze information stored in the working memory of the system and add new knowledge to the database. Even if linear order is not relevant for the inference process, new information is still indexed with textual clauses, so that the system is able to generate the original text along with annotation. A facilitator helps determining which expert, at a given point in time, has the most important insight or information to contribute to the problem's solution.

## 6 Evaluation

We are currently working on a corpus on 18 practice guidelines in French (several hundreds of frames), with the aid of doctors and experts of the domain. These experts have validated the approach, which is much more powerful than the sole identification of isolated text segments: the identification of the scope of conditional segments dramatically decrease the time spent by experts to validate the output of the system. We are in the process of evaluat-

ing the overall process on a relevant set of Practice Guidelines that have not been used during the implementation.

### 6.1 Evaluation criteria

A sequence is ok if the semantics of the sequence is preserved. For example *Chez l'obèse non diabétique (accord professionnel)* (*In the case of an obese person without any diabetes (professional approval)*), recognition is ok even if *professional approval* is not *stricto sensu* part of the condition. On the contrary, *Chez l'obèse* (*In the case of an obese person*) is not ok. The same criteria are applied for recommendations. We evaluate the scope of condition sequences by measuring, for if each recommendation is linked up with the appropriate condition sequence.

### 6.2 Manual annotation and inter-annotator agreement

Evaluation is made according to Practice Guidelines that have been manually annotated by two annotators: one is a domain expert (a doctor) and the other one by a non expert (a linguist). The task consists in (manually) deriving a tree structure from the original text document: each node is a condition and each leaf a recommendation. All the daughters of a node are under the scope of this node. Inter-annotator agreement is high (157 nodes out of 162 are annotated similarly, which mean above .96 agreement)

These results are encouraging and differ from other experiments on less stable data, for example, when people try to compute the scope of temporal adverbials in narrative texts (like *in 1999*). Temporal are known to open a frame but most of the time no clear boundary can be given. At the opposite, Practice Guidelines should lead to actions by the doctor and the scope of conditions needs to be clear inside the text. Inter-annotator agreement is here nevertheless especially high for this kind of task, especially when we consider the fact that the annotation was made by comparing the result of an expert with the result of a non expert. We thus assume that the scope of conditions is mainly expressed through linguistic cues which do not require, most of the time, domain-specific or expert knowledge. On the other hand, the very few cases where annotation differs between the expert and the non expert were clearly due to a lack of domain knowledge by the non expert.

### 6.3 Evaluation of the automatic recognition of basic sequences

The evaluation of basic segmentation gives the following results for the condition and the recommendation sequences.

**Conditions:**

	Without domain knowledge	With domain knowledge
P	1	1
R	.83	.86
P&R	.91	.92



## Recommendations:

	Without domain knowledge	With domain knowledge
P	1	1
R	.94	.95
P&R	.97	.97

Results are high for both categories, conditions as well as recommendations. The use of domain knowledge is limited and does not bring much benefit. However, this kind of knowledge is used to tag some titles corresponding to pathologies. These titles cannot be tagged without any specific knowledge and their recognition plays a crucial role since they can be assimilated to conditions. For example, the title *Hypertension artérielle (high arterial blood pressure)* is equivalent to a condition introduced by *in case of...* It is thus important to recognize and tag it accurately, since further recommendations are under the scope of this conditions. The number of such titles differs considerably from one Practice Guideline to another, and performance may be highly affected by an improper treatment of these cases.

Of course, not all errors have the same importance. Several recommendations may depend from a single condition, so that if the system failing to properly recognize the condition has a larger impact than if only one recommendation has been missed. We also observed that all the conditions and recommendations do not have the same importance (from a medical point of view) but it is very hard to take this into account for the evaluation.

### 6.4 Evaluation of the automatic recognition of the scope of conditions

The scope of conditions is computed with precision above .7 (we compare all the computed links with the reference; we thus obtain a score for precision but do not compute recall). This first result is encouraging, especially if one takes into account the large number of parameters that interfere in discourse processing. On the other hand, most of the successful cases correspond to simple configuration, where the scope of a condition is recognized by the default rule (default segmentation, see section 4.4).

The gain brought by more complex strategies involving the recognition of cohesive or rupture markers is limited. However, some interesting cases are solved due to the detection of cohesion or boundary cue. The system however fails to recognize extended scopes (beyond the default boundary) when the cohesion marks correspond to related lexical items (synonyms, hyponyms or hypernyms) or complex anaphora (nominal anaphora; hyponyms and hypernyms can be considered as a special case of nominal anaphora). Solving these (rare) complex cases would require “deep” domain knowledge and would be hard to implement.

## 7 Conclusion

We have presented in this paper a software dealing with the automatic segmentation of clinical Practice Guidelines. Our aim was to automatically fill an XML model from textual inputs. The software is able to process complex discourse structures and to compute the scope of conditional segments covering several propositions or sentences. Reported performance show that inter-annotator agreement is high for this task and that the system performs well compared to other implementation. Moreover, the system is only one trying to accurately solve the problem of the scope of conditions over several recommendations. We plan to apply our model to other languages and other kinds of texts in the future.

## References

- N. Asher and A. Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.
- R.C. Brownson, E.A. Baker, T.L. Leet, K.N. Gillespie. 2003. *Evidence-based public health*. Oxford University Press.
- M. Charolles. 2005. “Framing adverbials and their role in discourse cohesion: from connexion to forward labeling”. *Papers of the Symposium on the Exploration and Modelling of Meaning (Sem’05)*, Biarritz, France.
- R. Englemore and T. Morgan. 1988. *Blackboard Systems*. Addison-Wesley, USA.
- G. Georg and M.-C. Jaulent. 2005. “An Environment for Document Engineering of Clinical Guidelines”. *Proceedings of the American Medical Informatics Association*. Washington DC, USA. pp. 276–280.
- M.A.K. Halliday and R. Hasan. 1976. *Cohesion in English*. Longman, Harlow, UK.
- G. Kolata. 2004. “Program Coaxes Hospitals to See Treatments Under Their Noses”. *The New York Times*. Dec. 25, 2004.
- W.C. Mann and S. Thompson. 1988 “Rhetorical Structure Theory: Toward a Functional Theory of Text Organization.” In *Text* 8(3). pp. 243–281.
- D. Marcu. 1999. “A decision-based approach to rhetorical parsing”. *Proceedings of the 37th Meeting of the Association for Computational Linguistics (ACL’99)*. Maryland. pp. 365–372.
- D. Marcu. 2000. “The Rhetorical Parsing of Unrestricted Texts: A Surface-Based Approach”. *Computational Linguistic*. 26 (3). pp. 395–448.
- M-P. Péry-Woodley. 1998. “Signalling in written text: a corpus-based approach”. In M. Stede, L. Wanner & E. Hovy (Eds.), *Proceeding of the Coling ’98 Workshop on Discourse Relations and Discourse Markers*. pp. 79–85.
- R. Power, D. Scott and N. Bouayad-Agha. 2003. “Document Structure”. In *Computational Linguistics* 29(2). pp. 211–260.
- B. Séroussi, J. Bouaud, H. Dréau., H. Falcoff., C. Riou., M. Joubert., G. Simon, A. Venot. 2001. “ASTI : A Guideline-based drug-ordering system for primary care”. In *MedInfo*. n°84(1). pp. 528–532.
- R.N. Shiffman, B.T. Karras, A. Agrawal, R. Chen, L. Marenco, S. Nath. 2000. “GEM: A proposal for a more comprehensive guideline document model using XML”. *Journal of the American Medical Informatics Assoc.* n°7. pp. 488–498.

# Dynamic Iterative Ontology Learning

Christopher Brewster

José Iria

Ziqi Zhang

Fabio Ciravegna

Louise Guthrie

Yorick Wilks

Department of Computer Science,  
University of Sheffield, Sheffield, S1 4DP, UK  
`Initial.LastName@dcs.shef.ac.uk`

## Abstract

We present a novel approach to ontology learning which takes an iterative view of knowledge acquisition for ontologies. Current systems view the ontology learning process as single pipeline with one or more specific inputs and a single static output. Our approach is founded on three open-ended resources: a set of texts, a set of learning patterns and a set of ontological triples, and the system seeks to maintain these in equilibrium. As events occur which disturb this equilibrium, actions are triggered to re-establish a balance between the resources. We present a gold standard based evaluation of the final output of the system, the results of which are significantly better than those found in previous work.

## Keywords

Ontology Learning, Ontology Evaluation, Semantic Web, Knowledge Acquisition, Knowledge Management

## 1 Introduction

Ontologies have become the most commonly accepted form of knowledge representation in a wide range of fields including the Semantic Web, e-Science, e-Business, and Knowledge Management. The importance of reducing the manual effort involved in building them is undisputed. The core challenge in order to reduce this ‘knowledge acquisition bottleneck’ lies in learning ontologies from natural language texts, because, although there are other approaches (e.g. [8] where ontologies are learnt from software APIs), they have much more limited application.

An underlying assumption in many approaches to Ontology Learning (OL) from text is that the text corpus input to OL is, *a priori*, both representative of the domain in question and sufficient to build the ontology. This is, in our view, inadequate. For example, [13] write, regarding their system: “the main restriction [...] is that the quality of the corpus must be very high, namely, the sentences must be accurate and abundant enough to include most of the important relationships to be extracted”. In our view, requiring an exhaustive manual selection of the input texts defeats the very purpose of automating the ontology building process. Closely related to this is what

we consider to be the other fundamental failure of current approaches, which is to view the ontology learning process as single pipeline with one or more specific inputs and a single static output.

In this paper, we present a novel approach to ontology learning which takes an iterative view of knowledge acquisition for ontologies. Our approach is founded on three open-ended resources: a set of texts, a set of learning patterns and a set of ontological triples, and the system seeks to maintain these in equilibrium. Each resource may have additional items added to it: further documents can be added from an external repository or the web, further extraction patterns can be learnt, and further knowledge triples can be extracted from the documents. As events occur which disturb this equilibrium, actions are triggered that aim to re-establish the balance between the resources. The main advantage of our approach is its more accurate model of the way knowledge is continuously changing, uncertain and dependant on the evidence currently available and the confidence we have in that evidence.

This paper is organised as follows: In Section 2, we present some of the requirements of concerning OL, followed by a description of the system in Section 3. In Section 4, we describe the evaluation and this is followed by a discussion of the experiments. Related Work is presented in Section 6, followed by a Conclusion.

## 2 Requirements Analysis

A successful ontology learning method must take into account certain observations about knowledge and language: **1.** Knowledge is not monolithic, monotonic or universally agreed. It is uncertain, revisable, contradictory and differs from person to person. **2.** Knowledge changes continuously over time and so will be revised and re-interpreted continuously. **3.** Ontologies are inherently incomplete models of domains, but need to be maximally “fit for purpose.”. **4.** Texts assume the reader has a certain amount of background knowledge. The great majority of ontological knowledge is in this background knowledge, and not in the text. **5.** While it is easy to establish that some relationship exists between two terms, explicit defining contexts are relatively rare in texts.

The set of resources an OL system manipulates -

the text, the ontology, and the extraction patterns - are intrinsically incomplete at any given stage. The best possible input specification of the task for the OL system to perform is given by a seed ontology, a seed corpus and a seed pattern set. It also follows from the above that it is not possible to completely specify the task *a priori* - the ontology engineer should be able to intervene by pointing out correct/incorrect or relevant/irrelevant ontological concepts and documents, as the process runs, effectively delimiting the domain incrementally through examples. Given the dynamic nature of knowledge, our approach should allow for the continuous development of knowledge over time, as more resources are added. Therefore, another fundamental requirement of our approach is for the OL process to be viewed as an incremental rather than an one-off process - the output of one system run can be used as input to another run in order to refine the knowledge. Finally, the data sparsity problem necessitates the use of multiple sources of information.

### 3 The Abraxas Approach

Our incremental, weakly-supervised approach views OL as a process involving three resources: the corpus of texts, the extraction patterns set (conceived as a set of lexico-syntactic textual patterns), and the ontology (conceived as a set of RDF triples). The goal is to extend existing resources in terms of one another, always seeking a consistent overall state which we will name *equilibrium*. Our method allows equally creating an ontology given an input corpus, extending a corpus given an input ontology or deriving a set of extraction patterns given an input ontology and an input corpus. The overall system can be seen in Figure 1.

The initial input to the process serves both as a specification of the task and as seed data for a bootstrapping cycle where, at each iteration, a decision is made on which new candidate concept, relation, pattern or document to add to the domain. Such a decision is modelled via three unsupervised classification tasks that capture the interdependence between the resources: one classifies the suitability of a pattern to extract ontological concepts and relations in the documents; another classifies the suitability of ontological concepts and relations to generate patterns from the documents; and another classifies the suitability of a document to give support to patterns and ontological concepts. The notion of “suitability” is formalised by assigning the relationship of any resource to the domain a confidence value, which we will denominate “resource confidence” (RC).

#### 3.1 The Resource Confidence Measure

The Resource Confidence measure (RC) measures the confidence that the system has in a given resource i.e. item of knowledge, extraction pattern or document. The RC value for a knowledge triple reflects how confident the system is that it is a correct piece of knowledge, for an extraction pattern that the pattern will extract accurate pieces of knowledge, and for documents that the document provides valid knowledge triples. System added resources, whether documents,

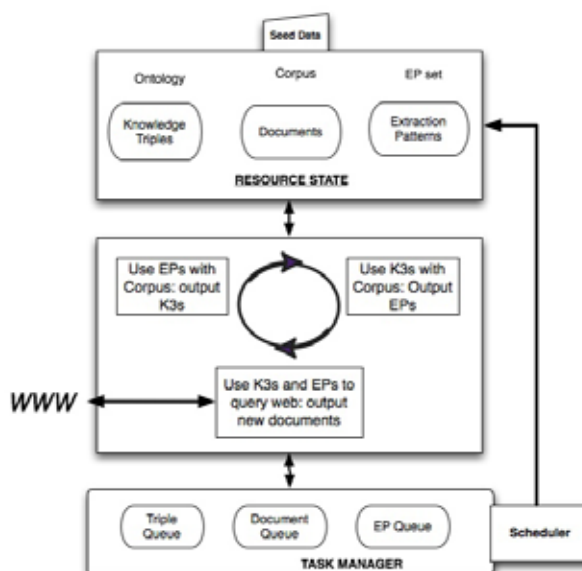


Fig. 1: Overview of the system

knowledge triples or extraction patterns are assumed to have varying degrees of confidence which is a function of the success or suitability of a given resource in deriving the corresponding other resource. Thus for each resource set, confidence for any resource item is defined in terms of the other resource sets. This means that for any given resource, there is a corresponding set of resource pairs with which it interacts.

The formulae for calculating the RC of any given resource are designed so that a) a single measure combines the effect of the other types of resources; b) the greater the sum of the confidence/RC values of the other resource pairs a given resource is associated with, the greater is the RC of that resource; c) the measure should take into account resource pairs not covered.

For example, for a given knowledge triple  $t_i$ , we aim to combine in one single measure the effect of both extraction patterns which extract the triple, and the documents that the triple is extracted from. An extraction pattern-document pair is defined as the instance of an extraction pattern applied to a given document. The measure favours knowledge triples that are the outcome of many extraction pattern-document pairs (instances) and favours triples that cover extraction pattern-document pairs with a high confidence.

Let  $O$  be the set of co-occurrences of resource pairs - in this case as we are calculating the RC of a triple, the relevant resource pairs are document-extraction pattern pairs. We can conveniently represent this as a triple e.g.  $o_1 = \{d_2, p_1, t_2\}$  which means that occurrence  $o_1$  refers to document  $d_2$  which has a match with EP  $p_1$  to extract knowledge triple  $t_2$ . In the following formulas,  $d_r$  and  $d_w$  are restricted to the specific document in question, while  $d_p$  and  $d_n$  sum over all documents. Let  $d_r$  and  $d_w$  be the number of correct and incorrect documents in the set of document-extraction pattern pairs which output the triple  $t_i$ , and  $d_p$  and  $d_n$  be the number of positive and negative documents in the set of document-extraction pattern pairs which output *all* triples.

$$d_r = \sum_{o \in O_t} RC(d_o) \quad (1)$$

$$d_w = \sum_{o \in O_t} (1 - RC(d_o)) \quad (2)$$

$$d_p = \sum_{o \in O} RC(d_o) \quad (3)$$

$$d_n = \sum_{o \in O} 1 - RC(d_o) \quad (4)$$

Similar functions can be defined analogously for  $p_r$ ,  $p_w$ ,  $p_p$  and  $p_n$ . For further details and examples cf. [1].

$r$  and  $w$  are defined in terms of the quantities defined in formulae 1 to 4 and the analogous formulae for  $p_r$ ,  $p_w$ ,  $p_p$  and  $p_n$ .  $r$  is defined as shown in Eq. (5), where  $d_r - d_n$  are defined as above. The quantity  $r$  trivially combines the contribution of both extraction patterns and documents by summing  $d_r$  and  $p_r$ . A refinement of the quantity, for ranking purposes, is obtained by adding the quotients, which favour triples that cover a greater number of positives, but less and less so as the number of negatives not covered increases.

$$r = d_r + \frac{(d_n - d_w)}{((d_n - d_w) + (d_p - d_r)) + 1} + \quad (5)$$

$$w = d_w + \frac{(p_n - p_w)}{((p_n - p_w) + (p_p - p_r))} + \quad (6)$$

$$d_p + \frac{(d_p - d_r)}{((d_p - d_r) + (d_n - d_w)) + 1} + \quad (6)$$

$$p_w + \frac{(p_p - p_r)}{((p_p - p_r) + (p_n - p_w))}$$

The quantity  $w$  is the symmetric of the formula for  $r$  as shown in Eq. (6). Then the Resource Confidence (RC) for a given Knowledge Triple (for example  $t_i$ ) is defined as shown in Eq. (7) which is merely the classic precision measure adapted for our purposes.

$$RC(t_i) = \frac{r}{(r + w)} \quad (7)$$

User-provided RC scores work as seeds and/or feedback to the system thereby optionally guiding the system as it runs. Extraction patterns are currently represented as described in [4]. The incompleteness of the corpus is tackled by iterative augmentation using the web or any other institutional repository as a corpus. Corpus augmentation in our approach consists of a set of methods that aim to incrementally add new documents to the corpus, such that documents with higher relevance to the domain are added first. Stopping criteria are established by setting a threshold on the lowest acceptable RC for each resource type, or by setting a threshold on the maximum number of iterations, without any new candidate resources for each resource type being obtained.

```

Corpus C = {d} a set of documents
Extraction Pattern Set P = {p} a set of extraction
patterns
Ontology O = {t} a set of knowledge triples

{
1. State (seed) data (C, P, O)
2. Candidates queues set to empty (C', P', O')
3a. Apply P and term recognition (using a Noun Phrase
chunker) to discover triples in C;
3b. Apply pattern induction to discover p in C;
3c. Download more texts by applying O with Ps;

4. Score discovered resources with RC;
5. Place each discovered resource into corresponding
candidate queue (CC, CEP, CT);
6. Pop the resource with the highest RC from the
candidate queues and add it to state (C, EP, T);
7. Apply rationalisation;
8. Re-score resources in C', P', O' and C, P, O;
9a. If a triple t has been added, instantiate P with t to
query the web and download more texts using the triple t;
9b. If an extraction pattern p has been added, apply p
over state C to discover new triples;
9c. If a document d has been added, apply P and term
recognition over the text to discover triples;
10. Go to Step 2;
}

```

Table 1: The Bootstrapping Algorithm

### 3.2 Bootstrapping Algorithm

The bootstrapping algorithm is shown in Table 1. Bootstrapping starts with the user providing some seed data (1,2). Initial processing includes applying the extraction patterns to the seed corpus to extract any available knowledge triples (3a), and learning new extraction patterns (3b). If the seed corpus is small, additional texts are obtained from the WWW by querying a search engine using the seed ontology and extraction patterns and added to the seed corpus (3c). A small corpus defines the domain weakly, in which case the RC scores would not correctly reflect the relevance of a resource to the domain.

The knowledge resources extracted by the initial processing are scored by applying the RC formula (4), and placed in the three resource queues (5). The queues contain candidate resources, sorted based on their RC in descending order, to be processed in following iterations.

The Scheduler component (see Figure 1) determines the following steps (6), in which the bootstrapping process polls the queues, and adds one resource to the system state at a time. Different schedulers implement different measures to determine which type of resources to be polled. In the experiment reported in this paper, the scheduler compares the RCs of the top-most resource in each queue, and adds the one with the highest RC to the state. Other measures which, for example, reflect how users intervene with the system and whether the user wants to supervise ontology learning, or corpus building, or pattern induction are also implemented, but not used in our current experiment.

Once a resource is added to the state, the bootstrapping applies rationalisation (7) and re-scores the state and candidate resources (8). Rationalisation rearranges the ontology so as to remove redundancy and make the ontology more coherent.

Following the addition of the resource, a new learning iteration is triggered (9). The system then contin-

ues cycling through the stages described above, (see Table 1) and iterates until stopping criteria are met.

## 4 Evaluation

Ontology evaluation is challenging topic in itself because knowledge cannot easily be enumerated, catalogued or defined *a priori* so as to allow for some sort of comparison to be made with the output of ontology tools. Various proposals have been made in the literature and an evaluation by Gold Standard (GS) was chosen in our case. For that purpose, we created a domain-specific hand-crafted ontology reflecting common sense knowledge about animals, containing 186 concepts up to 3 relations deep<sup>1</sup>. In order to compare the GS ontology with the computer generated one, we chose to follow the methodology proposed by Dellschaft and Staab [3]. The following metrics are thus used: Lexical Precision (LP) and Recall (LR) measure the coverage of terms in the GS by the output ontology; Taxonomic Precision (TP) and Recall (TR) aims to identify a set of characteristic features for each term in the output ontology and compare them with those of the corresponding term in the GS; Taxonomic F-measure (TF) is the harmonic mean of TP and TR, while Overall F-measure (TF') is the harmonic mean of TP, TR, LP and LR.

As a seed corpus we used a set of 40 texts from Wikipedia all entries concerning animals which were included in the GS ontology. All were entries for commonly known animals such as *hedgehog*, *lion*, *kangaroo*, *ostrich*, *lizard*, amounting to a little over 8000 words. Note there is a substantial gap between the number of animals initially covered in the articles and the number present in the GS ontology. The articles were pre-processed to remove the markup present in the originals.

A series of experiments were conducted, each time varying the seed knowledge input to the *Abraxas* system (in this paper we only present the one experiment, where Corpus = 40 Wikipedia texts, and Ontology = {dog ISA animal} - fuller details may be found in [1]). In all cases we used as a stopping criterion the Explicit Knowledge Gap (EKG) measure described in [6, 1]. This is a measure of the extent to which the ontology and the corpus are in equilibrium in the sense of the corpus providing explicit evidence for the items in the ontology. EKG is defined in Eq. 8 where  $E$  is the set of pairs of terms whose ontological relationship is explicit,  $\Pi$  is the set of pairs of terms in the corpus that are known to have some kind of ontological relationship on distributional grounds. The systems seeks to minimise EKG but in practice we use an empirically chosen threshold.

$$EKG = |E \cap \Pi| \quad (8)$$

We used the same set of 6 extraction patterns, shown in Table 2, which previous research had shown to have good precision [1]. Pattern learning was disabled in order to separate concerns - we intended to isolate the ontology learning process from the influence of pattern learning in these experiments, making results

<sup>1</sup> Publicly available from <http://nlp.shef.ac.uk/abraxas/>

NP(pl) such as NP*	NP(sg) is a kind of NP(sg)
NP(sg) or other NP(pl)	NP(sg) is a type of NP(sg)
NP(pl) and other NP(pl)	NP(pl) or other NP(pl)

**Table 2:** Extraction patterns used: NP = noun phrase, sg = singular, pl = plural.

LP	0.40	LR	0.48
TP	0.95	TR	0.70
TF	0.81	TF'	0.60

**Table 3:** Results obtained for experiment 1.

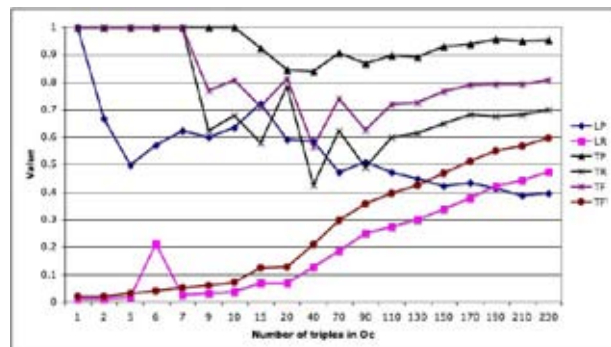
more comparable with those of the literature. For the same reasons, the system was tested in a completely unsupervised manner.

**Comparison with Gold Standard** Our initial experiment was with Case 1, running over approximately 500 iterations. The final results are shown in Table 3. Both the TF and TF' obtained are significantly better than equivalent results in the literature, which often achieve maximum scores around [0.3] for both precision and recall [2].

**Learning Curves** Figure 2 shows how the results vary over the number of iterations. We can see here that LR steadily increases reflecting the growing size of the ontology and correspondingly its overlap with the GS. In contrast, LP is in constant flux but with a tendency to decrease. TP varies between set limits of [1.0 - 0.84] indicating that concepts are generally inserted correctly into the hierarchy. TR is also a measure in considerable flux and manual analysis of the different output ontologies show that sudden insertion of parent nodes (e.g. *mammal* at iteration 9) make a substantial difference which gradually stabilises over further iterations. Over long numbers of iterations, this flux in TR seems to become less likely. We also observe a steady increase TF' in parallel with the increase in LR indicating that the system is doing better as it increases its coverage of the lexical layer of the GS ontology.

## 5 Discussion

The low LP and LR do not accurately reflect the real quality of the generated ontology. LP has a tendency to decrease because the system is using the Web as a corpus, so it will inevitably include items absent from the GS. On the other hand, manual inspection



**Fig. 2:** Evaluation measures (LP, LR, TP, etc.) plotted against the sequentially produced ontologies from the iterative process.



of the ontology showed that in 230 triples, there were 225 concepts of which only 14 could be clearly seen to belong to another domain (flour, book, farmer, plant etc.), and another 10 were borderline (predatory bird, domestic dog, wild rabbit, large mammal, small mammal, wild fowl, etc.). So a manual evaluation would suggest 201 correct terms or [0.89] precision. The gradually falling LP presents a challenge for ontology learning and may either need a different approach to evaluating this element or a need for techniques which focus the ontology more effectively.

The flux shown in the graph presented in Figure 2 in the early stages shows that in principle as more data is added to the system the output becomes more stable and consistent. The general tendency is for the measures to move upwards indicating a gradual but steady improvement over the progression of the iterations. These results are as was hoped and reflect the capacity of the system to adapt as the data added to the system changes the confidence values for individual items of knowledge. The high F measures for the system show that our approach has fundamental validity.

Given the high quality of the output of this approach the question arises whether this is really what is needed. Is this type of ontology too focussed and does it just succeed algorithmically to re-create the well-known tennis problem [11]? This can only be answered by further experimentation and evaluation, varying the parameters of the approach.

## 6 Related Work

For an over view of research in OL, please consult [9]. More extensive descriptions of related work can be found in [6, 1].

The original inspiration for using lexico-syntactic patterns is [5] and developed by many other authors since. A number of authors have worked on ways to build ontologies accessing resources beyond the original corpus, e.g. [2] experiment with using data from WordNet, the Web (in general) and the counts provided by Google; [10] introduced an approach for automatically acquiring hypernyms and hyponyms for any given term using search engines. The bootstrapping learning approach inspiration from [14], [12] and [4]. Combining the use of the Web as a corpus and the bootstrapping approach, Etzioni et al. have created the KnowItAll system to collect factual information for a given domain, and provided one module that learns taxonomic relations [7].

## 7 Conclusion

In this paper, we have presented an iterative dynamic and adaptive system for ontology learning. The system is designed to achieve a balance between three open ended resources, a corpus, an ontology and a set of extraction patterns. We have described the key principles that lead to the system design and the key aspects of the system architecture and shown in our evaluation that the system is able to generate domain specific ontologies of good quality ( $TF^1 = [0.5 - 0.6]$ ).

There are a number of objectives in our future work. First we plan to perform experiments to identify where the methodology fails especially concerning abstract concepts which are absent from the text collection. Secondly, we plan to fully evaluate the influence of pattern learning in the overall ontology learning process with a series of new experiments. Finally, we plan to investigate the application of our approach in important domains such as biomedical texts.

## References

- [1] C. Brewster. *Mind the Gap: Bridging from Text to Ontological Knowledge*. PhD thesis, Department of Computer Science, University of Sheffield, 2007.
- [2] P. Cimiano, A. Pivk, L. Schmidt-Thieme, and S. Staab. Learning taxonomic relations from heterogeneous sources of evidence. In *Ontology Learning from Text: Methods, Evaluation and Applications*, Frontiers in Artificial Intelligence. IOS Press, 2005.
- [3] K. Dellschaft and S. Staab. On how to perform a gold standard based evaluation of ontology learning. In I. F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, editors, *International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 228–241. Springer, 2006.
- [4] O. Etzioni, M. J. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web an experimental study. *Artif. Intell.*, 165(1):91–134, 2005.
- [5] M. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING 92)*, Nantes, France, July 1992, 1992.
- [6] J. Iria, C. Brewster, F. Ciravegna, and Y. Wilks. An incremental tri-partite approach to ontology learning. In *Proceedings of the Language Resources and Evaluation Conference (LREC-06)*, 22-28 May, Genoa, Italy, 2006.
- [7] A.-M. Popescu, A. Yates, and O. Etzioni. Class Extraction from the World Wide Web. In *Proceedings of the AAAI-04 Workshop on Adaptive Text Extraction and Mining (ATEM-04)*, San Jose, CA, July 2004.
- [8] M. Sabou. From software APIs to web service ontologies a semi-automatic extraction method. In S. A. McIlraith, D. Plexousakis, and F. van Harmelen, editors, *International Semantic Web Conference*, volume 3298 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 2004.
- [9] M. Shamsfard and A. A. Barforoush. The state of the art in ontology learning; a framework for comparison. *The Knowledge Engineering Review*, 18(04):293–316, 2004.
- [10] R. Sombatsrisomboon, Y. Matsuo, and M. Ishizuka. Aquisition of hypernyms and hyponyms from the WWW. In *Proc. of 2nd International Workshop on Active Mining (AM2003)*, pages 7–13, Maebashi, Japan, 2003.
- [11] M. Stevenson. Combining disambiguation techniques to enrich an ontology. In *Proceedings of the Fifteenth European Conference on Artificial Intelligence (ECAI-02) workshop on Machine Learning and Natural Language Processing for Ontology Engineering*, Lyon, France, 2002.
- [12] M. Thelen and E. Riloff. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 214–221, Philadelphia, PA, July 2002.
- [13] S.-H. Wu and W.-L. Hsu. SOAT a semi-automatic domain ontology acquisition tool from chinese corpus. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Academia Sinica, ACLCLP, and National Tsing Hua University, Taiwan, Morgan Kaufmann, August 2002.
- [14] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. Automatic acquisition of domain knowledge for information extraction. In *COLING 2000, 18th International Conference on Computational Linguistics, Proceedings of the Conference, 2 Volumes, July 31 - August 4, 2000, Universität des Saarlandes, Saarbrücken, Germany*, pages 940–946. Morgan Kaufmann, 2000.

# Multi-Word Units in Treebank-Based Probabilistic Parsing and Generation

Conor Cafferkey, Deirdre Hogan and Josef van Genabith  
National Centre for Language Technology (NCLT)  
School of Computing, Dublin City University  
Dublin 9, Ireland  
{ccafferkey,dhogan,josef}@computing.dcu.ie

## Abstract

Multi-word units (MWUs) such as named entities (NEs) and other fixed and semi-fixed expressions are important in information retrieval, extraction and question answering. It is also often assumed that MWUs are useful in parsing as their identification can reduce the overall complexity of the task. Despite this, we are not aware of any previous work on the use of MWUs in treebank-based probabilistic parsing, nor in the converse operation—probabilistic generation (or surface realisation). We present the results of several experiments using MWUs as a means to impose constraints on both probabilistic parsing and generation with automatically-acquired (treebank-based) grammars. In the case of generation from treebank-acquired Lexical Functional Grammar (LFG) f-structure approximations we show that modest improvements in accuracy can be made. Our experiments integrating the same MWUs in treebank-based probabilistic parsing yield smaller, but still statistically significant gains. We analyse the results and offer a number of explanations as to why the gains achieved are smaller than might be naively expected.

## Keywords

multi-word units, named entities, parsing, generation, surface realisation

## 1 Introduction and motivation

There exists a large and growing body of research on the identification and applications of multi-word units (MWUs) of various types. For example, the identification of named entities (NEs) has been the focus of a number of shared tasks and workshops [15, 16].

In this paper we explore the possibilities of making use of MWUs in both probabilistic parsing and generation (surface realisation from LFG f-structures). Intuitively, the use of MWUs in parsing and generation should reduce the complexity of the tasks as, given a particular string, identifying MWUs reduces the overall number of effective terminal tokens. In particular we expect that the identification of MWUs may be useful in resolving attachment ambiguities in parsing and imposing word-order constraints in generation, respectively.

We take Bikel’s [1] implementation of Collins’s model 2 [6] as the baseline for our parsing experiments. A history-based generator based on the PCFG-based generator of Cahill and van Genabith [3] is used as our baseline for surface realisation from treebank-acquired LFG f-structure approximations [2].

To date, there exists a surprisingly small amount of literature on the integration of MWUs in statistical parsing and generation. Nivre and Nilsson document the use of MWUs in deterministic dependency parsing of Swedish, showing modest but significant gains [13]. Kaplan and King describe the use of MWUs (specifically proper-noun named entities) in the context of LFG parsing using a large hand-crafted LFG grammar, again with clear gains [10]. We are not aware of any comprehensive evaluation of the use of MWUs as constraints in probabilistic parsing and surface realisation based on automatically-acquired (treebank-based) grammars.

The remainder of the paper is structured as follows: in Section 2 we describe the parsing technology used in our experiments. Section 3 describes the history-based generator used in our experiments. In Section 4 we describe the specific parsing and generation experiments carried out. Section 5 summarises our experimental results. Finally, Section 6 gives a more in depth discussion of our results and provides explanations as to why the gains established are not more pronounced.

## 2 Treebank-based probabilistic parsing

We carry out our parsing experiments using Bikel’s [1] implementation of Collins’s history-based lexicalised generative parsing model [6].

The past decade has seen considerable advances in probabilistic parsing models trained on corpora such as the Penn Wall Street Journal (WSJ) treebank [12]. Although the current state-of-the-art labeled bracket recall and precision is in the region of 90% [4, 6] there still exist many hurdles in the way of any further gains in parsing performance. Among such hurdles are the well-documented problems of PP-attachment and coordination.

There have also been limited investigations into incorporating chunking in treebank-based probabilistic parsing. For example, Glaysher and Moldovan [7] make use of a SVM-based chunker to constrain and

hence speed up the Collins parser.

We investigate the use of automatically identified multi-word units (specifically multi-word named entities and prepositional multi-word expressions) as a means to constrain the parser in a similar fashion with the aim of reducing complexity and hence improving parse quality.

### 3 Surface realisation from f-structures

We carry out the surface realisation experiments using an extension of the generation model of Cahill and van Genabith [3] using the treebank- and PCFG-based LFG approximations of Cahill et al. [2]. For generators which do not rely on hand-crafted grammars and are thus easily ported to new languages, this generator achieves state-of-the-art accuracy and coverage. The generation process takes as input an LFG f-structure [9] and outputs a sentence for that f-structure. LFG is a constraint-based theory of grammar, which analyses strings in terms of c(onstituency)-structure and f(unctional)-structure. C-structure is defined in terms of CFGs, and f-structures are recursive attribute-value matrices which represent abstract syntactic functions. See Figure 1 for an example of an LFG c-structure tree (on the left) linked to an LFG f-structure (on the right).

C-structures and f-structures are related in a projection architecture in terms of a piecewise correspondence  $\phi$ .<sup>1</sup> The correspondence is indicated in terms of the curvy arrows pointing from c-structure nodes to f-structure components in Figure 1. Given a c-structure node  $n_i$ , the corresponding f-structure component  $f_j$  is  $\phi(n_i)$ . F-structures and the c-structure/f-structure correspondence are described in terms of functional annotations on c-structure nodes (CFG grammar rules). An equation of the form  $(\uparrow F) = \downarrow$  states that the f-structure associated with the mother of the current c-structure node ( $\uparrow$ ) has an attribute (grammatical function) (F), whose value is the f-structure of the current node ( $\downarrow$ ). The up-arrows and down-arrows are shorthand for  $\phi(M(n_i)) = \phi(n_i)$  where  $n_i$  is the c-structure node annotated with the equation.<sup>2</sup>

The generation model of [3] maximises the probability of a tree given an f-structure (Eqn. 1), and the string generated is the yield of the highest probability tree. The generation process is guided by local information in the input f-structure: f-structure annotated CFG rules (LHS  $\rightarrow$  RHS) are conditioned on their LHSS and on the set of features/attributes  $Feats = \{a_i | \exists v_j \phi(LHS) a_i = v_j\}$ <sup>3</sup>  $\phi$ -linked to the LHS (Eqn. 2). Table 1 shows a generation grammar rule and conditioning features extracted from the example in Figure 1. The probability of a tree is decomposed into the product of the probabilities of the f-structure annotated rules (conditioned on the LHS and local

*Feats*) contributing to the tree. Conditional probabilities are estimated using maximum likelihood estimation.

grammar rule	local conditioning features
$S(\uparrow=\downarrow) \rightarrow NP(\uparrow_{SUBJ}) VP(\uparrow=\downarrow)$	$S(\uparrow=\downarrow), \{SUBJ, OBJ, PRED, TENSE\}$

**Table 1:** Example grammar rule (from Figure 1)

$$Tree_{best} := \operatorname{argmax}_{Tree} P(Tree|F-Str) \quad (1)$$

$$P(Tree|F-Str) := \prod_{\substack{X \rightarrow Y \text{ in Tree} \\ Feats = \{a_i | \exists v_j (\phi(X)) a_i = v_j\}}} P(X \rightarrow Y|X, Feats) \quad (2)$$

In the extension of the generator of [3] that is used for the experiments in this paper, all generation grammar rules are also conditioned on the mother f-structure feature label (GF). For example, from Figure 1, the conditioning context for the rule  $PRP(\uparrow=\downarrow) \rightarrow her$  is increased so that it includes the mother f-structure feature label OBJ. The probabilistic generation model is defined as:

$$P(Tree|F-Str) := \prod_{\substack{X \rightarrow Y \text{ in Tree} \\ Feats = \{a_i | \exists v_j (\phi(X)) a_i = v_j\} \\ (\phi(M(X))) GF = \phi(X)}} P(X \rightarrow Y|X, Feats, GF) \quad (3)$$

A chart generation algorithm based on that of Kay [11] generates phrase structure trees for the input f-structure.

## 4 Experimental setup

We now discuss our experiments incorporating multi-word units in the parsing and generation processes. We carry out experiments with MWUs from three different sources. First, we use the output from Chieu and Ng's maximum entropy-based named entity recognition (NER) system [5]. This system identifies four types of NE: person, organisation, location, and miscellaneous. Secondly, we use a dictionary of candidate multi-word prepositional expressions.<sup>4</sup> Finally, we carry out experiments with multi-word units extracted from the BBN Pronoun Coreference and Entity Type Corpus [17]. This supplements the Penn WSJ treebank with additional annotations of 29 named entity types, including nominal-type NES such as person, organisation, location, etc. as well as numeric types such as date, time, quantity and money. Since the BBN corpus data is very comprehensive and is hand-annotated we take this to be a gold standard, representing an upper bound for any gains that might be made by identifying multi-word NES in our experiments.<sup>5</sup> Table 2 gives examples of the various types of MWUs identified by the three sources.

<sup>1</sup> Our formalisation follows [8].

<sup>2</sup> M is the mother function on CFG tree nodes.

<sup>3</sup> In words, *Feats* is the set of top level features/attributes (those attributes  $a_i$  for which there is a value  $v_i$ ) of the f-structure  $\phi$  linked to the LHS.

<sup>4</sup> Based on a list from [mwe.stanford.edu](http://mwe.stanford.edu)

<sup>5</sup> It is possible that other types of MWUs might be more suited to the task than the NES identified by the BBN corpus, so further gains might in fact be possible.



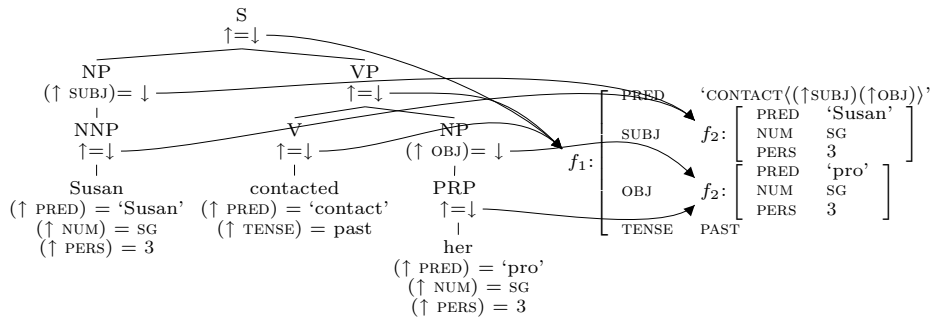


Fig. 1: C- and f-structures with  $\phi$  links for the sentence Susan contacted her.

MWU type	Examples
Names	Martha Matthews Yoshio Hatakeyama
Organisations	Rolls-Royce Motor Cars Inc. Washington State University
Locations	New York City New Zealand
Time expressions	October 19th two years ago the 21st century
Quantities	\$2.7 million to \$3 million about 25 % 60 mph
Prepositional expressions	in fact at the time on average

Table 2: Examples of some of the various types of MWU from our three sources

For our purposes we are not concerned with the distinctions between the different types of MWUs; we merely exploit the fact that they may be treated as atomic units in the parsing and generation models. In all cases we disregard MWUs that cross the original syntactic bracketing of the WSJ treebank. An overview of the frequencies and lengths of the various types of MWUs used in our experiments is presented in Table 3.

	average number	average length
Chieu and Ng NER	0.61	2.40
MWE list	0.10	2.48
BBN corpus	1.15	2.66

Table 3: Average number of MWUs per sentence and average MWU length in the WSJ treebank grouped by MWU source

In our parsing experiments which incorporate MWUs the WSJ treebank training and test data are modified (in fact, retokenised) such that the word tokens comprising MWUs are concatenated into single words (for example, *Bank of America* becomes *Bank\_of\_America*). The concatenated token assumes the part-of-speech tag of its head word constituent.<sup>6</sup> Figure 2 shows a tree fragment from the treebank (on the left) and the tree fragment after retokenisation (on

<sup>6</sup> We used the head-finding rules given in Collin’s thesis [6] to determine constituent heads.

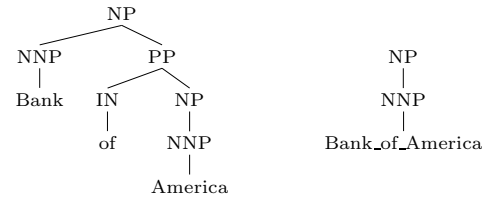


Fig. 2: Two different trees for the phrase Bank of America, with the treebank tree on the left and the tree after retokenisation on the right

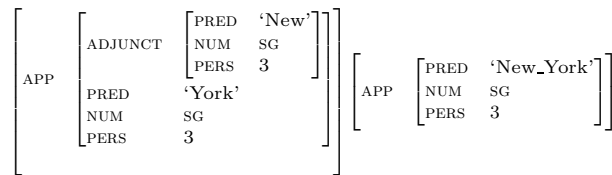


Fig. 3: Two different f-structures for the phrase New York, with the treebank-acquired f-structure on the left and the f-structure after retokenisation on the right

the right).

For generation we carry out two types of experiment. In the first, *type a*, the training and test sets are retokenised such that multi-word units are concatenated into single words. In the second, *type b*, only the test data is retokenised with no retokenisation of the training data. Figure 3 shows an f-structure fragment acquired from the treebank (on the left) and the tree fragment after retokenisation (on the right). Strings output by the generator are post-processed so that the concatenated words are converted back into sequences of word tokens.

## 5 Evaluation

All experiments were carried out on the WSJ treebank with sections 02-21 for training (39,832 sentences), section 24 for development (1,346 sentences) and section 23 for final test results (2,416 sentences). As previously noted, Bikel’s [1] parser was used for our parsing experiments while the LFG annotation algorithm of Cahill et al. [2] was used to produce the f-structure inputs for development and test sets the Cahill and

	Recall	Precision	F-score
Baseline	88.53	88.63	88.58
Best automatic MWUs	88.68	88.76	88.72
Best BBN MWUs	88.74	88.87	88.81

**Table 4:** Parsing results for test set (section 23), all sentence lengths

	BLEU	StringEd	Coverage
Baseline	67.24	69.89	99.88
Best automatic MWUs	67.81	70.36	99.92
Best BBN MWUs	68.82	70.92	99.96

**Table 5:** Generation results for test set (section 23), all sentence lengths

van Genabith generator [3].

## 5.1 Parsing results

Table 4 shows our best parsing results for section 23. For each result we present labeled bracket recall, precision and f-score measures against the retokenised gold file. Improvements are statistically significant at level 0.01 over the baseline according to a stratified shuffling test with 10,000 iterations.<sup>7</sup>

In Table 4, *Baseline* refers to the unaltered Bikel parser, with no MWU-retokenisation of the WSJ corpus data. *Best automatic MWUs* refers to our best results using automatically acquired MWUs. These were achieved using the NES identified by Chieu and Ng’s NER system combined with the list of candidate prepositional multi-word expressions. *Best BBN MWUs* refers to our best results using the BBN corpus named entities, achieved using proper-noun NES (those denoted ENAMEX).

## 5.2 Generation results

Table 5 shows our final generation results for section 23. For each test we present the BLEU score [14] as well as simple string accuracy and coverage. We use a bootstrap resampling method, popular for machine translation evaluation, with a resampling rate of 1,000 to measure the significance of improvements in BLEU scores.<sup>8</sup> We also calculate the significance of increases in simple string accuracy by carrying out a paired t-test on the mean difference of the simple string accuracy scores. For both tests improvements are significant at level 0.001 over the baseline.

In Table 5, *Baseline* refers to the history-based generator, as described in Section 3, not incorporating any type of MWU. *Best automatic MWUs* displays our best results using automatically-identified MWUs. These were achieved using experiment *type b*, described in Section 4, with the MWUs produced by Chieu and Ng’s NE recogniser [5]. The final row in Table 5 shows the results using the BBN corpus-derived multi-word units incorporated in a *type a* experiment.

<sup>7</sup> Script from [www.cis.upenn.edu/~dbikel/software.html](http://www.cis.upenn.edu/~dbikel/software.html)

<sup>8</sup> Scripts from [tinyurl.com/2b66vs](http://tinyurl.com/2b66vs)

## 6 Discussion

We now discuss the MWU experiments in more detail. Table 6 provides a breakdown of the parsing experiments on the development set (WSJ section 24) while Table 7 details the generation experiments on the same set. First, MWUs came from the named entity recogniser of [5], then we added the MWUs from the list of candidate prepositional multi-word expressions and finally we ran tests with MWUs extracted from the BBN corpus.

### 6.1 Parsing

In our initial experiments we observed, surprisingly, that parsing with the BBN NES gives performance nearly identical to parsing with the Chieu and Ng NES. To determine why this might be the case we ran additional experiments where we split the BBN NES into three broad categories: name expressions (denoted ENAMEX), number expressions (NUMEX) and time expressions (TIMEX). We observe that ENAMEX category clearly preforms best—NUMEX and TIMEX actually have a detrimental effect on parsing performance. Based on inspection of the data we believe that this is because the word groupings of the name expressions are more consistent with the syntactic bracketings of the WSJ treebank than those of the other categories.

Although using the BBN name expressions alone does yield better results than incorporating all types, the gains remain small. We identified several potential factors brought about by the retokenisation of the WSJ corpus data that might militate against higher results: the reduction in the number of syntactic bracketings; alterations to lexical distributions in the data; and the frequency of the MWUs.

In all of our parsing experiments the MWU-retokenisation of the corpus leads to an overall reduction in the number of possible syntactic bracketings (due to the reduction of the amount of effective word tokens). Fewer syntactic bracketings per parse tree means that a single mistake will be penalised more heavily by the bracketing recall and precision metrics. Table 8 shows the number of syntactic bracketings in the development set (WSJ section 24) per MWU source. The reduction in the number of brackets is actually quite modest, and it is therefore unlikely to be a major contributing factor to the small size of the improve-

	Recall	Precision	F-Score
Baseline	87.58	88.18	87.88
Chieu and Ng NES	87.65	88.18	87.91
+ MWE list	87.68	88.23	87.95
All BBN NES	87.52	88.35	87.93
BBN ENAMEX NES	87.73	88.28	88.01
BBN NUMEX NES	87.44	88.25	87.84
BBN TIMEX NES	87.44	88.07	87.76

**Table 6:** Parsing results for development set (section 24), all sentence lengths

		BLEU	StringEd	Coverage
Baseline		65.85	69.93	99.93
Type a	Chieu and Ng NES	65.81	70.34	99.93
	+MWE list	64.81	69.67	99.93
	BBN NES	67.24	71.46	99.93
Type b	Chieu and Ng NES	66.37	70.26	99.93
	+MWE list	66.28	70.21	99.93
	BBN NES	66.84	70.74	99.93

**Table 7:** Generation results for development set (section 24), all sentence lengths

	# brackets
Baseline	25,662
Chieu and Ng NES	25,633
BBN ENAMEX NES	25,589

**Table 8:** Number of brackets per gold file (section 24)

ments.

The MWU-retokenisation also leads to alterations to the lexical distribution of the corpus. When the constituent tokens of the MWUs are concatenated the number of rare lexical events (those observed, say,  $\leq 2$  times in the training data) increases quite substantially (Table 9). This is likely to have a detrimental effect on parse quality due to the lexicalised nature of the parsing model used (in such a parser increasing the number of infrequently occurring words increases the sparsity of the training data).

	1 occurrence	$\leq 2$ occurrences
Baseline	20,622	27,049
Chieu and Ng NES	26,949	33,945
BBN ENAMEX NES	26,703	33,528

**Table 9:** Number of rare words (those observed  $\leq 2$  times in sections 02-21)

Finally, the MWUs identified are in fact relatively infrequent (Table 3) and as such any gains brought by exploiting these units are likely to be quite small.

## 6.2 Generation

For generation, our first set of experiments (*type a*), where both training data and development set data were retokenised, produced the worst results for the automatically identified MWUs. Accuracy actually decreased for these experiments. In an error analysis of *type a* experiments with the Chieu and Ng MWUs,

we inspected those sentences where accuracy had decreased from the baseline. We found that for over half (51.5%) of these sentences, the input f-structures contained no multi-word units at all. The problem for these sentences therefore lay with the probabilistic grammar extracted from the MWU-retokenised training data. When the source of MWU was the BBN corpus, however, accuracy improved significantly over the baseline and the result is the highest accuracy achieved over all experiment types. We suspect that the low accuracies for the automatically acquired MWUs in the *type a* experiments are due to noisy MWUs which negatively affect the grammar (Chieu and Ng’s system achieved an f-score of 88.3% in the CONLL 2003 NER task [16]).

In order to avoid changing the grammar and thus risking side-effects which cause some heretofore likely constructions become less likely and vice versa, we ran the next set of experiments (*type b*) which leave the original grammar intact and alter the input f-structures only. These experiments were more successful overall and we achieved an improvement over the baseline for both BLEU and String Edit Distance scores with all MWU types. As can be seen from Table 7 the best score for automatically identified MWUs are with the Chieu and Ng MWUs (accuracy decreases marginally when we added the MWUs from the list of propositional MWE candidates).

## 7 Conclusion and future work

To the best of our knowledge, this paper presents the first study of the influence of MWU-preprocessing on state-of-the-art treebank-based probabilistic parsing and generation. We have shown that statistically significant gains can be made by exploiting MWUs as constraints in probabilistic parsing and generation.

Overall the gains achieved were small, implying that for the unit types used in our experiments MWU-preprocessing has relatively limited utility. On the

other hand, given the relatively small amount of modifications to the corpus incurred by marking up the MWUs, it stands to reason that any gains should also be small. There exists scope for investigations into the influence of other classes of MWU on parsing and generation.

## Acknowledgments

Research supported by the Irish Research Council for Science, Engineering and Technology: funded by the National Development Plan and Microsoft Research Limited.

## References

- [1] D. M. Bikel. Design of a multi-lingual, parallel-processing statistical parsing engine. In *Proceeding of the Human Language Technology Conference (HLT)*, 2002.
- [2] A. Cahill, R. O. M. Burke, J. van Genabith, and A. Way. Long-distance dependency resolution in automatically acquired wide-coverage pcfg-based lfg approximations. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, Barcelona, Spain, 2004.
- [3] A. Cahill and J. van Genabith. Robust PCFG-based generation using automatically acquired LFG approximations. In *Proceedings of the 44th ACL*, 2006.
- [4] E. Charniak. A maximum entropy-inspired parser. In *Proceedings of the 1st NAACL*, 2000.
- [5] H. L. Chieu and H. T. Ng. Named entity recognition with a maximum entropy approach. In *Proceedings of the CoNLL*, 2003.
- [6] M. Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.
- [7] E. Glaysher and D. I. Moldovan. Speeding up full syntactic parsing by leveraging partial parsing decisions. In *ACL*, 2006.
- [8] R. Kaplan. The formal architecture of lexical-functional grammar. In Dalrymple, Kaplan, Maxwell, and Zaenen, editors, *Formal Issues in Lexical-Functional Grammar*, pages 7–27. CSLI Publications, 1995.
- [9] R. Kaplan and J. Bresnan. Lexical functional grammar, a formal system for grammatical representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. MIT Press, Cambridge, MA, 1982.
- [10] R. M. Kaplan and T. H. King. Low-level mark-up and large-scale lfg grammar processing. In *Proceedings of the Lexical Functional Grammar Conference*, 2003.
- [11] M. Kay. Chart generation. In *Proceedings of the 34th ACL*, 1996.
- [12] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1994.
- [13] J. Nivre and J. Nilsson. Multiword units in syntactic parsing. In *Workshop on Methodologies and Evaluation of Multiword Units in Real-World Applications*, 2004.
- [14] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation, 2001.
- [15] E. F. Tjong Kim Sang. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158. Taipei, Taiwan, 2002.
- [16] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada, 2003.
- [17] R. Weischedel and A. Brunstein. *BBN pronoun coreference and entity type corpus*. Linguistic Data Consortium, 2005.

# Recompiling a knowledge-based dependency parser into memory

Sander Canisius and Antal van den Bosch  
ILK / Dept. of Communication and Information Sciences  
Tilburg University  
P.O. Box 90153, NL-5000 LE Tilburg, The Netherlands  
{*S.V.M.Canisius,Antal.vdnBosch*}@uvt.nl

## Abstract

Data-driven parsers tend to be trained on manually annotated treebanks. In this paper we describe two memory-based dependency parsers trained on treebanks that are automatically parsed by a knowledge-based parser for Dutch. When compared to training on a manual treebank of Dutch, the memory-based parsers exhibit virtually the same performance at the same amount of training material, and achieve markedly higher parsing accuracies when trained on more data. The first memory-based parser is based on a single classifier and operates in linear time, while the second parser employs constraint satisfaction inference (CSI) over three classifiers that each perform a parsing subtask. The non-linear CSI-based parser outperforms the linear parser. Based on this case study we discuss the possibilities of re-engineering knowledge-based parsers in memory.

## Keywords

Dependency parsing, memory-based learning, constraint satisfaction inference

## 1 Introduction

Within the last half century many computational natural language parsers have been designed and implemented. Until a decade ago, most available parsers were rule-based and manually built, drawing on explicit linguistic knowledge. For instance, for Dutch a prime example is the Alpino parser [7], implementing a HPSG-based stochastic attribute-value grammar. Probably the best parser for Dutch, Alpino is a typical modern example of a rule-based approach that has hybridized with a stochastic, data-driven approach. After a rule-based core generates possible parses for a given sentence (possibly hundreds or thousands), a stochastic component searches in this space of possibilities for the most likely parse, given a background collection of example parses, a so-called treebank. Using machine learning methods such as maximum entropy, this stochastic component can be efficiently trained and run [8, 7]. Alpino, available as an open source software system<sup>1</sup>, comes with both the

parser and the manually annotated treebank on which it was optimized.

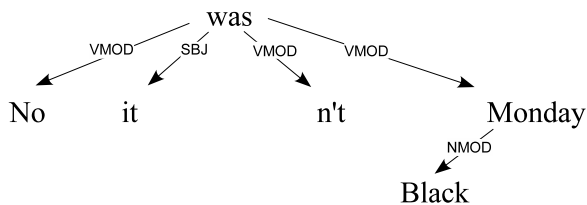
The Alpino treebank took several person years to annotate [11], and can now also be used to train any machine-learning-based or stochastic parser on. Nevertheless it has a limited size of about 262 thousand words (currently). Due to the distributional properties of words, sentence-level natural language processing systems based on machine learning tend to improve generalization performance when more data is available [1, 9]. Hence, it is relevant to investigate methods beyond manual annotation by which more annotated data can be harvested and employed.

One route that has not been explored earlier is to take the already existing parser and apply it to a large amount of digitally available unannotated Dutch text, and use the automatically parsed data as (additional) training examples. Of course these parses contain all the errors that Alpino makes, and without human inspection it cannot be known where the errors are. Training a supervised machine learning method on this partly erroneous data will lead to a system that may therefore never be better than the parser. At the same time, the trained system may in fact become as accurate as the parser itself in the long run; it may learn to behave exactly as Alpino would.

In this paper we present an attempt at recompiling the Alpino parser into two variants of a memory-based dependency parser, one simple and one more complex, which are not only trained on Alpino treebank data converted to dependency structures, but also on large amounts of unannotated texts parsed by Alpino. The two variants are tested on various types of text, to test their out-of-domain robustness. It is shown that the two memory-based parsers can improve beyond being trained on the manually annotated treebank, when texts parsed by Alpino are used as training data; having the manual treebank as part of the learning material does not even improve performance. Furthermore, both parsers are fast; the faster of the two processes at least 1,500 words per second, its processing time being linear in function of the length of the input sequence.

The paper is structured as follows. In Section 2 we formulate dependency parsing in a classification framework. We briefly describe IGTREE, a fast approximation of  $k$ -nearest neighbor classification, which is used as the classifier engine. In Section 3 we provide learning curves, error analyses, and measurements of memory usage and speed of the two parsers. We discuss

<sup>1</sup> Alpino: <http://www.let.rug.nl/~vannoord/alp/Alpino/>



**Fig. 1:** Dependency structure for the sentence "No it wasn't Black Monday"

our findings and compare them to the original Alpino parser in Section 4.

## 2 Algorithms

### 2.1 Dependency parsing as classification

The first parsing algorithm we present is a straightforward interpretation of dependency parsing as a classification task. Pairs of words are classified as to whether they are connected by a dependency, and if so by which relation type. The classification instances are represented by a small set of basic features including word forms and part-of-speech tags for the words themselves and those immediately surrounding them. Canisius et al. [3] propose a simple inference scheme for obtaining the most-likely dependency tree from the classified instances for a sentence.

Given a token for which the dependency relation is to be predicted, a number of classification cases have been processed, each of them indicating whether and if so how the token modifies one of the other tokens in the sentence. In case of classification errors, a token may have been classified as modifying more than one head. A valid dependency tree, however, does not contain such tokens. To resolve this issue, the candidate head tokens are ranked according to the classification confidence of the base classifier that predicted them, and the highest-ranked candidate is selected.

### 2.2 Constraint satisfaction inference for dependency structures

Our second parsing algorithm casts dependency parsing as a weighted constraint satisfaction problem. Constraint satisfaction inference (CSI) [4] uses standard classifiers to predict weighted soft-constraints on the structure of the parse tree. Constraints that are predicted each cover a small part of the complete structure, and overlap between them ensures that global output structure is taken into account, even though the classifiers only make local predictions in isolation of each other.

The constraints are predicted by a classifier, where the weight for a constraint corresponds to the classifier's confidence estimate for the prediction. For the current study, we trained three classifiers to predict three different types of constraints.

1.  $C_{dep}$ , suggests a dependency arc for inclusion in the parse tree. For the example tree in Figure 1, among others the constraint  $C_{dep}(\text{head}=\text{was}, \text{modifier}=\text{No}, \text{relation}=\text{VMOD})$  should be predicted.
2.  $C_{dir}$ , the relative position of the head of a word. The tree in Figure 1 will give rise to constraints such as  $C_{dir}(\text{modifier}=\text{Black}, \text{direction}=\text{RIGHT})$ .
3.  $C_{mod}$ , suggests that a word is modified by a certain type of relation. The constraints generated for the word `was` in Figure 1 would be  $C_{mod}(\text{head}=\text{was}, \text{relations}=\text{SBJ})$ , and  $C_{mod}(\text{head}=\text{was}, \text{relations}=\text{VMOD})$ .

With the above, a weighted constraint satisfaction problem is formulated that describes a dependency tree. Any off-the-shelf W-CSP solver could be used to obtain the best dependency parse. However, as a more time-efficient alternative we chose to use the CKY algorithm for dependency parsing [6]. This choice restricts the output space of the parser to projective trees only.

Of the two parsing algorithms, the simple classification approach can be expected to be faster than the CSI-based parser, and can be expected to be leaner in memory usage due to the fact that it only assumes one classifier as compared to the three classifiers required by the CSI parser. On the other hand, the extra effort spent by the CSI parser is expected to pay off in superior parsing quality.

### 2.3 IGTREE: A fast approximation of $k$ -NN classification

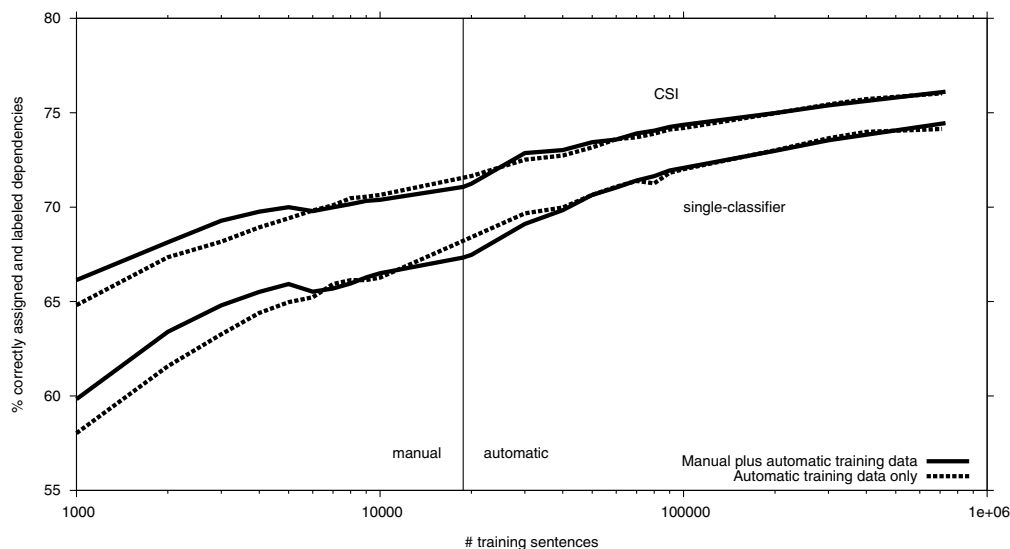
The classifier engine used in the above-mentioned three classifiers,  $C_{dep}$ ,  $C_{dir}$ , and  $C_{mod}$ , is IGTREE [5], an algorithm for the top-down induction of decision trees. IGTREE compresses a database of labeled examples (i.e. feature-value vectors with an assigned output class) into a lossless decision-tree structure that preserves the labeling information of all examples.

Classification in IGTREE occurs according to standard decision-tree classification; a new example (i.e. an unlabeled feature vector) is matched deterministically, top-down, against paths in the tree, until an end node is met, or no branch in the tree matches with the value at the particular feature tested at that node; the class label of that last visited node or end node is the classifier's prediction. As a normalized measure of confidence for the predictions made by IGTREE, needed in the inference step of the dependency parser, we divide the tree-node counts assigned to the majority class found at the last visited node, by the total counts assigned to all classes at that node.

## 3 Performance analyses

### 3.1 Generalization performance and coverage

As training material for the two memory-based dependency parsers we used all manually annotated data available in the Alpino Treebank [11], amounting to



**Fig. 2:** Learning curves in terms of the percentage of correctly labeled dependencies, trained on manual plus automatic data (solid line), or only on automatically parsed training data (dashed line), of both the simple classification-based parser (“single-classifier”, bottom two lines) and of the CSI-based parser (“CSI”, upper two lines).

18,791 sentences with 262,452 words. The treebank format is to a limited degree constituent-based, but contains all necessary information to convert each tree to a dependency structure. We did this using the CoNLL-X shared task [2] conversion software.

Subsequently, texts were collected that were automatically parsed by the Alpino parser [7]: ten thousand Dutch Wikipedia pages (about 179 thousand sentences, 2.2 million words), newspaper articles from the *Algemeen Dagblad* from the first half of 1994 (about 498 thousand sentences, 8.1 million words), and the unannotated parts of the *Eindhoven corpus* (33 thousand sentences, 551 thousand words), resulting in a corpus of approximately 710 thousand sentences and 10.8 million words. Instead of the part-of-speech tags furnished by Alpino, we re-tagged the corpus with the rich Spoken Dutch Corpus tagset, using a fast memory-based tagger [10].

Three variants of the two memory-based dependency parsers are trained. The first variant of both parsers is trained only on the manually annotated data; the second is trained exclusively on the automatically annotated data, while the third is trained on a concatenation of both training sets. Figure 2 displays learning curves in terms of correctly assigned and labeled dependencies, a commonly used evaluation metric [2], of the three variants, for both parsers. The x axes of the figure has a logarithmic scale and represents the number of training sentences. Two curves are plotted per parser rather than three, as the learning curve of the concatenated set continues at the point where the manual training set stops (i.e. at 18,791 sentences, indicated by the vertical bar).

The test set consists of 2,530 sentences (47,471 words) taken from the manually parsed section of the Eindhoven corpus (the *cdbl* part) that is held out from the training data; this is professionally written newspaper text with relatively long sentences, with many

subclauses and quotations.

For each parser, the two curves are remarkably similar; training a parser on automatically parsed training data leads to virtually the same accuracies as training on manually annotated data. Also, continuing training a parser on automatically parsed data does not cause the learning curve to regress.

As can be clearly observed from the learning curves, the CSI-based parser performs consistently better than the single-classifier parser, but with a diminishing gap as more training examples are available. At the maximum amount of training data, approximately 729 thousand sentences, the difference is about 1.6%.

The best scores of the two parsers trained exclusively on three variants of different kinds of training data, and tested on the aforementioned manually parsed test set, are displayed in Table 1. The table also includes accuracy scores on correctly assigned dependency relations regardless of the label (“unlabeled dependencies”), and on correctly assigned labels regardless of which word the word relates with (“label accuracy”). The parser trained exclusively on automatically parsed data is also tested along the same evaluation metrics on two different test sets that are part of the manually annotated training set, namely a set of 1,100 questions from the CLEF Dutch question-answering competition<sup>2</sup>, and a small test suite of 18 sentences used in a comparison of Dutch parsers in 2001. The parser trained on automatically parsed data performs at accuracy levels comparable to the scores on the first test set. From this it can be tentatively concluded that the parser indeed has a wide coverage.

### 3.2 Memory usage and speed

Table 2 summarizes the memory-usage measurements of the single-classifier parser and the CSI-based parser

<sup>2</sup> CLEF: <http://www.clef-campaign.org/>

Parser	Evaluation	Newspaper text			Questions	Test suite
		Manual	Automatic	Both	Automatic	Automatic
Single-classifier	Labeled dependencies	67.3	74.1	74.5	78.7	77.0
	Unlabeled dependencies	70.6	76.9	77.2	82.4	78.8
	Label accuracy	76.3	81.2	81.4	81.6	83.3
CSI-based	Labeled dependencies	71.1	76.0	76.1	80.6	79.9
	Unlabeled dependencies	75.2	79.3	79.4	84.5	82.5
	Label accuracy	77.2	81.2	81.2	82.7	83.6

**Table 1:** Best accuracies on test data of the single-classifier and CSI-based parser: the percentage of correctly assigned dependencies, with and without labeling, and the accuracy on labels only, tested on newspaper texts, a test set of questions, and a test suite of 18 hand-selected sentences.

Training set	Single-classifier	CSI-based
Manually annotated	3.5	8.9
Automatically parsed	33.7	87.6
Both	34.4	89.5

**Table 2:** Amount of memory used (Mb) by the single-classifier parser and the CSI-based parser with the two training set sizes and their combination.

when trained on maximum amounts of training data<sup>3</sup>. The footprint of the parser trained on manually annotated data is small (under 10 Mb), but this is at the cost of a lower performance. Trained on all available automatically-annotated and manually-annotated training data, the single-classifier parsers have a footprint of about 34 Mb, which can still be regarded reasonable in current computers. Their CSI-based counterparts have to claim memory for three classifiers rather than one, hence they have a larger memory need of about 89 Mb.

Typically, rule-based parsers become exponentially slower when parsing longer sentences. Alpino uses stochastic search to battle the problem, but the solution is only partial. To get an idea of the behavior of the memory-based approach with longer sentences, the speed and accuracy of both the single-classifier parser and the CSI-based parser was measured on different sentence lengths found in the first test set. Figure 3 shows both, measured on sentence lengths from 2 to 50. As the left graph of Figure 3 shows, shorter sentences are parsed more successfully, which is also typical for Alpino; the CSI-based parser furthermore outperforms the single-classifier consistently. The right graph of Figure 3 shows a perhaps more unexpected leveling of the speed of the single-classifier parser to about 1,500 words per second; sentences shorter than 20 words are processed faster. Earlier we noted that for each sentence, pairwise examples are generated ( $n(n-1)$ , to be exact), but we also constrained this (also with test sentences) to pairs of words within a range of twenty words from each other, as 99% of all relations in the training corpus occur within that range. This fixed constraint bounds the number of examples per sentence, making the relation between the sentence

length and the number of examples effectively linear.

The CSI-based parser is slower than the single-classifier parser for two reasons: first, it is based on three classifiers ( $C_{dep}$ ,  $C_{dir}$ , and  $C_{mod}$ ), rather than one. Second, the CSI-based parser performs a more complex inference step, the CKY algorithm, to arrive at a full dependency structure. Processing time of this algorithm is cubic in the length of the input. Beyond sentence length 10 the CKY procedure takes more time than the three (linearly processing) classifiers. Effectively, the speed appears to diminish at a linear rate, from about 700 words per second for very short sentences, via about 350 words per second at 20 words, the average length of sentences in the test set, to about 170 words per second at sentence length 40. Note that both parsers never fail to process a sentence.

## 4 Discussion

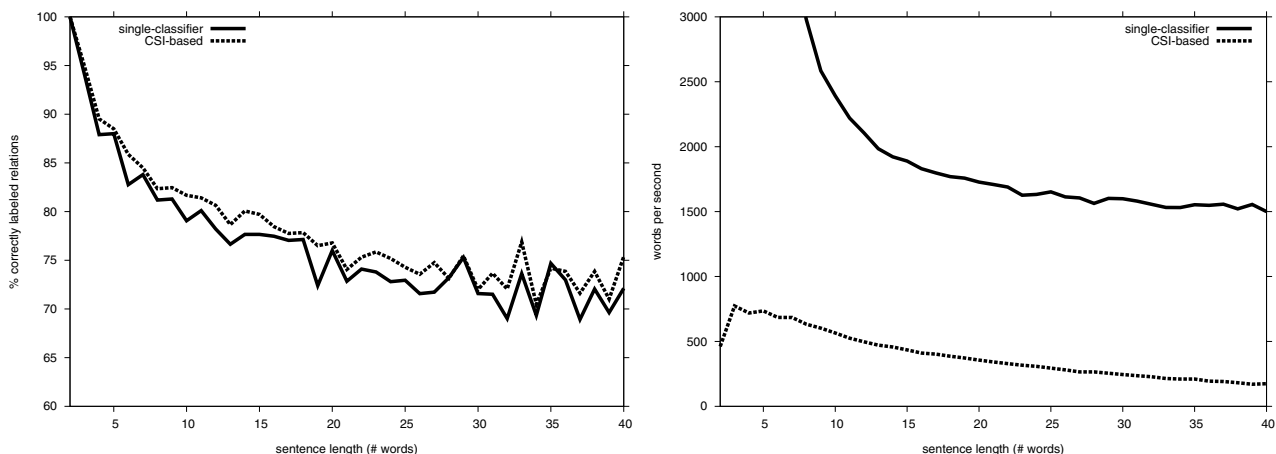
The experiments in this paper have shown that a manually written knowledge-based parser can to some extent be re-engineered as a memory-based parser, which performs similarity-based reasoning on examples of fragments of parses generated by the original parser. Recompile in memory can be quite fast when using IGTREE, a fast approximation of  $k$ -nearest neighbor classification, as the classifier engine. We developed a single-classifier parser operating in linear time, which processes sentences at speeds of at least 1,500 words per second. The second, more complex parser based on three classifiers and with a constraint-satisfaction inference step built in is slower; it only processes a few hundred words per second. The longer the sentence, the slower the CSI-based parser. Yet, there is no exponential increase in processing time with very long sentences. Furthermore, both parsers never fail to parse a sentence, and tests on three different test texts showed a robustly consistent level of performance.

A vexing question is whether we have actually built parsers that emulate, or may in the long run emulate, Alpino. It is clear that both parsers can never be pure emulations. Compared with Alpino, they may and most likely will produce different results on unseen text. For now it may be illustrative to compare evaluations on the same test texts.

Table 3 shows that Alpino still outperforms our best-performing parser by wide margins. On all three test sets the difference in performance score is over

<sup>3</sup> The hardware used for testing is equipped with Dual Core AMD Opteron 880 2,412 Mhz processors.





**Fig. 3:** Generalization accuracies in terms of percentages of correctly labeled dependencies (left) and words processed per second (right) of the single-classifier and CSI-based dependency parsers trained on the maximum amount of data, measured per sentence length from 2 to 50.

Test set	Alpino	CSI-based
Newspaper	86.8	76.1
Questions	93.7	80.6
Test suite	92.6	79.9

**Table 3:** Comparison of labeled dependencies score of Alpino and our best parser on the same test texts.

ten points. The classifiers used in our parsers only use extremely simple feature representations. To compensate for this, more training data is needed to match the performance of Alpino’s more sophisticated rule-based implementation. As we have shown in our experiments, the extra training data needed for narrowing this gap can be obtained from automatically parsed texts. However, the increase in performance with respect to the increase in data is only log-linear, and therefore large amounts of data will be needed to truly match the performance of Alpino.

On the positive side, our parsers do have an advantage in terms of memory and speed. Although the Alpino parser is quite memory-lean, it needs more memory with larger sentences. In contrast, our parsers have a static memory footprint (apart from an additional modest cubic-size buffer needed by the CSI-based parser). In terms of speed, Alpino can be exceptionally slow with long sentences due to its exponential components, and needs considerable search heuristics and even memory and time limits to keep within reasonable bounds; in contrast, our parsers appear to behave either linear (the single-classifier parser) or only mildly slower (the CSI-based parser) when processing longer sentences.

In future work we plan to optimize and improve the CSI-based parser. We also intend to continue training on more texts parsed by Alpino, as the end of that resource, and of the ensuing learning curve, is not in sight.

## Acknowledgments

We are indebted to Gertjan van Noord and his co-workers at the Rijksuniversiteit Groningen, The Netherlands, for their invaluable work on the Alpino parser and treebank. This work was supported by NWO, the Netherlands Organisation for Scientific Research, in the context of the NWO IMIX Programme and the NWO Vici project “Implicit linguistics”.

## References

- [1] M. Banko and E. Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 26–33. Association for Computational Linguistics, 2001.
- [2] S. Buchholz and E. Marsi. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL-X, the Tenth Conference on Computational Natural Language Learning*, New York, NY, 2006.
- [3] S. Canisius, T. Bogers, A. Van den Bosch, J. Geertzen, and E. Tjong Kim Sang. Dependency parsing by inference over high-recall dependency predictions. In *Proceedings of the Tenth Conference on Computational Natural Language Learning, CoNLL-X*, New York, NY, 2006.
- [4] S. Canisius and E. Tjong Kim Sang. A constraint satisfaction approach to dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1124–1128, Prague, Czech Republic, 2007.
- [5] W. Daelemans, A. Van den Bosch, and A. Weijters. iGtree: using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, 11:407–423, 1997.
- [6] J. Eisner. Bilexical grammars and their cubic-time parsing algorithms. *Advances in Probabilistic and Other Parsing Technologies*, pages 29–62, 2000.
- [7] R. Malouf and G. Van Noord. Wide coverage parsing with stochastic attribute value grammars. In *Proceedings of the IJCNLP-04 Workshop Beyond Shallow Analyses - Formalisms and statistical modeling for deep analyses*, 2004.
- [8] M. Osborne. Estimation of stochastic attribute-value grammars using an informative sample. In *Proceedings of the 18th International Conference on Computational Linguistics, COLING-2000*, 2000.
- [9] A. Van den Bosch and S. Buchholz. Shallow parsing on the basis of words only: A case study. In *Proceedings of the 40th Meeting of the Association for Computational Linguistics*, pages 433–440, 2002.
- [10] A. Van den Bosch, I. Schuurman, and V. Vandeghinste. Transferring PoS-tagging and lemmatization tools from spoken to written Dutch corpus development. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC-2006*, Trento, Italy, 2006.
- [11] L. Van der Beek, G. Bouma, R. Malouf, and G. Van Noord. The alpino dependency treebank. In *Selected Papers from the Twelfth Computational Linguistics in the Netherlands Meeting, CLIN-2001*, Amsterdam, 2001. Rodopi.

# Incorporating Syntax and Semantics in the Text Representation for Sentence Selection

Maria Fernanda Caropreso  
School of Information Technology and Engineering,  
University of Ottawa  
Ottawa, Ontario K1N 6N5 Canada  
[caropres@site.uottawa.ca](mailto:caropres@site.uottawa.ca)

Stan Matwin  
School of Information Technology and Engineering,  
University of Ottawa, Canada and  
Institute of Computer Science,  
Polish Academy of Sciences, Warsaw, Poland  
[stan@site.uottawa.ca](mailto:stan@site.uottawa.ca)

## Abstract

The basic Bag of Words representation generally used in Text Categorization loses important syntactic and semantic information contained in the documents. When the texts are of a short length this may be particularly problematic. In this paper we study the contribution of incorporating syntactic and semantic information into the representation in a Sentence Selection task in a genomics corpus. We analyze the use of a hierarchical technical dictionary created from the SwissProt Protein Knowledgebase. In our study, we either replace a gene or protein name by a generic term or add its SwissProt ancestor terms. Following our previous work, we introduce the hierarchical terms into a syntactic representation that uses relations between words in the sentences. We show that using hierarchical technical dictionaries together with syntactic relations is beneficial for our problem when using state of the art machine learning algorithms.

## Keywords

Machine Learning – Sentence Selection - Text Representation – Syntactic and Semantic Features

## 1. Introduction

Sentence selection (SSel) consists in identifying the relevant sentences for a particular purpose. This is a necessary step in many document-processing tasks, such as Text Summarization (TS) and Information Extraction (IE). The proportion of sentences considered relevant for the above tasks in a given document is usually low, making some pre-filtering a prerequisite.

Sentence selection can be considered a particular case of Automatic Text Categorization (ATC), which consists in automatically building programs capable of labeling natural language texts with categories from a predefined set. ATC is performed using standard Machine Learning methods in a supervised learning task. The standard text representation used in ATC is the Bag of Words (BOW), which consists of representing each document by the words that occur in it. This representation is also used in related tasks such as Information Retrieval (IR) and Information Extraction (IE). Different ways of expanding this representation have been tried on these areas of research, some of the expansions aiming to add some semantic or syntactic knowledge (see some related work in the next section).

Even though SSel and ATC are related, not all their characteristics are the same. One of the differences is that in SSel the sentences are short in length, with few words from the vocabulary occurring in each of them. This results in an even more sparse representation than in the ATC case. Another difference is that ATC is usually used to recognize the general topic of a document, while SSel concentrates on more specific details. Because of these differences, some variations to the standard representations and techniques usually used for ATC might be beneficial for SSel.

We address the task of sentence selection working on a corpus of texts on genetics. The sentences are short in length and the vocabulary of this corpus is highly specific. We believe that, because of these characteristics, the use of syntactic and semantic knowledge could be even more beneficial than in a collection of a more general nature.

Our work is devoted to identification of relevant sentences in scientific abstracts on genetics. Those abstracts are written in natural language and can be searched via the Internet using keyword queries. However, the queries would retrieve a large superset of relevant abstracts [9] from which we would like to identify the sentences that express an interaction between genes and/or proteins. Due to the continuous submission of new abstracts, this task becomes repetitive and time consuming. Because of that, automatic sentence selection is considered of interest to the scientific community. We automatically learn classifiers that categorize the sentences from the abstracts into two classes: those that describe an interaction between genes and/or proteins and those that do not. In those classifiers we study the usefulness of including syntactic and semantic knowledge in the text representation.

In the remainder of this paper we first introduce some related work and we present the details of our approach and our dataset. Afterwards we present the representations that we used and the experiments we performed together with their results and their analysis. We finish the paper presenting our conclusions and future work.

## 2. Related Work

The usefulness of syntactic and statistical phrases compared to the BOW was first studied by Fagan [4] in the IR context. In these experiments it was shown that

statistical phrases were not only easier to obtain but they also improved performance more than syntactic phrases.

In [7] Lewis compared different representations using either words or syntactic phrases (but not a combination of both) for IR and ATC. The results with the phrase representation showed no significant improvement with respect to the representation using the BOW. Mitra et al. [8] study the usefulness of linguistic knowledge for an IR system. The results indicate that the noun phrases are useful for lowly ranked answers but not so much for the highly ranked answers where the words alone perform well. Similar results were obtained in ATC by Furnkranz [5] when building syntactic phrases following some particular syntactic patterns learned from the data by an extraction system.

Caropreso et al. [2] studied the usefulness of statistical phrases (as opposed to single words) in ATC. The more discriminating phrases were added to the BOW. The experiments showed that the use of these phrases could in some cases improve the classification.

Cohen and Singer [3] studied the importance of introducing the order of the words in the text representation by defining position related predicates in an ILP system. This has been extended by Goadrich et al. [6] in recent research in the IE area, incorporating the order of noun phrases into the representation.

The use of hierarchies for the purpose of generalizing the vocabulary, and in particular the use of Wordnet in ATC, has been studied among others by Scot and Matwin [10]. They showed that word senses are not adequate to improve ATC accuracy.

Shatkay and Feldman [11] introduce various literature-mining methods, both in a general domain and within bioinformatics, including methods that make use of syntactic and semantic knowledge. They also present an information retrieval system and an information extraction system for finding specific information about genes.

### 3. Our Approach and Dataset

We study the usefulness of including syntactic and semantic knowledge in the text representation for the selection of sentences from technical genomic texts. In this specific context, the occurrence (or not) of specialized terms is expected to discriminate between sentences that contain information about genes and/or proteins interaction, and those that do not contain that information.

In our previous work [1], we showed that syntactic bi-grams (formed by words that are syntactically linked) provide extra information on whether two genes and/or proteins are interacting with each other. Such phrases were formed, for example, by an adjective modifying a noun, the main noun in the subject or object role of a sentence together with its verb, or the main noun in a prepositional phrase together with either the noun or verb it modifies. Using the syntactic bi-grams together with their single words, we represented the sentences and we evaluated the classification performance of this representation compared to the BOW. Our experiments included the machine learning algorithms Naïve Bayes

and Support Vector Machine (SVM), and they were performed using Weka [14].

It is understood in linguistics that syntactically related words express semantic concepts. By using syntactic bi-grams we are then already incorporating into the representation some basic semantics. In our previous work, we attempted to add some extra semantics with the help of technical dictionaries used to generalize the specific vocabulary by replacing the genes or protein names by the generic marker "geneprot". In our present work, we further enrich the representation by introducing more semantic knowledge in the form of a hierarchical dictionary to help with the specific vocabulary. This dictionary was created from a list of proteins and genes extracted from the SwissProt Protein Knowledgebase. From this source we obtained a classification of each of the proteins and genes, which became its ancestor in the hierarchy. All these ancestors were then related under a common root, the generic word "geneprot". We try different ways of introducing the information from this hierarchical dictionary into the text representation for our task, both in the representation that uses the syntactic information and in the basic representation without syntactic information. We again use the Naïve Bayes and SVM algorithms, and we compare the new results among themselves and with our previous ones.

Our experiments were done on a corpus created by the CADERIGE project. The examples that consist of only one sentence were automatically selected from MedLine abstracts with a query *Bacillus subtilis* transcription. The sentences were then pre-filtered to keep only those 932 that contain at least two names of either genes or proteins. The remaining sentences were manually categorized as positive or negative according to whether they describe or they do not describe a genomic interaction. The result was a balanced dataset with 470 positive and 462 negative examples. The vocabulary size is in the order of 3000 words. Some earlier work done on this corpus is presented in [9].

### 4. Representations

In this section we present the different ways we represented the sentences in order to capture the syntactic and semantic information. We start from the basic BOW representation and we then add the syntactic features as presented in the next sub-section. In these two representations we then study the inclusion of the semantic information provided in the hierarchical dictionary, trying different alternatives as presented in the second sub-section.

#### 4.1 Syntactic Representation

Given the characteristics of our task, we think that a richer representation that takes into consideration these characteristics would help to perform a better sentence selection. In particular, because the texts are so short, words are not disambiguated by the context. We believe that the syntactic information provided by a parser enriches the representation by showing the relations

among the words in the sentence, which, to certain extent, determine their senses.

We present here an example of the analysis performed by the Link Parser [13], the relations it recognized in our collection and how they are used in the text representation.

The Link Parser was selected for specifically providing the relation between words in the sentence by establishing a link between them. In order to create a syntactic representation we ran the parser on each sentence of the data collection, identified some syntactic links, such as the object of a verb, and we built syntactic bi-grams with the linked words. Out of the many links identified by the parser, we only took into consideration those links that we believe could help enrich our representation by bringing into the representation semantic relations relevant to the classification task (details on the links included can be found in [1]).

Given the first part of the fifth sentence of our collection:

"we isolated a temperature-sensitive sporulation defective mutant of the siga gene" (example 1)

the following are the links we identified among the set of links returned by the Link Parser: `isolated_we,` `mutant_isolated,` `mutant_temperaturesensitive,` `mutant_sporulation,` `mutant_defective,,` `gene_mutant,` `gene_siga.`

## 4.2 Hierarchical representation

While the syntactic representation goes some way towards producing a richer task representation, it lacks additional semantic knowledge. For this we turn to one of the several hierarchical knowledge bases available for our domain (eg. GeneOntology, Mesh, SwissProt.) In this way, our enriched representation on the one hand generalizes with respect to the BOW representation, and on the other hand enriches the representation semantically, which to some extent should alleviate the sparseness problem.

As previously presented (in section 3), the hierarchy we use was created from information contained in the Swiss Prot KnowledgeBase. It consists of a 3-level tree. The leaves are the gene or protein names and the root the generic term "geneprot". The intermediate level of the hierarchy is a classification of the gene or protein presented in the database.

For our experiments we generated different representations using this hierarchy, and compared their performance with the ones obtained when using the BOW, the basic representation using the gene or protein names, with and without the syntactic information, to which we refer on the results table as **names**. The new representations are:

a) the representations created by replacing the gene/protein names with the root of the hierarchy, referred as **repl\_root**,

b) the representation created by adding (instead of replacing) the root of the hierarchy for each gene/protein name, referred as **add\_root**,

c) the representation created by adding the first ancestors of the gene/protein name, referred as **add\_anc**,

d) the representation created by adding both the first ancestor of the gene/protein name and the root, referred as **add\_both**.

For the representations that use both syntactical and semantic information, new bi-grams were created to either replace or be added to the representation for each original bi-gram that contains a gene/protein name. For example, in the sentence presented before (example 1), we found the bi-gram "gene\_siga", therefore the new bi-grams "gene\_rna", and "gene\_geneprot" will be added to the representation that adds both the ancestor and the root of the hierarchy when considering the syntactic information.

## 5. Experiments and results

In this section we present the experiments we performed using the machine learning algorithm Naive Bayes (NB) from the Weka package.

As a baseline we use the basic representation, considering all the words that appear in any of the links, but without considering the links themselves and without using the technical dictionaries. We compare its performance with all our alternative representations, first considering only information on the hierarchical dictionary with its different variations in the representation, and then all the same variations including the recognized syntactic bi-grams (while keeping all the corresponding words.)

After learning and evaluating classifiers for the different representations, the results were compared using Accuracy, Precision, Recall and F1-measure. Given a contingency table for a two class problem, containing the real classification in the rows and the classifier's predictions in the columns, and in each entry the number of examples correctly or incorrectly predicted, as the following:

	Predicted Positives	Predicted Negatives
Real Positives	TP - True Positives	FN - False Negatives
Real Negatives	FP - False Positives	TP - True Negatives

the previous measures are defined as:

$$1 - \text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

$$2 - \text{Precision (Pr)} = TP / (TP + FP)$$

$$3 - \text{Recall (Re)} = TP / (TP + FN)$$

$$4 - F1 = 2 * Pr * Re / (Pr + Re)$$

According to preliminary experiments, the number of features was set in 1000, which resulted in the best F1 measure. The features were selected through filtering using the Information Gain measure. This was performed locally for each different fold or training set.

It is known that, over time, there is often a topic drift in the documents of a collection. In these cases, if N-fold cross-validation is used, it will not be sensitive to the effects of concept drift, because training and testing instances are spread over the entire time axis. It is therefore expected to obtain over optimistic results.

Using a time split is a protocol which more realistically evaluates the real use of the system when a classifier will be trained on instances available prior to the time of training and used to predict the class of new examples as they become available. We chose to use this approach to evaluate our representations, taking advantage of the knowledge that the sentences are ordered according to the date when the abstracts they belong to were submitted to PubMed.

Tables 1 shows the Accuracy, Precision, Recall and F1-measure obtained by the Naïve Bayes algorithm in a time related training/test split. The 60% of the sentences, which originally were part of the earlier abstracts, are used as training, while the remaining ones from the latest abstracts are used as testing.

We first observe that the use of the syntactic features importantly improved the accuracy of the classifiers (around 5% improvement), while only slightly improving the F1 measure in most cases.

The addition or replacement of the hierarchical information to the representation does not consistently affect the performance of the classifiers. However, the best accuracy and F1 measure, respectively 0.72 and 0.70, are obtained with the representation including the links and the top level of the hierarchical dictionary, the representation referred as `add_root` in the tables.

The correlation of the precision and recall measures is evident in Table 1. While the precision increases in the "Links" part of the table, the recall decreases in a similar proportion. This is why usually the F-measure, which is a weighted average of both Precision and Recall, is presented together with them. In our case we presented the F1 measure, which gives equal weight to both. In order to present another measure that relates the Precision and Recall values, we calculated the breakeven point, which is the value where precision and recall are equal, and it can be obtained by changing the classifier threshold. We did that with the NB algorithm and we found the BOW representation to have the lowest breakeven point at 0.64, while by adding the syntactic information, with or without the hierarchical information, it was 0.67, and the higher (and best) breakeven point of 0.69 was reached when adding only the hierarchical information.

To better understand the effects of the variations of the threshold in the Naïve Bayes algorithm, we show in Fig. 1 the different values of the F1 measure for a simplified 11 points threshold curve in the training/test split. We only show the representations on the first and last columns (`names` and `add_both`) of table 1, with and without the syntactic information, called here **Words**, **Links**, **Words\_Hier** and **Links\_Hier** respectively.

**Table 1.** Test set Acc., Pr., Re. and F1 for Naive Bayes

WORDS (BOW + hierarchical information)					
	names	r_root	a_root	a_anc	a_both
<b>Accuracy</b>	0.65	0.64	0.64	0.65	0.66
<b>Precision</b>	0.57	0.56	0.56	0.58	0.57
<b>Recall</b>	0.86	0.84	0.86	0.87	0.87
<b>F1</b>	0.68	0.67	0.67	0.69	0.69
LINKS (BOW + syntactic + hierarchical information)					
	names	r_root	a_root	a_anc	a_both
<b>Accuracy</b>	0.71	0.69	0.72	0.71	0.70
<b>Precision</b>	0.66	0.63	0.66	0.64	0.63
<b>Recall</b>	0.72	0.73	0.74	0.73	0.75
<b>F1</b>	0.69	0.68	0.70	0.68	0.69

We can observe that while the F1 measure is similar for the four representations at the 0.5 threshold (the default threshold used for the values presented in table 1), it presents considerable variations for other values of the threshold. In particular, for thresholds lower than 0.5 (which implies higher recall, and is shown on the left side of the graph in Fig. 1,) the two representations that make use of the syntactical information (**Links** and **Links\_Hier**) yield higher F1 measure values than the two representations that do not consider the syntactic information, and vice-versa.

In Fig. 2 we present the Precision/Recall curves for the same four representations as in figure 1. These curves confirm the observation that at high levels of recall, and in particular for recall values over 0.80, the representations that consider the syntactic information perform better than the ones that do not.

Fig. 2 also shows the differences in the breakeven point values mentioned before (observe the graph around the point 0.7 for both precision and recall). It is clear from the graph that the representation that takes into consideration the hierarchical semantic information (**Hier**) results in a higher precision not only at the breakeven point but on the whole interval of recall between the values of 0.4 and 0.8.

At levels of recall lower than 0.4, the representations containing the semantic information obtain lower precision than the ones that do not (the Words and Links on their own.)

## 6. Conclusions and Future Work

In this paper we have presented the problem of sentence selection from a genetics corpus and how we envisioned the contribution of semantic and syntactic knowledge in this task. We directly introduced semantic knowledge in the representation by either replacing or adding the ancestors of genes/proteins names according to a technical hierarchical dictionary. As introduced in section 3, syntactic relations were also incorporated in the representation, bringing additional semantic knowledge.

This was accomplished extending the set of features with bi-grams obtained from a syntactic parser.

Figure 1

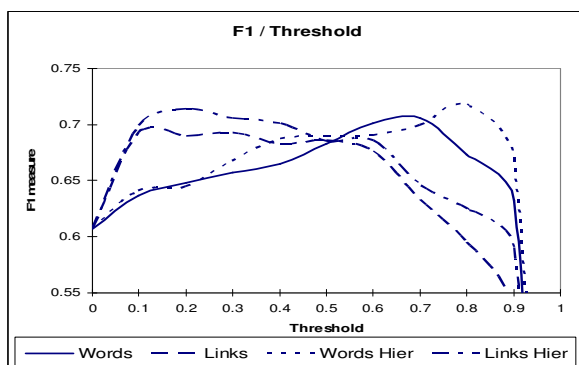
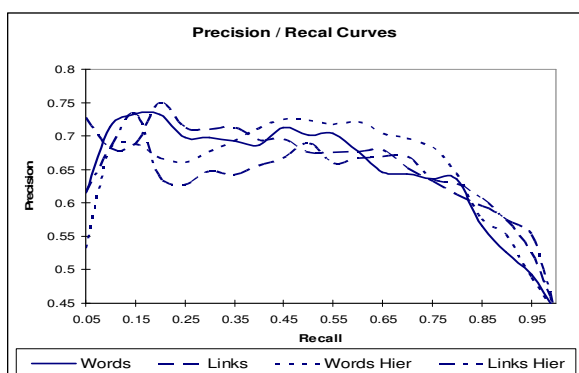


Figure 2



We have empirically showed that this syntactic and semantic knowledge is useful for sentence selection on this genetics corpus when using state of the art machine learning methods. Its use improved the classifiers' performance with respect to the basic BOW representation.

In a time based train/test split, we found the basic representation with only words to work well enough when relatively low values of recall are accepted. Adding the hierarchical semantic information brought the performance up for medium to high values of recall. When the highest values of recall are required, the representations that add the syntactic information to either of the previous ones perform the best.

In the future we plan to extend the use of semantic background knowledge to include other hierarchies of genes/proteins. One possible source for that could be the publicly available Mesh or Gene Ontology. We also plan to extend the use of syntactic knowledge by differentiating the links according to the kind of relation they denote (noun phrases, subject, etc.) and introducing morphological information (whether a word is a noun, an adjective, a verb, etc.) Finally, we plan to try this approach on a similar but larger dataset in the genetic abstracts context, as well as on a different domain on Legal documents.

## 7. Acknowledgements

The authors acknowledge the support of NSERC and the Ontario Centres of Excellence for this research.

## 8. References

- [1] Caropreso, M.F., Matwin, S. "Beyond the Bag of Words: a Text Representation for Sentence Selection" Proceedings of the 19th Canadian Conference on Artificial Intelligence. Québec City, Québec, Canada. June 2006.
- [2] Caropreso, M.F., Matwin, S. and Sebastiani, F. "A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization". In Amita G. Chin (ed.), Text Databases and Document Management: Theory and Practice, Idea Group Publishing, Hershey, US, 2001.
- [3] Cohen, W.W. and Singer, Y. (1999): Context-sensitive learning methods for text categorization in ACM Trans. Inf. Syst. 1999.
- [4] Fagan, J. L. Experiments in automatic phrase indexing for document retrieval: a comparison of syntactic and non-syntactic methods. PhD thesis, Department of Computer Science, Cornell University, Ithaca, US, 1987.
- [5] Furnkranz, J. A study using n-gram features for text categorization. Technical Report TR-98-30, Oesterreichisches Forschungsinstitut Artificial Intelligence, Wien, AT, 1998.
- [6] Goadrich, M., Oliphant, L. and Shavlik, J. Learning Ensembles of First-Order Clauses for Recall-Precision Curves: A Case Study in Biomedical Information Extraction. Proceedings of the Fourteenth International Conference on Inductive Logic Programming, Porto, Portugal. 2004.
- [7] Lewis D D, "Representation and Learning in Information Retrieval", Ph.D. dissertation, University of Massachusetts, 1992.
- [8] M. Mitra, C. Buckley, A. Singhal, and C. Cardie, "An Analysis of Statistical and Syntactic Phrases". 5TH RIAO Conference, Computer-Assisted Information Searching On the Internet, 200-214, 1997.
- [9] Ould, M., Caropreso, F., Manine, P., Nedellec, C., Matwin, S., "Sentence Categorization in Genomics Bibliography: a Naïve Bayes Approach", Informatique pour l'analyse du transcriptome, Paris, 2003.
- [10] Scott, S. and Matwin, S. Feature Engineering for Text Classification. Proceedings of ICML-99, 16th International Conference on Machine Learning, 1999.
- [11] Shatkay, H. and Feldman, R. Mining the Biomedical Literature in the Genomic Era: An Overview, Journal of Computational Biology (JCB), V.10, #6. 2003.
- [12] Siolas, G. Modèles probabilistes et noyaux pour l'extraction d'informations à partir de documents. Thèse de doctorat de l'Université Paris 6. July 2003.
- [13] Sleator, D. and Temperley, D. 1991. Parsing English with a Link Grammar. Carnegie Mellon University Computer Science technical report CMU-CS-91-196, October 1991
- [14] Witten, I. H. and Frank, E.. Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.
- [15] Zelikovitz, S. and Hirsh, H. Improving Text Classification with LSI Using Background Knowledge. Proceedings of CIKM-01, 10th ACM International Conference on Information and Knowledge Management. 2001.

# The BulTreeBank: Parsing and Conversion

Atanas Chanev  
University of Trento  
via Matteo del Ben 5, 38068 Rovereto, Italy &  
Fondazione Bruno Kessler-irst  
via Sommarive, 18, 38050 Povo-Trento, Italy  
*chanev@form.unitn.it*

Kiril Simov, Petya Osenova  
Linguistic Modelling Laboratory, IPP  
Bulgarian Academy of Sciences  
Acad. G. Bontchev st. 25A  
1113 Sofia, Bulgaria  
*{kivs,petya}@bultreebank.org*

Svetoslav Marinov  
School of Humanities and Informatics  
University College Skövde &  
Göteborg University, Graduate School of Language Technology  
Faculty of Arts, Box 200, Göteborg, Sweden  
*svetoslav.marinov@his.se*

## Abstract

Treebanks are often based on either of two grammatical formalisms: phrase structure (constituency) grammar or dependency grammar. However, sometimes it is necessary to transform treebank representations in order to test statistical parsers based on the alternative approach. In this paper we present new parsing results for Bulgarian by training two statistical parsers (constituency and dependency) on the BulTreeBank. We explore the interaction between constituency and dependency representations in both the constituency and the dependency parser using information based on the alternative formalism. We show that this interaction has a positive impact on parsing accuracy. We also investigate the relation between the BulTreeBank and one of its dependency variants which had been automatically derived from the original treebank.

## 1 Introduction

The practical utility of syntactic parsers in NLP has a high potential [18]. However, the state-of-the-art applications in key NLP areas often do not use parsing but rather implement N-gram language models. Besides being used in various tasks, parsing remains an interesting research question on its own (for example, both in 2006 and in 2007 the shared tasks of the Conference on Computational Natural Language Learning [6], [25] have been on dependency parsing<sup>1</sup>.)

Constituency parsing and dependency parsing are undoubtedly the two most common approaches to parsing natural languages. For a long time, constituency parsers such as [11] and [10] have been the state of the art for English. Furthermore, some constituency parsers have been ported from English to other languages such as Czech [12] or Chinese [1], among others. On the other hand, dependency parsers

have become increasingly popular, especially for languages with rather free word order [6].

Constituency and dependency-based parsers are similar in many ways. For example, they can be based on the same or similar parsing algorithms; statistical parsers can use the same techniques for learning etc. Another similarity between constituency and dependency parsers concerns the dependency parsing measures [19] which can also be used for evaluating constituency parsers (provided that head-dependent relations can be derived from the constituents in the treebank). Constituency parsers such as [11] use dependency information encoded in head-tables. However, dependency parsers often do not benefit from constituency information.

Without crossing dependencies, constituency grammar and dependency grammar are weakly equivalent [16, 15]. Dependency formalisms that allow crossing relations cannot be ‘transformed’ to constituency formalisms without using (some kind of) empty structures. Take, for example, the sentence fragment from the Penn treebank [21] “*The Soviet legislature approved a 1990 budget yesterday that halves its huge deficit...*” The dependency relation between ‘*approved*’ and ‘*yesterday*’, and the one between the heads of the phrase ‘*a 1990 budget*’ and the relative clause ‘*that halves its huge deficit*’ are crossing (Figure 1). In the original annotation of this sentence, there is an empty SBAR structure before ‘*yesterday*’ which points to the SBAR ‘*that halves its huge deficit*’.

This work addresses the practical aspect of the relation between the constituency and dependency formalisms. This relation might be interchangeability or complement. For a full scale interchangeability, a chosen formalism based on one of the approaches should be converted to a formalism based on the other approach and then converted back without any errors. Furthermore, both formalisms must be capable of representing in a sensible way the syntactic structure of the sentences from a large corpus. In the paper we present results on interchangeability between two representations of the Bulgarian treebank – BulTreeBank.

<sup>1</sup> <http://nextens.uvt.nl/~conll/>,  
<http://depparse.uvt.nl/depparse-wiki/SharedTaskWebsite/>

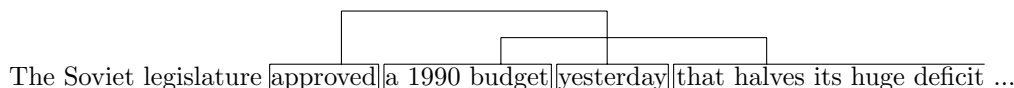


Fig. 1: Crossing relations in a sentence fragment from the Penn treebank [21]

The complement relation between the two formalisms is beyond the scope of the paper. It requires a joint model for simultaneously applications of both types of the linguistic knowledge.

In this paper we show updated results for parsing Bulgarian. The best settings for the dependency parser of [26] are used on another dependency conversion of the BulTreeBank [8]. Moreover, constituency information was included in a new parsing model which, employing gold standard phrase structure labels, outperformed the best dependency parser trained on the BulTreeBank [23]. Tests for constituency parsing of Bulgarian are also reported. Finally, we list the results of two conversion procedures from one of the dependency variants of the BulTreeBank back to constituency representations.

This paper is structured as follows: In Section 2 we briefly describe the BulTreeBank. Then, in Section 3, we review the measures that can be used to evaluate parsers. Our work on dependency parsing is described in Section 4. Section 5 is dedicated to our experiments with a statistical constituency parser. In Section 6 we report two conversion procedures for transforming a dependency variant of the BulTreeBank back to constituency representations. We conclude and list our plans for future work in Section 7.

## 2 The BulTreeBank

Currently the BulTreeBank [29, 28] comprises 214,000 tokens, a little more than 15,000 sentences. Each token is annotated with elaborate morphosyntactic information. The original XML format of the BulTreeBank is based on HPSG. Syntactic structure is encoded using a set of constituents with head-dependant markings. The phrasal constituents contain two types of information: the domain of the constituent (*NP*, *VP* etc.) and the type of the phrase (head-complement (*NPC*, *VPC* etc.), head-subject (*VPS*), head-adjunct (*NPA*, *VPA* etc.) and so forth.)

In almost every constituent the head daughter could be determined unambiguously. However, more specific rules are needed in some combinations of constituents. For example, in *NPs* of the type *NN*. The head might be the former or the latter noun depending on the semantics of the phrase. In such cases manual annotation of the head is necessary. Coordinations are considered to be non-headed phrases, where the grammatical function overrides the syntactic labels. We converted the BulTreeBank to Penn treebank bracketed format [21] for our tests on constituency parsing.

The BulTreeBank has been converted to dependency format using three different conversion procedures [8] (we will refer to the resulting treebanks using

the abbreviations BTBD-1, BTBD-2<sup>2</sup> and BTBD-3<sup>3</sup>, respectively). BTBD-1 is an extension of a previous conversion of part of the treebank that is described in [22]. BTBD-2 has been influenced by the annotation scheme of a dependency treebank of Italian - the Turin University Treebank [5]. BTBD-1 and BTBD-2 have been converted using a variant of the constituency-to-dependency conversion procedure described in [30].

The procedure used to convert the BulTreeBank to BTBD-3 is rule-based. It is based on an HPSG-compatible annotation scheme which has been designed according to the specific characteristics of the Bulgarian language. This is also the most popular dependency variant of the BulTreeBank. It has been parsed by 13 research teams at the CoNLL 2006 shared task on dependency parsing.

## 3 Parsing measures

Various measures have been used in the literature to evaluate parsers. One of these measures is the complete match (e.g. the number of correctly parsed trees divided by the total number of trees in the test set). However, this method cannot evaluate properly phrases (or dependency pairs) that have been parsed correctly but the trees that they belong to have been classified as incorrect. The PARSEVAL constituency measures (bracketing precision, bracketing recall and crossing brackets) [3] solve this problem for phrase structure grammar but they have been criticized for other demerits in [7, 17], among others.

The dependency parsing measures (labeled/unlabeled attachment score) proposed in [19] are an alternative. In this paper, we use the PARSEVAL F-measure (the harmonic mean of the PARSEVAL precision and recall) for evaluating constituency parsers. We also use labeled attachment score (LAS) for evaluating both constituency and dependency parsers. Note, that for constituency parsers, labeled attachment scores obtained using different head-tables are also different.

## 4 Dependency parsing

The dependency incarnations of the BulTreeBank have seen an increased attention recently as a valuable resource for training and testing statistical parsers. Marinov and Nivre [22] started off with a limited set of 5,000 sentences and reported labeled attachment score

<sup>2</sup> Software for converting the original BulTreeBank to BTBD-2 can be downloaded from: <http://depparse.uvt.nl/depparse-wiki/SoftwarePage/>

<sup>3</sup> More information on how to acquire BTBD-3 can be found on <http://www.bultreebank.org/dpbtb/>



of 72.9% on automatically assigned part-of-speech tags using MaltParser<sup>4</sup>, a shift-reduce dependency parser. The accuracy of the same dependency parser (i.e. [24]) trained on BTBD-1, BTBD-2 and BTBD-3 using gold standard part-of-speech tags was reported in [8]. Their best labeled attachment score was 79.5% achieved on BTBD-3. Both of these results were achieved with a memory-based learner [13] using part-of-speech tags, dependency and lexical information as learning features.

The BulTreeBank was used as an optional treebank at the CoNLL shared task in 2006 [6] and 13 different teams parsed it reporting results from 67.6% to 87.6% labeled attachment score. The best-performing parsers at the CoNLL 2006 shared task on dependency parsing – the two-staged parser of [23] and Maltparser of [26], clearly outperformed the parsing model reported in [8], achieving labeled attachment scores of 87.6% and 87.4%, respectively. Although [8] also have used Maltparser in their experiments, their version of the parser employed memory-based learning compared to Support Vector Machines (SVM) [9], used by [26]. SVM learning together with optimized feature models resulted in an increase of over 7 percentage points measured in labeled attachment score.

Our results on dependency parsing of the BulTreeBank are summarized in Table 1 together with other experiments reported in the literature. Firstly, we used the best feature model for BTBD-3 from [26] on BTBD-2, to test if feature model optimisation was generally robust regarding the chosen dependency annotations. Keeping all settings the same as in [26] but just changing the data we obtained 83.1% LAS. An improvement has been achieved, if this result is compared to the experiments of [8] on the same data set (79.2% LAS).

Trying to improve parsing, we decided to use constituency information as features in the learning model of the dependency parser, influenced by the constraint-based models in psycholinguistics such as [20]. Our idea was that if distinct types of information can bias human parsing decisions, then using such information for learning a statistical parser would increase its accuracy. We extracted the constituency information from the original treebank and added it as a separate layer in BTBD-2 and BTBD-3 using the following procedure:

```

if two or more new constituents have been
  opened before the token, associate the
  label of the constituent before the
  last to the token;
elsif one new constituent has been opened
  before the token, associate its label
  to the token;
else associate the default label (_) to
  the token

```

A constituent opened before  $word_i$  should be interpreted as a constituent which contains  $word_i$ , for  $i = 1$ , and, as additional condition for  $i > 1$ , does not contain  $word_{i-1}$  (where  $i$  is the position of the word in the sentence).

<sup>4</sup> <http://w3.msi.vxu.se/~nivre/research/MaltParser.html>

Pure Dependency Parsing			
Malt-MBL	Malt-SVM	Malt-SVM	best
BTBD-2	BTBD-2	BTBD-3	BTBD-3
79.2%	83.5%	87.4%	87.6%
Dependency + Constituency			
BTBD-2, Malt(SVM)		BTBD-3, Malt(SVM)	
90.6%		89.7%	

**Table 1:** Labeled attachment score of the parsers that we trained, compared to Malt-MBL [8], Malt-SVM on BTBD-3 [26] and the CoNLL 2006 best reported result for Bulgarian [23]

Take, for example the structure of the sentence from the BulTreeBank ‘Pravo na avtorstvo’ (*‘Right of authorship’*) which consists of an NPA:

*(S (NPA (N (Ncnsi Pravo))(PP (Prep (R na))(N (Ncnsi avtorstvo))))))*

The constituents which are opened before the first word are *S*, *NPA* and *N*. The constituent opened before the last is *NPA*, so it will be added as a label associated to the word ‘Pravo’. The constituent label associated with ‘na’ would be *PP*. As only one new constituent – *N* has been opened before the word ‘avtorstvo’, it would be associated to the last word in the sentence.

The addition of the constituency information in the parsing model has led to a labeled attachment score of 90.6% for BTBD-2 which is a significant increase of 3% compared to the best result from CoNLL 2006 shared task [23]. The same parsing model used on BTBD-3 has 89.7% labeled attachment score. These numbers are not comparable to the results reported at the CoNLL 2006 shared task and other parsing experiments, because we have used gold standard constituency information not only in the training set but also in the test set. Such information is not available in the typical parsing task. To overcome this demerit, we plan to use constituency information obtained using a constituency parser or a chunker for the test set instead of gold-standard constituents.

## 5 Constituency parsing

There have been a few studies on constituency parsing of Bulgarian (for an overview, the reader is referred to [8]). However, those parsers have only been partially evaluated. In this section we describe the first experiments on parsing the BulTreeBank using a statistical constituency parser. We then evaluate the results against gold-standard data. We used the multilingual statistical parsing engine of Dan Bikel [2]<sup>5</sup> which is an implementation and extension of Collins’ parser [11]. The parser has been set to parse several languages using treebank specific information in the form of a mapping table and a head-table. These tables can be easily replaced with other tables prepared for different treebanks/languages.

We have trained the parser for Bulgarian using a head table with default rules, i.e. the head child of a constituent is its leftmost child. This is our baseline

<sup>5</sup> <http://www.cis.upenn.edu/~dbikel/software.html>

Model	Parseval F	LAS
baseline	<b>80.4%</b>	75.8%
Head 1 POS map	77.7%	76.6%
Head 1	79.4%	<b>80.2%</b>
Head 2	78.1%	76.6%

**Table 2:** Results for constituency parsing of the *BulTreeBank*

parser	LAS
constituency	80.2%
dependency	79.2%

**Table 3:** Labeled attachment score (LAS) of the constituency parser (using the *BTBD-2* head-table for evaluation) compared to LAS of the dependency parser trained on *BTBD-2* by [8]

model. The initial results were encouraging as we obtained 80.4% PARSEVAL F-measure using the parser with default settings. For all the other models that we tested F-measure was 77.7% or higher but it never reached the accuracy of the baseline model.

We did several tests using the default settings of the parser but replacing the default head table (and the default mappings for one test). We used two distinct head-tables derived from the tables used in [8] to convert the *BulTreeBank* to *BTBD-2* and *BTBD-3*, respectively. Moreover, we manually mapped the part-of-speech tags of the *BulTreeBank* with those of the *Penn Treebank*.

One of the parsing models had a head table derived from *BTBD-2* and it did not use the mapping of the part-of-speech tags. This model gave the best LAS and the second best F-measure. Its F-measure is 1 percentage point lower than the F-measure of the baseline. On the other hand, the baseline model is the model with the lowest labeled attachment score and the highest F-measure. The results are shown in Table 2. The highest PARSEVAL F-measure and labeled attachment score are in bold.

The model with the highest LAS performed better than the dependency parser for Bulgarian described in [8]. A comparison between the accuracies of the two parsers can be found in Table 3.

## 6 Conversion

The relationship between constituency and dependency grammar has been addressed in [16, 15]. If crossing relations are excluded from a dependency grammar, it has the same power as a constituency grammar (i.e. both of them can weakly generate the same language). The relation between constituency and dependency grammar has also been explored in [14] with an attempt to combine them in a single formalism. Furthermore, some constituency treebanks (e.g. the second release of the *Penn treebank*) have a layer of grammatical relations. However, the annotation schemes of dependency treebanks have not usually been designed with the purpose of easy conversion to representations

based on the constituency approach. Non-projective relations cannot even be represented with constituents without using some kind of empty categories.

If constituency and dependency representations were interchangeable for a particular treebank, then the different kinds of parsers could be evaluated against it using the same measures. Furthermore, there would be a prerequisite to explore the advantages and demerits of the different parsing measures<sup>6</sup>. A weaker relation between particular constituency and dependency formalisms might be that of a complement. Then dependency information would be useful when included in phrase structure parsing models and constituency information would be useful when included in dependency parsers.

A conversion is needed when one wants to use a parser based on the alternative approach on a treebank. The conversion procedure can have two directions: from constituency to dependency or vice versa. A few conversion procedures have been described in the literature: from constituency to dependency (for an overview the reader is referred to [8]) and dependency to constituency, e.g. [12] as well as the inverted conversions of [30]. Since the *BulTreeBank* is HPSG-based, its phrase structures are similar to the representations in the *Penn treebank*. The *BulTreeBank* was converted from constituency to dependency in [8]. We aim to convert it back to constituency using a procedure derived from those described in [30] as well as using a rule-based approach.

### 6.1 From dependency to constituency

We have used two methods to convert *BTBD-3* back to constituency. One of them is based on a procedure described in [30]. It is treebank-neutral but it also needs treebank-specific resources in the form of three tables. The other conversion method is treebank-specific and is based on rules.

#### 6.1.1 Treebank-neutral method

The procedure described in [30] requires three tables: projection table, modification table and argument table. The projection table consists of projection rules. Each projection rule has a part-of-speech tag or a constituent on the left hand side and the constituent to which it is projected, on the right hand side. Only head-bearing categories can project to their parents. The projections must be unique, i.e. every part-of-speech tag or constituent can project to at most one constituent. Projections can be arranged in projection chains, i.e.

$$N_{cmsd} \rightarrow N \rightarrow NPA$$

This chain shows the *BulTreeBank* tag for a noun that is common, masculine, singular and definite which is projected to noun (*N*) and then to *NPA*.

The modification table lists the constituents which can modify every particular constituent on the left or

<sup>6</sup> For example, for two sets of parameters of a statistical parser *param1* and *param2*, the accuracy of the parser trained using *param1* can be greater than the accuracy of the parser trained using *param2*, if evaluated with one of the measures but less than it, if evaluated with the other measure.

on the right side. Some parts of speech (e.g. prepositions or verbs) can have up to a certain number of particular constituents as arguments. These relations are described in the argument table.

The only difference between a rule from the argument table and a rule from the modification table is that the former can specify the maximum number of arguments while the latter does not have a limit for the number of modifiers. In the BulTreeBank there is no need for constraints on the number of arguments of particular constituents. That is why we merged the modification table and the argument table.

The conversion algorithm is recursive. It begins from the root of the dependency graph, continues with its left children (from right to left) and then its right children (from left to right). It attaches one by one only complete subtrees built using the language-specific tables with a minimal number of projections. The full details of the algorithm can be found in [30].

While the conversion procedure described above is adequate for the Penn Treebank of English it is difficult to apply it to the BulTreeBank. Two kinds of factors can be distinguished for this difficulty: treebank-specific and language-specific. A treebank-specific factor concerns the trees of the BulTreeBank which are deeper, if compared to those of the Penn treebank. Combined together with language-specific factors such as pro-dropness and the relatively free word order of Bulgarian, this increased the number of rules of the modification table for the BulTreeBank, in comparison with the compact modification table for the Penn treebank of [30]. Furthermore, the one-projection-per-category projection table that we prepared for Bulgarian seemed inadequate for the many variants for projection of certain constituents in the BulTreeBank.

For example, in a common sentence in Bulgarian (with a subject and an object), the main verb would project to *V*, then to *VPC*, *VPS* and *S*. However, intransitive verbs do not take objects and if there are not other complements, *VPC* can be dropped from the projection chain as it is shown below.

*part of speech of the main verb*  $\rightarrow V \rightarrow VPS \rightarrow S$

Moreover, Bulgarian is a pro-drop language and a sentence can be without a subject. In such cases there is no need for the *VPS* constituent. If the sentence has an object (or another complement) but does not have a subject, then its projection chain according to the BulTreeBank annotation guide would be:

*part of speech of the main verb*  $\rightarrow V \rightarrow VPC \rightarrow S$

There are two other types of verb phrases that are used in the BulTreeBank – *VPA* and *VPF* (*VP* filler – takes an empty category as an argument) which complicate further the use of a projection table with unique projections.

These issues made the conversion of the BulTreeBank back to constituency using the method of [30] error-prone and unreliable. The converted treebank had only a subset of the constituents from the original treebank and the accuracy of the conversion (see Section 6.2) was significantly lower than the accuracy reported in [30] for the Penn Treebank.

### 6.1.2 Treebank-specific method

In addition to the treebank-neutral conversion we applied also a treebank-specific conversion which incorporated some minimal amount of linguistic knowledge about the annotation scheme of the treebank. This knowledge is the order of realization of the dependent constituents, which is: complements - subjects - adjuncts. Special rules were applied for coordination.

The first step was to construct the maximal constituent for each head in the sentence. This was done by bottom-up application of partial regular grammar which grouped together all the dependent elements of the same head and the head itself. For example, all modifiers of a nominal head were taken at once, or all the complements, the subject and the adjuncts were joined around the verbal head. The bottom-up application means that each dependent element of a head has to be a complete phrase. This means that if the dependent element is phrasal, the grammar constructs first the phrase and then adds it to the higher head and so forth to the complete coverage of the sentence.

For each of the constituents that need additional analysis, i.e. for a constituent

*(Adjunct Adjunct Subject Head Obj Indobj)*

we have to add the following structure

*(Adjunct (Adjunct (Subject (Head Obj Indobj))))*.

This task was performed by regular grammars, where for each type of dependent element there was one such grammar. The grammars were run again in bottom-up mode, but this time they were ordered according to the realization of the dependent elements as it was mentioned above.

The last step was to label the constituents. For this we constructed a set of rules. The rules determine the label of a given constituent on the basis of the head daughter and the dependent element. For example, if the head daughter is verbal and a non-head daughter is *Object*, then the constituent is annotated with the *VPC* element.

## 6.2 Measures and evaluation

To evaluate our conversions we used one of the measures for parsing, because, to our knowledge, no plausible measures for conversion have been proposed in the literature. For evaluating a single transformation only, a gold standard must be prepared manually. However, to evaluate both a transformation to another representation and a transformation back to the original treebank representation, one might simply evaluate the resulting treebank on the original treebank. Xia [30] reported 88% F-measure for converting the Penn Treebank to dependency representations, and then back to constituency representations.

We chose to use the PARSEVAL F-measure for evaluating the conversion from the original BulTreeBank to BTBD-3 of [8], on the one hand, and our own conversions in the other direction, on the other.

The accuracy of the conversion procedures is given in Table 4. In the case of the treebank-neutral conversion, 65% F-measure has been achieved for all the constituents. If evaluating only on the subset of constituents which the conversion procedure had been able to recognize, the number rose to 69.4%. The

Procedure/Constituents	Full set	Subset
Trebank-neutral	65%	69.4%
Trebank-specific	80.9%	-

**Table 4:** PARSEVAL F-measure for the conversion: *BulTreeBank*  $\rightarrow$  *BTBD-3*  $\rightarrow$  *BulTreeBank*

treebank-specific method has 80.9% F-measure. The reason for the better performance of the method is that its rules can assign constituents more reliably. For comparison, the conversion of the Penn Treebank in [30] with the method on which we based our treebank-neutral method has 88% F-measure.

The two tagsets for the constituency and the dependency variants of the treebank are different in their granularity. The constituency treebank is annotated with more than 60 syntactic tags, whereas the dependency treebank is annotated with only 18 tags. Thus, we could expect some information to be represented only implicitly in the dependency treebank. This is another reason why more treebank specific information used in the conversion has contributed to a better result.

The results from this study show that the constituency and dependency formalisms in the case of the *BulTreeBank* and *BTBD-3* (one of its dependency conversions) are not interchangeable. However, our results on dependency and constituency parsing support a weaker claim, namely that the constituency and dependency formalisms complement to one another. We may also assume that rule-based constituency parsing could benefit from the availability of dependency information. This is another way to view our treebank-specific conversion procedure.

PARSEVAL measures have not been designed for evaluating conversions. They have even been problematic when used to evaluate parsers. In future we envisage to define a measure for parsing (which may also be eligible for conversions) on the basis of well-formed fragments where well-formed fragments are defined as in DOP ([4], [27]). Comparing sets of fragments will minimize the impact of the errors high in the tree which are the source of the main criticism to PARSEVAL. Such a method will also help to localize the problematic cases.

## 7 Conclusion and future work

In this paper we described the parsing of the original *BulTreeBank* as well as two of its dependency conversions. We repeated the experiment of parsing *BTBD-3* with the parser and the best settings of [26] using *BTBD-2* instead, achieving better results than those described in [8]. We did experiments with a constituency parser which we evaluated against the *BulTreeBank* using constituency, as well as dependency measures.

The use of constituency information helped to increase the accuracy of the dependency parser. Together with the standard use of dependency information in the form of head-tables in constituency parsers with positive impact, our results only hint

that the future state-of-the-art parsers would probably use both constituency and dependency information to build syntactic structures (being constituency trees or dependency graphs). Regarding the conversion procedures, the treebank-specific method gave better results in the evaluation against the original treebank.

Further work is needed to find two annotation schemes (constituency and dependency) which can code in a plausible way the syntactic structures of the sentences in a large, balanced corpus. In the same time, it should be feasible to convert them to one another without any errors. Parsing accuracy can be improved further by parameter optimisation. Another way to make parsers more accurate is to use various kinds of automatically annotated linguistic information in their learning models. In addition, more plausible measures for evaluating parsers and conversions have to be designed and used in addition to the PARSEVAL and dependency measures.

## Acknowledgements

We would like to thank Alberto Lavelli for his valuable comments on a previous draft of this paper. The work reported here is partly supported by the project BIS-21++.

## References

- [1] D. Bikel. Design of a multi-lingual, parallel-processing statistical parsing engine. In *proceedings of HLT-2002*, 2002.
- [2] D. Bikel. Intricacies of Collins parsing model. *Computational Linguistics*, 30(4):497–511, 2004.
- [3] E. Black, S. P. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. L. Klavans, M. Liberman, M. P. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of the DARPA Speech and Natural Language Workshop*, 1991.
- [4] R. Bod. *Beyond Grammar: An experience-based theory of language*. CSLI Publications, California, USA, 1998.
- [5] C. Bosco. *A grammatical relation system for treebank annotation*. PhD thesis, University of Turin, 2004.
- [6] S. Buchholz and E. Marsi. CoNLL-X shared task on multilingual dependency parsing. In *Proc. of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, New York, 2006.
- [7] J. Carroll, G. Minnen, and T. Briscoe. Parser evaluation using a grammatical relation annotation scheme. In *A. Abeillé (ed.), Treebanks. Building and Using Parsed Corpora*. Dordrecht: Kluwer, 2003.
- [8] A. Chanev, K. Simov, P. Osenova, and S. Marinov. Dependency conversion and parsing of the *BulTreeBank*. In *Proc. of the LREC-Workshop Merging and Layering Linguistic Information*, 2006.
- [9] C.-C. Chang and C.-J. Lin. LIBSVM: A library for Support Vector Machines. URL: <http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf>, 2005.
- [10] E. Charniak. A maximum-entropy-inspired parser. In *Proc. of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Seattle, 2000.
- [11] M. Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.
- [12] M. Collins, J. Hajič, L. Ramshaw, and C. Tillmann. A statistical parser for Czech. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, College Park, 1999.

- [13] W. Daelemans and A. V. den Bosch. *Memory-based Language Processing*. Cambridge University Press, 2005.
- [14] M. Dras, D. Chiang, and W. Schuler. On relations of constituency and dependency grammars. *Journal of Language and Computation*, 2(2):281–305, 2004.
- [15] H. Gaifman. Dependency systems and phrase-structure systems. *Information and Control*, 8:304–337, 1965.
- [16] D. G. Hays. Dependency theory: A formalism and some observations. *Language*, 40:511–525, 1964.
- [17] S. Kübler and E. Hinrichs. From chunks to function-argument structure: A similarity-based approach. In *Proceedings of ACL-EACL 2001*, 2001.
- [18] M. Lease, E. Charniak, M. Johnson, and D. McClosky. A look at parsing and its applications. In *Proceedings of AAAI 2006*, 2006.
- [19] D. Lin. A dependency-based method for evaluating broad-coverage parsers. *Natural Language Engineering*, 4 (2):97–114, 1998.
- [20] M. C. MacDonald, N. J. Pearlmutter, and M. S. Seidenberg. Lexical nature of syntactic ambiguity resolution. *Psychological Review*, 101(4), 1994.
- [21] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19 (2), 1993.
- [22] S. Marinov and J. Nivre. A data-driven dependency parser for Bulgarian. In *Proceedings of TLT4*, pages 89–100, 2005.
- [23] R. McDonald, K. Lerman, and F. Pereira. Multilingual dependency analysis with a two-stage discriminative parser. In *CoNLL-X Shared Task on Multilingual Dependency Parsing*, 2006.
- [24] J. Nivre. *Inductive Dependency Parsing*. Springer, 2006.
- [25] J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, 2007.
- [26] J. Nivre, J. Hall, J. Nilsson, G. Eryigit, and S. Marinov. Labeled pseudo-projective dependency parsing with support vector machines. In *CoNLL-X Shared Task on Multilingual Dependency Parsing*, pages 221–225, 2006.
- [27] K. Simov. Grammar extraction and refinement from an HPSG corpus. In *Proc. of the ESSLLI Workshop on Machine Learning Approaches in Computational Linguistics*, pages 38–55, Trento, Italy, 2002.
- [28] K. Simov and P. Osenova. Practical annotation scheme for an HPSG treebank of Bulgarian. In *LINC2003*, 2003.
- [29] K. Simov, P. Osenova, A. Simov, and M. Kouylekov. Design and implementation of the Bulgarian HPSG-based treebank. *Journal of Research on Language and Computation – Special Issue*, pages 495–522, 2005.
- [30] F. Xia. *Automatic Grammar Generation from Two Different Perspectives*. PhD thesis, University of Pennsylvania, 2001.

# Confidence Measures and Thresholding in Coreference Resolution

John Chen, Laurie Crist, Len Enyon, Cassandre Creswell, Amit Mhatre, Rohini Srihari  
Janya, Inc.

1408 Sweet Home Road, Suite 1

Amherst, NY 14228

*{jchen,lcrist,leynon,ccreswell,amhatre,rohini}@janyainc.com*

## Abstract

Coreference resolution is an important component of information extraction systems. Machine learning methods have been found to perform quite well for this task, leading to research in a variety of such methods. In our current work, we explore the approach of increasing the performance of an existing pairwise coreference system by using a confidence measure in order to filter out low-scoring classifications. We explore several ways to define a confidence measure. Subsequently, we use a confidence measure in conjunction with thresholding. We find that a multiple threshold system, with the thresholds defined the right way, outperforms both the baseline and a single threshold system. We also discover that basing a threshold as close as possible to the evaluation metric is a good idea, and explore reasons why this might be so.

## Keywords

coreference resolution, machine learning, confidence estimation

## 1 Introduction

One important goal of information extraction (IE) is to extract information about entities of interest from a set of text documents automatically. Coreference resolution is important for IE, because it concerns the problem of determining which noun phrase mentions (NPs) in a document refer to the same real-world entity. Increasing the accuracy of coreference resolution in order to improve IE is the focus of this paper.

A common paradigm for coreference resolution casts it as a classification task that is solved through machine learning [13, 10, 11]. Specifically, the task is to determine whether or not two NPs corefer, where they both come from the same document. In order to decide, a machine learning algorithm uses features based on the NPs themselves, their immediate context, and their relative positions in the document. Subsequently, a clustering algorithm uses this information to group NPs into clusters, each cluster representing a particular entity.

Although such systems can perform quite well when evaluated on standard data sets, there is still room for improvement. Instead of taking the approach of trying different algorithms such as [8, 17], we take the

approach of assigning a basic confidence measure to each output of the machine learning classifier. We then use the measures in order to filter out low-confidence classifications and measure its effects on coreference and the rest of the system. The goal is to achieve a high-accuracy IE system.

The use of confidence measures to grade the output of a system has been well studied in the field of speech recognition [16]. In natural language processing, there has been recent interest in applying confidence measures to the output of machine translation systems [12, 15]. In this field, however, the use of confidence measures is usually embedded in some other application, such as guiding the search of a decoder [5, 3] or determining the quality of an automatically labeled example for bootstrapping [14]. To our knowledge, it is only in the latter sense that confidence measures have been applied to coreference resolution [7].

There are a number of ways that a confidence measure can be defined over our coreference resolution system. Once a confidence measure is defined, there are subsequently a number of ways in which the confidence measure may be employed in order to achieve our goal of a high-accuracy IE system. In our current work, we evaluate several ways to define a confidence measure. Subsequently, we use a confidence measure in order to filter out low-confidence classifications. We have experimented with several ways to do the filtering, and examined its impact on our IE system.

The outline of this paper is as follows. First, we describe the system on which we perform our experiments. Second, we provide background information about the experiments that we perform such as the corpora being used and details about performance measures. Third, we describe a series of experiments. Initial experiments evaluate different confidence measures while other experiments examine different ways to use a confidence measure to filter out low-scoring classifications. Finally, we present our conclusions and future work.

## 2 Baseline System

The text of the input document passes through a sequence of modules before our coreference system receives it. First, the text is tokenized and stemmed. Second, parts of speech are assigned similar to [2]. Third, WordNet [9] features as well as other lexical fea-

tures from gazetteers are assigned. Fourth, a named entity tagger along the lines of [4] tags the text. Fifth, a shallow parser chunks the text. Our coreference system receives the text in this form.

Following [13, 10], our coreference system consists of a classifier that determines whether two input NPs corefer, and a clusterer that relies on the classifier in order to group NPs into clusters representing entities. We divide NPs into three types: named entities (NEs), nominals, and pronominals. Our classifier is divided into two parts. The first part determines whether two NEs are coreferent; this is the *aliasing module*. Basically, it matches two NEs if they have been tagged as the same NE type (e.g. person, organization, location) by the NE tagger and if their strings match. The second part determines whether two NPs are coreferent when at least one of them is a pronominal or nominal; this is the *statistical coreference module*.

Our statistical coreference module uses the features in Figure 1 in order to decide whether two NPs are coreferent. Out of the two input NPs, the one that is found first in the document is called the *antecedent* and the other is called the *anaphor*. These features are similar to the ones that are found in [13, 10]. One difference from these other systems is that instead of a single model, we use a pair of models. One model is used when the anaphor is a pronominal, while the other model is used when the anaphor is a nominal. Both models use the same features, except that the nominal model lacks the feature of antecedent grammatical role.

Each of the pronominal model and the nominal model is a maximum entropy (ME) model as in [1]. Like [1], parameter estimation is performed using improved iterative scaling. Parameters are iteratively updated until none of the parameters' values change by  $1.0 \times 10^{-5}$  or 1500 iterations have elapsed.

A ME model defines a probability distribution  $P_{ME}(y|x)$  where  $y$  is a random variable that is TRUE if the input NPs are coreferent and  $x$  represents an instance of two input NPs and their context. Features in a ME model are defined as a set of indicator functions  $\{f_1, \dots, f_n\}$  over the domain  $(x, y)$ . An example of a feature is

$$f_{1043}(x, y) = \begin{cases} 1 & \text{if } y \text{ is TRUE and the genders} \\ & \text{of NPs in } x \text{ match.} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The probability is computed using:

$$P_{ME}(y|x) = Z(x) \prod_{i=1}^n \alpha_i^{f_i(x,y)} \quad (2)$$

where the  $\alpha_i$ 's are the model parameters and  $Z(x)$  is a normalization constant.

Given the results of the aliasing module and the statistical coreference module, clustering is performed. Clustering is performed in a greedy manner in that if either module proposes a link between two NPs, those NPs are placed in the same cluster.

## 3 Experimental Preliminaries

### 3.1 Corpora

We use the ACE 2004 training documents for both training and testing. Out of all of these documents, 70% have been selected to be our training corpus, about 10,700 words. The remainder comprise the test corpus, about 4,300 words. In all our experiments with ACE 2004 test data, we use the ground-truth mentions and value for entity type and subtype features. In a run-time production system, the input to the coreference module is the set of mentions detected by earlier modules along with features assigned automatically. As such, somewhat lower results are expected than in the experimental system where hand-annotated mentions and type features are used. The advantage of using hand-annotations as input is that it allows us to test the performance of coreference capabilities in isolation, independent of the result of the information extraction system.

In addition, we have an auxiliary test set in order to further validate the systems that we develop. The auxiliary test set consists of news articles from the content provider LexisNexis. It consists of about 300 articles.

### 3.2 Performance Measures

There are two ways that scoring of the output of the coreference system is performed. One way is *link accuracy* which is an accuracy measure that we define over all of the coreference links proposed by the system between NP pairs. Link precision is the percentage of all links that are suggested by the system that are actually links. Link recall is the percentage of actual links captured by the system out of all of the actual links captured by the version of the system with the highest link recall. Link F measure is the weighted harmonic mean of link precision and link recall. Another way to score output is *entity-constrained mentions (ECM) accuracy* [8]. This is a metric of the percentage of mentions that are in the *right* entity. It depends on the ACE scoring script to assign a mapping from key entities to output entities. Based on this mapping, a mention that is missing from an entity is a false negative; a mention that is present in the output but not in the key is a false positive. Any mentions in unmapped entities are false positives also. From this, the precision, recall, and F-measure of mentions in entities can be computed.

## 4 Experiments on Different Confidence Measures

A confidence measure  $\phi$  is used to evaluate a decision  $y$  made about the coreferentiality of  $x$ , an input pair of NPs. There are various ways in which it can be defined. Because our system clusters entity mentions in a greedy manner, we are interested in a confidence measure that may ameliorate the problems with this approach. In particular, we define it to be  $\phi(x, y) = P(y|C, x)$  where  $C$  represents the cluster in which  $y$

**Fig. 1:** Feature set that is used for statistical coreference. All of the features are used in the nominal model. All of the features except “antecedent grammatical role” are used in the pronominal model.

Feature Name	Feature Values
Anaphor mention type	<i>NE</i> if named entity; <i>Anaphor</i> if def. or dem. NP; <i>OtherPro</i> if relative pronoun; <i>Pronoun</i> if any other pronoun; <i>Nom</i> if any other kind of NP.
Antecedent mention type	Feature values are the same as Anaphor mention type.
Antecedent grammatical role	Grammatical role of antecedent, such as <i>subj</i> for subject, <i>obj</i> for object, etc.
Number match	<i>Yes</i> if anaphor and antecedent’s number matches; <i>No</i> if they do not; <i>Unk</i> if it cannot be determined.
Gender match	<i>Match</i> if anaphor and antecedent’s gender matches; <i>Mismatch</i> if they do not; <i>Unk</i> if they possibly mismatch; <i>Compat</i> if they possibly can match.
String match	<i>Match</i> if anaphor and antecedent’s surface strings entirely match; <i>PartMatch</i> if there is a substring match; <i>NotMatch</i> if there is entirely no match.
Entity type match	<i>Y</i> if anaphor and antecedent’s types (e.g. Person, Organization, etc.) match; <i>N</i> if they do not; <i>Unk</i> if it cannot be determined.
Entity subtype match	<i>Y</i> if anaphor and antecedent’s subtypes match; <i>N</i> if they do not; <i>Unk</i> if it cannot be determined.
Entity distance	Number of entities intervening between anaphor and antecedent.

belongs. We expand  $\phi(x, y)$  as follows using the chain rule:

$$\begin{aligned} \phi(x, y) &= P(y|C, x) \\ &= \frac{P(C|x, y)P(y|x)}{P(C|x)} \end{aligned} \quad (3)$$

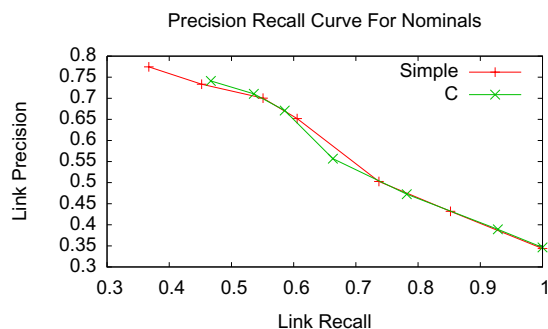
Depending on how we calculate the terms of Equation 3, we can define different confidence measures. In these experiments,  $P(y|x)$  is always computed using the maximum entropy model (Equation 2). Now, if we define  $\frac{P(C|x, y)}{P(C|x)} = 1$ , then we have a simple confidence measure  $\phi_{\text{simple}}(x, y)$ . We can define a more complex measure as follows. Let  $C$  be a function from an entity to its entity type, e.g. Person, Organization, etc. Let the input pair of NPs,  $x$ , be represented by their entity type. Given these stipulations, we can compute  $P(C|x, y)$  and  $P(C|x)$  during training using maximum likelihood estimation. We can then determine their values during testing by first finding NP clusters in the usual greedy manner and then using them to assign the new, more complex confidence measure  $\phi_C(x, y)$  using Equation 3.

We evaluate  $\phi_{\text{simple}}$  and  $\phi_C$  in terms of link accuracy. In particular, we use each confidence measure in conjunction with a threshold value  $\theta$  in order to control which pairs of NPs out of the ones that the statistical models decide are coreferent are actually treated by the system as coreferent. By varying  $\theta$ , the user is able to specify a precision versus recall tradeoff for link accuracy.

Evaluating them in terms of link F measure for various threshold levels  $\theta$  shows that no confidence measure performs consistently better. See Table 1. Similarly, if we evaluate them in terms of accuracy on nominal coreference, there is also no big difference. See Figure 2. On the other hand, evaluating them in terms of accuracy on pronominal coreference, we see that most of the time a system that uses  $\phi_C$  outperforms a system that uses  $\phi_{\text{simple}}$ .

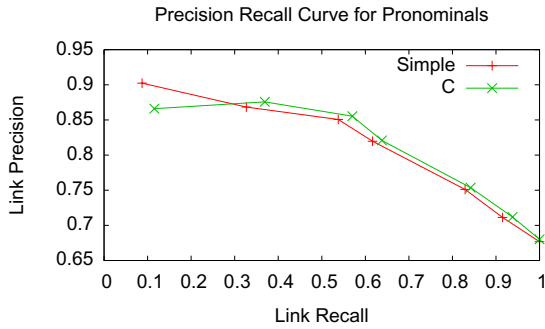
**Table 1:** Comparing at various threshold values  $\theta$  using Link F Measure, there is no clear difference between confidence measures  $\phi_{\text{simple}}$  and  $\phi_C$ .

$\theta$	$\phi_{\text{simple}}$	$\phi_C$
0.50	0.6876	0.6904
0.60	0.7189	0.7087
0.70	0.7243	0.7229
0.80	0.6795	0.6789
0.85	0.6454	0.6651
0.90	0.5031	0.5511
0.95	0.2834	0.3450



**Fig. 2:** Confidence measures  $\phi_{\text{simple}}$  and  $\phi_C$  perform comparably over nominals.





**Fig. 3:** Confidence measure  $\phi_C$  performs better than  $\phi_{simple}$  overall over pronominals.

**Table 2:** Increasing  $\theta$  shows the recall precision tradeoff in terms of link accuracy over all links. (TP=Number of True Positives; FP=Number of False Positives).

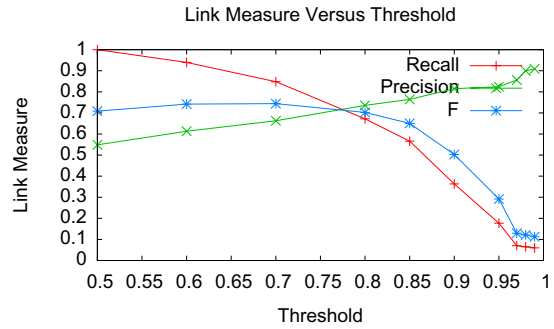
$\theta$	TP	FP	Link Recall	Link Precision	Link F
0.50	2722	3310	1.0000	0.5487	0.7086
0.60	1963	3110	0.9396	0.6130	0.7420
0.70	1429	2807	0.8480	0.6627	0.7440
0.80	798	2224	0.6719	0.7359	0.7025
0.85	578	1873	0.5659	0.7642	0.6502
0.90	273	1203	0.3634	0.8150	0.5027
0.95	125	586	0.1770	0.8242	0.2915
0.97	39	232	0.0701	0.8561	0.1296
0.98	24	215	0.0650	0.8996	0.1212
0.99	20	199	0.0601	0.9087	0.1128

## 5 Use of a Single Threshold

Given the mixed results in Section 4, in this section we will experiment with a confidence measure  $\phi(x, y)$  such that  $\phi(x, y) = \phi_{simple}$ . As before, we use confidence measure  $\phi$  in conjunction with a threshold value  $\theta$  in order to control which pairs of NPs the system decides are coreferent. In this section, instead of evaluating on NP links only, we determine whether this thresholding can be used to improve the accuracy of entity detection.

We calculate the link accuracy of coreference link creation for a range of values for  $\theta$ . We do this on training data that has undergone 20-fold cross validation. The results are shown in Table 2 and Figure 4. There is an expected tradeoff between recall and precision as  $\theta$  is increased. The optimal  $\theta$  is 0.70, leading to a link F measure of 0.7440.

Although we have seen that the optimal link F measure occurs when  $\theta = 0.70$ , we would like to confirm that this leads to the best entity detection. We would hope that it would lead to an improvement over the baseline system,  $\theta = 0.50$ , which is the threshold value that is used when coreference links are adopted by the system when  $P_{ME}(y = \text{TRUE}|x) > P_{ME}(y = \text{FALSE}|x)$ . In order to do so, we evaluated the output of the system on the test set using ECM accuracy for different values of  $\theta$ . The results are shown in Table 3.



**Fig. 4:** Optimal value for  $\theta$  over all links is 0.70.

**Table 3:** Optimal value of  $\theta = 0.70$  over all links leads to higher ECM accuracy.

$\theta$	ECM		
	Recall	Precision	F
0.50	0.619	0.619	0.619
0.70	0.627	0.627	0.627
0.90	0.576	0.576	0.576

The optimal setting  $\theta = 0.70$  does lead to better entity detection in terms of ECM accuracy.

## 6 Addition of Multiple Thresholds

Instead of parameterizing the system using one threshold  $\theta$ , an alternative approach parameterizes the system using different thresholds for different kinds of anaphora. We divide anaphora into *nominals*, *personal pronouns* (e.g. I, me, myself, he, him), and *other pronouns* including indefinite quantifier-type pronouns (e.g. some, three, another) and also demonstrative pronouns (e.g. this, that, these, those). The motivation for the split between nominals and pronouns is their different distributions in raw text, which accounts for their split close to the root of the decision tree that is grown in [10] and plays a role in our decision to model pronominals and nominals as separate ME models. The motivation for the split between personal pronouns and other pronouns is that the two kinds of pronouns behave different in terms of the types and saliency of antecedents that they occur with [6].

Now we calculate the link accuracy of coreference link creation for different thresholds on cross validated training data including  $\theta_{nom}$  (over links of nominal anaphora),  $\theta_{pers}$  (over links of personal pronouns), and  $\theta_{other}$  (over links of other pronouns.) The results are shown in Figures 5, 6, and 7. Notice that there is a clear difference in the behavior of the different types of NPs when the thresholds are varied. Optimal values for the thresholds according to link F measure are  $\theta_{nom} = 0.80$ ,  $\theta_{pers} = 0.50$ , and  $\theta_{other} = 0.70$ . Link F measure over all kinds of anaphora is 0.7820 when multiple thresholds are used, an improvement over the score when a single threshold was used (0.7440).

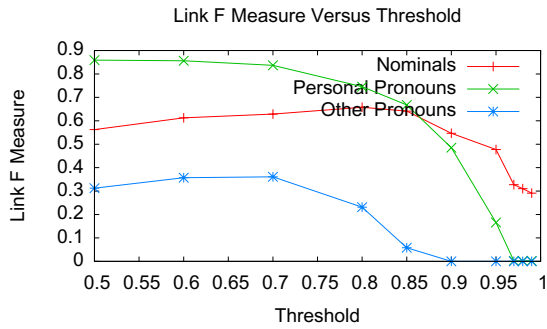


Fig. 5: Varying  $\theta$  by type of anaphor shows that optimal values for  $\theta$  differ for each type.

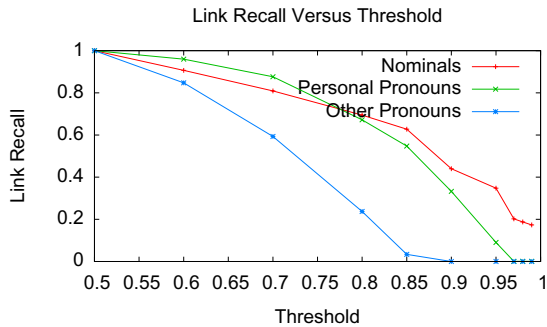


Fig. 6: Varying  $\theta$  show that link recall is also affected by the type of anaphor.

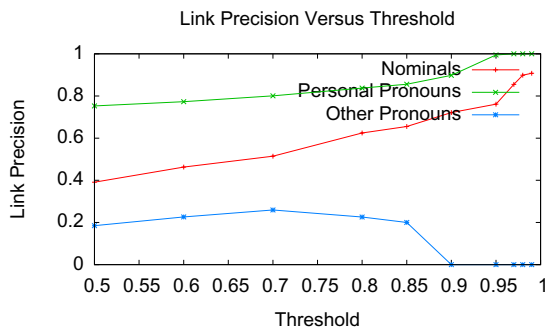


Fig. 7: In terms of precision, thresholding brings the most benefit to coreference involving nominal anaphors.

Table 4: Set of optimal threshold values, varying by type of anaphora, does give better ECM accuracy.

$\theta$ 's	ECM		
	Recall	Precision	F
baseline	0.620	0.620	0.620
optimal-1	0.627	0.627	0.627
optimal+	0.646	0.646	0.646

At this point, we would like to confirm that the optimal threshold values for  $\theta_{nom}$ ,  $\theta_{pers}$ , and  $\theta_{other}$  do in fact give better performance in entity detection. Accordingly, we evaluate the output of the system using ECM accuracy on the test data for different values of  $\theta$ . Let us define three different settings. The *baseline* setting uses  $\theta_{nom} = \theta_{pers} = \theta_{other} = 0.50$ . The *optimal-1* setting uses the thresholds that are found to be optimal given the restriction that all values are equal, namely  $\theta_{nom} = \theta_{pers} = \theta_{other} = 0.70$ . The *optimal+* setting uses the thresholds that are found to be optimal when these values are allowed to be different, in particular  $\theta_{nom} = 0.80$ ,  $\theta_{pers} = 0.50$ , and  $\theta_{other} = 0.70$ . The results are shown in Table 4. It shows that the addition of multiple thresholds is clearly beneficial in terms of ECM accuracy.

One might wonder what is the reason that having different thresholds for different kinds of anaphora yields better results. In order to start to shed light on this question, we examined a few of the clustering results. One typical example is shown in Figure 8. Not surprisingly, for low values of  $\theta$ , more NPs are clustered together, but for high values of  $\theta$  they are not. In this example, this behavior benefits nominal entity detection but detracts from pronominal entity detection. The fault that the default model chose to cluster the nominals “jews,” “mexicans,” and “palestinians” together might be because of their close proximity in the input text or that they match in grammatical number.

The problem with pronominal output when a high threshold is used is ostensibly because NPs are separated into different clusters when they should not be. On the other hand, one might believe that there should always be an increase in the ECM score because the system will not link NPs together unless it has a strong reason to do so. The resolution of this question seems to be in the scoring mechanism; when computing ECM accuracy, *all* of the NP mentions are by default assigned to some entity, even if the system did not link them to any other entity. Therefore, the ECM score does not necessarily increase when the threshold is raised is because of the disconnect between the means of thresholding, which assigns a high confidence to only a certain subset of NPs—those whose links were assigned high probabilities by the ME model, and the means of scoring, which scores by looking at all of the NPs.

	Pronominal Example	Nominal Example
<b>Correct Clustering</b>	newcomers they their	jews mexicans palestinians
<b>Theta=0.50 Clustering</b>	newcomers they their	jews mexicans palestinians
<b>Theta=0.90 Clustering</b>	newcomers they their	jews mexicans palestinians

**Fig. 8:** Representative example in which pronominal coreference resolution works better with a lower threshold while nominal coreference resolution works better with a higher one.

**Table 5:** Baseline is better in terms of profile strength but Optimal+ is better in terms of Mention Count.

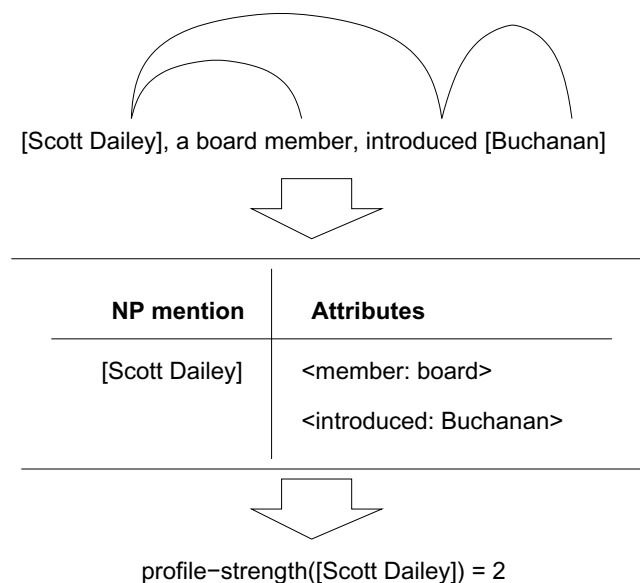
System	Profile Strength	Mention Count
Baseline	5.594	5.139
Optimal+	5.525	5.147

## 7 Extended Comparison of Different Systems

We performed experiments on the auxiliary test corpus. Because the auxiliary test corpus is not hand annotated with coreference information, these were performed by manually inspecting the output of two different systems: the *baseline* system, utilizing a single threshold of 0.5 and the *optimal+* system, utilizing the thresholds that were found to be optimal in Section 6.

We compared these two systems using two novel evaluation metrics: mention count and profile strength. *Mention count* is the simpler of the two. It is defined per entity as the number of entity mentions that the system treats as coreferent. This simple count differs from ECM in that it does not account for false positives or missing mentions. It has the advantage that it does not require hand-annotated key data for its computation. *Profile strength* is a measure that counts the number of descriptive elements that are associated with all of the entity mentions that the system treats as coreferent. Like mention count, it is defined on a per entity basis. Examples of descriptive elements include adjectives that modify an entity mention, verbs that take an entity mention as an argument, and mentions of other entities that are linked to the current entity in the same sentence. Unlike others, this metric is a measure of the “informativeness” of an entity as output by the system. See Figure 9.

Our analysis shows that the *optimal+* system seems



**Fig. 9:** Profile strength is a measure of the informativeness of an NP mention and is derived from the relations in which it participates.

**Table 6:** The number of entities with entirely “correct” descriptive elements is greater in Optimal+ than in Baseline in a manual examination of a small number of entities from the systems’ output.

System	False Positives	False Negatives
Baseline	11	0
Optimal+	2	4

to fare better in general than *baseline* using these metrics. Table 5 shows that *optimal+* is better in terms of mention count but *baseline* is better in terms of profile strength. However, a manual analysis of 14 entities generated by the system and chosen at random shows that the number of false positives for *baseline*, where a false positive is an entity with an incorrect descriptive element, is much higher than the corresponding number for *optimal+*. The difference between the number of false negatives for the two systems is quite a bit smaller. See Table 6.

## 8 Conclusions and Future Work

We have explored the use of confidence measures along with thresholding for the task of coreference resolution. We have defined several confidence measures over the output of a pairwise coreference resolution system. One confidence measure is defined in terms of the output of a greedy NP clustering algorithm, which has a small but noticeable impact on system performance when evaluated over pronominals. We have introduced a scheme of using a confidence measure along with thresholding for the task of coreference resolution. By letting a user of the system adjust the threshold, it allows more control over the kind of system output that is produced. We have proposed dividing anaphora into three types (nominal, personal pronoun, and other pronoun) and adopting a different threshold for each type. Our experiments show that this particular division leads to better performance than using one threshold for all anaphora. Our experiments have also used different methods to evaluate coreference resolution including link accuracy, ECM accuracy, and profile strength. The results suggest that when determining threshold values, it is a good idea to use a metric as close as possible to the ultimate evaluation metric. For example, determining the thresholding value using link accuracy did not always lead to higher ECM accuracy or higher profile strength. Along these lines, we have performed some qualitative analysis that suggests why this might be so.

In future work, we would like to examine other ways to determine threshold values as well as to examine the use of other confidence measures. As for the former, our current method uses a held out corpus in order to determine the threshold values. One possible alternative to using a held out corpus would be to estimate the accuracy of different threshold values by measuring the perplexity of thresholded examples when the model is applied to unlabeled data. As to the use of other confidence measures, we might look at the use of other ways to characterize clusters in the confidence measure.

## References

[1] A. L. Berger, S. D. Pietra, and V. J. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[2] E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):545–565, 1995.

[3] S. Caraballo and E. Charniak. New figures of merit for best-first probabilistic chart parsing. *Computational Linguistics*, 24(2):275–298, 1998.

[4] H. L. Chieu and H. T. Ng. Named entity recognition: A maximum entropy approach using global information. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan, 2002.

[5] J. Goodman. Global thresholding and multiple-pass parsing. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, 1997.

[6] J. Gundel, N. Hedberg, and R. Zacharski. Cognitive status and the form of referring expressions. *Language*, 69:274–307, 1993.

[7] S. Harabagiu, R. C. Bunescu, and S. J. Maiorano. Text and knowledge mining for coreference resolution. In *Proceedings of the Second Conference of the North American Chapter for the Association for Computational Linguistics (NAACL)*, 2001.

[8] X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, Main Volume, pages 135–142, Barcelona, Spain, July 2004.

[9] G. Miller. Wordnet: A lexical database. *Communications of the ACM*, 38(11):39–41, 1995.

[10] V. Ng and C. Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Philadelphia, PA, July 2002.

[11] S. P. Ponzetto and M. Strube. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL-06)*, New York, 2006.

[12] C. Quirk. Training a sentence-level machine translation confidence measure. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal, 2004.

[13] W. Soon, H. Ng, and D. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.

[14] M. Steedman, R. Hwa, S. Clark, M. Osborne, A. Sarkar, J. Hockenmaier, P. Ruhlén, S. Baker, and J. Crim. Example selection for bootstrapping statistical parsers. In *Proceedings of HLT-NAACL 2003*, 2003.

[15] N. Ueffing and H. Ney. Word-level confidence estimation for machine translation using phrase-based translation models. In *Proceedings of the Joint Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*, Vancouver, Canada, 2005.

[16] D. Willett, A. Worm, C. Neukirchen, and G. Rigoll. Confidence measures for hmm-based speech recognition. In *Proceedings of the Fifth International Conference on Spoken Language Processing (ICSLP)*, pages 3241–3244, Sydney, Australia, 1998.

[17] X. Yang, J. Su, and C. L. Tan. Improving pronoun resolution using statistics-based semantic compatibility information. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 165–172, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

# From Use Cases to UML Class Diagrams using Logic Grammars and Constraints

Henning Christiansen  
Roskilde University  
P.O. Box 260, DK-4000 Roskilde, Denmark  
*henning@ruc.dk*

Christian Theil Have  
IT University of Copenhagen  
Rued Langgaards Vej 7, DK-2300 København S, Denmark  
*cth@itu.dk*

Knut Tveitane  
IT University of Copenhagen  
Rued Langgaards Vej 7, DK-2300 København S, Denmark  
*knut@itu.dk*

## Abstract

We investigate the possibilities for automated transition from Use Cases in a restricted natural language syntax into UML class diagrams, by trying to capture the semantics of the natural language and map it into building blocks of the object oriented programming paradigm (classes, objects, methods, properties etc.). Syntax and semantic analysis is done in a framework of Definite Clause Grammars extended with Constraint Handling Rules, which generalizes previous approaches with a direct way to express domain knowledge utilized in the interpretation process as well as stating explicit rules for pronoun resolution. The latter involves an improvement of earlier work on assumptions with time stamps.

## Keywords

Application of NLP; Logic grammars and constraints; Domain specific analysis; Abduction and assumptions.

## 1 Introduction

Use cases are widely used to map requirements during inception and elaboration of a software development project. Mapping requirements is an important but difficult task, that can be impaired by lack of understanding and communication difficulties. According to [33], known as the Chaos Report, imprecise and incomplete requirements is a prevalent cause of software project failures. Often it seems that the stakeholders do not speak the same language. The engineer speaks in terms of design models while the domain expert defines the problem in domain specific, and often ambiguous, language within his frame of reference.

We suggest to bridge this gap by introducing an automatic and interactive system which translates a restricted, but naturally appearing, use case language into class diagrams in the UML notation [29].

The system maintains an up-to-date diagrammatical presentation of the current use case text in a window on the user's screen, cf. Fig. 1. This is intended to encourage an iterative mode of working, so as soon as the user adds a new or modifies an existing sentence,

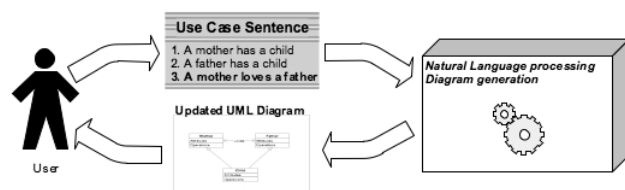


Fig. 1: *The system is used iteratively and interactively - each new sentence causes the UML diagram to be updated*

the consequences in terms of the object model is displayed immediately. This can aid the user in the process of understanding the current domain, including identifying possible misconceptions at an early stage. Possible applications include requirement engineering, brainstorming, prototyping and as a tool for teaching UML. The current version is limited to generation of class diagrams but points forward to the goal, which is to include also the dynamic aspects of use cases with generation of process diagrams, etc.

In this work, we reconcile and promote different methods for discourse analysis founded on logic programming technology, more specifically the familiar Definite Clause Grammars [30] and Constraint Handling Rules [17] (CHR). CHR adds a global resource, in the shape of a constraint store, and makes a sort of production-like rules available for controlling and utilizing this store, while maintaining a declarative framework. We indicate an improvement of earlier work concerned with so-called assumptions by adding time stamps, which make it possible to state rules for anaphora resolution in an explicit and logical form.

## 2 Related Work

**Similar Systems** Several authors have attempted to automate translation from specifications in natural language to code or diagrams, using either essentially formal language with a “natural” appearance or opportunistic parsing. We have not seen this related directly to Use Cases and UML, which is our approach.

Attempto Controlled English [18] is a system that

translates specification texts in a formal language of declarative sentences into first order logic discourse representation structures and optionally into Prolog. Sowa has defined a similar, but simpler, specification language called “Common Logic Controlled English” [31] which translates directly into first order logic (and vice versa).

The Metafor system [24] use opportunistic techniques and the semantically enriched lexicon *ConceptNet* [25] to derive and discover relations between classes and translate English sentences into code in Lisp or Python. Its input language supports a variety of narrative stances, past and present tense and anaphorical and indirect references. The authors [26] have coined the term “*programmatically*” to describe the transliteration process: “Programmatically semantics is a mapping between natural linguistic structures and basic programming language structures” [26]. We have adopted this terminology.

Examples of other approaches using NLP for requirement analysis are described by [15, 16, 19, 23].

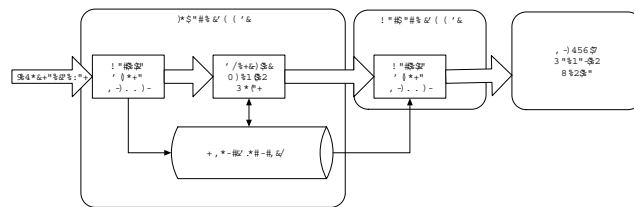
### Use Cases and Unified Modelling Language

Abbott [1] introduced a (manual) methodology for object-oriented program design that derives candidate classes, objects and operators from the syntactical elements of English sentences. Abbotts method, became an integral part of Booch’s “object-oriented design” [5] process, where an informal problem description is formalized through definition of objects and their attributes and operations. Jacobsen introduced the concept of Use Cases [21], which resembled Booch’s problem descriptions. Use Cases describe a users view of the system which is useful for “gaining an understanding of the problem” and “identifying candidate objects” [4]. Rumbaugh et. all [27] described the OMT notation including the “Class Association Diagram”, the precursor of the UML class diagram. Booch, Jacobson and Rumbaugh later defined the first draft for a “Unified Modeling Language” [7].

Use cases model the actors of a system and the flow of events between them. They describe *what* a system does without specifying how [6]. Even though use cases are written in natural language, only a subset of English is normally used. The UML User Guide [6] provides examples but no clear guidelines.

Cockburn [12] has suggested a semi-formal approach where each action description has a certain structured format. The CREWS guidelines for use case writing in [3] provides insight wrt. to the linguistic structures of use cases. The CP guidelines [13] was proposed as simpler set guidelines with similar expressiveness. The language suggested by the guidelines includes present tense *subject-verb-object* like sentences with no adverbs or adjectives. In essence the guidelines advocate avoiding all ambiguous language constructs. Our grammar is inspired by the CP guidelines, but allows more advanced pronoun usage.

**NLP using Logic Grammars** As shown in previous work [2, 8, 9, 10], CHR provides a straightforward implementation of abduction, and here we follow the principle of discourse analysis as abduction, introduced by [20] and now widely accepted: the meaning of a discourse is taken to be the set of “hidden” facts over which the discourse is faithfully created.



**Fig. 2: Architecture Overview.** The thick lines illustrate program flow and the thin lines illustrate the use of the constraint store.

Our work follows the tradition of logic programming based grammars, but extends previous work in different ways. Assumption Grammars [14] (AG) provide mechanisms that may cope with pronouns. Inspired by the work of [9, 10] that realizes AGs in CHR, we have extended with time stamps to make it possible to specify detailed scope and preference rules in CHR, which otherwise is a shortcoming of AGs. Creation of data and knowledge bases from text using logic grammars have been pursued by a variety of authors, e.g., [32, 18]. The model of “Meaning in Context” formalized by [11], which is based on CHR shows how domain knowledge utilized in the interpretation process can be expressed directly in CHR.

## 3 System Overview

### 3.1 Architecture

The system is implemented in a combination of Definite Clause Grammars (DCG) and CHR, with sentence meaning added gradually to the constraint store. This converted into the GraphViz language using a second DCG, and rendered as a UML class diagram using a GraphViz engine, and displayed to the user; see Fig. 2. Incrementality is simulated by parsing the entire text and drawing new diagram when a period which is added or changed. Only a rudimentary user interface exists at the moment, but the current prototype qualifies as proof of concept for our ideas.

### 3.2 Supported Language Constructs

The subset of natural English supported by our system needs to be sufficiently expressive as to cover the important entities and relations in an object oriented system description. Fig. 3 below shows the diagram for the example sentences.

#### 3.2.1 Basic Sentences

The basic sentence consist of a noun phrase followed by a simple verb phrase that contains an intransitive or transitive verb. We consider first verbs that imply an action to be performed by or on the subject of the sentence. The subject maps to a *class definition* in the object oriented programming paradigm. The verb maps to a *method* of the class represented by the subject. For a transitive verb, the object defines another class that serves as *argument* to the method. Example: “*The professor teaches. A student reads, writes projects and takes exams.*”

### 3.2.2 Property Sentences

Property sentences imply an ownership or containment relation, and are similar to the transitive sentences above, but use specific verbs such as “have”. The object may be plural and quantified. The quantification may be a numeral or a linguistic quantifier such as “some”, “most” or “every”. These sentences associate properties expressed by their object with the class(es) indicated by their subject. There are different approaches for representing these in object oriented programming languages. We have tried not to limit the flexibility, by maintaining as precise information as possible about the cardinality of the property. When exact number is given, this is preserved and a quantifier such as “some” is mapped into an undetermined cardinality denoted as “n”. When alternatives are given for the same property, the different cardinalities are aggregated into an interval; the details are spelled out in section 4.2 below. Example: “A professor has an office. The university has five study lines.”

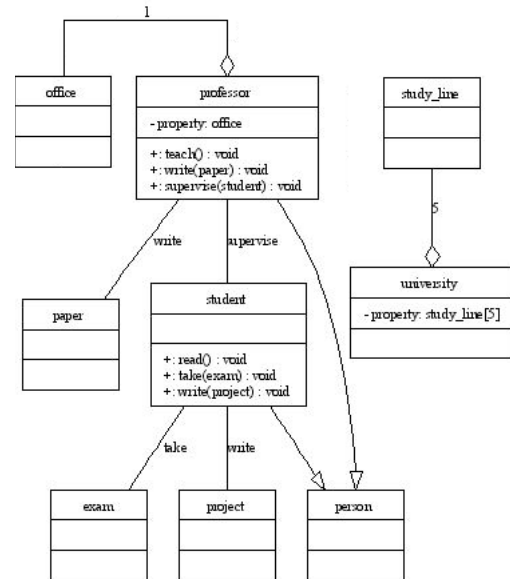


Fig. 3: Results from the collected sentences of sec. 3

### 3.2.3 Inheritance and Instantiation Sentences

Sentences formed with the verb “to be” relate single objects or classes to other classes. In “A student is a (kind of) person”, “person” is a superclass of “student”, and “student” a subclass of “person”. Multiple inheritance, where the subject is “a kind of” more than one class is also possible. In “John is a student”, the subject is a proper noun indicating a named entity (John) of the class “student”. In object oriented terminology, John is an *object* of class student. After an individual has been introduced with a designated class membership, it can be used as a *prototype* member of that class *C* in sentence forms above. Consider a sentence such as “John reads”. It looks straightforward at first glance, but its programmatic semantics is bit more complex. “John” maps to an object, “reads” to a method, but objects don’t define methods (or properties). The sentence must be understood to mean that the method belongs to the *class* the object is an instance of. Prototypes have also natural usages as arguments in basic sentences that introduces method In a suitable context, “Mary interviews Peter” carries the same programmatic semantics as “Teachers interview students”. Example: “Students are a kind of persons and professors are a kind of persons.”

### 3.2.4 Adjectives

The implied programmatic semantics of adjectives limits or specializes the meaning of a noun which may be reflected in the class diagrams in different ways. The term “large car” may imply a subclass of car, a boolean property “large”, or a property which may take different values, one of which is “large”. A non-trivial semantic analysis needs to be made, and we currently provide no solution.

### 3.2.5 Pronouns

We require that pronouns can be resolved in a unique way which is intelligible to the user; at the same time we should also, for the acceptance of the tool, allow a variety of natural patterns. As is well-known, pronoun

(and anaphora) resolution is one of the most difficult tasks in computational linguistics, cf. [28], and we have decided to use a simple heuristic, and reject any sentence for which it does not apply. Resolving, say, “he” considers the most recent occurrence of a male object which is found in a previous sentence *S*; however, if *S* contains two candidates, the pronoun application is claimed ambiguous. This excludes usages such as “Peter and Paul ... . He ...” and “Women are persons. They have two legs.” Example: “The professor writes papers and he supervises students.”

## 4 NLP Methods Applied

In the following, we assume a basic knowledge of DCG, and explain the CHR specifics that are used.

### 4.1 Overall Implementation Principles

We consider sentences that describe class hierarchy, e.g., “A dog is an animal”. The following grammar rule gives the overall structure.

```

sentence -->
    fc_noun_phrase(Number, _, subj, IdSub),
    subord_verb(Number, _),
    fc_noun_phrase(_, Number, obj, IdObj),
    {extends(IdSub, IdObj)}.

```

Notice that it produces no explicit output via arguments, but updates the constraint store by the calls to constraints, i.e., extends by abductive reasoning the constraint store with those facts that seem to be the reason, why the sentence can be stated. In this example, the rule depends on `extends(dog, animal)` and, since `extends` is declared as a constraints, it will be added to the constraint store if it was not there already. The grammar rules for noun phrases may, in a similar way, create the facts `class(dog)` and `class(animal)`. Thus the analysis of this sample sentence produces a store of three constraints that can

be converted in a straightforward way into the input language of Graphviz, which in turn produces a class diagram showing that the class of dogs is a subclass of animals.

The second attribute of noun phrases is called `CollectiveNumber`; for “a cat and a dog” it evaluates to `sing` and for “cats and dogs” to `plur`. Taking the collective number for the subject as the number for the object allows “A dog is an animal and a pet” and excludes “A dog is animals and pets”. Noun phrases are divided into different categories with particular restrictions; for example, `fc_noun_phrase` (for fully specified class) used above is a category which forbids pronouns and quantified expressions like “two cats” in this particular sort of sentences. Similar categories are defined for `indiv_noun_phrase` referring to particular objects (“She, Peter, and Paul”), `rc_noun_phrase` for restricted class (“Mary and the boys” assuming that Mary is a prototype for some class), and `q_noun_phrase` for quantified expressions such as “four legs and a tail”.

Noun phrases generate a representation of their contents, using a plus to combine conjunctive phrases. Here are some examples, assuming that Mary is a prototype woman and that “she” refers to Mary.

<code>fc_</code>	cats and dogs	<code>cat+dog</code>
<code>indiv_</code>	her, Peter, and Paul	<code>mary+peter+paul</code>
<code>rc_</code>	Mary and the boys	<code>woman+boy</code>
<code>q_</code>	a tail and some legs	<code>tail:1+legs:n</code>

The following CHR rules unroll constraints with composite arguments into individual constraints.

```
extends(A+B,C) <=> extends(A,C), extends(B,C).
extends(A,B+C) <=> extends(A,B), extends(A,C).
```

These are so-called simplification rules, triggered each time an `extends` constraint with a plus in one of its arguments appear in the store. They delete the constraint(s) matched by the left-hand side, the head, and add those on the right, the body.

## 4.2 Expressing Knowledge About Use Case Modeling

CHR can be used to express knowledge about the domain in question. We can illustrate this by the way we aggregate the constraints emerging from different statements about the same property. We use `property(car,wheels:4)` to express that a car has four wheels and `property(car,doors:(2..5))` to say that it has between 2 and 5 doors. Consider the following CHR rule which is part of the implementation.

```
property(C,P:N), property(C,P:M) <=>
  q_count(N), q_count(M), q_less_eq(N,M)
  | property(C,P:(N..M)).
```

It applies when the store contains two `property` constraints for the same class and property, provided the guard is true. The guard is an optional part between the arrow and the vertical bar which here refers to Prolog predicates written specifically for the purpose, so that, say, `q_count(5)`, `q_count(n)`, and `q_less_eq(2,n)` are true. So “Peter has a dog. Paul has five dogs” yields `property(man,dog:1)` and `property(man,dog:5)` which by the rule above

get replaced by `property(man,dog:(1..5))`. Another rule (not shown) combines different intervals for multiplicity into one.

## 4.3 Pronoun Resolution

Here we sketch briefly the approach inspired by the assumption principle of [14] but extended with a time stamp (here, sentence number) to realize the indicated principle. When, say, “Peter” is mentioned in sentence no. 7, a constraint `referent(sing,masc,peter,7)` is emitted, and an occurrence of “him” in sentence no. 10 gives rise to `expect_referent(sing,masc,X)`; the following rule attempts to bind X to the suitable value.

```
sentence_no(Now), referent(No,G,Id,T) \
  expect_referent(No,G,X) <=>
  T < Now,
  \+ ( find_constraint( referent(No,G,_,TMoreRecent),_)
    T < TMoreRecent, TMoreRecent\==Now)
  | ( find_constraint( referent(No,G,Id1,T), _)
    Id1\=Id -> X = error:pronoun:ambiguous(No,G,Now)
    ; X=Id ).
```

The rule is a so-called simpagation rule which, when applied, keeps the constraints before “\” in the store and removes the remaining ones up until the arrow. CHR does not allow negations in the head, so the test that time T designates the most recent `referent` (i.e., there is no other such with a more recent time stamp) is done in less elegant way in the guard. The body tests for ambiguity and may generate an error code. Finally, the following rule catches unresolved pronouns if, e.g., the whole text start with “He”.

```
sentence_no(Now) \ expect_referent(No,G,X) <=>
  X=error:pronoun:unresolved(No,G,Now).
```

The following grammar rule for using pronouns shows how the implemented `expect_referent` constraint can be used.

```
indiv_simple_noun_phrase(Num,Case,G,Id) -->
  pronoun(Num,Case,G),
  {expect_referent(Num,G,Id)}.
```

This example illustrates how relatively complicated contextual dependencies in logic grammars can be modeled in a fairly concise way using CHR. The use of prototypical individuals for classes is realized in a similar way. In “Mary is the boss. She manages the employees.”, the pronoun is resolved to `mary`, and a call to a constraint `expect_class(...mary...)` locates the class `boss` (i.e., if it is unique, otherwise an error code) and the constraint `method(boss,manage,employees)` is created.

## 4.4 From Constraint Stores to Diagrams

Another DCG is defined for the GraphViz input language. This grammar references the constraint store and generates a phrase as long as possible, thereby converting constraints into phrases to be given as input to GraphViz. This is straightforward and not described further. If, for example, the constraint store contains `class(man)` and `method(man,walk)`, the nonterminal `class_node` generates the phrase `man[ label = "{man||: walk(): void\1}"]`.



## 5 Conclusions and Future Work

We presented a system for analyzing a restricted natural language for use case writing, based on Definite Clause Grammars extended with Constraint Handling Rules. Our grammar captures candidate domain classes and their relations and visualize these using an UML class diagram. The syntax of the language is simple, but expressive enough to model a given domain. The language seems natural and expressive but avoids inherently ambiguous sentence elements such as adverbs and adjectives. By extending our grammar with Constraint Handling Rules, we are able to handle pronoun resolution with ambiguity detection, prototypical references (e.g. names) and allow the user to express knowledge about the domain, such as multiplicity, using simple prototypical sentences.

Our grammar captures information about the static world. This is precisely what is reflected in the UML class diagram. However, Use Cases normally also contain sentences about event flows as such “*If the light is red then the cars stop*” and “*They wait until the light is green*”. These sentences contain information of a dynamic nature that would typically be depicted in state or flow charts, or in UML, sequence diagrams. It would be a natural next step to extend the grammar to support such sentences. As we have indicated, CHR rules which include more “expert knowledge” about use case modeling can be added to the analysis, and this potential should be investigated further. However, this must be done with care: adding more intelligence to the system may help the user to realize properties of the world he is describing, but it may also destroy the transparency and incrementality exposed in the current prototypic.

**Acknowledgement:** This work is supported by the CONTROL project, funded by Danish Natural Science Research Council.

## References

- [1] R. J. Abbott. Program design by informal English descriptions. *Communications of the ACM*, 26(11):882–894, 1983.
- [2] S. Abdennadher and H. Christiansen. An experimental CLP platform for integrity constraints and abduction. In *Proceedings of FQAS2000, Flexible Query Answering Systems: Advances in Soft Computing series*, pages 141–152. Physica-Verlag (Springer), 2000.
- [3] C. B. Achour. Guiding scenario authoring. In *EJC*, pages 152–171, 1998.
- [4] E. V. Berard. *Be Careful With 'Use Cases'*. The Object Agency, Inc., August 1998.
- [5] G. Booch. Object-oriented development. *IEEE Transactions on Software Engineering*, 12(2):211–220, Feb. 1986.
- [6] G. Booch, I. Jacobson, and J. Rumbaugh. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
- [7] G. Booch and J. Rumbaugh. *Unified Method for Object-Oriented Development Version 1.0*. Rational Software Corporation, 1997.
- [8] H. Christiansen. A constraint-based bottom-up counterpart to definite clause grammars. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *RANLP*, volume 260 of *Current Issues in Linguistic Theory (CILT)*, pages 227–236. John Benjamins, Amsterdam/Philadelphia, 2004.
- [9] H. Christiansen. CHR Grammars. *Int'l Journal on Theory and Practice of Logic Programming*, 5(4-5):467–501, 2005.
- [10] H. Christiansen and V. Dahl. HYPROLOG: A new logic programming language with assumptions and abduction. In M. Gabbrieli and G. Gupta, editors, *ICLP*, volume 3668 of *Lecture Notes in Computer Science*, pages 159–173. Springer, 2005.
- [11] H. Christiansen and V. Dahl. Meaning in Context. In A. Dey, B. Kokinov, D. Leake, and R. Turner, editors, *Proceedings of Fifth International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-05)*, volume 3554 of *Lecture Notes in Artificial Intelligence*, pages 97–111, 2005.
- [12] A. Cockburn. Structuring use cases with goals. *Journal of Object-Oriented Programming*, Sept.Oct. 1997.
- [13] K. Cox and K. Phalp. Replicating the CREWS use case authoring guidelines experiment. *Empirical Software Engineering*, 5(3):245–267, 2000.
- [14] V. Dahl, P. Tarau, and R. Li. Assumption grammars for processing natural language. In *ICLP*, pages 256–270, 1997.
- [15] J. Drazan and V. Mencl. Improved processing of textual use cases: Deriving behavior specifications. In *Proceedings of SOFSEM 2007*, volume 4362 of *Lecture Notes in Computer Science*. Springer, 2007.
- [16] A. Fantechi, S. Gnesi, G. Lami, and A. Maccari. Application of linguistic techniques for use case analysis. In *RE*, pages 157–164. IEEE Computer Society, 2002.
- [17] T. Frühwirth. Theory and practice of constraint handling rules, special issue on constraint logic programming. *Journal of Logic Programming*, 37(1-3):95–138, Oct. 1998.
- [18] N. E. Fuchs. Attempto controlled english. In *WLP*, pages 211–218, 2000.
- [19] H. M. Harmain and R. J. Gaizauskas. Cm-builder: A natural language-based case tool for object-oriented analysis. *Autom. Softw. Eng.*, 10(2):157–181, 2003.
- [20] J. R. Hobbs, M. E. Stickel, D. E. Appelt, and P. A. Martin. Interpretation as abduction. *Artif. Intell.*, 63(1-2):69–142, 1993.
- [21] I. Jacobson. Object oriented development in an industrial environment. In *OOPSLA*, pages 183–191, 1987.
- [22] K. P. Karl Cox and M. Shepperd. Comparing use case writing guidelines. In *Seventh International Workshop on Requirements Engineering (RE'01)*, June 2001.
- [23] N. Kiyavitskaya, N. Zeni, L. Mich, and J. Mylopoulos. Experimenting with linguistic tools for conceptual modelling: Quality of the models and critical features. In F. Meziane and E. Métais, editors, *NLDB*, volume 3136 of *Lecture Notes in Computer Science*, pages 135–146. Springer, 2004.
- [24] H. Liu and H. Lieberman. Toward a programmatic semantics of natural language. In *VL/HCC*, pages 281–282. IEEE Computer Society, 2004.
- [25] H. Liu and P. Singh. Conceptnet: A practical commonsense reasoning toolkit, May 02 2004.
- [26] Liu, Hugo and Lieberman, Henry. Programmatic semantics for natural language interfaces. In *Proceedings of ACM CHI 2005 Conference on Human Factors in Computing Systems*, volume 2 of *Late breaking results: short papers*, pages 1597–1600, 2005.
- [27] M. E. S. Loomis, A. V. Shah, and J. E. Rumbaugh. An object modeling technique for conceptual design. In J. Bézuvin, J.-M. Hullot, P. Cointe, and H. Lieberman, editors, *ECOOP '87, European Conference on Object-Oriented Programming*, volume 276 of *Lecture Notes in Computer Science*, pages 192–202. Springer-Verlag, 1987.
- [28] R. Mitkov. *Anaphora Resolution*. Longman (Pearson Education), 2002.
- [29] Object Management Group. *Unified Modeling Language (UML), version 2.0*. Object Management Group, Framingham, Massachusetts, Oct. 2004.
- [30] F. C. N. Pereira and D. H. D. Warren. Definite clause grammars for language analysis—A survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, 13(3):231–278, 1980.
- [31] J. F. Sowa. *Common Logic Controlled English*, 2004. Draft, <http://www.jfsowa.com/clce/specs.htm>.
- [32] P. Tarau, K. D. Bosschere, V. Dahl, and S. Rochefort. Logimoo: An extensible multi-user virtual world with natural language control. *J. Log. Program.*, 38(3):331–353, 1999.
- [33] The Standish Group. The CHAOS report, 1994.

# Better Training for Function Labeling

Grzegorz Chrupala  
Dublin City University  
Dublin, Ireland  
*gchrupala@computing.dcu.ie*

Nicolas Stroppa  
Dublin City University  
Dublin, Ireland  
*nstroppa@computing.dcu.ie*

Josef van Genabith  
Dublin City University  
Dublin, Ireland  
*josef@computing.dcu.ie*

Georgiana Dinu  
Eberhard Karls Universität  
Tübingen, Germany  
*gdinu@sfs.uni-tuebingen.de*

## Abstract

Function labels enrich constituency parse tree nodes with information about their abstract syntactic and semantic roles. A common way to obtain function-labeled trees is to use a two-stage architecture where first a statistical parser produces the constituent structure and then a second component such as a classifier adds the missing function tags.

In order to achieve optimal results, training examples for machine-learning-based classifiers should be as similar as possible to the instances seen during prediction. However, the method which has been used so far to obtain training examples for the function labeling classifier suffers from a serious drawback: the training examples come from perfect treebank trees, whereas test examples are derived from parser-produced, imperfect trees.

We show that extracting training instances from the reparsed training part of the treebank results in better training material as measured by similarity to test instances. We show that our training method achieves statistically significantly higher f-scores on the function labeling task for the English Penn Treebank. Currently our method achieves 91.47% f-score on the section 23 of WSJ, the highest score reported in the literature so far.

## Keywords

function labeling, machine-learning

## 1 Introduction

Treebanks such as the English or Chinese Penn Treebanks are collections of syntactic parse trees. The trees include extra information in addition to constituent bracketing and labeling. In this paper we focus on the function labels (also known as function tags). The function labels used in the Penn treebanks fall into several types. Grammatical labels are used to encode the grammatical function of the constituent. Form-function labels are used to indicate the semantic class of adjuncts and discrepancies between form and function. There is also a label used for topicalization, and

several other miscellaneous labels. Detailed information about the label sets can be found in the annotation guidelines for the respective treebanks (Bies, 1995; Xue and Xia, 2000). Table 1 provides a summary of labels used in the English and Chinese treebanks.

Widely used statistical parsers, such as those of (Collins, 1999; Charniak, 2000), which use treebanks as training data to parse unseen sentences, do not include function labels in the parse trees they produce. However, pure constituency trees may be insufficient for many NLP tasks - often something closer to semantic information is required. Grammatical functions and semantic roles such as those encoded in form-function labels are a step towards this deeper, abstract representation. Thus an important task is to be able to produce parses which include the richer annotations provided by function labels.

In this paper we review approaches to producing parse trees with function labels and present our research on the impact of different training methods in a two-stage processing architecture where we use machine learning techniques to train classifiers which add function labels to bare constituent trees such as those output by Charniak's or Collins' parsers.

In a multi-stage processing pipeline the optimal training input for the downstream stages is important. Ideally the training at stage  $n + 1$  should be performed on input from stage  $n$ : e.g. a parsing model which uses automatically POS-tagged input should be trained on tags produced by the POS tagger used to preprocess the raw input, rather than gold tags. In practice pipeline architectures this has been violated.

For example, in the case of function labeling, the two-stage models used in previous work have all used "perfect" treebank trees to train the function labeler even though the labeler operates on "imperfect" trees output by the parser. This is presumably due to the fact that the function labels we want to learn are attached to nodes in the treebank trees. Unfortunately, those nodes do not necessarily correspond to constituents in the trees produced by the parser.

The main contribution of our paper consists in presenting a theoretically sound method of training on parser output rather than treebank trees for the function labeling task and investigating the effect of several versions of this approach on the results as compared against the baseline method which uses perfect tree-

Label	Meaning	ETB	CTB
<b>Clause types</b>			
IMP	imperative		✓
Q	question		✓
<b>Syntactic labels</b>			
LGS	logical subject	✓	✓
PRD	predicate	✓	✓
PUT	complement of put	✓	
SBJ	surface subject	✓	✓
IO	indirect object		✓
OBJ	direct object		✓
FOC	focus		✓
<b>Miscellaneous labels</b>			
CLF	it-cleft	✓	
HLN	headline	✓	✓
TTL	title	✓	✓
CLR	closely related	✓	
APP	appositive		✓
PN	proper noun		✓
SHORT	short form		✓
WH	WH-phrases		✓
<b>Semantic (form-function) labels</b>			
ADV	adverbial	✓	✓
BNF	benefactive	✓	✓
DIR	direction	✓	✓
EXT	extent	✓	✓
LOC	locative	✓	✓
MNR	manner	✓	✓
NOM	nominal	✓	
PRP	purpose or reason	✓	✓
TMP	temporal	✓	✓
CND	condition		✓
IJ	interjective		✓
VOC	vocative	✓	✓
<b>Topicalization</b>			
TPC	topicalized	✓	✓

**Table 1:** *Function labels in the English and Chinese Penn Treebanks*

bank trees. We show that using the better-motivated methods helps to improve the quality and quantity of training material available to the machine-learning algorithm.

In Section 2 we describe previous approaches to the function labeling task. In Section 3 we present our improved method of obtaining appropriate training material for function labeling. In Section 4 we present experimental results for English and Chinese, and in Section 5 we conclude and suggest possible future research.

## 2 Previous Work

There are two main approaches to obtaining parse trees with function label information:

- Two-stage systems, where “bare” parse trees are enriched with function labels in a postprocessing step (Blaheta and Charniak, 2000; Jijkoun and de Rijke, 2004; Chrupala and van Genabith, 2006),
- Modifying the parser’s internals to output function labels (Musillo and Merlo, 2005; Gabbard

et al., 2006).

Blaheta and Charniak (2000) use a probabilistic model with feature dependencies encoded by means of feature trees to add English Penn II Treebank function labels to the output of Charniak’s parser. They report an f-score of 87.277% on correctly parsed constituents, and 88.472% on original treebank trees from WSJ section 23.

Jijkoun and de Rijke (2004) describe a method of learning function labels, empty nodes and coindexations from the English Penn II Treebank trees. They transform trees to dependencies and use memory-based learning to transform the dependency graphs. One of the transformations is node renaming, which adds function labels to parser output. They report an f-score of 88.5% for the task of function tagging on correctly parsed constituents on WSJ section 23.

Chrupala and van Genabith (2006) compare the performance of three machine learning algorithms on function labeling of the Spanish Cast3LB treebank (Civit and Martí, 2004) against the baseline which uses a modified version of Bikel’s parser (Bikel, 2002) to directly learn and output function-labeled nodes. They evaluate their results in a task-based setting by using the resulting function-labeled trees to produce LFG f-structures, and report a 2.67% improvement in f-score over the baseline for this task.

Musillo and Merlo (2005) extend the Henderson parser (Henderson, 2003) and model function labels as both expressions of the lexical semantics properties of a constituent and as syntactic elements whose distribution is subject to structural locality constraints. This improves their parsing score and function labeling score on the grammatical and semantic label classes in the English Penn II Treebank.

Gabbard et al. (2006) describe a two stage parser which builds Penn Treebank analyses including both function labels and empty categories and coindexations. Function labeling is performed during the first stage: they modify Bikel’s implementation of Collins’ parsing model to enable it to output function labels. They report 88.96% f-score on correctly parsed constituents on WSJ section 23.

## 3 Methods

We use the two-stage architecture, in which the first stage consists of bare constituency parsing using a statistical parsing model and the second stage decorates constituent labels with function labels. The labeler is a machine-learning classification model. Our focus is to investigate ways of improving the performance of the classifier by extracting more and better quality training examples from the available resources.

By improving the quality of training material we mean making it more similar to the instances that the model has to classify during prediction, i.e. we will try to better approximate the standard assumption made in most machine-learning research that instances (in training and test) are *independently and identically distributed* (i.i.d.), in particular, they should be drawn from the same probability distribution.

In the previous two-stage approaches (Blaheta and Charniak, 2000; Jijkoun and de Rijke, 2004; Chrupala

and van Genabith, 2006) this assumption is violated in that the training instances are feature vectors extracted from nodes in the “perfect” parse trees from the treebank, whereas at prediction time the model has to classify instances extracted from nodes in imperfect parser output, which can and does contain a certain proportion of errors (incorrect bracketings or incorrect constituent labels).

We propose to alleviate this issue by using training material which is extracted from the trees obtained by reparsing the training portion of the treebank and using the (imperfect) trees output by the parser rather than the original treebank trees. We still need the original treebank trees in order to assign classes (function labels) to the training instances extracted from parser output. We do this by matching node-spans between automatically parsed trees and gold trees in the training set. We only extract training instances from those nodes in the automatically parsed tree for which there is a node with the same span in the gold tree, from which we can obtain the function label.

### 3.1 Baseline Method

Our baseline method uses a simple two-stage architecture: constituency parsing, followed by function labeling. The first stage is performed by the constituency parsing model, obtained by training a statistical parser on the training portion of the treebank. The output of this stage, sentences parsed into bare constituency trees, are the input to the second stage component, i.e. the function labeler. The labeler is trained, in the baseline method, on the original “perfect” trees from the training portion of the treebank.

#### 3.1.1 Features

Each node to label is represented as a fixed-length vector of features encoding categorical, configurational and lexical information about the node and its context. We use the following features:

1. Node constituent label
2. Node head word’s part of speech tag
3. Node head word
4. Node’s head-sister’s constituent label
5. Node’s head-sister’s head word’s part of speech
6. Node’s head-sister’s head word
7. Node’s alternative head word’s part of speech tag (alternative head is the head of the second child for PPs)
8. Node’s alternative head word
9. Node’s yield length
10. Node’s mother’s constituent label
11. Node’s grandmother’s constituent label
12. Offset to node’s head sister

Plus the following:

- Features 1,2,3,7,8,9 for the preceding sister node
- Features 1,2,3,7,8,9 for the following sister node

Those features are binarized (i.e. each feature:value pair is mapped to a new boolean feature), and the examples (i.e. the feature vectors) are used to train a classifier. There is one minor complication: in principle a node can be decorated with more than one function label (although labels belonging to the same group are (usually) mutually exclusive). Thus we could train

a separate classifier for each label, or a separate classifier for each label group, or simply treat the label set on the node as an atomic class. In the experiments reported below we used the first method, i.e. we train a separate binary classifier for each function label, and combine their output to add a set of function labels to each node.

### 3.2 Evaluation Metrics

Evaluating the performance of a function labeling system is not entirely straightforward. Since the constituency trees output by the parser are not identical to the gold-standard treebank trees, one cannot report simple labeling accuracy. Blaheta and Charniak (2000) decide to measure accuracy (for their with-null metric) and f-score (for the no-null metric) over the *correctly parsed* subset of nodes, i.e. those nodes that subtend the correct portion of the string and have the correct constituent label. In this work we use the same metric.

### 3.3 Training on Parser Output

Using the metric described above, since we are evaluating only the correctly parsed subset of nodes, one might naively expect that the score should be the same for labeling both the parser output and the perfect treebank trees. However, the results reported in (Blaheta and Charniak, 2000) show that the performance is over 1% better for the treebank trees. The authors convincingly explain that the likely cause is that although the focus node to be labeled is correctly parsed, the neighbouring context nodes that some features depend on may be incorrect.

This fact serves as our motivation for extracting training examples from treebank sentences parsed by the same parser that is used to parse unseen test data. Our hypothesis is that training instances obtained in this way are going to be more similar to test instances than the ones extracted from perfect treebank trees and thus will better approximate the i.i.d assumption. We expect that the machine learning algorithm will perform better on test instances which are more similar to those used for training; for example it might be able to weight down features which depend on incorrect characteristics of the parse trees, as such features will be less reliable as class predictors.

Our improved training example extraction procedure is as follows: sentences in the training portion of the treebank are reparsed. Then we follow the algorithm presented in Figure 1 to extract training instances. The function `INSTANCES` returns training instances from a parse tree  $T$  given the reference treebank gold tree  $T'$  for the same sentence. For each node  $n$  in  $T$  we check whether there exist one or more nodes with the same span and constituent label in the corresponding  $T'$  (line 3)<sup>1</sup>. The function `INSTANCE` takes the union of the function label sets (`FUNCLABELS( $n'$ )`) found on the nodes in the gold tree  $T'$  and returns this set (as a class  $\mathcal{C}$ ) together with the feature vector `FEATURES( $n$ )` corresponding to node  $n$ .

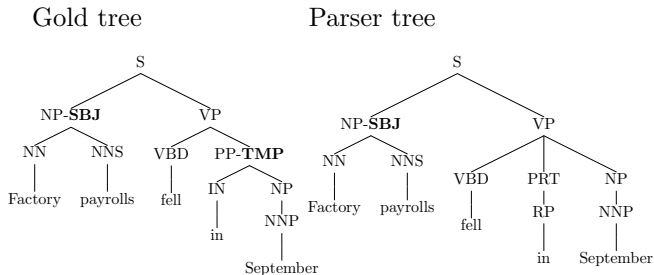
<sup>1</sup> The square bracket notation denotes multisets.

```

1 INSTANCES( $T, T'$ ) =
2  $\mathcal{N} \leftarrow \{ \text{NODESPEC}(n) \mid n \in \text{NODESET}(T') \}$ 
3  $\mathbf{I} \leftarrow [ \text{INSTANCE}(n, T') \mid n \in \text{NODEMULTISET}(T) \wedge \text{NODESPEC}(n) \in \mathcal{N} ]$ 
4 return  $\mathbf{I}$ 
5 INSTANCE( $n, T'$ ) =
6  $\mathcal{C} \leftarrow \bigcup \{ \text{FUNCLABELS}(n') \mid n' \in \text{NODESET}(T') \wedge \text{NODESPEC}(n') = \text{NODESPEC}(n) \}$ 
7 return  $\langle \text{FEATURES}(n), \mathcal{C} \rangle$ 
8 NODESPEC( $n$ ) =  $\langle \text{NODESPAN}(n), \text{NODECONSTITUENTLABEL}(n) \rangle$ 

```

**Fig. 1:** Algorithm for extracting training instances from a parser tree  $T$  and gold tree  $T'$



**Fig. 2:** Example gold and parser tree

Figure 2 illustrates this algorithm: in effect we transfer function labels from nodes in the gold tree to matching nodes in the parser tree. Matching nodes are those whose constituent label and span are the same. In the example tree the SBJ function label is transferred but the TMP is not since there is no matching node in the parser tree due to a parsing error.

A problem with our method as described so far is that we use a constituency parsing model trained on sections 2-21 of WSJ to reparse those same sections so that we can extract training material from them. Obviously it is very likely that the resulting parse trees will be closer to gold trees than will be the case for test sentences taken from WSJ section 23. It would be advisable to extract input for our labeling model from the treebank trees reparsed with parsing models trained on material from which those trees are excluded. We did not do this for the experiments on the English data with Charniak’s parser, due to technical difficulties encountered when attempting to retrain this parser. However, for the experiments on the Chinese data with Bikel’s parser we did 10-fold-cross-training, that is we divided the training material into 10 parts and parsed each part in turn with the model trained on the remaining 9 parts. We report the results on the Chinese data in section 4.

### 3.3.1 Instance Similarity

We tried to verify our prediction that the instances extracted using our method would be more similar to test instances. As a simple metric of similarity, we compare instance overlap between the training set and the test set. Instance overlap is the cardinality of the intersection of the multiset of instance feature vectors used for training and the multiset of instance feature vectors used for testing. For multisets defined as tuples  $(A, f)$  with the underlying set  $A$  and the multiplicity function  $f : A \rightarrow \mathbb{N}$  which assigns to each element the number of times it occurs, multiset cardinality is

	Instance count	Overlap
Test	44,113	—
Baseline	741,833	9,067
Reparse	712,973	10,022

**Table 2:** Instance counts and instance overlap against test for the English Penn Treebank training set ( $n$  is the number of trees in  $n$ -best list used)

defined as:

$$|(A, f)| = \sum_{a \in A} f(a),$$

and multiset intersection as:

$$(A, f) \cap (B, g) = (A \cap B, a \mapsto \min(f(a), g(a))).$$

We use both the baseline method where examples are extracted from gold trees, and our improved training method to obtain training examples from sections 2-21 of the Wall Street Journal part of the English Penn Treebank and compare both against instances extracted from the parsed sentences taken from section 23. For parsing the test sentences and the training sentences we used the Charniak parser.

Table 2 summarizes the comparison. Even though our method produces a lower total number of instances than the baseline (since we only extract instances from correctly spanning nodes) it still shares 955 instances more with the test set than the baseline.

To further test our conjecture about our method giving better training examples we calculated mean Hamming distance between training examples and test examples. Hamming distance counts the number of features at which two vectors differ:

$$d^h(\mathbf{v}, \mathbf{w}) = \sum_{i=1}^{|\mathbf{v}|} \mathbf{v}_i \neq \mathbf{w}_i. \quad (1)$$

We calculate the mean distance between the collection of test instances  $\mathbf{T}$  and the collection of training instances  $\mathbf{U}$  as:

$$\bar{d}^h(\mathbf{T}, \mathbf{U}) = \frac{1}{|\mathbf{T}| \times |\mathbf{U}|} \sum_{\mathbf{t} \in \mathbf{T}} \sum_{\mathbf{u} \in \mathbf{U}} d^h(\mathbf{t}, \mathbf{u}). \quad (2)$$

As shown in Table 3, against the test set derived from section 23 of WSJ we get mean Hamming distance of 15.1483 for the baseline method and 15.1283 for our method (for comparison the mean distance of the test set against itself is 15.099). According to this metric examples obtained by our method are more similar to test examples.

	Mean distance to Test
Test	15.0999
Baseline	15.1483
Reparse	15.1283

**Table 3:** Mean Hamming distance scores for the English Penn Treebank training set

## 4 Experimental Results

In this section we present evaluation results on the function labeling task for two datasets:

- Section 23 of the WSJ portion of the English Penn II Treebank, with models trained on data extracted from sections 2-21. Section 22 was used for development. The Charniak parser<sup>2</sup> was used for constituency parsing.
- Articles 271 to 300 of the Penn Chinese Treebank 5, with models trained on data extracted from articles 26 to 270. Articles 1-25 were used for development. We follow (Levy and Manning, 2003) in adopting this test/training/development split. The Bikel parser<sup>3</sup> was used for constituency parsing.

For both datasets we used the LIBSVM library (Chang and Lin, 2001) which implements the Support Vector Machines algorithm (Vapnik, 1998).

### 4.1 Experiments with the English Penn Treebank

Table 4 summarizes evaluation results for the function labeling task on the English Penn II Treebank. There is a clear increase in f-score over the baseline for our method, which gives a relative error reduction of almost 8.5% over the baseline. The approximate randomization test (Noreen, 1989) with  $10^6$  shuffles obtained a  $p$ -value of  $10^{-7}$  for the baseline versus our method, showing that the improvement is statistically significant.

Our results (91.47% f-score) are the best scores published to date on the function labeling task evaluated on parser output on the section 23 of WSJ: 87.27% in (Blaheta and Charniak, 2000), 88.5% in (Jijkoun and de Rijke, 2004) and 88.96% in (Gabbard et al., 2006)<sup>4</sup>

Table 5 shows the performance broken down per function label. Although performance on three labels (LOC, LGS and PRP) drops, the rest of the labels show the same score or benefit from our training method.

### 4.2 Experiments with the Penn Chinese Treebank

For the Chinese Treebank we performed experiments evaluating the impact of using our basic method and

<sup>2</sup> Available at <ftp://ftp.cs.brown.edu/pub/nlparser/>

<sup>3</sup> Available at <http://www.cis.upenn.edu/~dbikel/software.html#stat-parser>

<sup>4</sup> Not all of those scores are exactly comparable to ours or to each other. The score in (Jijkoun and de Rijke, 2004) is on trees transformed into dependencies. Gabbard et al. (2006) use Bikel’s parser to produce the trees whereas we use Charniak’s.

	Precision	Recall	F-score
Baseline	92.28	89.14	90.68
Reparse	93.07	89.92	91.47

**Table 4:** Function labeling evaluation on parser output for WSJ section 23

Label	Freq. in test	Baseline	Reparse
SBJ	4148	98.27	98.27
TMP	1303	91.19	91.52
PRD	1025	68.35	91.26
LOC	1024	89.45	89.06
CLR	635	68.98	68.93
ADV	419	85.98	89.36
DIR	293	68.98	71.20
TPC	267	86.50	96.02
PRP	207	68.35	67.95
NOM	199	95.02	95.58
MNR	178	76.12	77.62
LGS	166	88.10	88.10
EXT	105	87.72	88.24
TTL	61	74.42	74.42
HLN	52	18.18	26.23
DTV	19	66.67	66.67
PUT	10	66.67	66.67
CLF	3	—	—
BNF	2	—	—
VOC	1	—	—

**Table 5:** Per-tag performance of baseline and when training on reparsed trees

also the variation with cross-training on the function labeling task.

The results we obtained are somewhat contradictory: we saw an improvement in performance using both on the development set (articles 1-25), but on the test set (articles 271-300) the basic method shows practically no improvement whereas cross-training actually leads to results worse than for the baseline.

Table 6 shows the results for the development set which are consistent with our findings so far: our method outperforms the baseline by 0.18%. Additionally, we observe that adding cross-training produces a further increase in the f-score of 0.3%.

However, as can be seen in Table 7, for the test set our predictions are not borne out: with cross-training we actually obtain a lower score than the baseline (−0.32%); without cross-training the score is only marginally better than the baseline (+0.01%).

We performed an approximate randomization test for both the development set and the set, testing the baseline against our method with cross-training. For the development set we obtained a  $p$ -value of 0.13; for the test set the  $p$ -value was 0.08 — this suggests that neither the improvement for the development set nor the decrease in f-score for the test set are statistically significant.

It would be interesting to repeat our experiments for Chinese using larger data sets. There are two reasons why we want to do that. First, testing on a larger test set would offer a higher confidence in the significance of the observed performance scores. Second, we suspect that one reason that our approach did not

	Precision	Recall	F-score
Baseline	88.35	84.64	86.46
Reparse	88.54	84.82	86.64
Reparse + x-train	89.11	84.88	86.94

**Table 6:** Function labeling evaluation for the CTB on the parser output for the development set

	Precision	Recall	F-score
Baseline	91.46	90.13	90.79
Reparse	91.39	90.23	90.80
Reparse + x-train	91.53	89.43	90.47

**Table 7:** Function labeling evaluation for the CTB on the parser output for the test set

show consistent improvement across both the development set and the test set might be related to the relatively small amount of training material we used, for both training the parser and the function labeling model. Thus parse quality is rather low, and since we only exploit correctly parsed nodes in extracting training instances for labeling, the amount of training data available decreases even further. We also suspect that parse quality for Chinese may be lower than for English even while holding training set size constant, reflecting the smaller amount of work which has gone into research on parsing Chinese. Testing those conjectures remains an area for future investigation.

It remains to be seen whether using our approach with training sets comparable in size with the one we used for English would more show more consistent benefits for Chinese.

## 5 Conclusions and Future Work

We have presented a method to perform training in a sound manner in the two-stage function labeling model and investigated the impact of our proposal on the function labeling task. Our approach improves the similarity of the training material to the test instances as measured by instance overlap and mean Hamming distance.

We have consistently found substantial statistically significant improvements on the English Penn Treebank data, and a more mixed picture for the Chinese Penn Treebank sentences. We would like to better understand what factors influence the effect of our proposed training methods on function labeling performance: we plan to study this issue in more detail in future.

It should also be possible to apply our findings to other tasks where training examples are typically extracted from perfect trees whereas the test data is produced automatically and contains errors. Using parser output instead and exploiting several of the most probable trees could be beneficial in those situations.

## Acknowledgements

We gratefully acknowledge support from Science Foundation Ireland grant 04/IN/I527 for the research reported in this paper.

## References

- Bies, Ann (1995). Bracketing guidelines for Treebank II style Penn treebank project. Technical report, University of Pennsylvania.
- Bikel, Dan (2002). Design of a multi-lingual, parallel-processing statistical parsing engine. In *Human Language Technology Conference (HLT)*. San Diego, CA, USA.
- Blaheta, Don and Charniak, Eugene (2000). Assigning function tags to parsed text. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 234–240. San Francisco, CA, USA.
- Chang, Chih-Chung and Lin, Chih-Jen (2001). LIBSVM: a library for Support Vector Machines (version 2.31).
- Charniak, Eugene (2000). A maximum-entropy-inspired parser. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 132–139. San Francisco, CA, USA.
- Chrupala, Grzegorz and van Genabith, Josef (2006). Using machine-learning to assign function labels to parser output for Spanish. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 136–143. Sydney, Australia.
- Civit, Montserrat and Martí, Maria Antonia (2004). Building Cast3LB: A Spanish treebank. *Research on Language and Computation*, 2(4):549–574.
- Collins, Michael (1999). *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.
- Gabbard, Ryan, Kulick, Seth, and Marcus, Mitchell (2006). Fully parsing the Penn treebank. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 184–191. New York City, USA.
- Henderson, James (2003). Inducing history representations for broad coverage statistical parsing. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 24–31. Morristown, NJ, USA.
- Jijkoun, Valentin and de Rijke, Maarten (2004). Enriching the output of a parser using memory-based learning. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 311–318. Barcelona, Spain.
- Levy, Roger and Manning, Christopher (2003). Is it harder to parse Chinese, or the Chinese treebank? In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 439–446. Morristown, NJ, USA.
- Musillo, Gabriele and Merlo, Paola (2005). Lexical and structural biases for function parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 83–92. Vancouver, British Columbia.
- Noreen, Eric W. (1989). *Computer intensive methods for testing hypotheses*. A Wiley-Interscience Publication, New York.
- Vapnik, Vladimir N. (1998). *Statistical Learning Theory*. Wiley-Interscience.
- Xue, Nianwen and Xia, Fei (2000). The bracketing guidelines for the Penn Chinese treebank. Technical report, University of Pennsylvania.

# Evaluating large-scale Knowledge Resources across Languages

Montse Cuadros  
TALP Research Center  
Universitat Politècnica de Catalunya  
Barcelona, Spain  
cuadros@lsi.upc.edu

German Rigau  
IXA NLP Group  
Euskal Herriko Unibersitatea  
Donostia, Spain  
german.rigau@ehu.es

Mauro Castillo  
Departamento de Computación e Informática  
Universidad Tecnológica Metropolitana  
Santiago de Chile, Chile  
mcast@utem.cl

## Abstract

This paper presents an empirical evaluation in a multilingual scenario of the semantic knowledge present on publicly available large-scale knowledge resources. The study covers a wide range of manually and automatically derived large-scale knowledge resources for English and Spanish. In order to establish a fair and neutral comparison, the knowledge resources are evaluated using the same method on two Word Sense Disambiguation tasks (Senseval-3 English and Spanish Lexical Sample Tasks). First, this study empirically demonstrates that the combination of the knowledge contained in these resources surpasses the most frequent sense classifier for English. Second, we also show that this large-scale topical knowledge acquired from one language can be successfully ported to other languages.

## Keywords

Large-scale knowledge resources, lexical semantics, evaluation, WordNet, Word Sense Disambiguation

## 1 Introduction

Using large-scale knowledge bases, such as WordNet (WN) [9], has become a usual, often necessary, practice for most current Natural Language Processing (NLP) systems. Even now, building large and rich enough knowledge bases for broad-coverage semantic processing takes a great deal of expensive manual effort involving large research groups during long periods of development. In fact, hundreds of person-years have been invested in the development of wordnets for various languages [21]. For example, in more than ten years of manual construction (from 1995 to 2006, that is from version 1.5 to 3.0), WN passed from 103,445 semantic relations to 235,402 semantic relations<sup>1</sup>. That is, around one thousand new relations per month. But this data does not seem to be rich enough to support advanced concept-based NLP applications directly. It seems that applications will not scale up to working in

<sup>1</sup> Symmetric relations are counted only once.

open domains without more detailed and rich general-purpose (and also domain-specific) semantic knowledge built by automatic means.

Fortunately, during the last years the research community has devised a large set of innovative methods and tools for large-scale automatic acquisition of lexical knowledge from structured and unstructured corpora. Among others we can mention eXtended WN [17], large collections of semantic preferences acquired from SemCor [2, 3] or acquired from British National Corpus (BNC) [15], large-scale Topic Signatures for each synset acquired from the web [1] or acquired from the BNC [6]. Obviously, all these semantic resources have been acquired using a very different set of processes, tools and corpora, resulting on a different set of new semantic relations between synsets. In fact, each semantic resource has different volume and accuracy figures when evaluated in a common and controlled framework [7]. However, as far as we know, no empirical study has been carried out trying to see how these semantic resources complement each other.

Furthermore, since this knowledge is language independent (knowledge represented at the semantic level as relations between synsets), to date no empirical evaluation has been performed showing to which extent these large-scale semantic resources acquired from one language (in this case English) could be of utility for another (in this case Spanish).

This paper is organized as follows. First, we introduce the multilingual semantic resources compared in the evaluation. In section 3 we present the multilingual evaluation framework used in this study. Section 4 describes the results when evaluating these large-scale semantic resources on English and section 5 on Spanish. Finally, section 6 presents some concluding remarks and future work.

## 2 Multilingual Knowledge Resources

Our evaluation covers a wide range of large-scale semantic resources: WordNet (WN) [9], eXtended WordNet [17], large collections of semantic preferences ac-



Source	#relations
Princeton WN1.6	138,091
Selectional Preferences from SemCor	203,546
New relations from Princeton WN2.0	42,212
Gold relations from eXtended WN	17,185
Silver relations from eXtended WN	239,249
Normal relations from eXtended WN	294,488
<b>Total English</b>	<b>934,771</b>
<b>Total Spanish</b>	<b>517,279</b>

**Table 1:** *Semantic relations uploaded in the MCR*

quired from SemCor [2, 3] or acquired from the BNC [15], large-scale Topic Signatures for each synset acquired from the web [1] or SemCor [11].

Although these resources have been derived using different WN versions, using the technology for the automatic alignment of wordnets [8], most of these resources have been integrated in a common resource called Multilingual Central Repository (MCR) [4] maintaining the compatibility among all the knowledge resources which use a particular WN version as a sense repository. Furthermore, these mappings allow to port the knowledge associated to a particular WN version to the rest of WN versions.

## 2.1 Multilingual Central Repository

The Multilingual Central Repository (MCR) [4] is a result of the 5th Framework MEANING project<sup>2</sup>. The MCR follows the model proposed by the EuroWordNet project. EuroWordNet [21] is a multilingual lexical database with wordnets for several European languages, which are structured as the Princeton WN.

The MCR constitutes a natural multilingual large-scale linguistic resource for a number of semantic processes that need large amounts of multilingual knowledge to be effective tools. The MCR also integrates WN Domains [14], new versions of the Base Concepts and the Top Concept Ontology, and the SUMO ontology [20]. The current version of the MCR contains 934,771 semantic relations between synsets, most of them acquired by automatic means. This represents almost four times larger than the Princeton WN (235,402 unique semantic relations in WN 3.0). Table 1 shows the number of semantic relations between synset pairs in the MCR. As the current version of the Spanish Wordnet do not have translation equivalents for all the English synsets<sup>3</sup>, the total number of ported relations is around a half of the English ones.

Hereinafter we will refer to each resource as follows:

**WN** [9]: This resource uses the direct relations encoded in WN1.6 and WN2.0 (for instance, *tree#n#1-hyponym->teak#n#2*). We also tested WN<sup>2</sup> (using relations at distance 1 and 2), WN<sup>3</sup> (using relations at distances 1 to 3) and WN<sup>4</sup> (using relations at distances 1 to 4).

**XWN** [17]: This resource uses the direct relations encoded in eXtended WN (for instance, *teak#n#2-gloss->wood#n#1*).

<sup>2</sup> <http://nipadio.lsi.upc.es/~nlp/meaning>

<sup>3</sup> Currently, the Spanish WN has translation equivalents to English for 62,720 synsets.

<i>political_party#n#1</i>	2.3219
<i>party#n#1</i>	2.3219
<i>election#n#1</i>	1.0926
<i>nominee#n#1</i>	0.4780
<i>candidate#n#1</i>	0.4780
<i>campaigner#n#1</i>	0.4780

**Table 2:** *Topic Signatures for party#n#1 obtained from Semcor (6 out of 719 total word senses)*

**WN+XWN:** This resource uses the direct relations included in WN and XWN. We also tested (WN+XWN)<sup>2</sup> (using either WN or XWN relations at distances 1 and 2, for instance, *tree#n#1-related->wood#n#1*).

**spBNC** [15]: This resource contains 707,618 selectional preferences acquired for subjects and objects from BNC.

**spSemCor** [3]: This resource contains the selectional preferences acquired for subjects and objects from SemCor (for instance, *read#v#1-tobj->book#n#1*).

**MCR** [4]: This resource uses the direct relations included in MCR but in the experiments below we excluded spBNC because of its poor performance. Thus, MCR contains the direct relations from WN, XWN, and spSemCor but not the indirect relations of (WN+XWN)<sup>2</sup>. We also tested MCR<sup>2</sup> (using relations at distance 1 and 2), which also integrates (WN+XWN)<sup>2</sup> relations.

## 2.2 Topic Signatures

Topic Signatures (TS) are word vectors related to a particular topic [13].

For this study, we use two different large-scale TS. The first constitutes one of the largest available semantic resource with around 100 million relations (between synsets and words) acquired from the web [1]. The second has been derived directly from SemCor.

**TSWEB**<sup>4</sup>: Inspired by the work of [12], these TS were constructed using monosemous relatives from WN (synonyms, hypernyms, direct and indirect hyponyms, and siblings), querying Google and retrieving up to one thousand snippets per query (that is, a word sense), extracting the words with distinctive frequency using TFIDF. For these experiments, we used at maximum the first 700 words.

Since this is a semantic resource between word-senses and words, it is not possible to port these relations to Spanish without introducing a large amount of noise.

**TSSEM:** These TS have been constructed using the part of SemCor having all words tagged by PoS, lemmatized and sense tagged according to WN1.6 totalizing 192,639 words. For each word-sense appearing in SemCor, we gather all sentences for that word sense, building a TS using TFIDF for all word-senses co-occurring in those sentences.

In table 2, there is an example of the first word-senses we calculate from *party#n#1*.

<sup>4</sup> <http://ixa.si.ehu.es/Ixa/resources/sensecorpus>

The total number of relations between WN synsets acquired from SemCor is 932,008. In this case, due to the smaller size of the Spanish WN, the total number of ported relations is 586,881.

### 3 Evaluation framework

In order to compare the knowledge resources described in the previous section, we evaluated all these resources as Topic Signatures (TS). This simple representation tries to be as neutral as possible with respect to the resources used.

All knowledge resources are evaluated on a WSD task. In particular, in section 4 we used the noun-set of Senseval-3 English Lexical Sample task which consists of 20 nouns and in section 5 we used the noun-set of the Senseval-3 Spanish Lexical Sample task which consists of 21 nouns. For Spanish, the MiniDir dictionary was specially developed for the Senseval-3 task. Most of the MiniDir word senses have links to WN1.5 (which in turn are linked by the MCR to the Spanish WN). All performances are evaluated on the test data using the fine-grained scoring system provided by the organizers. We use the noun-set only because TSWEB is available only for nouns, and the English Lexical Sample uses the WordSmyth dictionary [18] as a sense repository for verbs instead of WN.

Furthermore, trying to be as neutral as possible with respect to the resources studied, we applied systematically the same disambiguation method to all of them. Recall that our main goal is to establish a fair comparison of the knowledge resources rather than providing the best disambiguation technique for a particular knowledge base.

A common WSD method has been applied to all knowledge resources. A simple word overlapping counting is performed between the TS and the test example<sup>5</sup>. The synset having higher overlapping word counts is selected. In fact, this is a very simple WSD method which only considers the topical information around the word to be disambiguated. Finally, we should remark that the results are not skewed (for instance, for resolving ties) by the most frequent sense in WN or any other statistically predicted knowledge.

## 4 English evaluation

### 4.1 Baselines for English

We have designed a number of basic baselines in order to establish a complete evaluation framework for comparing the performance of each semantic resource on the English WSD task.

**RANDOM:** For each target word, this method selects a random sense. This baseline can be considered as a lower-bound.

**SemCor MFS (SEMCOR-MFS):** This method selects the most frequent sense of the target word in SemCor.

**WordNet MFS (WN-MFS):** This method selects the most frequent sense (the first sense in WN1.6) of

Baselines	P	R	F1
TRAIN	65.1	65.1	65.1
TRAIN-MFS	54.5	54.5	54.5
WN-MFS	53.0	53.0	53.0
SEMCOR-MFS	49.0	49.1	49.0
RANDOM	19.1	19.1	19.1

**Table 3:** *P, R and F1 results for English Lexical Sample Baselines*

the target word. WN word-senses were ranked using SemCor and other sense-annotated corpora. Thus, WN-MFS and SemCor-MFS are similar, but not equal.

**TRAIN-MFS:** This method selects the most frequent sense in the training corpus of the target word.

**Train Topic Signatures (TRAIN):** This baseline uses the training corpus to directly build a TS using TFIDF measure for each word sense. Note that in WSD evaluation frameworks, this is a very basic system, a baseline. However, in our evaluation framework, this "WSD baseline" should be considered as an upper-bound.

Table 3 presents the precision (P), recall (R) and F1 measure (harmonic mean of recall and precision) of the different baselines. In this table, TRAIN has been calculated with a vector size of at maximum 450 words. As expected, RANDOM baseline obtains the poorest result. The most frequent senses obtained from SemCor (SEMCOR-MFS) and WN (WN-MFS) are both below the most frequent sense of the training corpus (TRAIN-MFS). However, all of them are far below to the TS acquired using the training corpus (TRAIN).

### 4.2 Evaluating each resource on English

Table 4 presents ordered by F1 measure, the performance of each knowledge resource and its average size of the Topic Signature per word-sense. The average size of a knowledge resource is the length of the word list associated to a synset on average. Obviously, the best resources would be those obtaining better performances with a smaller number of related words per synset. The best results for precision, recall and F1 measures are shown in bold. We also mark in italics those derived resources applying non-direct relations. Surprisingly, the best results are obtained by TSSEM (with F1 of 52.4). The lowest result is obtained by the knowledge directly gathered from WN mainly because of its poor coverage (recall of 18.4 and F1 of 26.1). Also interesting, is that the knowledge integrated in the MCR although partly derived by automatic means performs much better in terms of precision, recall and F1 measures than using them separately (F1 with 18.4 points higher than WN, 9.1 than XWN and 3.7 than spSemCor).

Despite its small size, the resources derived from SemCor obtain better results than its counterparts using much larger corpora (TSSEM vs. TSWEB and spSemCor vs. spBNC).

Regarding the baselines, all knowledge resources surpass RANDOM, but none achieves neither WN-MFS, TRAIN-MFS nor TRAIN. Only TSSEM obtains

<sup>5</sup> We also consider multiword terms.

KB	P	R	F1	Av. Size
TSSEM	<b>52.5</b>	<b>52.4</b>	<b>52.4</b>	103
<i>MCR</i> <sup>2</sup>	45.1	45.1	45.1	26,429
MCR	45.3	43.7	44.5	129
spSemCor	43.1	38.7	40.8	56
<i>(WN+XWN)</i> <sup>2</sup>	38.5	38.0	38.3	5,730
<i>WN+XWN</i>	40.0	34.2	36.8	74
TSWEB	36.1	35.9	36.0	1,721
XWN	38.8	32.5	35.4	69
<i>WN</i> <sup>3</sup>	35.0	34.7	34.8	503
<i>WN</i> <sup>4</sup>	33.2	33.1	33.2	2,346
<i>WN</i> <sup>2</sup>	33.1	27.5	30.0	105
spBNC	36.3	25.4	29.9	128
WN	44.9	18.4	26.1	14

**Table 4:** *P*, *R* and *F1* fine-grained results for the resources evaluated individually on English.

better results than SEMCOR-MFS and is very close to the most frequent sense of WN (WN-MFS) and the training (TRAIN-MFS).

Regarding other expansions and combinations, the performance of WN is improved using words at distances up to 2 (F1 of 30.0), and up to 3 (F1 of 34.8), but it decreases using distances up to 4 (F1 of 33.2). Interestingly, none of these WN expansions achieve the results of XWN (F1 of 35.4). Finally,  $(WN+XWN)^2$  performs better than  $WN+XWN$  and  $MCR^2$  slightly better than  $MCR^6$ .

### 4.3 Combining resources

In order to evaluate more deeply the contribution of each knowledge resource, we also provide some results of the combined outcomes of several resources. The combinations are performed following three different basic strategies [5].

**Direct Voting (DV):** Each semantic resource has one vote for the predominant sense of the word to be disambiguated and the sense with most votes is chosen.

**Probability Mixture (PM):** Each semantic resource provides a probability distribution over the senses of the word to be disambiguated. These probabilities (normalized scores) are summed, and the sense with the highest score is chosen.

**Rank-Based Combination (Rank):** Each semantic resource provides a ranking of senses of the word to be disambiguated. For each sense, its placements according to each of the methods are summed and the sense with the lowest total placement (closest to first place) is selected.

#### 4.3.1 Combining two resources

Table 5 presents the F1 measures with respect these three methods when combining two different resources. The combinations are ordered by the result of the rank-based combination. The best result which corresponds to the rank-based combination of MCR and TSSEM<sup>7</sup> is shown in bold.

<sup>6</sup> No further distances have been tested.

<sup>7</sup> Note that in this case, some information appearing in SemCor could be counted twice, as we are not removing duplicated relations

KB	PM	DV	Rank
MCR+TSSEM	52.3	45.4	<b>52.7</b>
$MCR+(WN+XWN)^2$	47.8	37.8	51.5
$(WN+XWN)^2+TSSEM$	51.0	41.7	50.5
TSSEM+TSWEB	51.0	42.2	49.4
MCR+TSWEB	48.9	37.6	48.6
$(WN+XWN)^2+TSWEB$	41.5	34.3	45.4

**Table 5:** *F1* fine-grained results for the 2 system-combinations

KB	PM	DV	Rank
$MCR+TSSEM+(WN+XWN)^2$	52.6	37.9	<b>54.6</b>
MCR+TSWEB+TSSEM	54.1	37.2	53.3
$MCR+TSWEB+(WN+XWN)^2$	49.8	33.3	52.1
$(WN+XWN)^2+TSSEM+TSWEB$	51.5	36.1	51.5

**Table 6:** *F1* fine-grained results for the 3 system-combinations

Regarding the combination method applied, the probability-mixture and the rank-based methods behave similarly (each method wins in three of the six combinations), and obtaining better results than the direct-voting method. Hereinafter, we use the rank-based measure for comparing results.

Interestingly, only in two cases the ensemble of resources makes worse the individual results. Both cases involve TSSEM (F1 of 52.4) when combined with TSWEB (F1 of 49.4) and  $(WN+XWN)^2$  (F1 of 50.5). However, for the rest of the cases, it seems that each resource provides some kind of knowledge not provided by the others. For instance, the knowledge contained in  $(WN+XWN)^2$  seems to be not represented in the MCR. Furthermore, despite  $(WN+XWN)^2+TSWEB$  obtains the lower results (F1 of 45.4) when combining two resources, the individual contribution to the ensemble is impressive (5.4 points with respect  $(WN+XWN)^2$  and (9.4 points with respect to TSWEB). However, the larger increment corresponds to  $MCR+(WN+XWN)^2$  (F1 of 51.5, 6.0 points higher than MCR and 13.25 higher than  $(WN+XWN)^2$ ), indicating that both resources contain complementary knowledge. In fact, there is some knowledge contained in the MCR not present in TSSEM (because the small increment of 0.3 points with respect TSSEM alone).

Regarding the baselines, none of the combinations achieves the most frequent sense of WN (WN-MFS with F1 of 53.0). However, several of them surpass the most frequent sense of SemCor (SEMCOR-MFS with F1 of 49.1). In particular, the combinations including information from SemCor (TSSEM or MCR).

#### 4.3.2 Combining three resources

Table 6 presents the F1 measure results with respect these three methods when combining three different semantic resources. The combinations are ordered by the result of the rank-based combination. The best result which corresponds to the rank-based combination of MCR  $(WN+XWN+spSemCor)$ , TSSEM and  $(WN+XWN)^2$  is presented in bold.

KB	PM	DV	Rank
MCR+(WN+XWN) <sup>2</sup> +TSWEB+TSSEM	53.1	32.7	<b>55.5</b>

**Table 7:** *F1 fine-grained results for the 4 system-combinations*

Regarding the combination method applied, the rank-based method seems to be similar to probability-mixture (winning in two of the four combinations, losing in one and having a tie in one). Again, both strategies are superior to the direct-voting method.

Considering only the rank-based combination, in general, the combination of three knowledge resources obtains slightly better results than using only two or one resource. In this case, only one ensemble of resources makes worse the individual results. This case involves again TSSEM (F1 of 52.4) when combined with (WN+XWN)<sup>2</sup>+TSWEB (F1 of 45.4). However, for the rest of the cases, again it seems that the combination of resources integrates some knowledge not provided by the resources individually. In this case, the larger increase corresponds to MCR+TSWEB+(WN+XWN)<sup>2</sup> (F1 of 52.1, 16.1 points higher than TSWEB, 12.1 points higher than (WN+XWN)<sup>2</sup>, and 7.6 points higher than MCR). Furthermore, there is some knowledge contained in the MCR+(WN+XWN)<sup>2</sup> not present in TSSEM (because an small increment of 2.2 points with respect TSSEM alone).

In fact, all these combinations outperform the most frequent sense of SemCor (F1 of 49.1), and two combinations of three resources surpass the most frequent sense of WN (WN-MFS with F1 of 53.0): MCR+TSWEB+TSSEM (F1 of 53.3) and MCR+TSSEM+(WN+XWN)<sup>2</sup> (F1 of 54.6), and the later is also slightly over the most frequent sense of the training (F1 of 54.5). Obviously, this result should be highlighted since in the all-words tasks most current supervised approaches rarely surpass the simple heuristic of choosing the most frequent sense in the training data, despite taking local context into account [10].

#### 4.3.3 Combining four resources

Table 7 presents the F1 measure results with respect the three methods when combining the four different semantic resources. In bold is presented the best result which corresponds to the rank-based combination of MCR, TSSEM, TSWEB and (WN+XWN)<sup>2</sup>.

It seems that the rank-based has better behavior than direct-voting or probability-mixture methods.

Considering only the rank-based combination, as expected, the combination of the four knowledge resources obtains better results than using only three, two or one resource. Again, it seems that the combination of resources provides some kind of knowledge not provided by each of the resources individually. In this case, 19.5 points higher than TSWEB, 17.25 points higher than (WN+XWN)<sup>2</sup>, 11.0 points higher than MCR and 3.1 points higher than TSSEM.

Regarding the baselines, this combination outperforms the most frequent sense of SemCor (SEMCOR-

Baselines	P	R	F1
TRAIN	81.8	68.0	74.3
MiniDir-MFS	67.1	52.7	59.2
RANDOM	21.3	21.3	21.3

**Table 8:** *P, R and F1 fine-grained results for Spanish Lexical Sample Baselines*

MFS with F1 of 49.1), WN (WN-MFS with F1 of 53.0) and, the training data (TRAIN-MFS with F1 of 54.5). This fact indicates that the resulting combination of large-scale resources encodes the knowledge necessary to behave as a most frequent sense tagger for English. Furthermore, it is also worth mentioning that the most frequent synset for a word, according to the WN sense ranking is very competitive in WSD tasks, and it is extremely hard to improve upon even slightly [16].

## 5 Spanish evaluation

### 5.1 Spanish Baselines

As well as for English, we have designed a number of basic baselines in order to establish a complete evaluation framework for comparing the performance of each semantic resource when evaluated on the Spanish WSD task.

**RANDOM:** For each target word, this method selects a random sense. This baseline can be considered as a lower-bound.

**Minidir MFS (Minidir-MFS):** This method selects the most frequent sense (the first sense in Minidir) of the target word. Since Minidir is a special dictionary built for the task, the word-sense ordering corresponds to their frequency in the training data. Thus, for Spanish, Minidir-MFS is equal to TRAIN-MFS.

**Train Topic Signatures (TRAIN):** This baseline uses the training corpus to directly build a Topic Signature using TFIDF measure for each word sense. As for English, this baseline can be considered as an upper-bound of our evaluation.

Note that the Spanish WN do not encodes word-sense frequency information and for Spanish there is no all-words sense tagged corpora available of the style of Italian<sup>8</sup>.

In the Spanish evaluation only sense-disambiguated relations can be ported without introducing extra noise. For instance, TSWEB has not been tested on the Spanish side. TSWEB relate synsets to words, not synsets to synsets. As this resource is not word-sense disambiguated, when translating the English words to Spanish, a large amount of noise would be introduced (Spanish words not related to the particular synset).

Table 8 presents the precision (P), recall (R) and F1 measure of the different baselines. As for English, TRAIN has been calculated with a vector size of at maximum 450 words. As expected, RANDOM baseline obtains the poorest result and the most frequent sense obtained from Minidir (Minidir-MFS, and also TRAIN-MFS) is far below the TS acquired using the training corpus (TRAIN).

<sup>8</sup> <http://multisemcor.itc.it/>

Knowledge Bases	P	R	F1	Av. Size
MCR	46.1	<b>41.1</b>	<b>43.5</b>	66
WN <sup>2</sup>	56.0	29.0	42.5	51
(WN+XWN) <sup>2</sup>	41.3	41.2	41.3	1,892
TSSEM	33.6	33.2	33.4	208
XWN	42.6	27.1	33.1	24
WN	<b>65.5</b>	13.6	22.5	8

**Table 9:** *P*, *R* and *F1* fine-grained results for the resources evaluated individually on Spanish.

## 5.2 Evaluating each resource on Spanish

Table 9 presents ordered by F1 measure, the performance of the knowledge resources and its average size per word-sense. In bold appear the best results for precision, recall and F1 measures. WN obtains the highest precision (P of 65.5) but due to its poor coverage (R of 13.6), the lowest result (F1 of 22.5). Also interesting, is that the knowledge integrated in the MCR outperforms in terms of precision, recall and F1 measures the results of TSSEM, possibly indicating that the knowledge currently uploaded in the MCR is more robust than TSSEM and that the topical knowledge gathered from a sense-annotated corpus of one language can not be directly ported to another language. Possible explanations of these low results could be the smaller size of the resources (approximately a half size) and the differences in the evaluation frameworks, including the dictionary, sense distinctions and mappings.

Regarding the baselines, all knowledge resources surpass RANDOM, but none achieves neither Minidir-MFS (equal to TRAIN-MFS) nor TRAIN.

## 6 Conclusions and further work

To our knowledge, this is the first time to show that a very simple WSD system based on topical knowledge gathered from several semantic resources outperforms the Most Frequent Sense classifiers in the SensEval-3 English lexical-sample task. Obviously, more sophisticated approaches could be devised [19]. Furthermore, since these resources represent semantic relations at the conceptual level, can be also successfully ported to and evaluated in other languages.

It is our belief, that accurate WSD systems would rely not only on sophisticated algorithms but on knowledge intensive approaches. The results presented in this paper suggests that much more research on acquiring and using large-scale semantic resources should be addressed.

It seems that the combination of publicly available large-scale resources encodes the knowledge necessary to behave as a most frequent sense tagger for English. We plan to empirically validate this hypothesis in all-words tasks.

Further experiments in the cross-lingual scenario are also needed to clarify the different behaviours of the MCR and TSSEM, maybe using the Italian WN (also integrated in the MCR) and MultiSemCor.

## 7 Acknowledgements

We want to thank the valuable comments of the anonymous reviewers. This work has been partially supported by the projects KNOW (TIN2006-15049-C03-01) and ADIMEN (EHU06/113).

## References

- [1] E. Agirre and O. L. de Lacalle. Publicly available topic signatures for all wordnet nominal senses. In *Proceedings of LREC*, Lisbon, Portugal, 2004.
- [2] E. Agirre and D. Martinez. Learning class-to-class selectional preferences. In *Proceedings of CoNLL*, Toulouse, France, 2001.
- [3] E. Agirre and D. Martinez. Integrating selectional preferences in wordnet. In *Proceedings of GWC*, Mysore, India, 2002.
- [4] J. Atserias, L. Villarejo, G. Rigau, E. Agirre, J. Carroll, B. Magnini, and P. Vossen. The meaning multilingual central repository. In *Proceedings of GWC*, Brno, Czech Republic, 2004.
- [5] S. Brody, R. Navigli, and M. Lapata. Ensemble methods for unsupervised wsd. In *Proceedings of COLING-ACL*, pages 97–104, 2006.
- [6] M. Cuadros, L. Padró, and G. Rigau. Comparing methods for automatic acquisition of topic signatures. In *Proceedings of RANLP*, Borovets, Bulgaria, 2005.
- [7] M. Cuadros and G. Rigau. Quality assessment of large scale knowledge resources. In *Proceedings of EMNLP*, 2006.
- [8] J. Daudé, L. Padró, and G. Rigau. Validation and Tuning of Wordnet Mapping Techniques. In *Proceedings of RANLP*, Borovets, Bulgaria, 2003.
- [9] C. Fellbaum, editor. *WordNet. An Electronic Lexical Database*. The MIT Press, 1998.
- [10] V. Hoste, W. Daelemans, I. Hendrickx, and A. van den Bosch. Evaluating the results of a memory-based word-expert approach to unrestricted word sense disambiguation. In *Proceedings of the Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, pages 95–101, 2002.
- [11] S. Landes, C. Leacock, and R. Tengi. Building a semantic concordance of english. In *WordNet: An electronic lexical database and some applications*. MIT Press, Cambridge, MA., 1998, pages 97–104, 2006.
- [12] C. Leacock, M. Chodorow, and G. Miller. Using Corpus Statistics and WordNet Relations for Sense Identification. *Computational Linguistics*, 24(1):147–166, 1998.
- [13] C. Lin and E. Hovy. The automated acquisition of topic signatures for text summarization. In *Proceedings of COLING*, 2000. Strasbourg, France.
- [14] B. Magnini and G. Cavaglia. Integrating subject field codes into wordnet. In *Proceedings of LREC*, Athens. Greece, 2000.
- [15] D. McCarthy. *Lexical Acquisition at the Syntax-Semantics Interface: Diathesis Alternations, Subcategorization Frames and Selectional Preferences*. PhD thesis, University of Sussex, 2001.
- [16] D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. Finding predominant senses in untagged text. In *Proceedings of ACL*, pages 280–297, 2004.
- [17] R. Mihalcea and D. Moldovan. extended wordnet: Progress report. In *Proceedings of NAACL Workshop on WordNet and Other Lexical Resources*, Pittsburgh, PA, 2001.
- [18] R. Mihalcea, T. Chloviski, and A. Killgariff. The senseval-3 english lexical sample task. In *Proceedings of ACL/SIGLEX Senseval-3*, Barcelona, 2004.
- [19] R. Navigli and P. Velardi. Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(7):1063–1074, 2005.
- [20] I. Niles and A. Pease. Towards a standard upper ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, pages 17–19. Chris Welty and Barry Smith, eds, 2001.
- [21] P. Vossen, editor. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, 1998.

# A Rule-Based Morphological Disambiguator for Turkish

Turhan Daybelge  
Department of Computer Engineering  
Bilkent University  
Bilkent 06800, Ankara, Turkey  
daybelge@cs.bilkent.edu.tr

Ilyas Cicekli  
Department of Computer Engineering  
Bilkent University  
Bilkent 06800, Ankara, Turkey  
ilyas@cs.bilkent.edu.tr

## Abstract

Part-of-speech (POS) tagging is the process of assigning each word of an input text into an appropriate morphological class. Automatic recognition of parts-of-speech is very important for high level NLP applications, since it would be usually infeasible to perform this task manually in practical systems. One approach to POS tagging uses morphological disambiguation which selects the most suitable morphological parse for each word from the set of parses that is assigned to that word by the morphological analyzer. Accurate POS tagging is not a simple task in general. It even becomes harder for agglutinative languages like Turkish; the number of morphological parses associated with each word in a text is usually much larger than that is for non-agglutinative languages such as English. This is due to the ambiguous nature of such languages. In this paper, we introduce an effective rule based morphological disambiguator for Turkish.

**Keywords:** part-of-speech tagging, morphological disambiguation.

## 1. Introduction

Part-of-speech (POS) tagging is the process of assigning each word of a given text into an appropriate lexical class (part of speech) such as noun, verb, adjective, etc. One approach to POS tagging is the reduction of the problem of tagging to more general morphological disambiguation problem. Once a suitable morphological parse is selected for each word from its possible morphological parses, it is trivial to detect lexical categories of words since this information is already contained in morphological parses.

In morphological disambiguation, the morphological analyzer produces all possible morphological parses for each word in the text, and a single morphological parse is tried to be selected from the set of parses assigned to that word. Unlike the ideal case, the morphological disambiguator sometimes may not select a single parse, and the selection of the best subset of parses can be aimed. Turkish part-of-speech tagger described in this paper is actually a morphological disambiguator that aims to select the best subset of the morphological parses if it cannot select a single morphological parse for a word.

Like many applications that deal with great amounts of data, it is infeasible to manually handle parts-of-speech tagging for NLP applications that require tagging of large corpora. Automatic recognition of parts-of-speech is very

important for high level NLP applications such as machine translation. Although 100% accurate POS tagging is not possible in practice, highly effective systems for English are available currently. Although the effective POS taggers are available for widely studied languages such as English, the effective POS taggers are not available for the most of the languages that got less attention, and Turkish is one of these languages. In this paper, we present an effective morphological disambiguator for Turkish. The developed Turkish morphological disambiguator is planned to be used as a part of an example-based machine translation system between English and Turkish [4,5]. The developed Turkish morphological disambiguator is also integrated with a graphical user-interface so that it can be used as a morphological annotator tool for Turkish texts.

Due to the inherent morphological level ambiguity of Turkish, POS tagging and morphological disambiguation in general are much more complicated processes for Turkish. Agglutinative nature of Turkish makes the number of morphological parses for each word larger when it is compared to English. The number of possible inflectional and derivational suffixes for Turkish nouns and verbs is much higher, and this leads to the more morphological level ambiguity in Turkish words. According to [7], about 80% of Turkish words have more than one morphological parse.

There can be many different reasons for the morphological ambiguities in a Turkish word. For example, the word “*kitabın*” has the following two possible morphological parses:

kitab+Noun+A3sg+P2sg+Nom    your book  
kitab+Noun+A3sg+Pnon+Gen    of the book

Here the ambiguity in the word “*kitab-ın*” is due to the phonetic similarity of the genitive suffix in the second parse and the second singular possessive suffix in the first parse. Both of them are realized as the suffix “*ın*” at surface level. Similarly, nouns with the accusative suffix and the third singular possessive suffix usually have the same surface form.

The finding the just POS tags of Turkish words will not be enough for the most NLP applications in Turkish. We have to find the actual intended morphological parse of the word. For this purpose, we have developed the Turkish morphological disambiguator presented in this paper. This morphological disambiguator tries to find the intended morphological parse of each word. If it cannot select the

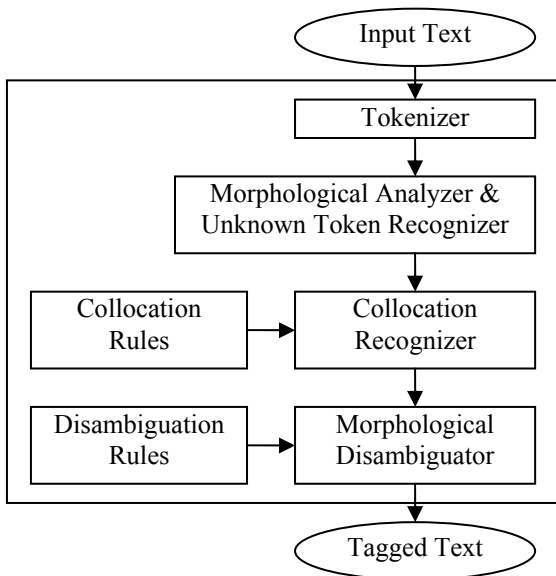


Figure 1. Architecture of the morphological disambiguator

intended morphological parse of the word, it tries to select the smallest subset of the morphological parses by eliminating some of the illegal parses.

The rest of the paper is organized as follows. Section 2 summarizes the related work in POS tagging including the previous works in Turkish POS tagging. We present the general architecture of our rule-based Turkish morphological disambiguator in Section 3. In Section 4, the performance of the presented Turkish morphological disambiguator is evaluated. We give the concluding remarks in Section 5.

## 2. Related Work on POS Tagging

There are two broad categories of POS tagging algorithms which are rule-based taggers and stochastic taggers. Rule based taggers contain a database of hand crafted rules that are designed to minimize ambiguity when applied in a certain order on each word in the text. Statistical POS taggers (also known as stochastic taggers), use a training corpus to calculate the likelihood of co-occurrence of all ordered pairs of tags. By training a probabilistic model such as HMM, the tagger tries to disambiguate any given new text. Since we do not have reliable morphological tagged huge corpus for Turkish, we have decided to develop a rule-based morphological disambiguator.

The earliest algorithms for automatic part-of-speech tagging were the rule-based ones. The tagger TAGGIT that was an aid in tagging the famous Brown Corpus was a rule-based one [6]. Stochastic techniques have proven to be more successful compared to pure rule-based ones. Church [3] presented a stochastic method that achieved over 95% accuracy. Also Cutting [6] presented a part-of-speech tagger based on a hidden Markov model that enables robust and accurate tagging with only a lexicon and some unla-

beled training text requirements. Brill [2] presented a rule based POS tagger which used a transformation based method that learns its rules from a training corpus. Current trend in morphological disambiguation and POS tagging is blending machine learning techniques and statistic methods into rule based approaches.

Oflazer and Kuruöz [7], developed a Turkish POS tagger that uses local neighborhood constraints, heuristics and limited amount of statistical information. Oflazer and Tür [8] developed a system that combines corpus independent, linguistically motivated handcrafted constraint rules, constraint rules that are learned via unsupervised learning from a training corpus, and additional statistical information from the corpus to be morphologically disambiguated. Our morphological disambiguator is a rule-based system, and its rules are similar to the rules of the system presented in [7,8].

## 3. Morphological Disambiguator

Our main aim was the development of an easy to use, modern, portable and publicly available effective morphological disambiguator for Turkish. Our morphological disambiguator is purely rule-based currently, but we plan to extend it with automatic rule learning capability in the near future when the reliable morphologically tagged Turkish corpus is available. In fact, we are planning to use the developed morphological disambiguator as an annotation tool in the creation of this kind of corpus.

The developed rule-based morphological disambiguator is implemented in Java programming language, and it communicates with Turkish morphological analyzer that is developed in PCKIMO environment [1]. Our morphological disambiguator has an easy to use graphical user interface but can also be used as a command line tool. The main architecture of the morphological disambiguator is given in Figure 1. It takes an input Turkish text and produces the morphologically tagged text.

Our morphological disambiguator takes an input text, and the input text is first divided into its tokens by the tokenizer. In this way, the text is represented as a sequence of tokens. Then the morphological analyzer is run on each token and a list of morphological parses is associated with each word. Then the unknown word recognizer is run for those tokens for which the morphological analyzer has returned an empty list. The unknown token recognizer associates each unknown word with a set of morphological parses. Then collocation recognizer detects the word sequences that constitute some special meaning when they are used together, and packs them into composite tokens. Lastly the morphological disambiguator is run on the token sequence, which detects and eliminates improper morphological parses using context sensitive rules.

In our system, we have used a morphologic analyzer for Turkish that is developed using PC-KIMMO environment. The morphological level description of Turkish that was

previously used in the Xerox INFL system that is developed at Bilkent University [9] has been recently ported to PC-KIMMO environment. The re-implemented system has more root words, and can handle some extra constructs such as different number constructs. The total number of the root words is more than 30,000.

After the tokenization of an input text, the tokens created by the tokenizer are sent to the morphological analyzer. After the morphological analysis, each token is assigned one or more morphological parses. For example, the results of the morphological analyzer for the token “*yarıřmada*” are as follows.

1. *yarıř*+Verb+Pos^DB+Noun+Inf2+A3sg+Pnon+Loc
2. *yarıř*+Verb+Pos+ASPECT\*PR-CONT+A3sg

After the morphological analysis there may be some tokens that are not assigned any parses such as some foreign proper nouns. These tokens are currently handled by the unknown token recognizer module. The unknown token recognizer also uses PC-KIMMO as a backend. In order to find suitable morphological parses for unknown tokens, it applies some root substitution methods that use phonetic rules of Turkish. As a simple example we can give the token “*talkshowu*” (his talkshow). The foreign word “*talkshow*” is not included in the lexicon of the morphological analyzer, so it is an unknown token. After the morphological analyzer and the unknown token recognizer, all of the tokens of the input text are associated with a set of parses.

### 3.1 Collocation Recognizer

The collocation recognizer takes the morphologically analyzed text and tries to detect and combine certain lexicalized and non-lexicalized collocations. The need for such a processing arises from the fact that a group of words, when appeared subsequently in a sentence, may behave as a multiword construct with a totally or partially different function compared to its individual members in that sentence. A typical example is the construct “*gelir gelmez*”:

- *gelir*. (He comes.)
- *gelmez*. (He does not come.)
- *gelir gelmez* ayrıldık. (We left as soon as he comes.)

Here words “*gelir gelmez*”, when used together, function in that sentence as an adverb whereas the words are inflected verbs when considered individually. There are a number of other non-lexicalized forms which are in general in the form *w+x w+y*, where *w* is the duplicated string of a root and certain suffixes, and *x* and *y* are possibly different sequences of other suffixes.

The collocation recognition is performed according to the rules given in the collocation rules file, which contains 334 rules currently. A collocation rule is sequence of token constraints and an action statement. If the sequence of token constraints matches a sequence of tokens in the text that is analyzed, the action in the action statement is applied. An action statement provides a template using which

the collocation recognizer can combine the tokens in the matched sequence into a single composite token. For example, the rule that handles the collocation “*gelir gelmez*” is follows:

```
<collocationRule> <constraint> <parse>
_R+Verb+Pos+Aor+A3sg </parse> </constraint>
<constraint> <parse> _R+Verb+Neg+Aor+A3sg
</parse> </constraint> <action>
%1 %2+Adverb+When </action> </collocationRule>
```

A constraint does not always have to declare a parse to be matched, but also token readings can be matched. This kind of rules is especially used for detecting lexicalized collocations. It is also possible use regular expressions when writing token constraints. Token matching by regular expressions is case sensitive while the ordinary token matching is case insensitive.

### 3.2 Morphological Disambiguator

Morphological analysis of a Turkish word usually results in more than one morphological parse. This ambiguity is due to the agglutinative nature of the language. The morphological disambiguator module, using a set of context sensitive and handcrafted rules, aims to reduce the number of parses associated with each word.

Disambiguation is performed using two types of disambiguation rules, namely *choose* and *delete* rules. These rules are applied only if a word is in the specified context of the rule. By being in the context, we mean that the surrounding words match the constraints of the rule. A disambiguation rule must target a token, i.e. the token that this rule aims to disambiguate. A rule can also specify neighboring tokens, each described by an offset value, i.e. the relative position of the neighbor according to the target.

A high percentage of disambiguation rules in our system are similar to the rules in [7,8]. Our morphological disambiguator uses more capable and descriptive formatting for disambiguation rules, and the number of disambiguation rules in our system is higher when compared to that of the previous work [7,8]. Currently, the total number of the disambiguation rules is 342. 289 of them are choose-rules, and 53 of them are delete-rules.

Most of the choose rules in this file are motivated by the grammatical constraints of Turkish; so they are independent from the text category. When choose rules are applied to a certain word, if the constraints of the rule are satisfied, then the target token and its ambiguous neighbors are disambiguated at once. For the noun phrase “*çocuğun kitabı*” (the child’s book), the morphological analyzer returns us the following parses:

```
çocuğun
1. çocuk+Noun+A3sg+Pnon+Gen (correct parse)
2. çocuk+Noun+A3sg+P2sg+Nom
kitabı
1. kitap+Noun+A3sg+Pnon+Acc
2. kitap+Noun+A3sg+P3sg+Nom (correct parse)
```



The tokens of this noun phrase can be disambiguated by the following choose-rule:

```
<chooseRule> <neighbour offset="-1">
<parse>A3sg+Gen</parse> </neighbour>
<target> <parse stemAllowed="false"> Noun+P3sg
</parse> </target> </chooseRule>
```

After applying the rule given above on this noun phrase, not only the word “*kitabı*” is disambiguated, but also the appropriate parse for its neighbor “*çocuğun*” is chosen.

Another set of rules, called *delete-rules*, are also used in the disambiguation process. Delete rules are mainly used to remove very rare parses of some common words. Delete rules only affect the word that is being disambiguated, and they work only in a non-ambiguous context. An example delete rule is given below:

```
<deleteRule> <target>
<token>biz</token> <parse>Noun</parse>
</target> </deleteRule>
```

The rule above drops the very infrequent noun parse of the word “*biz*” in favor of the remaining pronoun parse.

The rules in the disambiguation rules file are grouped according to their function. They are also ordered according to their generality; the more a rule is stricter (specific), the higher in the file it would appear. The order of the rules is very important, because if the ordering is wrong, then the disambiguation will produce more wrong results.

#### 4. Evaluation

In order to evaluate the performance of our morphological disambiguator, we created a test set. The test set consists of 15 randomly selected Turkish newspaper articles from online newspapers. First, the selected articles are hand tagged so that the results of the morphological disambiguator can be compared with these hand tagged articles in order to evaluate its results. Initially there were 2454 tokens in the test set. The human expert detected 77 collocations in the test set, and there were 2370 tokens (single or composite) after all collocations are hand tagged. 329 of these 2370 tokens are punctuation tokens, and 2041 of them were non-punctuation tokens. Each of 2370 tokens is correctly tagged with a single correct parse by the human expert. The human expert also selected a correct parse for the tokens that are unhandled by the morphological analyzer (unknown tokens).

Each token is assigned a set of morphological parses by the morphological disambiguator. We expect that one of these parses to be the correct one. A token is *fully disambiguated* if the disambiguator has dropped all parses except the correct one. We call the token *correctly disambiguated* if its multiple parses contain its correct parse.

We used the common precision and recall metrics in order to evaluate our morphological disambiguator. Precision measures the ratio of appropriate parses received from the morphological disambiguator to the total number of parses,

**Table 5. The results after the morphological analyzer and unknown token recognizer**

# of parses	1	2	3	4	5	6	7	8	9	10	11	12
# of tokens	1340	701	190	157	29	16	1	10	1	1	0	8

**Table 6. The results after the collocation recognizer**

# of parses	1	2	3	4	5	6	7	8	9	10	11	12
# of tokens	1304	674	172	155	28	16	1	10	1	1	0	8
# of corr. dis. toks.	1304	674	172	155	28	16	1	10	1	1	0	8
Number of Collocations											77	
Total Number of Tokens											2370	
Total Number of Parses											4226	
Number of Corr. Disamg. Tokens											2370	
<b>Precision</b>											<b>56.1%</b>	
<b>Recall</b>											<b>100%</b>	

**Table 7. The results after applying choose-rules**

# of parses	1	2	3	4	5	6	7	8	9	10	11	12
# of tokens	1820	382	70	72	7	5	1	6	1	0	0	6
# of corr. dis. toks.	1796	380	67	71	6	5	1	6	1	0	0	6
Total Number of Parses											3283	
Number of Corr. Disamg. Tokens											2339	
<b>Precision</b>											<b>71.2%</b>	
<b>Recall</b>											<b>98.7%</b>	

**Table 8. The results after applying delete rules**

# of parses	1	2	3	4	5	6	7	8	9	10	11	12
# of tokens	2010	271	56	22	3	7	0	1	0	0	0	0
# of corr. dis. toks.	1984	266	53	21	2	7	0	1	0	0	0	0
Total Number of Parses											2873	
Number of Corr. Disamg. Tokens											2334	
<b>Precision</b>											<b>81.2%</b>	
<b>Recall</b>											<b>98.5%</b>	

while the recall measures the ratio of correctly disambiguated tokens to the total number of tokens.

After the morphological analyzer and the unknown token recognizer steps of the disambiguator, there were 2454 to-

kens and there were 4383 parses for those tokens. The distribution of the tokens into the number of parses can be seen in Table 5.

Then, the collocation recognizer is executed and its results are given in Table 6. The collocation recognizer correctly found all of the 77 collocations. So, we can say that our collocation recognizer worked with 100% accuracy for this set. Although our collocation recognizer worked with 100% accuracy for this set, it can miss some collocations in a larger test set. We believe that our collocation recognizer may not be complete, but its coverage is very high. According to the results given in Table 6, the parses of each token contain its correct parse (100% recall), and 56.1% of the all parses in the result set are correct (56.1% precision). The results in Table 6 also indicate that the average number of parses per token is 1.78 ( $=2370/4226$ ), and a token can have maximum 12 parses. These measurements are the values before the disambiguation process.

We measured the precision and recall levels after applying choose and delete rules. The results after applying choose and delete rules are given in Tables 7 and 8. The precision increases from 56.1% to 71.2% by applying the choose rules by only sacrificing a small recall amount of 1.3%. The average number of parses per token also drops to 1.39 after the application of choose rules.

Finally, we apply delete rules in order to drop rare parses of tokens and achieve a precision of 81.2% and the recall becomes 98.5%. The average number of parses per token also drops to 1.21 after the application of delete rules. This is the overall performance of our morphological disambiguator. As a result, our disambiguator reduces the level of ambiguity from 1.78 parses per token to 1.21 parses per token with 81.2% precision and 98.5% recall values.

In general, precision and recall are inversely proportional to each other, i.e. it is usual to sacrifice from recall in order to improve precision. As it can be seen from the results, the decrease in recall is small when compared to the much significant increase in the precision.

## 5. Conclusion

In this paper, we introduced our effective rule-based morphological disambiguator for Turkish. Part-of-speech tagging is one of the low level disambiguation problems of NLP domain and although many highly accurate algorithms are available today, it still remains as an open research area especially for languages such as Turkish. Turkish, because of its agglutinative structure, has a higher ambiguity in the morphological level when compared to English. The morphological disambiguation of Turkish texts will reduce the burden in higher level NLP applications such as machine translation [4,5].

An advantage of our morphological disambiguator is that it uses a very flexible rule format for both the collocation recognition and the morphological disambiguation processes. This enables us to easily develop more rules when

need arises and fine tune the behavior of the morphological disambiguator. But manually maintaining the rule files may become cumbersome as the number of rules gets large. This is due to the fact that the order of rules affects the effectiveness of the morphological disambiguator. Today, many successful algorithms are neither purely rule-based nor statistical but follow a hybrid approach that combines the best properties of the two with some machine learning approaches. These taggers can usually learn new rules by analyzing relatively small sized training corpuses and can achieve great accuracy values. Although, the morphological disambiguator developed during this project is a pure rule-based tagger with no learning capabilities, it follows a very modular approach that can easily be extended with other capabilities such as automatic rule learning in the future.

As a future work, we are planning to morphologically tag a huge Turkish corpus using our annotator tool. The researchers can use this corpus for different applications. In fact, we are planning to extend our morphological disambiguator with the statistical and automatic rule learning capabilities using this corpus.

## 6. References

- [1] E. L. Antworth. PC-KIMMO: A Two-level Processor for Morphological Analysis. *Summer Institute of Linguistics*, Dallas, Texas, 1990.
- [2] Eric Brill, A simple rule-based part of speech tagger, *Proceedings of the third conference on Applied natural language processing*, March 31-April 03, 1992, Trento, Italy
- [3] Kenneth Ward Church, A stochastic parts program and noun phrase parser for unrestricted text, *Proceedings of the second conference on Applied natural language processing*, February 09-12, 1988, Austin, Texas
- [4] Ilyas Cicekli, Learning Translation Templates with Type Constraints, in: *Proceedings of Example-Based Machine Translation Workshop*, MT Summit X, Phuket, Thailand, September 2005, pp:27-34.
- [5] Ilyas Cicekli, and H. Altay Güvenir, Learning Translation Templates from Bilingual Translation Examples, in: *Recent Advances in Example-Based Machine Translation*, Carl, M., and Way, A. (eds), The Kluwer Academic Publishers, Boston, 2003, pp:247-278.
- [6] Doug Cutting , Julian Kupiec , Jan Pedersen , Penelope Sibun, A practical part-of-speech tagger, *Proceedings of the third conference on Applied natural language processing*, March 31-April 03, 1992, Trento, Italy
- [7] Kemal Oflazer, İlker Kuruöz, Tagging and morphological disambiguation of Turkish text, *Proceedings of the fourth conference on Applied natural language processing*, October 13-15, 1994, Stuttgart, Germany
- [8] Kemal Oflazer, Gökhan Tür, Combining Hand-crafted Rules and Unsupervised Learning in Constraint-based Morphological Disambiguation. *Proceedings of the ACL-SIGDAT Conference on Empirical Methods in Natural Language Processing*, May 1996, Philadelphia, PA, USA.
- [9] Kemal Oflazer, Two-level Description of Turkish Morphology, *Literary and Linguistic Computing*, Vol. 9, No:2, 1994.

# Effectively Realizing the Inferred Message of an Information Graphic

Seniz Demir and Sandra Carberry  
Dept. of Computer Science  
University of Delaware  
Newark, DE 19716  
{demir, carberry}@cis.udel.edu

Stephanie Elzer  
Dept. of Computer Science  
Millersville University  
Millersville, PA 17551  
elzer@cs.millersville.edu

## Abstract

Information graphics, such as bar charts and line graphs, that appear in popular media generally have a message that they are intended to convey. We have developed a Bayesian network that analyzes the communicative signals in an information graphic and produces a logical representation of the graphic's intended message. However, the output produced by the Bayesian network is deficient for producing natural language text. This paper presents our solution to several aspects of this problem: identifying an appropriate referent for the dependent axis, determining when to enumerate the bar labels in a message, and identifying the ontological category for the bar labels. An evaluation study shows that our methodology produces reasonable text that is much better than several baseline strategies.

## Keywords

Natural language processing, corpus analysis, graph understanding, text generation

## 1 Introduction

Information graphics (such as bar charts and line graphs) are non-pictorial graphics that depict attributes of entities and relations among them. Although some information graphics are only intended to display data[18], the overwhelming majority of information graphics that appear in magazines and newspapers have a communicative goal or intended message. For example, the graphic in Figure 1 ostensibly is intended to convey that the percentage of GM's net earnings produced by its finance unit increased substantially in the second quarter of 2003 in contrast with the decreasing trend from the third quarter of 2002 to the first quarter of 2003. We developed a Bayesian system that exploits the communicative signals in an information graphic to produce a logical representation of the graphic's intended message[7]. However, the logical representation produced by the Bayesian system is deficient for producing text, and additional information must be extracted from the graphic if a useful summary is to be constructed.

Clark[3] contends that language is not just text and utterances, but instead includes any deliberate signal that is intended to convey a message. Thus, under Clark's definition, information graphics are a form of language. Our work shows that methodologies typically used in processing utterances and text (such as

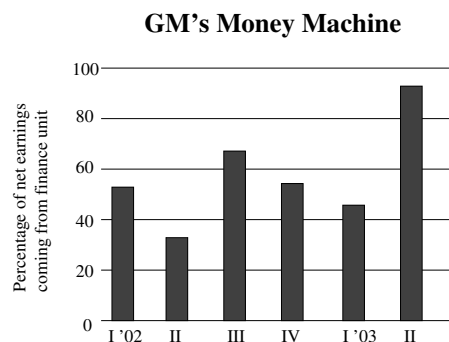


Fig. 1: Graphic with a cute caption

extraction of communicative signals, corpus analysis to identify common patterns, algorithms based on insights from the corpus analysis, and modification of existing software, such as parsers, to fit the needs of the problem) can also be successful on non-stereotypical forms of language.

Section 2 shows the importance of recognizing the intended message of an information graphic, Section 3 discusses related research, and Section 4 gives an overview of our system that hypothesizes the intended message of an information graphic. Section 5 then discusses the problems that we had to address in order to construct text from the logical representations of the hypotheses produced by our message recognition system. Sections 6, 7 and 8 present our solutions to these problems. Section 9 discusses examples of how our system constructs text capturing the graphic's primary message, and Section 10 presents an evaluation of the resultant text and discusses future work.

## 2 Importance of Understanding Information Graphics

Information graphics, such as the two graphics in Figure 5, are important knowledge resources that could be used for many purposes, such as devising proposals for legislation on identity theft. To be useful, such graphics must be accessible from a digital library based on what the graphic conveys.

What about graphics in multimodal documents? We conducted a corpus study to determine the extent to which information graphics are redundant in a multimodal document[1]. We found that in over 60% of the instances in our analyzed corpus, little or none of the graphic's message was captured by the article's ac-

companying text. Yet the graphic's message played an important role in achieving the discourse purpose[12] of the overall document. Thus information graphics cannot be ignored, and effective summarization of a multimodal document must take into account the messages conveyed by its information graphics.

Given that information graphics cannot be ignored, it is imperative that individuals with sight-impairments be provided with a means of accessing the graphic's content. Although researchers have attempted to convey information graphics via an alternative modality (such as touch or sound), these approaches have serious limitations, such as requiring expensive equipment or requiring that the user develop a mental map of the graphic, something that is very difficult for users who are congenitally blind[14]. Our approach differs significantly from previous work: we are developing an interactive natural language system that provides the user with a brief summary of the graphic's message and then responds to followup questions requesting further detail about the graphic.

For all of the above reasons, it is important that a system be able to recognize a graphic's message. However, as shown by Corio and LaPalme[4] and our own corpus study[6], naturally occurring captions are often very general and of limited utility in identifying the graphic's message. For example, the caption on the graphic in Figure 1 captures little of what the graphic conveys — namely, a contrast between recent performance of GM's finance unit and the trend over the preceding quarters. Thus it is essential that a system be devised for recognizing the message conveyed by an information graphic.

### 3 Related Work

Research has addressed the problem of generating information graphics and accompanying captions[4, 15, 16]. In graphics generation, the system is given a set of data along with one or more communicative goals, and it designs a graphic that achieves these goals. Our problem is different: we are given the information graphic and must identify its communicative goal (the message that it conveys) by reasoning about the communicative signals in the graphic. Futrelle and Nikolakis[10] developed a constraint grammar for parsing vector-based visual displays and producing structured representations of the elements comprising the display, but Futrelle's goal is to produce a graphic that serves as a simpler representation of one or more graphics in a document[9]. Our work is the first to address the problem of *understanding* an information graphic — i.e., recognizing the message that it conveys.

### 4 Graph Understanding System

We have developed a Bayesian system[7] that treats information graphics as a form of language and hypothesizes a graphic's intended message. The system takes as input an xml representation of the visual image (produced by a visual extraction module) that specifies the graphic's axes, the bars, their heights, colors, labels, any special annotations, the caption, etc. We have identified three kinds of communicative signals that appear in bar charts:

- the relative effort required for different perceptual tasks; for example, it is easier to determine the rank of an entity in a bar chart if the bars are sorted according to height than if they appear in alphabetical order of their labels. AutoBrief[15] contended that graphic designers construct graphics so that *important* perceptual tasks (the ones necessary for achieving the graphic's communicative goal) are as easy as possible. Thus the relative effort required for different perceptual tasks serves as evidence about which tasks the viewer is intended to perform in deciphering the graphic's message.
- the salience of entities in the graphic; for example, coloring a bar differently from other bars in a bar chart makes the bar salient, as does mentioning its label in the caption. Our hypothesis is that salient entities play a significant role in a graphic's message.
- the presence of suggestive verbs (such as *rising*) in a graphic's caption

Our system, described in [7], extracts this evidence from a given bar chart and enters it into a Bayesian network which hypothesizes the graphic's intended message. Leave-one-out cross validation on a corpus of 110 bar charts showed that our system has a success rate of 79.1% in identifying the graphic's message. Although our current system is limited to bar charts, we believe that our methodology is extensible to other kinds of information graphics.

### 5 Problems in Message Realization

Our Bayesian system produces a logical representation of a graphic's message; this representation consists of the message type (such as *Maximum* for messages which convey that a particular entity has the largest value in a bar chart), and the parameters of the message (such as the bar with the largest value in the case of the *Maximum* message type). For example, the system produces *Maximum(First\_Bar)* for the graphic in Figure 2. Reiter and Dale[17] argue that templates are appropriate for many natural language problems. Since the syntactic variability in our target messages is limited, we use templates for generation, with one template defined for each of our 12 message types.

For the graphic in Figure 2, the natural language output should ideally be "*Tennis has the highest number of past nominees for the Laureus World Sports Awards among the sports listed*". However, several problems arise:

1. the appropriate referent for the dependent axis (in this case, *number of past nominees for the Laureus World Sports Awards*) is not part of the logical representation and is not explicitly given in the graphic.
2. for some message types (such as *Maximum* in the above example), a decision must be made regarding when the labels should be enumerated in the natural language text and when only the ontological category of the labels should be given.

### Tennis players top nominees

The nominees for the 2003 Laureus World Sports Awards will be announced today. Sports that have had the most nominees:

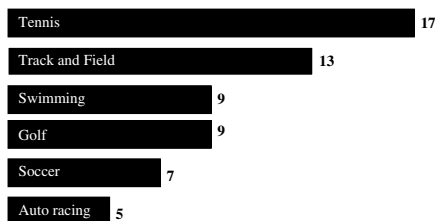


Fig. 2: Graphic from USA Today

- the ontological category of the independent axis labels (in the above example, *sports*) must be inferred from the bar labels.

Section 6 presents our corpus study and solution to the difficult problem of realizing an appropriate referent for the dependent axis, Section 7 discusses how Gricean maxims motivated our decision about when to enumerate the bar labels, and Section 8 describes how we identify the ontological category of the labels.

## 6 Measurement Axis Descriptor

The first, and most serious problem, encountered in generating the message conveyed by an information graphic is the identification of an appropriate referent for the dependent axis. We will refer to this referent as the **measurement axis descriptor**. We undertook a corpus analysis in order to identify where the measurement axis descriptor appears in a graphic and to motivate heuristics for extracting it.

### 6.1 Corpus analysis

We analyzed 107 simple bar charts from articles in newspapers and popular magazines such as Newsweek and Business Week. We observed that graphics contain a set of component texts, in addition to the bar labels, that are visually distinguished from one another (e.g by placement, blank lines, or different fonts), which we refer to as **text levels**. Table 1 lists the various text levels, along with how often they appeared in the graphs in our corpus. In composite graphs (graphs consisting of multiple individual graphs as in Figure 3), Overall\_Caption is the text that appears at the top of the overall group and serves as a caption for the whole set; Overall\_Description is additional text, distinguishable from the caption, that is common to the set of graphics and often elaborates on them. Caption

Text level	Freq. of Occurrences
Overall_Caption	31.8%
Overall_Description	17.8%
Caption	99.0%
Description	54.2%
Text_In_Graphic	39.3%
Dependent_Axis_Label	18.7%
Text_Under_Graphic	7.5%

Table 1: Text levels in bar charts

### Tallying Up the Hits

Yahoo once relied entirely on banner ads. Now it's broadened its business mix.

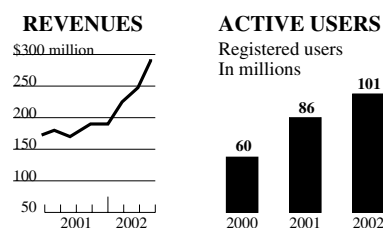


Fig. 3: A composite graphic from Newsweek<sup>1</sup>

and Description serve the same roles for an individual graphic. For example, in Figure 3, the Overall\_Caption is "Tallying Up the Hits" and the Overall\_Description is "Yahoo once relied entirely on banner ads. Now it's broadened its business mix". The Caption for the bar chart at the right of Figure 3 is "Active Users" and the Description is "Registered users In millions". For the graphic in Figure 2, the Caption is "Tennis players top nominees" and the Description is "The nominees for the 2003 Laureus World Sports Awards will be announced today. Sports that have had the most nominees:". Text\_In\_Graphic is any text within the borders of a graphic, such as "U.S. Biotech Revenues, 1992-2001" in Figure 4. Dependent\_Axis\_Label is the label (if any) on the dependent axis of a bar chart, such as "Revenues(in billions)" in Figure 4. Lastly, Text\_Under\_Graphic is any text under a graphic; such text generally starts with a marker symbol (such as \*). As our evaluation in Section 10 shows, no single text level provides an acceptable measurement axis descriptor for all graphics. For example, Text\_In\_Graphic is a better measurement axis descriptor than Dependent\_Axis\_Label for the graphic in Figure 4, but the bar chart in Figure 3 does not have any Text\_In\_Graphic.

The goal of our corpus study was to identify how to construct a measurement axis descriptor for a graphic. Two annotators analyzed each of the 107 graphics in our corpus and determined the ideal measurement axis descriptor. We then analyzed each of the graphics to identify where the ideal measurement axis descriptor appeared. In 55.1% of the graphics, the ideal measurement axis descriptor appeared as a unit in a single text level, but in 36.5% of these instances, the text level contained additional information. For example, in the graphic at the left of Figure 5, the ideal measurement axis descriptor is "identity-theft complaints" which is part of the Caption "Identity-theft complaints are skyrocketing". In 44.9% of the graphics, pieces of the measurement axis descriptor had to be extracted from more than one text level and melded together. In these instances, the ideal measurement axis descriptor can be viewed as consisting of a **core** or basic noun phrase from one text level that must be augmented with text from another level (or in some cases, from text in the accompanying article). For example, for the bar chart at the right of Figure 3, "registered users"

<sup>1</sup> This figure displays two of the five individual graphs comprising the composite graphic that appeared in Newsweek.

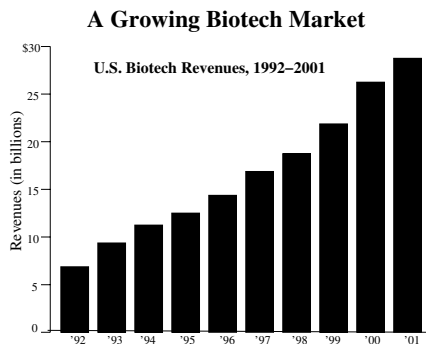


Fig. 4: Graphic from Business Week

is the core of the ideal measurement axis descriptor which is “Yahoo’s registered users”.

We also found that, with the exception of Text\_Under\_Graphic (which typically serves as a footnote providing detail about the core), the ordering of text levels in Table 1 forms a hierarchy of textual components, with Overall\_Caption and Dependent\_Axis\_Label respectively at the top and bottom of the hierarchy, such that the core generally appears in the lowest text level present in the graphic. For example, for every graphic in our corpus containing a Dependent\_Axis\_Label that was not just a scale or unit indicator (such as *millions* or *dollars*), the Dependent\_Axis\_Label contained the core. Where the Dependent\_Axis\_Label was absent from a graphic or was only a scale or unit indicator, but the graphic contained a Text\_In\_Graphic component, the core appeared in Text\_In\_Graphic in 35 of 39 graphics. Similar observations held for the text levels higher in the hierarchy. In retrospect, this is not surprising since text levels lower in the hierarchy are more specific to the graphic’s content and thus more likely to contain the core of the ideal measurement axis descriptor.

We also observed the presence of cues, such as the phrase “Here is” or a terminating colon punctuation mark, suggesting that a text level contains the core of the measurement axis descriptor. For example, the phrase “Here is” realized as a contraction in the sentence “Here’s the monthly construction spending”, suggests that the subsequent noun phrase tells what the graphic is presenting and thus contains the core of the measurement axis descriptor.

During the corpus analysis we observed three ways in which a core from one text level was augmented to produce the ideal measurement axis descriptor:

- Expansion of the noun phrase: nouns in the core of the descriptor were replaced with a noun phrase which had the same noun as its head. For example, in the graphic in Figure 4, the Dependent\_Axis\_Label contains the core (*Revenues*) but the ideal measurement axis label is “U.S. Biotech Revenues” appearing at a higher text level; this ideal measurement axis descriptor can be viewed as an expansion of the core.
- Specialization of the noun phrase: the core was augmented with a proper noun which specialized the descriptor to a specific entity. For example, in the graphic at the right of Figure 3, the ideal measurement axis descriptor “Yahoo’s registered

## A GROWING CONSUMER MENACE

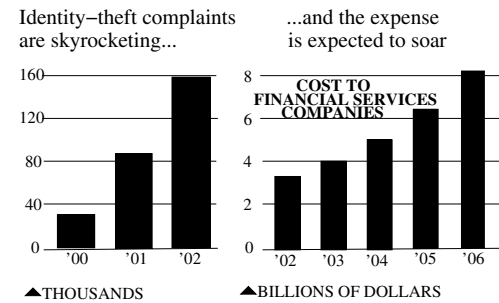


Fig. 5: Graphic from Business Week

users” consists of the core “registered users” augmented with the proper noun “Yahoo” that appears in the Overall\_Description.

- Addition of detail: Text\_Under\_Graphic typically serves as a footnote to give specialized detail about the graphic. If the Text\_Under\_Graphic begins with a marker, such as an asterisk, and the core is followed by the same marker, then Text\_Under\_Graphic adds detail to the core. For example, in the graphic in Figure 6, “unit costs” is the core but the ideal measurement axis descriptor must also contain the information from Text\_Under\_Graphic, namely “U.S. only, one available seat flown one mile, year ending June 2002”.

## 6.2 Methodology

Our methodology for constructing a measurement axis descriptor is based on the insights gained from our corpus analysis. First, preprocessing extracts the scale and unit indicators from the text levels or from labels on the dependent axis (for example, the label 30% would indicate that *percent* is the unit of measurement). Next heuristics are used to construct the core of the measurement axis descriptor by extracting a noun phrase from a text level of the graphic. Three kinds of augmentation rules, corresponding to the three kinds of augmentation observed in our corpus, are then applied to the core to produce the measurement axis descriptor. Finally, if the measurement axis descriptor does not already contain the unit of measurement (such as *percent*), the phrase indicating the unit of measurement is appended to the front of the measurement axis descriptor.

### 6.2.1 Heuristics

We developed 9 heuristics for identifying the core of the measurement axis descriptor. The application of the heuristics gives preference to text levels that are lower in the hierarchy, and the heuristics themselves take into account the presence of cue phrases, special characters, and the presence and position of noun phrases in a text level. The first heuristic only applies to the Dependent\_Axis\_Label:

- **Heuristic-1**: If the Dependent\_Axis\_Label contains a noun phrase that is not a scale or unit indicator, that noun phrase is the core of the measurement axis descriptor.

The second heuristic only applies to Text\_In\_Graphic:

- **Heuristic-2:** If Text\_In\_Graphic consists solely of a noun phrase, then that noun phrase is the core; otherwise, if Text\_In\_Graphic is a sentence, the noun phrase that is the subject of the sentence is the core.

The remaining heuristics are then applied, in order, to a text level, starting with the Description text level; if a core is not identified at one text level, the heuristics are applied, in order, to the next higher text level in the hierarchy. Space limitations preclude listing all of the heuristics, but the following are two representative heuristics, in addition to Heuristic-1 and Heuristic-2 presented above. Heuristic-5 is based on observations about punctuation that suggests the presence of the core of the measurement axis descriptor, and Heuristic-8 is based on observations about the location of the core when it is part of a full sentence.

- **Heuristic-5:** If a fragment at the text level consists solely of a noun phrase followed by a colon (:), and the noun phrase is not just a proper noun, that noun phrase is the core.
- **Heuristic-8:** The core is the noun phrase preceding the verb phrase in the current sentence at the text level.

In some graphics, what is extracted as the core is conflated with a reference to the ontological category of the bar labels. If the core's head noun matches the ontological category of the bar labels, then that noun cannot be the measurement axis descriptor; thus if the noun is modified by a subsequent relative clause or a phrase beginning with *with*, then the nouns and subsequent prepositional phrases in the modifier are instead collected as the core. For example, consider Figure 2. Our heuristics would initially extract *"Sports that have had the most nominees"* as the core of the measurement axis descriptor; since *sports* is the category of the bar labels, *"nominees"* becomes the core.

### 6.2.2 Augmentation Rules

Augmentation rules correspond to the three kinds of augmentation observed during corpus analysis (expansion, specialization, and addition of detail), along with addition of the unit of measurement (such as *percent*). In expanding the core, the system examines text levels higher in the hierarchy than the text level from which the core was extracted; if a noun phrase appears with the same head noun as a noun in the core, and the noun phrase does not consist of just an adjective and the head noun, then the noun in the core is replaced with the larger noun phrase.

To specialize the noun phrase, the system determines whether 1) there is only one proper noun at all text levels higher in the hierarchy than the text level from which the core was extracted, or 2) there is only one proper noun in the Overall\_Caption or the Caption; if one of these two criteria are satisfied and the proper noun is not a bar label in the graphic, then the possessive form of that proper noun is appended to the front of the core. (The reason for treating the Overall\_Caption and Caption differently from the other text

### SOUTHWEST'S BIG COST ADVANTAGE...

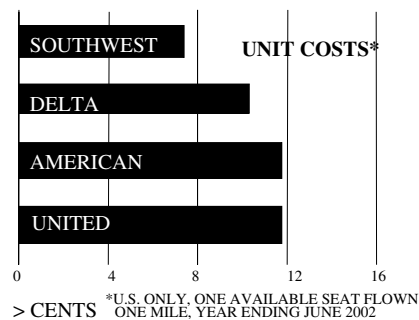


Fig. 6: Graphic from Business Week

levels is that a single proper noun at these levels refers to the content of the graphic, whereas a proper noun at other levels, such as Description, may be part of an elaboration or comparison with the graphic's content and thus not necessarily specialize the core.)

To add detail to the core, the system determines whether the core was followed by a special marker, such as an asterisk, in the text level from which it was extracted; if so, the system searches for text preceded by the same marker and appends it to the core as a bracketed expression.

Finally, the measurement axis descriptor that has been constructed is checked to determine if it already gives the unit of measurement (as identified from the graphic during preprocessing); if not, a phrase indicating the unit of measurement is added to the front of the measurement axis descriptor.

### 6.3 Implementing the Heuristics

The heuristics must examine the parses of a graphic's text levels in order to identify the core of the measurement axis descriptor and to apply the augmentation rules. We experimented with both NP chunkers and parsers. NP chunkers are biased toward noun phrases and produced unsatisfactory results for text levels that consisted of full sentences. Thus we adopted Charniak's maximum entropy based parser[2], but had to address its bias toward imperative sentences over sentence fragments. Fragments are common as Text\_In\_Graphic, Caption, and Overall\_Caption. When the Caption or Overall\_Caption is a fragment, it may begin with a noun that can also be used as a verb. An example is *"Cost to financial services companies"* which appeared as the Caption on one of the graphics in our corpus. Unfortunately, a bias toward full sentences causes the parser to parse such fragments as imperative sentences with words such as *cost* identified as verbs. However, imperative sentences are rarely seen in the textual components of graphics. We solved this bias problem with rules such as the following:

If WordNet[8] indicates that the first word of the input can be used as both a verb and a noun or as both a verb and an adjective, precede the input with "The" before sending it to the parser.

These rules forced the parser to prefer noun phrase fragments over imperative sentences.

## 7 Applying Gricean Maxims

Grice's Maxim of Quantity[11] states that one's discourse contribution should be as informative as necessary for the purposes of the exchange but not more so. Joshi, Webber, and Weischedel[13] showed that a system should not only produce correct information but should also prevent the user from drawing false inferences from the system's responses. If our system were to enumerate all entities involved in a comparison message, the response might be lengthy and the enumeration of little utility to the user. (To address instances in which the user wants the additional lengthy detail, both our system for blind individuals and our digital libraries application will include a facility for followup questions.) On the other hand, if the system never enumerates the entities (even when there are only a few), the user may make the false inference that there are too many to list. Thus we set a cut-off  $C$ , such that if the number of entities involved in a Maximum or Get-Rank message exceeds  $C$ , they are not enumerated. For the examples in this paper,  $C$  is set to 4, but we are performing human subject experiments to identify the most appropriate  $C$  value.

## 8 Identifying the Ontological Category

Although the dependent axis does not specify the ontological category for the bar labels, identifying the category results in better natural language than merely using a generic referent; for example, compare the phrase "...among the sports listed" with the phrase "...among the entities listed" in producing natural language text for the message conveyed by the graphic in Figure 2. We use OpenCyc ontology version v0.7.8b[5] to identify the ontological categories of bar labels.<sup>2</sup> For each bar label, all ontological categories it belongs to are identified. The most general and common category for at least two of the bar labels is identified as the ontological category.

## 9 Processing Examples

Our methodology for producing natural language text from the logical representation of a graphic's intended message has been implemented and tested on examples from many publications. Input to the natural language system is the logical representation of the graphic's message and the xml representation of the graphic's components, including the text at the various text levels.

The following examples illustrate how our system generates a graphic's message as natural language text. For the graphic in Figure 2, Heuristic-5 initially identifies the noun phrase "sports that have had the

<sup>2</sup> If we could determine that the bar labels in a graphic cover all elements of a given category, then the generated message could for example say "among airlines" instead of "among the airlines listed". However, we have not found an existing ontology that would allow us to reliably make such a determination.

most nominees" as the core. However, its head noun "sports" matches the ontological category of the bar labels; consequently, the noun "nominees" in the relative clause modifying "sports" becomes the core. The augmentation rule for specialization finds that "Laureus World Sports" is the only proper noun in the text levels and forms "Laureus World Sports's nominees". After adding a pre-fragment representing the unit of measurement, the measurement axis descriptor becomes "The number of Laureus World Sports's nominees". Using the template for the Maximum message type, our system generates "The bar chart titled 'Tennis players top nominees' shows that the number of Laureus World Sports's nominees is highest for Tennis among the sports listed." Our generated natural language for this example is imperfect and will be discussed further in Section 10.

For the graphic in Figure 4, Heuristic-1 identifies "Revenues" in Dependent\_Axis\_Label as the core. Since the core and the Text\_In\_Graphic, "U.S. Biotech Revenues", have the same head noun, the augmentation rule for expansion produces "U.S. Biotech Revenues" as the augmented core. Using the template for the Increasing-Trend message type, our system renders the following natural language text: "The bar chart titled 'A Growing Biotech Market' shows that the dollar value of U.S. biotech revenues had a rising trend from 1992 to 2001."

For the graphic in Figure 6, our system uses Heuristic 2, the augmentation rule for adding detail, and the template for the Minimum message type to produce the natural language text "The bar chart titled 'Southwest's Big Cost Advantage' shows that the cent value of unit costs (u.s. only one way available seat flown one mile, year ending june 2002) is lowest for Southwest among the entities listed: Southwest, Delta, American, and United." Note that in this instance, our system was unable to identify the ontological category of the bar labels and therefore used the generic term "entities". Note also that the bar labels were enumerated in this message since the number of bars did not exceed our cutoff of 4, whereas only the ontological category of the bar labels was given for the graphic in Figure 2.

## 10 Evaluation and Future Work

The quality of our generated text is largely dependent on how well we identify an appropriate measurement axis descriptor. Thus we constructed a test corpus consisting of 202 randomly selected bar charts from 19 different newspapers and magazines, along with their accompanying articles. We ran our system for each of the graphics and the resultant output was rated by two evaluators. The evaluators each assigned a rating from 1 to 5 to the system's output; if the evaluators differed in their ratings, then the lowest rating was recorded. For comparison, three baselines were computed, consisting of evaluations of the text that would be produced using each of the following three text levels as the measurement axis descriptor: Dependent\_Axis\_Label, Text\_In\_Graphic, and Caption. For the baselines, if the evaluators differed in their rating of the resultant output, the higher rating was recorded, thereby biasing our evaluation toward better scores



for the baselines (in contrast with the scores for our system’s output, where the lower score was recorded when the evaluators differed). We did not compute a baseline comparison using the text in Description as the measurement axis descriptor since that text level is most often full sentences and thus would generally produce very poor results.

5	<b>excellent text</b>
4	<b>very good:</b> very understandable but awkward
3	<b>good:</b> contains the right information but is hard to understand
2	<b>poor:</b> missing important information
1	<b>very bad</b>

The results of our evaluation are presented in Table 2. They show that our system produces natural language text that rates midway between good and very good. It is particularly noteworthy that our methodology performs much better than any of the baseline approaches.

However, further work is needed to improve our results. We need to resolve pronominal references within the text in a graphic and between texts in composite graphics, and we need to examine the internal organization/relation between graphs in composite graphics to identify the full referent of definite noun phrases. For example, for the bar chart on the right of Figure 5, the noun phrase “*the expense*” (and thus the noun phrase “*cost to financial services companies*” since *cost* and *expense* refer to the same entity) must be specialized so that it is “*the expense of identity-theft complaints*”, thereby leading to a measurement axis descriptor that is “*cost to financial services companies of identity-theft complaints*”. We also must take the tense of the text in the graphic into account in constructing a measurement axis descriptor. For example, the ideal measurement axis descriptor for the graphic in Figure 2 would indicate that the graphic is displaying the number of past nominees for the Laureus Sports Award, not current nominees. But the Description does not explicitly state “*past nominees*” and this must be inferred from the past tense in “*Sports that have had the most nominees*”.

## 11 Conclusion

This paper has presented our work on realizing the intended message of a simple bar chart. We have shown how Gricean maxims dictate the amount of information included in the summary and how the OpenCyc ontology is used to generate meaningful categories. We also presented our corpus analysis that explored where the ideal measurement axis descriptor appears in a graphic, discussed the insights that we gained from the corpus analysis, and presented our strategy for constructing a measurement axis descriptor by identifying a core descriptor and then augmenting it to obtain an appropriate measurement axis descriptor. Evaluation of our implemented system shows that our methodology generally produces reasonable text and that it performs far better than any of three baseline approaches. Moreover, our work illustrates how NLP methodologies can be successfully applied to non-stereotypical forms of language such as information graphics.

Approach	Evaluation score
Our system	3.574
<i>Baseline-1:</i> Dependent_Axis_Label	1.475
<i>Baseline-2:</i> Text_In_Graphic	1.757
<i>Baseline-3:</i> Caption	1.876

Table 2: Evaluation of generated text

## Acknowledgements

We would like to thank Dr. Vijay Shanker for his valuable suggestions on addressing the problems encountered with the parser. This material is based upon work supported by the National Science Foundation under Grant No. IIS-0534948.

## References

- [1] S. Carberry, S. Elzer, and S. Demir. Information graphics: An untapped resource for digital libraries. In *the Proc. of SIGIR’06*, pages 581–588, 2006.
- [2] E. Charniak. A maximum-entropy-inspired parser. In *the Proc. of NAACL’02*, pages 132–139, 2002.
- [3] H. Clark. *Using Language*. Cambridge University Press, 1996.
- [4] M. Corio and G. Lapalme. Generation of texts for information graphics. In *the Proc. of the 7th European Workshop on Natural Language Generation EWNLG’99*, pages 49–58, 1999.
- [5] CycL.Cycorp. URL: <http://www.cyc.com>.
- [6] S. Elzer, S. Carberry, D. Chester, S. Demir, N. Green, I. Zukerman, and K. Trnka. Exploring and exploiting the limited utility of captions in recognizing intention in information graphics. In *the Proc. of ACL’05*, pages 223–230, 2005.
- [7] S. Elzer, S. Carberry, I. Zukerman, D. Chester, N. Green, and S. Demir. A probabilistic framework for recognizing intention in information graphics. In *the Proc. of IJCAI’05*, pages 1042–1047, July 2005.
- [8] C. Fellbaum. *WordNet: An electronic Lexical Database*. The MIT Press, 1998.
- [9] R. Futrelle. Summarization of diagrams in documents. In I. Mani and M. Maybury, editors, *Advances in Automated Text Summarization*. MIT Press, 1999.
- [10] R. Futrelle and N. Nikolakis. Efficient analysis of complex diagrams using constraint-based parsing. In *the Proc. of the Third International Conference on Document Analysis and Recognition*, 1995.
- [11] H. P. Grice. Logic and conversation. *Speech Acts*, 3:41–58, 1975.
- [12] B. Grosz and C. Sidner. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, 12(3):175–204, 1986.
- [13] A. Joshi, B. Webber, and R. Weischedel. Living up to expectations: Computing expert responses. In *the Proc. of the National Conference on Artificial Intelligence*, pages 169–175, 1984.
- [14] A. Kennel. Audiograf: A diagram-reader for the blind. In *the Proc. of the Second Annual ACM Conference on Assistive Technologies*, pages 51–56, 1996.
- [15] S. Kerpedjiev and S. Roth. Mapping communicative goals into conceptual tasks to generate graphics in discourse. In *the Proc. of IUI’00*, pages 60–67, 2000.
- [16] V. Mittal, J. Moore, G. Carenini, and S. Roth. Describing complex charts in natural language; a caption generation system. *Computational Linguistics*, 34:431–468, 1998.
- [17] E. Reiter and R. Dale. Building applied natural language generation systems. *Natural Engineering*, 3(1):57–87, March 1997.
- [18] J. Yu, J. Hunter, E. Reiter, and S. Sripada. Recognising visual patterns to communicate gas turbine time-series data. In *the Proc. of the Twenty-second SGAI International Conference on Knowledge Based Systems and Applied Artificial Intelligence*, pages 105–118, 2002.

# Towards an Optimal POS Tag Set for Arabic Processing

Mona T. Diab  
Center for Computational Learning Systems  
Columbia University  
mdiab@cccls.columbia.edu

**Abstract:** Devising an appropriate part of speech (POS) tag set for a language is crucial for higher level computational processing. In this paper, we present a new POS tag set (ERTS) for Arabic that is proven more useful and functional for higher order processing of the language. ERTS comprises 75 tags derived from the full morphological tag set adopted in the detailed POS tag set in the Arabic Treebank. It expands on the standard POS tag set, RTS, which comprises 25 tags. The new tag set expands on the morphological features for nominals in the language. We evaluate the efficacy of the new POS tag set through a POS tagger and also in the context of a task based evaluation, namely, Base Phrase Chunking. The ERTS-POS tagger achieves an accuracy of 96.13%. We show that ERTS helps an Arabic BPC system achieve an  $F_{\beta=1}$  of 96.33% compared to 95.51% only obtained by using RTS alone or in conjunction with explicit independent encoding of morphological features.

## Keywords

POS tag set, Arabic Language, POS Tagging, Base Phrase Chunking

## 1 Introduction

Devising an appropriate part of speech (POS) tag set for a language is crucial for higher level computational processing. Settling on what is the right level of granularity for a tag set is typically an empirical question that is highly correlated with a specific task or set of tasks. In morphologically rich languages such as Arabic, this problem is even more pronounced since there are many morphological features that are both explicitly and implicitly encoded in the surface orthography. The writing system for Arabic is mostly underspecified for short vowels which are bearers of many of the morphological feature information. In fact, naturally occurring short vowels in newswire text amount to only 1.5% of the words [8].

We present a new POS tag set (ERTS) for Arabic processing. The POS tag set expands on the current standard POS introduced by the LDC, RTS, which is part of the Arabic Treebank (ATB) distribution [14]. RTS comprises 25 tags. ERTS expands on the nominal categories [Adjectives, Nouns, Proper Nouns] increasing the tag set to 75 tags. ERTS is derived from the full morphological tag set represented in the detailed POS tag set in the ATB. ERTS encodes definiteness, gender and number information onto the basic RTS. In this paper, we show that though ERTS has three times the number of tags in the POS tag set RTS, yet the same (using the same features and underlying

machinery) discriminative POS tagger trained on the RTS yields comparable performance to it when trained on the ERTS set. The ERTS-POS tagger achieves an accuracy of 96.13% compared to 96.15% for the RTS-POS tagger. Moreover, in a task based evaluation, we show that ERTS yields better results when used as a feature in a Base Phrase Chunking (BPC) system. ERTS-BPC achieves an  $F_{\beta=1}$  of 96.33% compared to 95.41% obtained by RTS-BPC.

The paper is laid out as follows: Section 2 illustrates some of facts about the Arabic language; Section 3 discusses related work in devising POS tag sets for morphologically rich languages; Section 4 illustrates the ERTS POS tag set; Section 5 describes our approach for ERTS evaluation; Section 6 presents the experimental setup, results and discussions.

## 2 Arabic Language

Arabic is a Semitic language.<sup>1</sup> It is known for its templatic morphology where words are made up of roots and affixes. Clitics agglutinate to words. For instance, the surface word *و بحسنتهم* *wbHsnAthm*<sup>2</sup> ‘and by their virtues [fem.]’, can be split into the conjunction *w* ‘and’, preposition *b* ‘by’, the stem *HsnAt* ‘virtues [fem.]’, and possessive pronoun *hm* ‘their’.

From the morphological standpoint, Arabic exhibits rich morphology. Similar to English, Arabic verbs are marked explicitly for tense, voice and person, however in addition, Arabic marks verbs with mood (subjunctive, indicative and jussive) information. For nominals (nouns, adjectives, proper names), Arabic marks case (accusative, genitive and nominative), number, gender and definiteness features. Moreover, closed class words such as object pronouns and possessive pronouns inflect for person, gender and number. From the syntactic standpoint, Arabic, different from English, is considered a pro-drop language, where the subject of a verb may be implicitly encoded in the verb morphology. Hence, we observe sentences such as *اكل البرتقال* *Akl AlbrtqAl* ‘ate-[he] the-oranges’, where the verb *Akl* encodes that the subject is a 3rd person masculine singular. This sentence is equivalent to *هو اكل البرتقال* *hw Akl AlbrtqAl* ‘he ate the-oranges’. In the Arabic Treebank (ATB), we observe that 40% of all sentences are pro-dropped for subject [13]. Arabic exhibits more complex noun phrases than in English mainly to express possession. These constructions are known as

<sup>1</sup> Other Semitic languages include Hebrew and Amharic

<sup>2</sup> We use the Buckwalter transliteration scheme to show romanized Arabic [2].

*idafa* constructions. In these complex structures an indefinite noun may be followed by a definite noun. For example, *رجل البيت rjl Albyt* ‘man the-house’ meaning ‘man of the house’. Therefore, Arabic does not have a special prepositional use to express possession in a manner similar to English. Due to these differences, among others, Arabic morphology plays a crucial role in encoding different syntactic configurations: For instance, the possible verb-argument orders in the form of case markings; or the possible agreement markers on nouns and their preceding adjectives. Therefore, designing a functional and practical tag set that captures the nuances of Arabic morphology is crucial.

It is worth noting that some of the case, mood and voice features are marked only using short vowels. Depending on the genre and domain, Arabic orthography is underspecified for short vowels in varying degrees. For example, if the genre of the text is religious such as the Quran, or pedagogical such as children’s books, the orthography would be fully specified for all the short vowels to enhance readability and disambiguation. However, the majority of other genres lack of such explicit marking. Accordingly, the design of an appropriate POS tag set, in the absence of Arabic diacritizer, should be sensitive to the surface orthography in the language.

### 3 Related Work

There are several published research efforts reporting on studies of what constitutes the optimal tag set for a morphologically rich language. [10] applies a machine learning approach to discover the most relevant morphological components to be included in the tag set for the Czech language. For Spanish, [5] experiment with different morphological features adding them incrementally to the Spanish POS tag and evaluating them in the context of syntactic parsing. Similar studies exist for Turkish [3] and Hindi [6]. The main theme in this line of research is trying to codify the most frequently prominent morphological features of the language in the tag set itself.

This study aims to do the same for Arabic. Arabic is a rich morphological language. Fully diacritized words are explicitly marked for case, gender, number, definiteness, mood, person, voice, tense, aspect and among other features. These morphological features are exhibited in the full tag set, FULL, provided in the ATB. FULL comprises over 2000 tag types. There exists a system that produces the full morphological POS tag set, MADA, with very high accuracy, 95% [9].<sup>3</sup> However, MADA relies crucially on the existence of an extensive morphological analyzer. But more importantly, given current parsing technology, such morphological tags have been shown to not be very useful since they are extremely sparse.<sup>4</sup> It should be noted, however, that it is extremely useful to have these morphological tags in order to induce features. Accordingly, the LDC introduced the reduced tag set (RTS) of 25 tags. RTS is designed to maximize the performance of Arabic syntactic parsing. It is designed to be as close as possible to the tag set devised for

the English Wall Street Journal set. RTS masks case, mood, gender, person, definiteness for all categories. It maintains voice and some tense information for verbs (excluding future), and some number information for nouns, namely, marking plural vs. singular for nouns and proper nouns. Therefore, in the process it masks duality for nouns and number for all adjectives. We note the existence of variations on FULL and RTS in the literature [11], however none of these tag sets is as standardized as those produced by the LDC. Moreover, none of them was tested and evaluated in a task based evaluation.

### 4 Current POS tag set

The goal of this work is to find the optimal POS tag set for Arabic processing. Hence we examine the morphological features present in Arabic as expressed in the Arabic treebank. Since our base POS tag set is RTS, we focus on the morphological features that are being masked by RTS, namely, CASE, MOOD, DEFINITENESS, GENDER, NUMBER and PERSON. We examine the frequency of overt morphological features indicating one of these features in the surface orthography.<sup>5</sup> Hence, our guiding principle in deciding on the extensions to RTS rely on what is specified in FULL as a morphological feature coupled with salience of the morphological feature in the surface orthography of the words in text.<sup>6</sup> For this current study we only focused on open class words.<sup>7</sup> Taking all these into consideration, we study the ATB data, specifically ATB3v2. We calculate the statistics over the manually annotated and disambiguated POS tagged corpus. We make the following observations: The majority (80.7%) of CASE is expressed via short vowels that are not overtly present in the surface orthography in newswire MSA. All the overt cases of CASE (19.3%) are indefinite and they are overtly marked with nunation diacritic in the surface orthography. Likewise for all the MOOD cases, all the mood morphological markers are expressed via short vowels.<sup>8</sup> All the DEFINITENESS cases are overt in the surface form, either in the form of definites with *Al* Determiner (59.17% of the time), or through the *idafa* construction (40.8% of the time). As for indefinites, the majority of them is overt except for 2.1% of the indefinite data. All the cases where the GENDER is explicitly marked in the morphological features, the gender is overtly present in the orthography. There are 54947 cases of FEM marking in the morphological features as expressed

<sup>5</sup> There are clear inter-dependencies between some of these morphological features, but we adopt a simplifying assumption of independence among the features.

<sup>6</sup> We do not assume the existence of a morphological analyser in this work, even though we depend on FULL in the initial design of the POS tag set.

<sup>7</sup> We ignore morphological features on pronouns, whether object pronouns or possessive pronouns. Hence, both categories were consistently mapped to PRP.

<sup>8</sup> It should be noted that in the case of jussive mood, the form of some of the verbs changes morphologically and orthographically when the verb’s underlying form ends with what is dubbed in Arabic grammar as a weak letter [A, w, y]. However, without a mechanism relating the different forms of the verb to each other via roots and derivational patterns, it is quite difficult to surmise what the underlying form of a verb is.

<sup>3</sup> Excluding CASE and MOOD features.

<sup>4</sup> Dan Bikel, personal communication.

by FULL, and all of them have an overt feminine marker stem finally.<sup>9</sup> There are 5192 cases marked with MASC gender, all of them are plural and dual cases. All the nominals morphologically marked with NUMBER in the FULL tag set have an overt number marker. Only feminine singular entities are marked with SG. As for plural and dual numbers, they are marked on either feminine or masculine entities and there is an overt morphological NUMBER marker in the surface orthography. PERSON is marked on object pronouns, possessive pronouns, and subject inflections on verbs. PERSON is one of the highly confusable morphological features. The first person singular and the third person feminine singular are the same as a subject inflection for instance. In this study, we ignore the PERSON morphological feature even though it is explicitly present in the orthography, however, it by and large, affects a closed class set.<sup>10</sup> Taking into consideration these observations and the realistic underspecification of some of the morphological features in naturally occurring text, we devise a POS tag set that extends the existing RTS and masks some of the morphological features present in FULL. The new tag set comprises 75 tags explicitly marking GENDER, NUMBER and DEFINITENESS on nominals while maintaining the already existing features present on RTS.

Accordingly, the new tag set, ERTS, is derived from the FULL tag set where there is an explicit specification in the FULL tag for the different features encoded. DEFINITENESS is marked as a binary feature with a present (D) or an absent one. GENDER is marked with an F or an M or nothing, corresponding to Fem and Masc or the absence of gender marking, respectively. Number is encoded with (Du) for dual or an (S) for plurals or the absence of any marking for singular.<sup>11</sup> For example, Table 1 contrasts some words with the FULL morphological tag and their corresponding RTS and ERTS definitions.

## 5 Evaluation Methodology

In this paper, we present two evaluations: the impact of extending the POS tagset from RTS with 25 tags to ERTS with 75 tags on the POS tagger performance; the impact of ERTS in a task based evaluation in the context of base phrase chunking (BPC).

This paper is an extension to the Diab et al. (2007) work. We approach the POS tagging and the BPC problems as classification problems using a discriminative approach. We adopt a unified tagging perspective for both tasks. We address them using the same SVM experimental setup which comprises a standard SVM as a multi-class classifier.

We use the YAMCHA sequence model on the SVMs to take advantage of the context of the items being

<sup>9</sup> The stem could be either the full word, or the word excluding an enclitic.

<sup>10</sup> We also do not examine the other tenses present in the data such as the Future tense on verbs.

<sup>11</sup> We acknowledge that by not marking singulars, we conflate uncountables. Moreover, since all the nominals marked with SG in FULL are feminine nominals, the gender marking coupled with no feature encoding on the POS tag implicitly indicates that it is singular.

compared in a vertical manner in addition to the encoded features in the horizontal input of the vectors. Accordingly, in our different tasks, we define the notion of context to be a window of fixed size around the segment in focus for learning and tagging.

**POS tagging approach:** The ERTS tag set comprises 75 tags. For the current system, only 57 tags are instantiated. We adopt the POS tagger in Diab et al. 2007 based on the RTS tag set using the same machinery set up and features. We adopt the YAMCHA sequence model based on the TinySVM classifier [12]. The tagger trained for ERTS tag set (similar to that used for RTS) uses lexical features of +/-4 character n-grams from the beginning and end of a word in focus. The context for YAMCHA is defined as +/-2 words around the focus word. The words before the focus word are considered with their ERTS tags. The kernel is a polynomial kernel. We adopt the one-vs-all approach for classification, where the tagged examples for one class are considered positive training examples and instances for other classes are considered negative examples [1].

**Base Phrase Chunking approach:** In this task, we use a setup similar to that of [12] and Diab et al. (2007), with the IOB annotation representation. Inside *I* a phrase, Outside *O* a phrase, and Beginning *B* of a phrase. We designate 10 types of chunked phrases. The chunk phrases identified for Arabic are *ADJP*, *ADVP*, *CONJP*, *INTJP*, *NP*, *PP*, *PREDP*, *PRTP*, *SBARP*, *VP*. Thus the task is a one of 21 classification task (since there are *I* and *B* tags for each chunk phrase type, and a single *O* tag).

The training data is derived from the ATB using the ChunkLink software.<sup>13</sup> ChunkLink flattens the tree to a sequence of base (non-recursive) phrase chunks with their IOB labels. For example, a token occurring at the beginning of a noun phrase is labeled as *B-NP*. The following Table 2 Arabic example illustrates the IOB annotation scheme:

Tags	<i>B-VP</i>	<i>B-NP</i>	<i>I-NP</i>	<i>O</i>
Arabic	وقع	مساء	الجمعة	.
Translit	wqE	msA'	AljmEp	.
Gloss	happened	night	the-Friday	.

Table 2: An Arabic IOB annotation example

The BPC context is defined as a window of +/-2 tokens centered around the focus word where all the features for the specific condition are used and the tags for the previous two tokens before the focus token are also considered.

In our BPC experiments, we vary two factors in our feature sets: the POS tag set, and the presence or absence of explicit morphological features. We have three possible tag sets: RTS, ERTS and FULL. We define a set of 6 morphological features (and their possible values): CASE (*ACC*, *GEN*, *NOM*, *NULL*), MOOD (*Indicative*, *Jussive*, *Subjunctive*, *NULL*), DEF (*DEF*,

<sup>13</sup> <http://ilk.uvt.nl/sabine/chunklink>

		Gloss	FULL	RTS	ERTS
حصيلة	<i>HSyIp</i> <sup>12</sup>	‘results’	NOUN+NSUFF_FEM_SG+CASE_IND_NOM	NN	NNF
نهائية	<i>nhA}yp</i>	‘final’	ADJ+NSUFF_FEM_SG+CASE_IND_NOM	JJ	JJF
حادث	<i>HAdv</i>	‘accident’	NOUN+CASE_DEF_ACC	NN	NNM
النار	<i>AlnAr</i>	‘the-fire’	DET+NOUN+CASE_DEF_GEN	NN	DNNM
الجماعي	<i>AljmAEy</i>	‘group’	DET+ADJ+CASE_DEF_GEN	JJ	DJJM
شخصين	<i>\$xSyn</i>	‘two-persons’	NOUN+NSUFF_MASC_DU_GEN	NN	NNMDu

Table 1: Examples of POS tag sets

INDEF, NULL), NUM (Sing, Dual, Plural, NULL), GEN (Fem, Masc, NULL), PER (1, 2, 3, NULL).

From the intersection of the two factors, we devise 10 different experimental conditions. The conditions always have one of the POS tag sets and either no explicit features (noFeat), all explicit features (allFeat), or some selective features of: CASE, MOOD and PERSON (CASE\_MOOD\_PER), or DEFINITENESS, GENDER, and NUMBER (DEF\_GEN\_NUM). Therefore, the experimental conditions are as follows: RTS-noFeat, RTS-allFeat, RTS-CASE\_MOOD\_PER, RTS-DEF\_GEN\_NUM, ERTS-noFeat, ERTS-allFeat, ERTS-CASE\_MOOD\_PER, ERTS-DEF\_GEN\_NUM, FULL-noFeat, FULL-allFeat.

## 6 Experiments and Results

### 6.1 Data

The dev, test and training data are obtained from ATB1v3, ATB2v2 and ATB3v2 [14]. We adopt the same data splits introduced by [4]. The corpora are all news genre.

We use the unvocalized Buckwalter transliterated version of the ATB. For both POS tagging and BPC, we use the gold annotations of the training and test data for the preprocessing required. Hence, for POS tagging, the training and test data are both gold tokenized. And for BPC, the POS tags, the morphological features, and, the tokenization are all gold. We derive the gold ERTS deterministically from the FULL set for the BPC results reported here.

The IOB annotations on the training and gold evaluation data are derived using the modified *Chunklink* output [7].

### 6.2 SVM Setup

We use the default values for YAMCHA with the C parameter set to 0.5. The tool accepts multiple features. It has a degree 2 polynomial kernel. YAMCHA adopts a one-vs-all binarization method.

### 6.3 Evaluation Metric

Standard metrics of Accuracy (Acc.), Precision, Recall, and  $F_{\beta=1}$ , on the test data are utilized. For both POS tagging and BPC, we use the CoNLL shared task evaluation tools.<sup>14</sup>

<sup>14</sup> <http://cnls.uia.ac.be/conll2003/ner/bin/conlleval>

## 6.4 Results

### 6.4.1 POS Tagging Results

Table 3 shows the results obtained with the YAMCHA based POS tagger, POS-TAG, and the results obtained with a simple baseline, BASELINE. BASELINE is a supervised baseline, where the most frequent POS tag associated with a token from the training data is assigned to it in the test set, regardless of context. If the token does not occur in the training data, the token is assigned the NN tag as a default tag.

POS system	Acc.%
RTS-POS-TAG	96.15
ERTS-POS-TAG	96.13
BASELINE	86.5

Table 3: Results of POS-TAG on two different tag sets RTS and ERTS

Both POS-TAG systems clearly outperform the most frequent baseline. The results obtained clearly indicate that ERTS-POS-TAG with 57 instantiated tags yields a comparable result to RTS-POS-TAG. Looking closely at the data, the worst obtained results were for the NO\_FUNC category, as it is randomly confusable with almost all POS tags. Then, the imperative verbs are mostly confused with passive verbs 50% of the time, however the data only comprises 8 imperative verbs. VBN, passive verbs, yields an accuracy of 68% only. However the most frequent baseline for VBN is 21%. VBN is a most difficult category to discern in the absence of the passivization diacritic which is naturally absent in unvowelized text. The overall performance on the nouns and adjectives is relatively high, however, the errors in this categories are almost always present due to the inherent ambiguity in nominals and the fact that almost all Arabic adjectives could be used as nouns.<sup>15</sup>

### 6.4.2 Base Phrase Chunking (BPC) Results

Table 4 illustrates the overall obtained results by the BPC system over the different experimental conditions.

All the  $F_{\beta=1}$  results yielded by ERTS POS tag set outperform their counterparts using the RTS POS tagset. In fact, ERTS-noFeat condition outperforms all other conditions in our experiments.

<sup>15</sup> This inherent ambiguity leads to inconsistency in the ATB gold annotations.

Cond.	$F_{\beta=1}$	Cond.	$F_{\beta=1}$
RTS-noFeat	95.41	ERTS-noFeat	<b>96.33</b>
RTS-CASE_MOOD_PER	95.73	ERTS-CASE_MOOD_PER	<b>96.32</b>
RTS-DEF_GEN_NUM	95.8	ERTS-DEF_GEN_NUM	<b>96.33</b>
RTS-allFeat	95.97	ERTS-allFeat	<b>96.25</b>
FULL-noFeat	96.29	FULL-allFeat	96.22

**Table 4:** Overall  $F_{\beta=1}$  results yielded for the different BPC experimental conditions

We note that adding morphological features to the RTS POS tag set helps the performance slightly as we see a sequence of small jumps in performance from RTS-noFeat (95.41) to RTS-allFeat (95.97). However adding these features to the ERTS and FULL conditions does not help. In fact, in both the allFeat conditions, we note a slight decrease. This suggests that the features are not adding much information over and above what is already encoded in the POS tag set, and, in fact adding the explicit morphological features might be adding noise.

There is no significant difference between using ERTS and FULL in the overall results. However, we note that ERTS slightly outperform the FULL conditions. This may be attributed to the consistency introduced by ERTS over FULL, i.e., if FULL is not consistent in assigning CASE or MOOD or PER, for instance, ERTS, being insensitive to these features is able to mask these inconsistencies present in the FULL tag set.

## 7 Conclusions and Future Work

We presented a new POS tag set for Arabic processing, ERTS. ERTS comprises more tags than the reduced tag set RTS. ERTS explicitly encodes some of the salient morphological features of the Arabic language on nominals. We present an evaluation of ERTS in the context of POS tagging and in the context of higher level syntactic processing in the form of base phrase chunking. The results obtained illustrate that increasing the tag set three-fold from 25 tags in RTS to 75 in ERTS, we see no significant difference in the POS tagger performance. The new POS tagger achieves an overall accuracy of 96.13% compared to 96.15% on RTS using the same underlying machinery and features. This result suggests that ERTS is close to an optimal level of linguistic representation for the rich morphological features overtly encoded in the surface orthography. Moreover, we illustrate the impact of the ERTS on BPC. ERTS-BPC outperforms RTS-BPC and FULL-BPC. In fact, explicitly encoding the morphological features together with RTS does not improve over using ERTS alone. The yielded results indicate that the BPC process is sensitive to the POS tag choice. ERTS seems to capture the right level of linguistic information sufficient for the BPC process. For future work, we would like to experiment with ERTS in the context of other tasks such as full syntactic parsing and information extraction. In the near future, our next set of experiments will examine closely other tense features on verbs and more specific encoding of features on pronouns.

## Acknowledgements

The author is partly funded by DARPA Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

## References

- [1] E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multi-class to binary: A unifying approach for margin classifiers. In *Proc. 17th International Conf. on Machine Learning*, pages 9–16. Morgan Kaufmann, San Francisco, CA, 2000.
- [2] T. Buckwalter. Buckwalter Arabic Morphological Analyzer Version 1.0, 2002. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Catalog No.: LDC2002L49.
- [3] R. Cakici. Automatic induction of a ccg grammar for turkish. In *Proceedings of ACL Student Research Workshop*, pages 73–78, Ann Arbor, Michigan, 2005.
- [4] D. M. D. N. H. O. R. S. S. Chiang. Parsing arabic dialects. In *Proceedings of EACL 2006*, pages 369–376, Trento, Italy, April 2006.
- [5] B. Cowan and M. Collins. Morphology and reranking for the statistical parsing of spanish. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 795–802, Vancouver, British Columbia, Canada, October 2005.
- [6] S. Dandapat, S. Sarkar, and A. Basu. Automatic part-of-speech tagging for bengali: An approach for morphologically rich languages in a poor resource scenario. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 221–224, Prague, Czech Republic, June 2007.
- [7] M. Diab. Improved arabic base phrase chunking with a new enriched pos tag set. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, pages 89–96, Prague, Czech Republic, June 2007.
- [8] M. G. Diab, Mona and N. Habash. Arabic diacritization in the context of statistical machine translation. In *MT Summit*, Copenhagen, Denmark, September 2007.
- [9] N. Habash and O. Rambow. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 573–580, Ann Arbor, Michigan, June 2005.
- [10] J. Hajič and B. Hladká. Tagging inflective languages: Prediction of morphological categories for a rich, structured tagset. In *COLING-ACL*, pages 483–490, 1998.
- [11] S. Khoja. Apt: Arabic part-of-speech tagger. In *Proc. of the NAACL Student Research Workshop*, 2001.
- [12] T. Kudo and Y. Matsumoto. Fast methods for kernelbased text analysis, 2003.
- [13] M. Maamouri, A. Bies, T. Buckwalter, M. Diab, N. Habash, O. Rambow, and D. Tabessi. Developing and using a pilot dialectal arabic treebank, 2006.
- [14] M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki. The penn arabic treebank : Building a large-scale annotated arabic corpus, 2004.

# Semantic Parsing of Modern Standard Arabic

Mona T. Diab  
CCLS  
Columbia University  
mdiab@cs.columbia.edu

Alessandro Moschitti  
DIT  
University of Trento  
moschitti@dit.unitn.it

## Abstract

Shallow approaches to text processing have been garnering a lot of attention recently. Specifically, shallow approaches to semantic processing are making large strides in the direction of efficiently and effectively deriving tacit semantic information from text. Semantic Role Labeling (SRL) is one such approach. SRL is the task by which arguments of a predicate are identified and classified. In this paper, we present a system for Arabic SRL. To our knowledge, this is the first system to address the problem of semantic parsing of Arabic. Our SRL system is an SVM based system using polynomial kernels. The system is evaluated on the released SEMEVAL 2007 development and test data. Given the size of the training data, the obtained results are very promising. The Arabic SRL system yields an  $F_{\beta=1}$  score of 94.06% on argument boundary detection and an overall  $F_{\beta=1}$  score of 81.43% on the complete semantic role labeling task using test data.

## Keywords

Semantic Role Labeling, Arabic Language

## 1 Introduction

Shallow approaches to text processing have been garnering a lot of attention recently. Specifically, shallow approaches to semantic processing are making large strides in the direction of efficiently and effectively deriving tacit semantic information from text. Semantic Role Labeling (SRL) is one such approach. With the advent of faster and powerful computers, more effective machine learning algorithms, and importantly, large data resources annotated with relevant levels of semantic information **FrameNet** [1] and **PropBank** corpora [11], we are seeing a surge in efficient approaches to SRL [3].

SRL is the process by which predicates and their arguments are identified and their roles defined in a sentence. For example, in the English sentence, ‘John likes apples.’, the predicate is ‘likes’, the first argument is the subject ‘John’, and the second argument is the object ‘apples’. ‘John’ bears the semantic role label *agent* and ‘apples’ bears the semantic role label *theme*. The labels may differ depending on the linguistic theory or annotated resource adopted. In **FrameNet**, for instance, ‘John’ is labeled the *liker* while ‘apples’ is labeled *likee*. However, in **PropBank**, ‘John’ is labeled *ARG0* and ‘apples’ is labeled *ARG1*.

There is a widely held belief in the NLP and computational linguistics communities that identifying and defining the roles of the arguments of predicates in a sentence has a lot of potential for and is a significant step toward improving important applications such as document retrieval, machine translation, question answering and information extraction [16]. However, effective ways for seeing this belief come to fruition requires a lot more research investment.

To date, most of the reported SRL systems are for English. Naturally, since most of the data resources exist for this language. We do see some headway for other languages such as German and Chinese [6, 19]. The systems for the other languages follow the successful models devised for English, e.g. [7, 8, 4, 20, 18, 14, 21, 9]. However, no SRL systems exist for Arabic.<sup>1</sup> With the release of the SEMEVAL 2007 Task 18 data,<sup>2</sup> from the **Pilot Arabic Propbank**, this map is about to change [5].

In this paper, we present a system for semantic role labeling for modern standard Arabic. To our knowledge, it is the first SRL system for a semitic language in the literature. It is based on a supervised model that uses support vector machines (SVM) technology for argument boundary detection and argument classification. It is trained and tested using the pilot Arabic Propbank data released as part of the SEMEVAL 2007 data. Given the lack of a reliable deep syntactic parser, in this research, we used gold trees from the Arabic Tree Bank (ATB) [13]. The system yields an F-score of 94.06% on the sub task of argument boundary detection and an F-score of 81.43% on the complete task, i.e. boundary plus classification.

This paper is laid out as follows: Section 2 presents facts about the Arabic language especially in relevant contrast to English; Section 3 presents the approach and system adopted for this work; Section 4 presents the experimental setup, results and discussion.

## 2 Arabic Language

Arabic is a very different language from English in several respects that are critical for a task such as SRL. Arabic is a semitic language. It is known for its templatic morphology where words are made up of roots and affixes. Clitics agglutinate to words. For instance, the surface word **وبحسنتهم** *wbHsnAthm*<sup>3</sup> ‘and by their

<sup>1</sup> In this paper, we use Arabic to refer to Modern Standard Arabic (MSA).

<sup>2</sup> <http://nlp.cs.swarthmore.edu/semeval/>

<sup>3</sup> We use the Buckwalter transliteration scheme to show romanized Arabic [2].

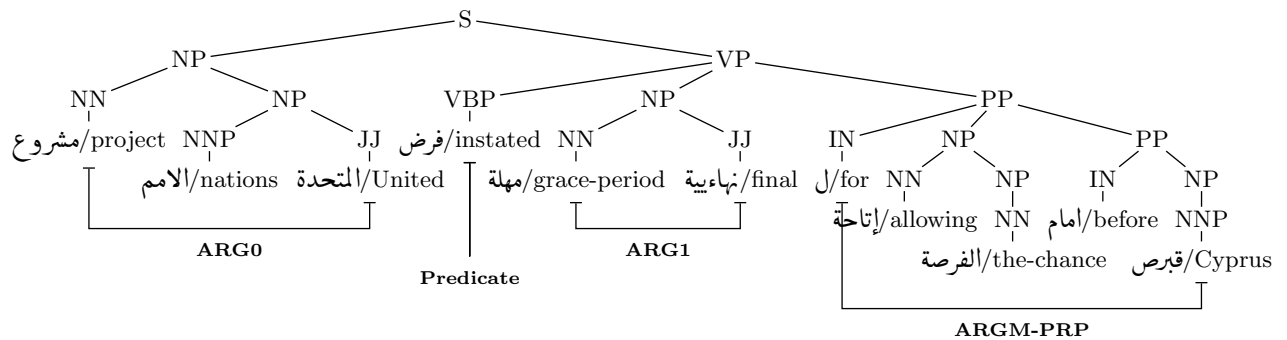


Figure 1: A syntactic parse tree of an Arabic sentence.

virtues[fem.]’, can be split into the conjunction *w* ‘and’, preposition *b* ‘by’, the stem *HsnAt* ‘virtues [fem.]’, and possessive pronoun *hm* ‘their’.

From the morphological standpoint, Arabic exhibits rich morphology. Similar to English, Arabic verbs are marked explicitly for tense, voice and person, however in addition, Arabic marks verbs with mood (subjunctive, indicative and jussive) information. For nominals (nouns, adjectives, proper names), Arabic marks case (accusative, genitive and nominative), number, gender and definiteness features. Depending on the genre of the text at hand, not all of those features are explicitly marked on naturally occurring text.

Arabic writing is known for being underspecified for short vowels. Some of the case, mood and voice features are marked only using short vowels. The amount of short vowels explicitly present in writing depends on genre and domain. For example, if the genre of the text were religious such as the Quran, or pedagogical such as children’s books, the orthography would be fully specified for all the short vowels to enhance readability and disambiguation.

From the syntactic standpoint, Arabic, differently from English (as well as, German and Chinese, for that matter) is considered a pro-drop language, where the subject of a verb may be implicitly encoded in the verb morphology. Hence, we observe sentences such as *اكل البرتقال Akl AlbrtqAl* ‘ate-[he] the-oranges’, where the verb *Akl* encodes that the subject is a 3rd person masculine singular. This sentence is exactly equivalent to *هو اكل البرتقال hw Akl AlbrtqAl* ‘he ate the-oranges’. In the Arabic Treebank (ATB), we observe that 40% of all sentences are pro-dropped for subject [12].

Also Arabic is different from English in that it exhibits a larger degree of free word order. For example, Arabic allows for both subject-verb-object (SVO) and verb-subject-object (VSO) argument orders.<sup>4</sup> In the ATB, we observe an equal distribution of both VSO and SVO orders each equally 30% of the time. An example of an SVO sentence is *الرجال اكلوا البرتقال AlrjAl AklwA AlbrtqAl* ‘the-men ate-them the-oranges’, this is contrasted with *اكل الرجال البرتقال Akl AlrjAl AlbrtqAl* ‘ate the-men the-oranges’.

Arabic exhibits more complex noun phrases than in English mainly to express possession. These constructions are known as *idafa* constructions. In these complex structures an indefinite noun maybe followed by a definite noun. For example, *رجل البيت rjl Albyt*

‘man the-house’ meaning ‘man of the house’. Therefore, Arabic does not have a special prepositional use to express possession in a manner similar to English.

### 3 A basic SRL system for Arabic

The previous section has shown some main differences between Arabic and English suggesting that an optimal model should take into account specific characteristics of Arabic. However, a remarkable amount of research has already been done in SRL and we can capitalize on it to design a basic and effective SRL system for Arabic. The idea is to use the technology developed for English language and verify if it is suitable for a preliminary system for Arabic.

Our adopted SRL models use Support Vector Machines to implement a two step classification approach, i.e. boundary detection and argument classification. Such models have been already investigated in [17, 15] and their description is hereafter reported.

#### 3.1 Predicate Argument Extraction

The extraction of predicative structures is based on the sentence level. Given a sentence, its predicates, as indicated by verbs, have to be identified along with their arguments. This problem is usually divided in two subtasks: (a) the detection of the target argument boundaries, i.e. the span of the argument words in the sentence, and (b) the classification of the argument type, e.g. *Arg0* or *ArgM* for Propbank annotation style or *Agent* and *Goal* for the FrameNet annotation style.

The standard approach to learn both the detection and the classification of predicate arguments is summarized by the following steps:

1. Given a sentence from the *training-set*, generate a full syntactic parse-tree;
2. let  $\mathcal{P}$  and  $\mathcal{A}$  be the set of predicates and the set of parse-tree nodes (i.e. the potential arguments), respectively;
3. for each pair  $\langle p, a \rangle \in \mathcal{P} \times \mathcal{A}$ :
  - extract the feature representation set,  $F_{p,a}$ ;
  - if the subtree rooted in  $a$  covers exactly the words of one argument of  $p$ , put  $F_{p,a}$  in  $T^+$

<sup>4</sup> MSA less often allows for OSV, or OVS.



Feature Name	Description
Predicate	Lemmatization of the predicate word
Path	Syntactic path linking the predicate and an argument, e.g. NN↑NP↑VP↓VBX
Partial path	<i>Path</i> feature limited to the branching of the argument
No-direction path	Like <i>Path</i> , but without traversal directions
Phrase type	Syntactic type of the argument node
Position	Relative position of the argument with respect to the predicate
Voice	Voice of the predicate, i.e. active or passive
Head word	Syntactic head of the argument phrase
Verb subcategorization	Production rule expanding the predicate parent node
Head word POS	POS tag of the argument node head word (less sparse than Head word)
Syntactic Frame	Position of the NPs surrounding the predicate
First and last word/POS	First and last words and POS tags of candidate argument phrases

**Table 1:** Standard linguistic features employed by most SRL systems.

(positive examples), otherwise put it in  $T^-$  (negative examples).

For instance, in Figure 1, for each combination of the predicate *imposed* with the nodes NP, S, VP, VPB, NNP, NN, PP, JJ or IN the instances  $F_{instated,a}$  are generated. In case the node *a* exactly covers "project nations United", "grace-period final" or "for allowing the chance before Cyprus",  $F_{p,a}$  will be a positive instance otherwise it will be a negative one, e.g.  $F_{instated,IN}$ .

The  $T^+$  and  $T^-$  sets are used to train the boundary classifier. To train the multi-class classifier,  $T^+$  can be reorganized as positive  $T_{arg_i}^+$  and negative  $T_{arg_i}^-$  examples for each argument  $i$ . In this way, an individual ONE-vs-ALL classifier for each argument  $i$  can be trained. We adopt this solution, according to [17], since it is simple and effective. In the classification phase, given an unseen sentence, all its  $F_{p,a}$  are generated and classified by each individual classifier  $C_i$ . The argument associated with the maximum among the scores provided by the individual classifiers is eventually selected.

The above approach assigns labels independently, without considering the whole predicate argument structure. As a consequence, the classifier output may generate overlapping arguments. Thus, to make the annotations globally consistent, we apply a disambiguating heuristic that selects only one argument among multiple overlapping arguments.

### 3.2 Features

The discovery of relevant features is, as usual, a complex task. However, there is a common consensus on the set of basic features. These standard features, firstly proposed in [7], refer to unstructured information derived from parse trees. e.g. *Phrase Type*, *Predicate Word* or *Head Word*.

For this preliminary Arabic SRL system, we adopt the features described in Table 1 presented in [7, 17, 21]. For example, the *Phrase Type* indicates the syntactic type of the phrase labeled as a predicate argument, e.g. NP for *Arg1* in Figure 1. The *Parse Tree Path* contains the path in the parse tree between the predicate and the argument phrase, expressed as a sequence of nonterminal labels linked by direction (up or down) symbols, e.g. VPB ↑ VP ↓ NP for *Arg1* in Figure 1. The *Predicate Word* is the surface form of the verbal predicate, e.g. *imposed* for all arguments.

## 4 Experiments

In these experiments, we investigate if the technology proposed in previous work for automatic SRL of English texts is suitable for Arabic SRL systems. From this perspective, we test each SRL phase, i.e. boundary detection and argument classification, separately.

The final labeling accuracy that we derive using the official CoNLL evaluation metrics, [3] along with the official development and test data of SEMEVAL provides a reliable assessment of the accuracy achievable by our overall SRL model.

### 4.1 Experimental setup

We use the dataset released in the SEMEVAL 2007 Task 18 on Arabic Semantic Labeling, which in turn is derived from the Pilot Arabic Propbank [5]. Such data covers the 95 most frequent verbs in the Arabic Treebank III ver. 2 (ATB). The ATB consists of MSA newswire data from Annhar newspaper from the months of July through November of 2002. All our experiments are carried out with gold trees.

An important characteristic of the dataset is the use of unvowelized Arabic in the Buckwalter transliteration scheme. The data comprises a development set of 886 sentences, a test set of 902 sentences, and a training set of 8,402 sentences. The development set comprises 1725 argument instances, the test data comprises 1661 argument instances, and training data comprises 21,194 argument instances. These instances are distributed over 26 different role types as described in Table 2. The training instances of the boundary detection task also include parse-tree nodes that do not correspond to correct boundaries. Such nodes/instances amount to more than 700K instances. For efficiency considerations, we experiment with a randomly sampled set of 350K instances of them. The experiments use SVM-light software [10]. For the boundary classifier, we use a polynomial kernel with the default regularization parameter and a cost-factor equal of 1.

### 4.2 Model Parameterization

In this phase, we tune the SVM-based classifiers on the development set. Several parameters could be investigated, e.g. the cost-factor (-j option in SVM-light) or the trade-off between generalization error and margin (-c option). However, although such parameters have been shown to be critical to the classification accuracy, they provide little information on the system outcome, (i.e. the relation between them and the data is rather

	#train	#dev	#test
ARG0	6,328	227	256
ARG0-STR	70	8	5
ARG1	7,858	702	699
ARG1-PRD	38	2	3
ARG1-STR	172	23	13
ARG2	1,843	191	180
ARG2-STR	32	5	4
ARG3	164	13	12
ARG4	15	0	4
ARGM	79	6	1
ARGM-ADV	994	103	115
ARGM-BNF	53	5	7
ARGM-CAU	89	12	11
ARGM-CND	38	6	3
ARGM-DIR	25	3	1
ARGM-DIS	56	8	5
ARGM-EXT	21	0	1
ARGM-LOC	711	82	61
ARGM-MNR	623	85	55
ARGM-NEG	529	76	39
ARGM-PRD	77	14	12
ARGM-PRP	343	42	27
ARGM-TMP	1,347	96	107
R-ARG0	0	14	36
R-ARG1	0	1	3
Total	21,194	1,725	1,661

**Table 2:** Distribution of training, development and test instances on the different role types.

obscure). Moreover, if they are too accurately tuned they may cause overfitting.

Given these premises, we opt to study another relevant parameter, namely the degree of polynomial in the kernel function. This, as suggested in [14], allows us to study if the feature combinations (e.g. feature conjunctions) help in learning the differences between different argument types. For example, in [17, 14], adding the conjunction of two features improved the basic model by almost 5 percent points.

Figure 2.a reports the F1 of the SVM boundary classifier on the development set. We note that as we introduce conjunctions, i.e. a degree larger than 2, the F1 increases by more than 3 percentage points. Thus, not only the English features are meaningful for Arabic but also their combinations are important, revealing that both languages share an underlying semantic structure.

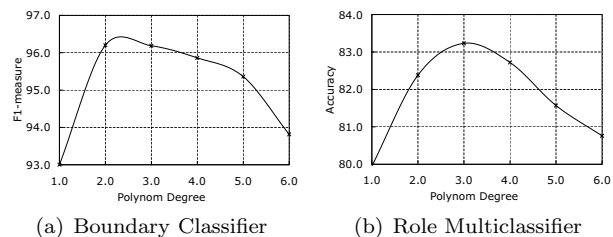
Figure 2.b reports the accuracy of the SVM multiclassifier according to different polynomial degrees. Such plots synthesizes the average impact of the polynomial degree among the accuracy of all arguments. Again, we note that a degree larger than 2 produces an improvement of more than 3 percent points. Thus, for the design of Arabic SRL system based on SVMs, a polynomial kernel seems appropriate.

### 4.3 Results

The previous results should only be considered as an indicative outcome since: (a) optimal parameters derived on the development set may differ from those of the test set and (b) they only refer to the classification of instances which is a easier task than sentence labeling. Indeed, the classifier can correctly classify a tree node as a correct boundary but then if it classifies another node (e.g. its parent) as a correct boundary there will be an overlap. The correctness of the final sentence annotation will depend on the choice made for resolving such overlap.

Therefore, the only way to derive realistic results is to annotate a sentence and compare it with the annotation of the gold standard. This can be reliably done by means of the official CoNLL evaluator, available at <http://www.lsi.upc.es/~srlconll/soft.html>.

Table 3 shows the F1 obtained with the above procedure on the development and test data. As ex-



**Figure 2:** Impact of the Polynomial Kernel degree.

pected, we note that the F1 on the development set, i.e. 93.68%, is lower than the highest value of the plot in Figure 2.a, i.e. about 96%. Also, it is slightly lower than the result on the test set, i.e. 94.06%. This confirms that to classify nodes is different than to annotate sentences and also suggests that the test data is *easier* than the development set.

Similar behavior is observed for the role classification task in Table 4. Again, the overall F1 on the development set, i.e. 77.85% is lower than the highest value of the plot in Figure 2.b, i.e. about 83%. Also, it is lower than the result on the test set 81.43%. Therefore, the test data is definitely *easier* than the development set.

## 5 Discussion

The obtained results are quite promising given that the available data is small compared to the English data sets. The Arabic training data only comprises a total of 21k argument instances compared to a typical English set of 250k instances. We acknowledge the caveat that we use gold parses and the typical English SRL system reports results on automatic parses as well. The fact that the test data yields a better performance across the board suggests that the test set is easier or more consistent than the dev set.

Our best scores are obtained on ARG0 (96.69) and ARG1 (90.79) followed by ARGM-NEG (88.37), ARGM-TMP (86.83), ARGM-LOC (76.6), ARGM-PRP (75) and ARG2 (72.28). We observe that these are the most frequent argument types in the training data.

Hence, in general, the F1 of the arguments seems to follow the English SRL behavior as their lower value depends on the lower number of available training examples (compare with Table 2). However, we note that there is not always a direct correlation between the number of training instances and the performance. For instance there are more instances for ARG2 than ARGM-LOC, but ARG2 yields a lower F-score. This may be attributed to the fuzzy definition of what constitutes an ARG2 argument. ARG2 is always confusable with the ARGM-PRP argument.

Regarding the F1 of individual arguments, we note that, as for English SRL, ARG0 shows high values, i.e. 95.42% and 96.69% on the development and test sets, respectively. On the contrary, ARG1 seems more difficult to be classified in Arabic than in English as it usually reaches an F1 close to the F1 of ARG0 whereas in this Arabic setting the F1 of ARG1 is only 89.83% (compare with 95.42% of ARG0). This may be attributed to the different possible orderings of the structure VSO vs VOS vs SVO. Also, it may be attributed

	Precision	Recall	$F_{\beta=1}$
Dev.	97.85%	89.86%	93.68
Test set	97.85%	90.55%	94.06

**Table 3:** Boundary detection F1 derived with CoNLL evaluator on the development and test sets.

	Precision	Recall	$F_{\beta=1}$
Overall	81.31%	74.67%	77.85
ARG0	94.40%	96.48%	95.42
ARG1	91.69%	88.03%	89.83
ARG1-PRD	50.00%	50.00%	50.00
ARG1-STR	20.00%	4.35%	7.14
ARG2	60.51%	61.78%	61.14
ARG3	66.67%	15.38%	25.00
ARGM	100.00%	16.67%	28.57
ARGM-ADV	46.39%	43.69%	45.00
ARGM-CND	66.67%	33.33%	44.44
ARGM-DIS	60.00%	37.50%	46.15
ARGM-LOC	69.00%	84.15%	75.82
ARGM-MNR	63.08%	48.24%	54.67
ARGM-NEG	87.06%	97.37%	91.93
ARGM-PRD	25.00%	7.14%	11.11
ARGM-PRP	85.29%	69.05%	76.32
ARGM-TMP	82.05%	66.67%	73.56

**Table 4:** Argument classification derived with CoNLL evaluator on the development set.

	Precision	Recall	$F_{\beta=1}$
Overall	84.71%	78.39%	81.43
ARG0	96.50%	96.88%	96.69
ARG0-STR	100.00%	20.00%	33.33
ARG1	92.06%	89.56%	90.79
ARG1-STR	33.33%	15.38%	21.05
ARG2	70.74%	73.89%	72.28
ARG3	50.00%	8.33%	14.29
ARGM-ADV	64.29%	54.78%	59.15
ARGM-CAU	100.00%	9.09%	16.67
ARGM-CND	25.00%	33.33%	28.57
ARGM-LOC	67.50%	88.52%	76.60
ARGM-MNR	54.17%	47.27%	50.49
ARGM-NEG	80.85%	97.44%	88.37
ARGM-PRD	20.00%	8.33%	11.76
ARGM-PRP	85.71%	66.67%	75.00
ARGM-TMP	90.82%	83.18%	86.83

**Table 5:** Argument classification derived with CoNLL evaluator on the test set.

to the fact that ARG0 is more predictable in the pro-drop cases (due to the fact that we are using gold parses which explicitly mark pro-drop nodes) amounting to 40% of the ATB data.

## 6 Conclusion

In this paper, we presented a first system for Arabic SRL system. The system yields results that are significantly better than the baseline, with 94.06% for argument boundary detection and 81.43% on argument classification.

For future work, we would like to experiment with explicit morphological features and different POS tag sets. Moreover, the results presented here are based on gold parses. We would like to experiment with automatic parses and shallower representations such as chunked data. Finally, we would like to experiment with more sophisticated kernels, i.e. the tree kernels described in [14], i.e. models that have shown a lot of promise for the English SRL process.

## Acknowledgements

Mona Diab is partly funded by DARPA Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA. Alessandro Moschitti has been partially funded by CCLS of the Columbia University.

## References

- [1] C. F. Baker, C. J. Fillmore, and J. B. Lowe. The Berkeley FrameNet project. In *COLING-ACL '98: Proceedings of the Conference, held at the University of Montréal*, 1998.
- [2] T. Buckwalter. Buckwalter Arabic Morphological Analyzer Version 1.0, 2002. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Catalog No.: LDC2002L49.
- [3] X. Carreras and L. Màrquez. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, Ann Arbor, Michigan, 2005.
- [4] J. Chen and O. Rambow. Use of deep linguistic features for the recognition and labeling of semantic arguments. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Sapporo, Japan, 2003.
- [5] M. Diab, M. Alkhalifa, S. ElKateb, C. Fellbaum, A. Mansouri, and M. Palmer. Semeval-2007 task 18: Arabic semantic labeling. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, Prague, Czech Republic.
- [6] K. Erk and S. Pado. Shalmaneser – a toolchain for shallow semantic parsing. 2006. Proceedings of LREC.
- [7] D. Gildea and D. Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002.
- [8] D. Gildea and M. Palmer. The necessity of parsing for predicate argument recognition. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics*, Philadelphia, PA, USA, 2002.
- [9] A. Haghighi, K. Toutanova, and C. Manning. A joint model for semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, Ann Arbor, Michigan, June 2005.
- [10] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, 1999.
- [11] P. Kingsbury and M. Palmer. Propbank: the next level of treebank. In *Proceedings of Treebanks and Lexical Theories*, 2003.
- [12] M. Maamouri, A. Bies, T. Buckwalter, M. Diab, N. Habash, O. Rambow, and D. Tabessi. Developing and using a pilot dialectal arabic treebank, 2006.
- [13] M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki. The penn arabic treebank : Building a large-scale annotated arabic corpus, 2004.
- [14] A. Moschitti. A study on convolution kernels for shallow semantic parsing. In *proceedings of the 42<sup>th</sup> Conference on Association for Computational Linguistic*, Barcelona, Spain, 2004.
- [15] A. Moschitti, A.-M. Giuglea, B. Coppola, and R. Basili. Hierarchical semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, Ann Arbor, Michigan, 2005.
- [16] A. Moschitti, S. Quarteroni, R. Basili, and S. Manandhar. Exploiting syntactic and shallow semantic kernels for question answer classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, 2007.
- [17] S. Pradhan, K. Hacioglu, V. Krugler, W. Ward, J. H. Martin, and D. Jurafsky. Support vector learning for semantic argument classification. *Machine Learning*, 60:1-3:11–39, 2005.
- [18] S. Pradhan, K. Hacioglu, W. Ward, J. H. Martin, and D. Jurafsky. Semantic role parsing: Adding semantic structure to unstructured text. In *Proceedings of the International Conference on Data Mining (ICDM-2003)*, Melbourne, USA, 2003.
- [19] H. Sun and D. Jurafsky. Shallow Semantic Parsing of Chinese., 2004. In Proceedings of NAACL-HLT.
- [20] C. A. Thompson, R. Levy, and C. Manning. A generative model for semantic role labeling. In *14th European Conference on Machine Learning*, 2003.
- [21] N. Xue and M. Palmer. Calibrating features for semantic role labeling. In D. Lin and D. Wu, editors, *Proceedings of EMNLP 2004*, Barcelona, Spain, 2004.

# Determining Ambiguity Classes for Part-of-Speech Tagging

Markus Dickinson  
Indiana University  
1021 E. Third Street  
Bloomington, IN 47405  
*md7@indiana.edu*

## Abstract

We examine how words group together in the lexicon, in terms of ambiguity classes, and use this information in a redefined tagset to improve POS tagging. In light of errors in the training data and a limited amount of annotated data, we investigate ways to define ambiguity classes for words which consider the lexicon as a whole and predict unknown uses of words. Fitting words to typical ambiguity classes is shown to provide more accurate ambiguity classes for words and to significantly improve tagging performance.

## Keywords

Part-of-speech tagging, Corpus annotation

## 1 Introduction

From one perspective, part of speech (POS) tagging is a task which attempts to assign a morphosyntactic label to each token in a text. From another, it is an attempt to say which word instances belong to the same class, i.e., function in the same way. The effect of this is that a tag serves to group word types together; thus, a tag can be thought of as shorthand for a set of words [5, 23]. Depending on the tagset, these word sets can be disparate; a set may contain words which are all adjectives, but some are only predicative, while some take obligatory complements. Viewed in this way, we can ask whether the POS tags in a tagset actually capture the relevant distinctions.

If the same POS tag for one collection of words behaves differently than for another, the tagset can be redefined to improve tagging (cf. [15]), given that the success of a tagger depends in part on what distinctions it learns [11, 14]. Because tags represent sets of words, to redefine a tagset, one can examine the regularities in the lexicon, in order to see whether the collections of words are appropriately grouped.

The regularities we focus on involve which ambiguity class to assign to a word, i.e., the set of “possible” tags. Ambiguity classes capture the distinctions which make tagging non-trivial. Pinpointing the most prominent classes to be disambiguated groups words with the same difficulties together and places the focus on the approximately 3% of tagging cases which a tagger gets wrong and which affect parsing (cf. [17, 10, 6]).

To determine a word’s ambiguity class, it seems like we can simply extract it from the data, but this is problematic. First, because of errors in the annotated training data, a word might have too many “possible” tags, some of which are impossible. Secondly, with limited annotated data, many possible tags are never observed. Finally, even if we had sufficient error-free data, some tags are still quite rare and not completely indicative of a word. A word can mostly pattern like other words, but with some exceptions. Thus, determining what class a word belongs to becomes an issue of primary importance and the focus of this paper. This is a relevant issue not only for complete disambiguation, but also for multi-tagging tasks, where a word may have more than one tag (e.g., [6]).

We here investigate grouping words by ambiguity classes in the context of POS tagging, and we use a notion of *typicality* to overcome the three problems outlined above. Typical ambiguity classes model the regularities, ignoring the exceptions; new tags are predicted based on a word’s similarity to a typical class; and tags which are atypical may be erroneous. Our starting point is a method of tagset modification for POS annotation error correction, described in section 2, since it uses ambiguity classes to deal with difficult tagging cases. In section 3, we turn to our POS tagging model: after filtering tags in section 3.1, we describe how to identify typical ambiguity classes in section 3.2 and subsequently merge classes in section 3.3, thereby predicting unknown uses of tags. For each step, we witness a gradual improvement in tagging accuracy, resulting in significant improvement. This is achieved despite basing the changes on information in the lexicon and not on contextual information.

## 2 Tagset modification

Using a modified tagset to deal with common ambiguities, Dickinson [12] develops a tagging method to correct POS annotation errors. Influenced by the “confusing parts of speech,” or “difficult tagging distinctions,” in POS annotation guidelines [21], the method is based on the idea that knowing the problematic distinction for a given corpus position can assist in tagging it.

The crucial insight is that the guideline diagnostics used, in the case of the Penn Treebank [16], to tell, e.g., RP (particle) from IN (preposition) are not the same as the ones used to tell RP from RB (adverb). These RP uses have differences in distribution based

on which distinction is involved, and thus the set of RP words can be subgrouped.

To do this, the tagset is altered while training, replacing each relevant tag with a *complex ambiguity tag*, indicating that word’s *ambiguity class* and the tag at that corpus position. At a given corpus position, a word is given a complex ambiguity tag if it applies; otherwise, it retains its simple tag. This tag-splitting method (cf. [1]) results in examples like (1a) becoming (1b) in the Wall Street Journal (WSJ) part of the Penn Treebank 3.

- (1) a. ago/RB
- b. ago/<IN/RB,RB>

Two constraints are used to determine the distinctions, or ambiguity classes, for words. First, low-frequency tags are filtered from consideration for an ambiguity class, in order to deal with some errors. For example, *an* uses the simple tag DT (determiner) instead tags with the class COMMA/DT because the comma tag only occurs once out of 4211 times. Secondly, only the ambiguity classes for the positions flagged by an error detection phase [13] are considered. Thus, a variation between JJ (adjective) and PRP (personal pronoun) for *ours* is not put into the model because such a variation never occurs for errors.

### 3 Selecting ambiguity classes

We choose to adapt this framework for POS tagging work since the emphasis on ambiguity classes finds regularities beyond the distinctions in the tagset. POS tagging, however, differs in crucial ways from error correction. First, training data and testing data are disjoint for tagging, whereas they are identical for error correction (i.e., the entire corpus is used for both), forcing us to consider unknown uses of words in ambiguity class assignment. Secondly, whereas automatic correction focuses only on positions flagged as potential errors, POS tagging is for an entire text, giving a large number of distinctions. In assigning ambiguity classes for POS tagging, therefore, we need new criteria to determine what words group together. Instead of asking whether it is involved in an error, we suggest typicality as a criterion for the relevance of an ambiguity class: is it a common distinction?

Following Toutanova et al. [25], we use the WSJ corpus merged data, sections 00-18 for training, sections 19-21 for development, and sections 22-24 for testing. All tagset modification is done to training data only, and tags are mapped back to Penn Treebank tags for evaluating tagger precision (see section 4.2).

We could assign ambiguity classes based on all possible tags for a word (cf. [7]), but this will not generalize well. The problem is that this method results in too many specific classes, which serves to isolate words in the lexicon. With 280 ambiguity classes and 887 total tags, we find unique classes like JJ/JJR/RB/RBR/VB for the word *further*. To better group words together, we need to limit what ambiguity classes are possible.

In fact, we observe that adding data makes this problem worse. We cumulatively calculated the set of ambiguity classes in the corpus, section by section, as

shown in figure 1. The set of ambiguity classes grows indefinitely, albeit slowly. As more and more instances are added to the corpus, there is a greater tendency for rare cases to emerge and for errors to be introduced.

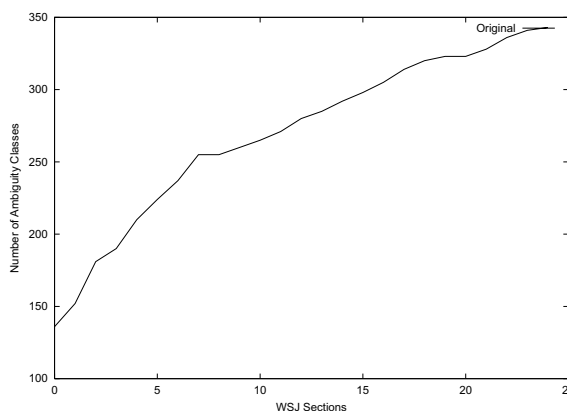


Fig. 1: Growth rate of ambiguity classes

Our task thus becomes one of restricting the ambiguity class for each word. We find that restrictions specific to an individual word (e.g., filtering) are insufficient (section 3.1), requiring global restrictions which consider the lexicon as a whole (section 3.2). Fitting words to these global, typical classes indicates which unseen tags are appropriate (section 3.3).

#### 3.1 Filtering

The first restriction directly addresses the problems of erroneous tags and low-frequency tags that are not indicative of a word by filtering out tags occurring less than 10% of the time for a word (cf. [9]). As an example of how this handles errors, instead of CD/DT/JJ/NN/NNP/VBP for *the*—which has five erroneous tags—we have only the correct DT. As an example of a non-indicative tag, the word *all* varies between DT, PDT (predeterminer), and RB, but RB accounts for 3.7% of the cases (38/1017); after filtering, we obtain DT/PDT. It is not that RB is wrong; it is that DT and PDT are its most prototypical uses and by restricting our focus to only DT and PDT, we are able to group *all* with its capitalized form *All*. Thus, *all* now has three possible tags: <DT/PDT,DT>, <DT/PDT,PDT>, and RB.

This tagging model uses 155 ambiguity classes, but two problems remain. First, we still have some highly specific classes; many words with similarities to other words pattern only like themselves. For instance, *Put* (which only appears 17 times) is alone in having the ambiguity class JJ/NN/VB/VBD/VBN. Furthermore, many classes seem ad hoc; ambiguity classes like \$/NNP (for the token *C*) are not problematic variations for annotators [21]. Secondly, it is not clear how filtering by itself is a sufficient test of indicativeness.

### 3.2 Identifying typical tag classes

Filtering is a local task, but to evaluate whether a word's ambiguity class captures a true regularity, i.e., is like others, we need to consider the whole lexicon. Since we want to capture regularities—i.e., repeated uses of the same class—we use a frequency-based criterion, to determine if an ambiguity class is *typical*.

For this (*Typical*) model, after filtering tags, we keep only the ambiguity classes with more than  $n$  tokens, where  $n$  is empirically determined. So, a class such as JJ/RB, which has 59 word types realized with 5054 tokens, is ranked above NN/NNP (common/proper noun), with 378 types and 4217 tokens. Such a token-based measure best reflects the prevalent patterns in the training corpus (i.e., the typical classes) and which decisions the tagger sees repeatedly. For the remainder of this discussion, we will use the model with  $n$  equal to 400, which uses 38 ambiguity classes (see section 4 for a comparison of different values of  $n$ ).

As a side effect, we discover that some regularities are word-specific; in other words, some classes are essentially lexicalized. For example, *that* is the only DT/IN/WDT word, with 7699 tokens, and thus tagging an item as <DT/IN/WDT,DT> is the same as tagging it <*that*,DT>. Others have shown such lexicalization to be useful in tagging (e.g., [20]). Our approach differs by automatically finding words which do not pattern with any others, and because we filter out non-indicative tags, there is a slight difference between our “lexicalized” classes and the general notion of lexicalization. Consider the word *like*, the only IN/VB word; in our model, it has four possible tags: <IN/VB,IN>, <IN/VB,VB>, JJ, and VBP. The JJ and VBP (present tense verb) cases get grouped with other JJ and VBP cases, instead of receiving the tags <*like*,JJ> and <*like*,VBP>. It patterns uniquely in being the only word which can be a preposition (IN) and a verb (VB); the other uses can be grouped with other corpus instances.

**Examining the lexicon** This model is an improvement, but it is inadequate. Similar words are often in different classes. For example, *explained* is VBD/VBN (past tense verb/past participle), while *classified* is JJ/VBD/VBN and *accomplished* is JJ/VBN, yet all seem to have the same possibilities. JJ never appears in the training data for *explained*, yet it is a possible tag. Consider also a word like *accepted*: it varies between JJ, VBD, and VBN, yet JJ is only 1 of the 41 occurrences, so the ambiguity class becomes VBD/VBN after filtering. Yet, this instance of JJ is correct (*future have become an accepted/JJ part of the financial landscape*). In cases like this, it becomes apparent that basing indicativeness on individual frequency is inadequate: the tag is neither more nor less indicative of *accepted* than VBD or VBN. Many verbs that are VBD/VBN, whether because of filtering (cf. *accepted*) or lack of observation (cf. *explained*), should also have JJ as a possible tag.

We thus want some way to predict tags not observed in the training data and to overcome excessive filtering. The solution seems to be in performing a limited amount of merging of classes; for example, JJ/VBN, VBD/VBN, and JJ/VBD/VBN can be combined into

the superset class JJ/VBD/VBN.

### 3.3 Merging ambiguity classes

After grouping words by typical ambiguity classes, we then merge classes, based on which tags are predictable from which other tags (*Merge* model). To find the mappings from one ambiguity class into another, superset class, we calculated the ambiguity class for every word in a portion of the training data (sections 00-15), and observed which tags are added for each word in some held-out data (sections 16-18). For example, 18 NN/VB word types become NN/VB/VBP words when adding more data. With 16 sections for the base set of ambiguity classes, this ensures a relatively stable set of ambiguity classes, and using the held-out data ensures that we capture the relevant property: which tag is predictable from an ambiguity class? Once we automatically deduced the mappings (e.g., NN/VB  $\mapsto$  NN/VB/VBP), we use them to merge ambiguity classes together.

To account for noise and idiosyncratic behavior, we use a few simple restrictions: 1) The resulting ambiguity class must be a typical class. 2) The mapping occurs for at least two words in the held-out data since single-occurring mappings are not general. 3) The class is not very fertile, i.e., does not generate lots of tag possibilities. Specifically, no more than three other tags are allowed. 4) Only the highest-ranking mapping is used. For example, the twice-occurring VB/VBP  $\mapsto$  IN/VB/VBP is not used because VB/VBP already has a mapping. With this method, we also merge single tags into ambiguity classes—e.g., VB  $\mapsto$  VB/VBP. The full set of mappings can be seen in figure 2.

Original	New
NNPS	NNPS/NNS
VB	VB/VBP
VBP	VB/VBP
VBD	VBD/VBN
VBN	VBD/VBN
VBG	NN/VBG
VBZ	NNS/VBZ
JJ/VBN	JJ/VBD/VBN
VBD/VBN	JJ/VBD/VBN
NN/VB	NN/VB/VBP
NN/VBP	NN/VB/VBP
VB/VBP	NN/VB/VBP
NNP/NNPS	NNP/NNPS/NNS
NNP/NNS	NNP/NNPS/NNS

Fig. 2: Mappings for merging classes

This merging serves to counteract some filtering, by putting some filtered tags back into ambiguity classes. On the one hand, we filtered JJ from the ambiguity class for *accepted*, making it VBD/VBN, because JJ appears only once out of 41 times. Now, it gets put back in, making the class JJ/VBD/VBN. On the other hand, we filtered RBR (comparative adverb) from the ambiguity class of *trimmed*, making it VBD/VBN, because it occurs once out of 15 times: RBR is erroneous, and it stays out of the ambiguity class. By using a criterion other than frequency, we can begin to separate errors from rare instances, giving a good first step in

having more selective filtering, instead of simply filtering out low-frequency tags from a tagging model (e.g., [9, 22, 4]). This is especially important for methods which depend upon rare but correct instances [8].

Additionally, we predict tags for words which never had that tag in the data, but should have. The example of *trimmed* is another case where JJ is appropriate, and its class becomes JJ/VBD/VBN.

This method of merging can obviously be improved. We almost definitely will over-generalize since we do not take morphology or tag distributions into account—e.g., not all VBD words are also VBN (cf. *went*). Still, it is important to remember that these assignments are currently being used only as an indication of possibilities; in most contexts, we do not expect VBN to be a legitimate tag for *went*.

**Tag prediction** Merged ambiguity classes can now predict the presence of possible tags for a word because they may contain tags a word lacks. To add these tags directly to a tagging model is straightforward for a tagger with a transparent lexicon. For every word with a complex ambiguity class, we add a count of one for any tag which is predicted to appear but does not (*Merge+*). For example, the word *cheer* originally varied between NN and VB, with one occurrence of each. We now add a count of one for VBP since NN/VB/VBP is its merged ambiguity class. A count of one makes the tagger aware that this tag is possible without making any further claims.

The prediction of unknown tag uses is in the spirit of Toutanova and Manning [26], who “augmented [a tag dictionary] so as to capture a few basic systematic tag regularities that are found in English. Namely, for regular verbs the *-ed* form can be either a VBD or a VBN and similarly the stem form can be either a VBP or VB.” Our predictions, however, arise from a data-driven analysis of word groupings, instead of being hand-encoded. Either way, lexicon augmentation can be used as a sanity check on filtering noisy data.

## 4 Evaluation

There are two ways to evaluate the resulting models. First, to gauge whether the ambiguity classes are capturing true facts about these words, or whether they are over- or under-generalizing, some degree of qualitative analysis is needed. Secondly, to gauge the effectiveness of better groupings in the lexicon, we will see how the ambiguity classes affect the quality of POS tagging. This is only one way to use these groupings, however; given the confounding factor of being integrated into an already complicated tagging model, both kinds of evaluation are important.

### 4.1 Quality of ambiguity classes

To determine the quality of the ambiguity classes used, we need a test bed of words with all of their truly possible tags. Thus, we sampled 100 lexical entries (from sections 00-18), removed their tags, and hand-annotated the set of possible tags. To guide this process, we first gathered the list of all (unaltered) ambiguity classes from the lexicon, so that the annotator

could first mark a word’s most prominent tags and then consult the list to see which other tags are general possibilities.

We then took the entries from the original lexicon for the 100 words and compared their possible tags to the hand-created set. We found that 49 words matched this set, while 51 were missing tags. Thus, we can see that the task of predicting tags for known words is a high priority for POS lexicon coverage: over half the word types are missing at least one tag.

The Merge(+) model, on the other hand, has 39 such undergeneralizations with only one overgeneralization (*describes*, predicted to be NNS/VBZ instead of VBZ only), correctly changing words like *smile* from NN/VB to NN/VB/VBP and *bottling* from VBG to NN/VBG. There were also six cases which were closer to a correct distribution, even if they were still missing tags. The word *responding*, for example, was originally VBG, but is now NN/VBG; although its complete set of possible tags is JJ/NN/VBG, it is now improved. With 18 total improved words, we are successfully adding more possible tags, without adding much noise.

### 4.2 POS tagging results

Having shown that the ambiguity classes are successfully capturing the range of a word’s tag possibilities, we want to test the effectiveness of using them to group words for POS tagging. Corpus positions are assigned complex ambiguity tags where appropriate for training, using the splitting framework from section 2, and the complex tags are mapped back to their simple tags for evaluation. Thus, if the tagger assigns *ago* the tag <IN/RB,RB>, we map it to RB in order to compare it against the benchmark.

**Development data** As a baseline for the development data, the default version of the Hidden Markov Model (HMM) tagger TnT [3] obtains a precision of 96.48%. Using filtering at 10% to assign ambiguity classes for tag splitting, the tagging model has 96.63% precision on the development data, showing that a first pass at using ambiguity classes provides better performance.

Testing the different ambiguity class models, we present the results side-by-side in figure 3. The best results are for the Merge+ model, with a token cutoff of 400, giving a precision of 96.71%, an improvement gained by making fewer, more general classes than the Typical model and by extrapolating the ambiguity classes directly to the lexical entries. It is also important to note the overall trends: each model slightly improves upon the previous one, for all cutoff levels.

**Testing data** After developing the different models, we ran them on the testing data (sections 22-24 of the WSJ) for  $n = 400$ . As shown in figure 4, we see the same improvements as with the development data, demonstrating that the improvements are not specific to one data set. Using McNemar’s Test [18], the results for Merge+ are significantly higher ( $p < .001$ ) than for the Baseline.

$n$	Typical		Merge		Merge+
	Pre.	AC	Pre.	AC	Pre.
100	96.64%	56	96.65%	48	96.68%
200	96.64%	47	96.65%	41	96.69%
300	96.66%	42	96.66%	36	96.70%
400	96.66%	38	96.67%	33	<b>96.71%</b>
500	96.65%	34	96.66%	30	96.70%

**Fig. 3:** Results on sections 19-21 of the WSJ (Pre. = precision, AC = number of ambiguity classes)

Model	Development	Testing
Baseline	96.48%	96.46%
Typical	96.66%	96.65%
Merge	96.67%	96.66%
Merge+	96.71%	96.70%

**Fig. 4:** Results on sections 22-24 of the WSJ

We also wanted to see how applicable our models are to other genres of text. As with better unknown word tagging, predicting unknown uses of known words mitigates the need for more training data, by filling in some gaps of what has not been observed. Such methods are potentially more applicable to other genres of text: tag uses in one genre may not appear in another. Thus, we tested the models on the Brown corpus part of the Penn Treebank, as shown in figure 5. Using McNemar’s Test, the results for Merge+ are significantly higher ( $p < .001$ ) than for the Baseline.

Model	Precision
Baseline	94.60%
Typical	94.79%
Merge	94.81%
Merge+	94.93%

**Fig. 5:** Results on the Brown corpus

We see the same trends here, showing the methods developed here improve even on another corpus. The percentage gain is still somewhat small—0.33% from Baseline to Merge+—but with a larger corpus, we can better see the impact of the improvement, obtaining a reduction of 1550 errors (24,815-23,265).

**Discussion** The increase in tagging precision, from 96.46% to 96.70% on the testing data, is only 0.24% and is below the state-of-the-art precision of 97.33% by Shen et al. [24] on the same data. What is important to notice, however, is not the absolute accuracy of the particular method used here, but that we have seen significant improvement by examining how words group in the lexicon. Further, we have improved coverage of the lexicon by fitting words to typical ambiguity classes. This has the potential to make any tagger better represent possible tags for words.

We have obtained an improvement in performance without encoding a variety of features or changing a tagging algorithm. In fact, we have only generalized patterns of tags found across the lexicon; we

have not used any contextual information. The principles behind these techniques are applicable to any tagging method; how they are applied to a tagger depends upon the tagger, however. The tagset alteration method works with HMMs because there is a direct interpretation of tag splitting and merging, namely that they correspond to state splitting and merging. For a decision tree tagger, it might be best to use the ambiguity classes as nodes in the decision tree (cf. [17]). Similarly, it is not yet clear how these techniques interact with tagging methods which have their own smoothing and error correction capabilities.

A criticism of this work might be that it is language-specific or tagset-specific. In that all languages have ambiguous words, the claim about language-specificity has to be empirically determined. However, taggers which encode highly specific features are language-specific (and likely tagset-specific). Consider, for example, how Toutanova and Manning [26] determine whether a word is a particle (RP): “the current word is often used as a particle, and ... there is a verb at most 3 positions to the left, which is ‘known’ to have a good chance of taking the current word as a particle.” This language-specificity is not altogether a bad thing, as general tagging algorithms have, at least for English, seemed to have hit an upper bound, and language-specific features may be necessary to improve. The approach outlined here, however, uses no hand-encoded knowledge and does not increase the complexity of the tagging algorithm.

As for tagset-specificity, which ambiguity class a word has is clearly dependent upon the tagset used, but it is less clear how these methods work with tagsets having different degrees of ambiguity or capturing different morphosyntactic properties. Modifying these methods for another tagset could tell us about how its tags interact and whether a better organization of the lexicon is needed. Using a corpus with a tagset that can be mapped to smaller tagsets (see, e.g., [2]) could more precisely determine the properties which make this tagset modification successful.

## 5 Summary and Outlook

We have investigated ways to assign ambiguity classes to words in order to overcome errors in the training data and a limited amount of annotated training data, thereby leading to a more robust lexicon and improvements in POS tagging. In order to make ambiguity class definitions work: 1) we defined typical ambiguity classes based on their frequency of occurrence, merging classes when appropriate; and 2) we made individual words conform to these classes, by using filtering and adding missing counts to lexical entries.

A benefit of the method is that it can target words or ambiguities of interest. After all, classification by ambiguity classes works well when narrowing in on error classes. Future work can investigate exactly which classes are useful for POS tagging and why, specifically examining whether these more specific tags provide more informative contexts (cf. [2, 12]). Experimenting on other tagsets and corpora can test the effects of tagset design [11] and provide feedback on annotation schemes. This examination can also help



address the rather arbitrary thresholds used for ambiguity class selection. Instead, one can attempt to more robustly cluster words in the lexicon, using not only lexicon information, but contextual information to distinguish the ambiguity classes (cf. [23]). Additionally, given that there are annotation errors in the evaluation data, qualitative analysis of the tagging results is needed [19].

Aside from continuing to improve the assignment of ambiguity classes, this work could impact other POS taggers where an ambiguity class represents a word or is a feature (e.g., [7, 9]), or where a word is assigned more than one tag [6]. These methods could also be adapted for any annotation task using a lexicon.

**Acknowledgments** Thanks to Adriane Boyd and Detmar Meurers for comments on an earlier draft and to Stephanie Dickinson for statistical advising. This material is based upon work supported by the National Science Foundation under Grant No. IIS-0623837.

## References

- [1] T. Brants. Estimating markov model structures. In *Proceedings ICSLP-96*, pages 893–896, Philadelphia, PA, 1996.
- [2] T. Brants. Internal and external tagsets in part-of-speech tagging. In *Proceedings of Eurospeech*, Rhodes, Greece, 1997.
- [3] T. Brants. TnT – a statistical part-of-speech tagger. In *Proceedings of ANLP 2000*, pages 224–231, Seattle, WA, 2000.
- [4] E. Brill and M. Pop. Unsupervised learning of disambiguation rules for part of speech tagging. In K. W. Church, editor, *Natural Language Processing Using Very Large Corpora*, pages 27–42. Kluwer Academic Press, Dordrecht, 1999.
- [5] A. Clark. Combining distributional and morphological information for part of speech induction. In *Proceedings of EACL-03*, pages 59–66, 2003.
- [6] J. R. Curran, S. Clark, and D. Vadas. Multi-tagging for lexicalized-grammar parsing. In *Proceedings of ACL-06*, pages 697–704, 2006.
- [7] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. A practical part-of-speech tagger. In *Proceedings of ANLP-92*, pages 133–140, Trento, Italy, 1992.
- [8] W. Daelemans, A. van den Bosch, and J. Zavrel. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34:11–41, 1999.
- [9] W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. MBT: A memory-based part of speech tagger-generator. In *Proceedings of VLC-96*, pages 14–27, Copenhagen, 1996.
- [10] M. Dalrymple. How much can part of speech tagging help parsing? *Natural Language Engineering*, 12(4):373–389, 2006.
- [11] H. Déjean. How to evaluate and compare tagsets? a proposal. In *Proceedings of LREC-00*, Athens, 2000.
- [12] M. Dickinson. From detecting errors to automatically correcting them. In *Proceedings of EACL-06*, pages 265–272, Trento, Italy, 2006.
- [13] M. Dickinson and W. D. Meurers. Detecting errors in part-of-speech annotation. In *Proceedings of EACL-03*, pages 107–114, Budapest, Hungary, 2003.
- [14] D. Elworthy. Tagset design and inflected languages. In *Proceedings of the ACL-SIGDAT Workshop*, Dublin, 1995.
- [15] A. MacKinlay and T. Baldwin. Pos tagging with a more informative tagset. In *Proceedings of ALTW 2005*, pages 40–48, Sydney, Australia, 2005.
- [16] M. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.
- [17] L. Marquez, L. Padro, and H. Rodriguez. A machine learning approach to POS tagging. *Machine Learning*, 39(1):59–91, 2000.
- [18] Q. McNemar. Note on the sampling error of the difference between correlated proportions. *Psychometrika*, 12:153–157, 1947.
- [19] L. Padro and L. Marquez. On the evaluation and comparison of taggers: the effect of noise in testing corpora. In *Proceedings of COLING/ACL-98*, pages 997–1002, 1998.
- [20] F. Pla and A. Molina. Improving part-of-speech tagging using lexicalized HMMs. *Natural Language Engineering*, 10(2):167–189, 2004.
- [21] B. Santorini. Part-of-speech tagging guidelines for the Penn Treebank project (3rd revision, 2nd printing). Technical Report MS-CIS-90-47, The University of Pennsylvania, Philadelphia, PA, June 1990.
- [22] H. Schmid. Part-of-speech tagging with neural networks. In *Proceedings of COLING 94*, pages 172–176, Kyoto, Japan, 1994.
- [23] H. Schütze. Distributional part-of-speech tagging. In *Proceedings of EACL-95*, pages 141–148, Dublin, Ireland, 1995.
- [24] L. Shen, G. Satta, and A. K. Joshi. Guided learning for bidirectional sequence classification. In *Proceedings of ACL-07*, pages 760–767, 2007.
- [25] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging using a cyclic dependency network. In *Proceedings of HLT-NAACL 2003*, pages 252–259, 2003.
- [26] K. Toutanova and C. D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of EMNLP/VLC-2000*, Hong Kong, 2000.

# Sometimes Less Is More: Romanian Word Sense Disambiguation Revisited

Georgiana Dinu  
University of Tübingen  
gdinu@sfs.uni-tuebingen.de

Sandra Kübler  
Indiana University  
skuebler@indiana.edu

## Abstract

Recent approaches to Word Sense Disambiguation (WSD) generally fall into two classes: (1) information-intensive approaches and (2) information-poor approaches. Our hypothesis is that for memory-based learning (MBL), a reduced amount of data is more beneficial than the full range of features used in the past. Our experiments show that MBL combined with a restricted set of features and a feature selection method that minimizes the feature set leads to competitive results, outperforming all systems that participated in the SENSEVAL-3 competition on the Romanian data. Thus, with this specific method, a tightly controlled feature set improves the accuracy of the classifier, reaching 74.0% in the fine-grained and 78.7% in the coarse-grained evaluation.

## Keywords

Word Sense Disambiguation, Romanian, memory-based learning

## 1 Introduction

Recent approaches to Word Sense Disambiguation (WSD) generally fall into two classes: (1) information-intensive approaches and (2) information-poor approaches. The typical features that are used in information-intensive approaches are the part-of-speech tags of the ambiguous word, the surrounding words with their part-of-speech (POS) tags, as well as collocational features from a larger context. If available, additional information such as document type, named entity information, or syntactic information are added. These approaches use supervised learning with a separate classifier for each ambiguous word. Additionally, the best results are achieved by combining different classifiers into an ensemble, in which the final decision is based on the votes of the different classifiers. The other end of the spectrum generally restricts the available information to types of information that can be extracted from small amounts of text without running into data-sparseness problems. Generally, such systems use a combined approach for all words and only a single algorithm to solve the problem.

From the description above, it is already clear which category is expected to perform better: supervised learning with the maximum of information and with an ensemble classifier. In contrast, our hypothesis is that for some classifiers, a reduced amount of data

is more beneficial than the full range of features that have been used in the past.

As data set, we chose the Romanian data from the SENSEVAL-3 competition [7]. This data set is rather small with regard to both the number of ambiguous words (39) and the number of instances for each word (between 19 and 266). Such a size can be expected for many languages for which there is no financial interest. The supervised learning method we chose is memory-based learning (MBL), a  $k$ -nearest neighbor approach. It bases the classification of a new instance on the  $k$  most similar instances found in the training data. This approach has been shown to be successful for a range of problems in NLP [1, 2] Daelemans et al. argue that MBL has a suitable bias for such problems because it allows learning from atypical and low-frequency events, thus enabling a principled approach to the treatment of exceptions and sub-regularities in language. Another advantage of MBL lies in the fact that it can work with complete words as feature values. As a consequence, however, MBL is also sensitive to large numbers of features that are only relevant for the classification of specific instances but not for all instances. This is the case even when features are weighted. This last characteristic of MBL suggests that a good balance between too much and too little information must be found, which in turn makes it a good candidate for our approach.

In the following sections, we show that MBL combined with a restricted set of features of three context words to each side of the ambiguous word, their POS tags, the closest verbs, nouns, and prepositions on both sides, lead to competitive results. We then employ two feature selection methods to further optimize the feature set. The results show that forward selection, which selects a smaller feature set, leads to optimal results, reaching an accuracy of 74.0% in the fine-grained and 78.7% in the coarse-grained evaluation, outperforming all systems that participated in the SENSEVAL-3 competition on the Romanian data.

## 2 Related Work

In building a supervised WSD system, one of the main decisions is the choice of a classifier. Memory-based learning (MBL) is a supervised learning method that has been successfully used in WSD after a difficult start: Mooney [8] reports the first experiment using a simple nearest neighbor method in a comparison of different machine learning methods for disambiguating the word *line*. He attributes the low performance

CT <sub>k</sub>	the token at position $k$ [-3..3] relative to the target word; CT <sub>0</sub> : target word
CP <sub>k</sub>	the POS tag of the token at position $k$
VA	the first verb found after the target word
VB	the first verb found before the target word
NA	the first noun found after the ambiguous word
NB	the first noun found before the ambiguous word
PA	the first preposition found after the ambiguous word
PB	the first preposition found before the ambiguous word

**Table 1:** *The complete list of features used in the experiments*

of this approach to the fact that it did not use feature weighting. Escudero et al. [4] show later that one of the problems for the nearest neighbor approach was the high number of context features, which resulted in a very sparse feature matrix. After they introduced feature weighting and collapsed the context features into one set-valued feature (and modified the similarity metric accordingly to calculate a set-based similarity), they showed that the nearest neighbor method outperforms the naive Bayes model, Mooney’s best performing model. Veenstra et al. [10] present a system that competed successfully in SENSEVAL-1. They use context features (word form and POS tag of the ambiguous word and 2 words on either side) as well as keyword features and definition features. For keyword features, the most informative words from the context are used. Veenstra et al.’s results show that the optimal settings depend on the individual ambiguous words. There is no optimal setting that works equally well for all words. Mihalcea [6] shows that even if feature weighting methods are used, memory-based learning is susceptible to irrelevant or redundant features. She improved her results for the SENSEVAL-2 English lexical sample task by using forward selection. This method reduces the number of features on average to 3.7 for nouns, 4.4 for adjectives, and to 4.5 for verbs.

Lee and Ng [5] thoroughly investigate which knowledge sources are relevant for WSD. They used four different classifiers and the SENSEVAL-1 and SENSEVAL-2 English data. Their findings show a trend that classifiers perform best when all features are offered to the systems. The Support Vector Machines classifier and AdaBoost perform best without feature selection while the naive Bayes and the decision tree classifier profit from feature selection. The only experiment in which a classifier performs best on a restricted set of features is the combination of the decision tree classifier with SENSEVAL-2 data, but the difference to the results on all features is rather small (57.2% for only collocational features versus 56.8% for all features). These findings suggest that a complete feature set provides an optimal setting for WSD.

WSD for Romanian was one of the tasks in SENSEVAL-3. In the competition, seven systems were evaluated. We will concentrate on the three best performing systems here: SWAT-HK-boost, SWAT-HK [11] and the Duluth system [9]. SWAT-HK-boost is a boosting approach that used context features and bigrams and trigrams of words and parts of speech. SWAT-HK is an ensemble voting approach based on SWAT-HK-boost and four other classifiers, using the same feature set as SWAT-HK-boost. The Duluth system uses an ensemble of three decision trees, each trained on a different set of features, word bigrams,

word unigrams, and word co-occurrence features. Note that all three best-performing systems use a combination of simpler classifiers. SWAT-HK-boost reaches a fine-grained accuracy of 72.7%, SWAT-HK 72.4%, and the Duluth system 71.4%. Since all systems described here attempted all words, precision and recall are identical, and we only report accuracy.

### 3 Experiments

For all experiments reported here, we used the SENSEVAL-3 Romanian lexical sample data [7], which consists of labeled examples for 39 ambiguous words: 25 nouns, 9 verbs, and 5 adjectives<sup>1</sup>. In order to allow a comparison of our experiments to systems that participated in SENSEVAL, we used the designated training and test sets. The senses, with an average of 8.8 fine-grained senses per word (4.7 coarse-grained), are manually extracted from a Romanian dictionary.

The experiments reported here were conducted with TiMBL [3], a memory-based learning system. TiMBL was used with the following settings: the IB1 algorithm, Gain Ratio for feature weighting, and  $k = 1$ . For evaluation, a leave-one-out cross-validation was performed.

As reported above, the experiments were conducted with a rather restricted feature set: We used lexical and POS information of the ambiguous word and of a context of three words on both sides, as well as information concerning the closest verbs, nouns, and prepositions in the sentence. Table 1 lists the complete set of features.

For each word, an optimal set of features is determined. We performed experiments with **forward** and **backward selection**. Initially, a pool of features containing all the features is generated. Forward selection starts the selection process by selecting a single feature from the pool, running the classifier with this single feature. Then the feature with the highest accuracy is selected. In the next step, the second feature is selected based on combinations of the selected feature and the remaining features in the pool. Features are added as long as accuracy improves. Backward selection starts with the complete pool of features. In the first step, experiments are conducted removing one of the features. Then the feature whose absence results in the highest improvement in accuracy is removed permanently. The process of removing features continues as long as accuracy improves or remains stable.

The forward selection experiment is similar to the experiment that Mihalcea [6] performed for the

<sup>1</sup> For the list of words and characteristics, cf. Tables 5 to 7.

forward selection										
feature	NA	CT <sub>0</sub>	NB	CT <sub>1</sub>	CT <sub>-1</sub>	CP <sub>0</sub>	CT <sub>2</sub>	CP <sub>-1</sub>	CT <sub>-2</sub>	VB
# words	28	25	24	19	18	18	14	15	13	12
backward selection										
feature	CP <sub>1</sub>	CP <sub>-1</sub>	CT <sub>1</sub>	CP <sub>-2</sub>	PB	NA	VB	CP <sub>2</sub>	CT <sub>-1</sub>	CP <sub>0</sub>
# words	28	27	25	23	23	22	22	21	20	19

**Table 2:** *The most commonly selected features in per-word feature selection*

	fine	coarse	POS	forward	backward
baseline (MFS)	58.5	62.8	nouns	7.4	9.9
all features	71.2†	76.4†	verbs	5.0	11.0
backward selection	72.7†	77.4*	adjectives	6.8	7.2
forward selection	<b>74.0*</b>	<b>78.7*</b>	overall	6.8	9.8

**Table 3:** *Results for the feature-selection experiments; all differences are significant at the 0.05 (\*) / 0.005 (†) level, McNemar*

SENSEVAL-2 English lexical sample task. Note, however, that Mihalcea used a larger feature pool including collocation information, sense specific keywords, named entity information, and syntactic information.

## 4 Results

The evaluation of the experiments was performed with the SENSEVAL scoring software, which provides coarse-grained and fine-grained accuracies.

### 4.1 Feature Selection

Table 3 gives the results of the selection process. The baseline reported here is computed by assigning the most frequent sense (MFS), as computed from the training data, to the test instances. It is evident that TiMBL, even without much optimization of the parameter settings, outperforms the baseline significantly. Classification accuracy can be further improved when these system parameters are optimized. However, this is irrelevant for the experiments reported here.

The results also show that WSD for Romanian profits from both feature selection methods, with forward selection outperforming backward selection. Our starting hypothesis was that irrelevant or redundant features harm TiMBL’s performance. A look at the average number of features after feature selection shows that this is true. Table 4 reports the average number of features used for the different selection algorithms and POS categories. From a total of 20 features, forward selection uses only approximately 7 features and backward selection 10. From these results, we can conclude that not all of the features of the original set are helpful for the task and that TiMBL suffers from irrelevant or redundant features despite the use of a feature weighting mechanism. Additionally, backward selection does not restrict the number of features as much as forward selection does. The forward selection results are comparable to the findings of Mihalcea [6],

**Table 4:** *Feature selection and number of features*

where a similar selection algorithm on SENSEVAL-2 English data improves the average performance by 3.9% in nouns and verbs, and 5.4% in adjectives.

The selection experiments can also be used to answer a linguistically relevant question: Which features provide the best information for WSD? Table 2 reports the features used in classifying the most words and the number of words for which the feature was used (out of a total of 39 words). It is surprising to see that the two selection methods prefer different types of features: While forward selection prefers word forms over POS information, backward selection has a more balanced distribution, favoring POS tags as the most often used features.

As reported before, the near context is a very good indicator for a word’s sense. The words surrounding the target word seem to be most helpful for disambiguation, and their relevance decreases with an increasing distance from the target word. The nouns preceding and following the ambiguous word as well as the word form of the ambiguous word itself play a very important role. This is a general trend for all words, irrespective of their parts of speech. The last feature may be surprising since one could assume that the forms would be very similar considering that there is a separate classifier for each ambiguous word. However, Romanian is an inflected language, so that the word form can provide information on some morphological and syntactic features, especially in the absence of further linguistic analysis.

Adjectives are special in that they are biased towards choosing features extracted from preceding context (preceding noun, preceding tokens), unlike verbs or nouns, which prefer an extraction window centered around the target word. On average, a noun chooses 3 features from the left context and 3 from the right. For verbs, its on average 2 words on each side while an adjective chooses 3.4 features from the left context and 2.2 from the right. Part of the explanation for the last number can be found in the fact that in Romanian, both predicative and attributive adjectives follow the constituents they modify, which presumably are important indicators for the sense of the adjective.

One of the extreme examples of words that were dis-

word	translation	no. senses (f/c)	size	MFS (f)	MFS (c)	acc. (f)	acc. (c)
ac	needle	16/7	127	50.8	50.8	73.8	75.4
accent	accent	5/3	172	73.6	77.0	89.7	93.1
actiune	action	10/7	261	39.8	39.8	61.7	85.2
canal	channel	6/5	134	68.2	68.2	69.7	75.8
circuit	circuit	7/5	200	49.5	50.5	59.4	65.3
circulatie	circulation	9/3	221	45.6	45.6	59.4	68.4
coroana	crown	15/11	252	58.7	61.9	77.0	77.8
delfin	dolphin	5/4	31	100	100	80.0	80.0
demonstratie	demonstration	6/3	229	64.3	64.3	73.0	73.0
eruptie	eruption	2/2	54	40.7	40.7	81.5	81.5
geniu	genius	5/3	106	72.2	77.8	64.8	70.4
nucleu	nucleus	7/5	64	78.8	78.8	81.8	81.8
opozitie	opposition	12/7	266	96.3	96.3	95.5	95.5
perie	brush	5/3	46	79.2	95.8	75.0	95.8
pictura	painting	5/2	221	47.7	47.7	75.7	81.1
platforma	platform	11/8	226	38.8	38.8	58.6	58.6
port	port	7/3	219	51.9	51.9	81.5	83.3
problema	problem	6/4	262	44.3	44.3	69.5	69.5
proces	process	11/3	166	62.2	64.6	81.7	82.9
reactie	reaction	7/6	261	83.2	83.2	83.2	83.2
stil	style	14/4	199	60.4	80.2	62.4	76.2
timbru	stamp	7/3	231	94.0	99.1	94.8	98.3
tip	type	7/4	263	76.3	76.3	87.8	89.3
val	wave	15/9	242	85.1	85.1	87.6	88.4
valoare	value	23/9	251	63.2	75.2	72.8	85.6
total	-	8.9/4.9	-	63.8	66.2	75.9	80.6

**Table 5:** MBL with per-word forward-feature selection: nouns

ambiguated using a very small number of features is the verb *câștiga* (to win). By only using the word form of the verb and the word form of the following noun, disambiguation accuracy increases from 52.2% (MFS) to 72.2%. An examination of the training data provides an explanation for this extreme behavior: This word has five senses but the predominant two senses are to gain material benefits, and to win a sports competition (or a contest, a trial). The following noun (NA) is a very good sense indicator in this case since in most cases this feature contains the object of the verb: *bani*, *dolari*, *mărci* or *lei* (money or various currencies) for the first sense, and *partida*, *derby*, *meci* (sport competitions) for the second sense. Thus, this single feature increases accuracy from 50.2% (MFS) to 66.9% on the training data. Additionally, the word form (CT<sub>0</sub>) of the ambiguous word helps to distinguish the two senses. For example, the first person plural form *caștigăm* is predominantly used within the winning a sport competition sense, as ‘our team (we) won the game’. CT<sub>0</sub> is thus the feature that brings the second best improvement, increasing accuracy from 66.9% to 71.8%. Adding any of the other features results in accuracy drops varying between 0.5% and 12%, suggesting that for this word, all these features provide irrelevant information.

## 4.2 Comparison with SENSEVAL-3 Participants

In contrast to most state-of-the-art WSD systems, our approach uses a rather impoverished feature set. It contains neither collocational features nor syntactic or global features. Thus, the conjecture is that the system should be at a disadvantage when compared

system	fine	coarse
feature selection MBL	<b>74.0</b>	<b>78.7</b>
SWAT-HK-boost [11]	72.7	77.1
Duluth [9]	71.4	75.2

**Table 6:** System comparison

to systems that had access to such data sources. A comparison with two of the best 3 systems in the SENSEVAL-3 competition, the SWAT-HK-boost system [11], and the Duluth system [9], shows that this is not the case (cf. Table 6). On the contrary, our memory-based system (with default parameter settings) outperforms both systems on this task<sup>2</sup>. The difference to the SWAT-HK-boost system is statistically significant (McNemar), on the 0.05 level.

One reason why we did not use collocational features is that collocations tend to increase the number of features by at least an order of magnitude, with most of the features having zero values for each example. Escudero et al. [4] show that such a selection of features harms the performance of  $k$ -nearest neighbor approaches. Since their suggested solution, a set-based approach in calculating the similarity of feature values, is not available in TiMBL, we decided not to use this type of information.

## 4.3 Results for Individual Words

Table 5 gives the results of the forward selection experiment for the individual nouns and Table 7 for verbs

<sup>2</sup> Wicentowski et al. [11] report a fine-grained accuracy of 73.3% for SWAT-HK-boost after an error was corrected.

word	translation	no. senses (f/c)	size	MFS (f)	MFS (c)	acc. (f)	acc. (c)
Verbs							
castiga	win	5/4	227	52.2	52.2	72.2	72.2
citi	read	10/4	259	82.3	90.8	82.3	89.2
cobori	descend	11/6	252	47.7	75.8	68.0	85.2
conduce	drive	7/6	265	55.2	56.0	81.3	82.1
creste	grow	14/6	209	43.7	43.7	72.8	74.8
desena	draw	3/3	54	81.5	81.5	81.5	81.5
desface	untie	11/5	115	27.6	32.8	53.4	56.9
fierbe	boil	11/4	83	32.6	37.2	48.8	58.1
indulci	sweeten	7/4	19	40.0	80.0	60.0	80.0
total	-	8.7/4.6	-	53.9	61.5	72.3	77.9
Adjectives							
incet	slow	6/3	224	41.6	41.6	79.6	79.6
natural	natural	12/5	242	23.6	51.2	67.5	74.8
neted	smooth	7/3	34	41.2	52.9	41.2	41.2
oficial	official	5/3	185	53.1	53.1	72.9	72.9
simpļu	simple	15/6	153	36.6	36.6	46.3	48.8
total	-	9/4	-	38.1	46.4	66.8	69.4

**Table 7:** MBL with per-word forward-feature selection: verbs and adjectives

and adjectives. Compared to the MFS baseline, nouns achieve a net gain of 12.1% (14.4% coarse-grained) and verbs 18.4% (16.4% coarse). Adjectives are disambiguated best for the Romanian task, achieving an accuracy gain of 28.7% (23% coarse). The error reduction rates for fine-grained scores are 33.4% for nouns, 40% verbs and 46.3% for adjectives.

## 5 Conclusion and Future Work

We have shown that when using a memory-based classifier for WSD, the feature set needs to be tightly controlled. In contrast to other experiments, the MBL classifier achieved optimal results with on average seven features per word. The most important features are the noun following the ambiguous word, the word form of the ambiguous word, and the preceding noun: features that can easily be retrieved. The experiments show that forward selection allows a greater reduction of features: on average seven features as compared to an average of ten features for backward selection. This is another indication that MBL suffers from irrelevant or redundant features. These findings are partially in line with the findings of Lee and Ng [5] for a naive Bayes and a decision tree classifier: both show an increased performance when feature selection is performed. However, the initial feature set that Lee and Ng used was much larger than the one used in the present study. A logical explanation for the differences between the results reported here and Lee and Ng's findings can be found in the differences of the classifiers used in the experiments. All classifiers Lee and Ng used in their experiments are based on greedy learning approaches while MBL is a lazy learning approach. There is a slight chance, however, that the results reported here are due to idiosyncrasies in the Romanian data set. For this reason, the next step is to test the same combination of classifier and features on data sets for different languages. Another reason for the success of this combination may be a consequence of the rather limited size of the training data. Therefore, the combination suggested here needs to be

tested on larger data sets with controlled data sizes.

## References

- [1] W. Daelemans and A. van den Bosch. *Memory Based Language Processing*. Cambridge University Press, 2005.
- [2] W. Daelemans, A. van den Bosch, and J. Zavrel. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34:11–43, 1999. Special Issue on Natural Language Learning.
- [3] W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. TiMBL: Tilburg memory based learner – version 5.1 – reference guide. Technical Report ILK 04-02, Induction of Linguistic Knowledge, Computational Linguistics, Tilburg University, 2004.
- [4] G. Escudero, L. Márquez, and G. Rigau. Naive Bayes and exemplar-based approaches to Word Sense Disambiguation revisited. In *Proceedings of the 14th European Conference on Artificial Intelligence, ECAI'2000*, pages 421–425, Berlin, Germany, 2000.
- [5] Y. K. Lee and H. T. Ng. An empirical evaluation of knowledge sources and learning algorithms for Word Sense Disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Philadelphia, PA, 2002.
- [6] R. Mihalcea. Instance based learning with automatic feature selection applied to Word Sense Disambiguation. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan, 2002.
- [7] R. Mihalcea, V. Năstase, T. Chklovski, D. Tătar, D. Tufiş, and F. Hristea. An evaluation exercise for Romanian Word Sense Disambiguation. In *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 29–32, Barcelona, Spain, 2004.
- [8] R. J. Mooney. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing EMNLP*, pages 82–91, Philadelphia, PA, 1996.
- [9] T. Pedersen. The Duluth lexical sample systems in SENSEVAL-3. In *SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs*, pages 203–08, Barcelona, Spain, 2004.
- [10] J. Veenstra, A. van den Bosch, S. Buchholz, W. Daelemans, and J. Zavrel. Memory-based Word Sense Disambiguation. *Computers and the Humanities, Special Issue on Senseval, Word Sense Disambiguation*, 34(1/2):171–177, 2000.
- [11] R. Wicentowski, G. Ngai, D. Wu, M. Carpuat, E. Thomforde, and A. Packer. Joining forces to resolve lexical ambiguity: East meets West in Barcelona. In *SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs*, pages 262–264, Barcelona, Spain, 2004.

# Recognition and Transliteration of Bengali Named Entities: A Computational Approach

Asif Ekbal and Sivaji Bandyopadhyay  
Department of Computer Science and Engineering,  
Jadavpur University,  
Kolkata, India 700032

## Abstract

The paper reports about the development of a Named Entity Recognition (NER) system in Bengali using a tagged Bengali news corpus and the subsequent transliterations of the recognized Bengali Named Entities (NEs) into English. The Hidden Markov Model (HMM) based NER system has been trained with a corpus of 62,280 word forms. Initially, a HMM based part of speech (POS) tagger has been used to tag this training set with the 26 POS tags. A modified joint source-channel model has been used along with a number of alternatives to generate the English transliterations of Bengali NEs. The NER system has demonstrated an average Recall, Precision and F-Score values of 89.62%, 78.47% and 83.63%, respectively with the 6-fold cross validation tests. Evaluation of the proposed transliteration models demonstrate that the modified joint source-channel model performs the best with a Word Agreement Ratio (WAR) of 75.4% for person names, 73.6% for location names and a Transliteration Unit Agreement Ratio (TUAR) of 91.7% for person names and 73.6% for location names during Bengali to English (B2E) transliteration.

## Keywords

Named Entity Recognition (NER), Transliteration, Hidden Markov Model (HMM), Joint Source -Channel Model.

## 1. Introduction

Named entities (NE) hold a very important place in many Natural Language Processing (NLP) application areas. Proper identification, classification and transliteration of named entities are very crucial and pose a very big challenge to the NLP researchers.

The problem of correct identification of NEs is specifically addressed and benchmarked by the developers of Information Extraction System, such as the GATE system [8]. The current trend in NER is to use the machine-learning approach, which is more attractive in that it is trainable and adoptable and the maintenance of a machine-learning system is much cheaper than that of a rule-based one. The representative machine-learning approaches used in NER are mainly HMM (BBN's *IdentiFinder* in [7]), Maximum Entropy (New York University's *MEME* in [3]) and Conditional Random Fields [2]. The works, carried out already in the area of NER, are all in non-Indian languages. In Indian languages particularly in Bengali, the work in NER can be found in [4] and [5]. The proposed

HMM based NER, reported here, outperforms both the earlier systems developed in [4][5].

The NE machine transliteration algorithms presented in this work have been evaluated with person and location names. A machine transliteration system that is trained on person and location names is very important in a multilingual country like India where large number of person and location name collections like census data, electoral roll and railway reservation information must be available to multilingual citizens of the country in their own vernacular. The transliteration models introduced in Ref. [9] and [10] are modified in [6], where a modified joint source-channel model has been proposed for the transliteration of Bengali named entities into English and vice-versa. The present work differs from [9][10] in the sense that identification of the transliteration units in the source language is done using regular expressions and no probabilistic model is used. Moreover, the proposed model differs in the way the transliteration units and the contextual information are defined. No linguistic knowledge is used in [9][10], whereas the present work uses linguistic knowledge in the form of possible conjuncts and diphthongs in Bengali and their representations in English.

## 2. Named Entity Recognition in Bengali

Bengali is one of the widely used languages all over the world. It is the seventh popular language in the world, second in India and the national language of Bangladesh. NE identification in Indian languages in general and in Bengali in particular is difficult and challenging. In English, the NE always appears with capitalized letter but there is no concept of capitalization in Bengali. A tagged Bengali news corpus, developed from the archive of a widely read Bengali news paper available in the web, has been used in this work for the NER. At present, the corpus contains around 34 million word forms in ISCII (Indian Standard Code for Information Interchange) and UTF-8 format. The location, reporter, agency and different date tags in the tagged corpus help to identify the location, person, organization and miscellaneous names, respectively. A portion of the corpus has been used to develop the HMM based NER. The objective of this system is to identify NEs in Bengali and to classify them into person name, location name, organization name,

miscellaneous names and “none- of -the -above” category. Miscellaneous names include date, time, percentage and monetary expressions. The following is the set of NE tags, defined to apply HMM in NER:

PER: Single-word person name, LOC: Single-word location name, ORG: Single-word organization name, MISC: Single-word miscellaneous name, B-PER: Beginning of a multi-word location name, I-PER: Internal of a multi-word location name, E-PER: End of a multiword person name, B-LOC: Beginning of a multi-word location name, I-LOC: Internal of a multi-word location name, E-LOC: End of a multi-word location name, B-ORG: Beginning of a multi-word organization name, I-ORG: Internal of a multi-word organization name, E-ORG: End of a multi-word organization name, B-MISC: Beginning of a multi-word miscellaneous name, I-MISC: Internal of a multi-word miscellaneous name, E-MISC: End of a multi-word miscellaneous name, NNE: Words that are not named entities (“none-of-the-above” category).

## 2.1 HMM Based Named Entity Recognition

The goal of NER is to find a stochastic optimal tag sequence  $T = \{t_1, t_2, t_3, \dots, t_n\}$  for a given word sequence  $W = \{w_1, w_2, w_3, \dots, w_n\}$ . Generally, the most probable tag sequence is assigned to each sentence following the Viterbi algorithm [1]. The named entity tagging problem becomes equivalent to searching for  $\text{argmax}_T P(T) * P(W | T)$ , by the application of Bayes’ law.

We have used trigram model, i.e., the probability of a tag depends on two previous tags, and then we have,  $P(T) = P(t_1 | \$) * P(t_2 | \$, t_1) * P(t_3 | t_1, t_2) * P(t_4 | t_2, t_3) * \dots * P(t_n | t_{n-2}, t_{n-1})$ , where an additional tag ‘\$’ (dummy tag) has been introduced to represent the beginning of a sentence. Due to sparse data problem, the linear interpolation method has been used to smooth the tri-gram probabilities as follows:  $P'(t_n | t_{n-2}, t_{n-1}) = \lambda_1 P(t_n) + \lambda_2 P(t_n | t_{n-1}) + \lambda_3 P(t_n | t_{n-2}, t_{n-1})$  such that the  $\lambda$ s sum to 1. The values of  $\lambda$ s have been calculated by the method given in (Brants 2000)[11].

To make the Markov model more powerful, *additional context dependent features* have been introduced to the emission probability in this work that specifies the probability of the current word depends on the tag of the previous word and the tag to be assigned to the current word. Now,  $P(W | T)$  is calculated by the equation:

$$P(W | T) \approx P(w_1 | \$, t_1) * P(w_2 | t_1, t_2) * \dots * P(w_n | t_{n-1}, t_n).$$

So, the emission probability can be calculated as:

$$P(w_i | t_{i-1}, t_i) = \frac{\text{freq}(t_{i-1}, t_i, w_i)}{\text{freq}(t_{i-1}, t_i)}$$

Here also the smoothing technique is applied rather than using the emission probability directly. The emission probability is calculated as:  $P(w_i | t_{i-1}, t_i) = \theta_1 P(w_i | t_i) +$

$\theta_2 P(w_i | t_{i-1}, t_i)$ , where  $\theta_1, \theta_2$  are two constants such that all  $\theta$ s sum to 1. The values of  $\theta$ s should be different for different words. But the calculation of  $\theta$ s for every word takes a considerable time and hence  $\theta$ s are calculated for the entire training corpus. In general, the values of  $\theta$ s can be calculated by the same method that was adopted in calculating  $\lambda$ s. The trigram model has been used in the present work to apply Viterbi algorithm [1] for finding best state sequence.

## 2.2 Handling the Unknown Words

Handling of unknown words is an important issue in NE tagging. Viterbi [1] algorithm attempts to assign a tag to unknown words. For words, which have not been seen in the training set,  $P(w_i | t_i)$  is estimated based on features of the unknown words, such as whether the word contains a particular suffix. The list of suffixes has been prepared for Bengali. At present there are 435 suffixes; many of them usually appear at the end of different NEs and non-NEs. A null suffix is also kept for those words that have none of the suffixes in the list. Other than these suffixes, the lists of suffixes that may occur with person names (e.g., -বাবু [-babu], -দা [-da], -দি [-di] etc.) and location names (e.g. -ল্যান্ড, [-land] - পুর [-pur], -লিয়া [-lia] etc.) have been kept. The probability distribution of a particular suffix with respect to a specific tag is generated from all words in the training set that share the same suffix. Two additional features that cover the numbers and symbols have been considered also. A lexicon, developed in an unsupervised way from the tagged Bengali news corpus, has been used in order to handle the unknown words further. Lexicon contains the root words and their basic part of speech information such as: noun, verb, adjective, adverb, pronoun and indeclinable. The lexicon has 45,000 entries and does not contain NEs. If an unknown word is found to appear in the lexicon, then most likely it is not a named entity.

## 3. Named Entity Transliteration

A number of transliteration models have been proposed in [6] that can generate the English transliteration from a Bengali word and the vice-versa. The work was mainly for the person names and the proposed models were not compared against any *baseline* model. Here, the transliteration models are extended to handle both person and location names. The Bengali NE is divided into Transliteration Units (TUs) with patterns  $C^+M$ , where C represents a consonant or a vowel or a conjunct and M represents the vowel modifier or matra. An English NE is divided into TUs with patterns  $C^+V^*$ , where C represents a consonant and V represents a vowel. The TUs are considered as the lexical units for machine transliteration. The system considers the Bengali and English contextual information in the form of collocated TUs simultaneously to calculate the plausibility of transliteration from each Bengali TU to various English candidate TUs and chooses



the one with maximum probability. The system learns the mappings automatically from the bilingual training sets guided by linguistic knowledge. The output of this mapping process is a decision-list classifier with collocated TUs in the source language and their equivalent TUs in collocation in the target language along with the probability of each decision obtained from a training corpus. The machine transliteration of the input Bengali word is obtained using direct orthographic mapping by identifying the equivalent English TU for each Bengali TU in the input and then placing the English TUs in order.

The regular expression based alignment technique has been considered in the present work as it is deterministic and seems to be more appropriate for English and other Indian languages due to comparable orthography. The process considers linguistic knowledge in terms of possible conjuncts and diphthongs in Bengali and their corresponding English representations in order to make the number of TUs on both source and target sides equal at the time of training of the models.

### 3.1 Transliteration Models

The various proposed models differ in the nature of collocational statistics used during machine transliteration process. All the models except the *baseline model* are basically the variations of the joint source-channel model in respect of the contextual information considered. In the *baseline model*, English consonant or sequence of consonants is represented as Bengali consonant or conjunct or a sequence of consonants. English vowels are represented as either Bengali vowels or as a matra (vowel modifier). English diphthongs are represented as vowel/semi-vowel-matra combination in Bengali. This *baseline* is used in case the system does not find a source TU pattern in the TU alignment table. The various transliterations models are defined below:

- Model A

In this model, no context is considered in either the source or the target side. This is essentially the monogram model.

K

$$P(B,E) = \prod_{k=1} P(\langle b, e \rangle_k)$$

- Model B

This is essentially a bigram model with previous source TU, as the context.

K

$$P(B,E) = \prod_{k=1} P(\langle b, e \rangle_k | b_{k-1})$$

-Model C

This is essentially a bigram model with next source TU as the context.

K

$$P(B,E) = \prod_{k=1} P(\langle b, e \rangle_k | b_{k+1})$$

- Model D

This is essentially the joint source-channel model where the previous TUs in both the source and the target sides are considered as the context.

K

$$P(B,E) = \prod_{k=1} P(\langle b, e \rangle_k | \langle b, e \rangle_{k-1})$$

-Model E

This is basically the trigram model where the previous and the next source TUs are considered as the context

K

$$P(B,E) = \prod_{k=1} P(\langle b, e \rangle_k | b_{k-1}, b_{k+1})$$

- Model F

In this model, the previous and the next TUs in the source and the previous target TU are considered as the context. This is the modified joint source-channel model.

K

$$P(B,E) = \prod_{k=1} P(\langle b, e \rangle_k | \langle b, e \rangle_{k-1}, b_{k+1})$$

### 3.2 Bengali to English Machine Transliteration

Translation of named entities is a tricky task: it involves both translation and transliteration. Transliteration is commonly used for named entities, even when the words could be translated: [ওয়াল স্ট্রিট (*wall street*) is translated to *Wall Street* (literal translation) although ওয়াল (*wall*) and স্ট্রিট (*Street*) are vocabulary words]. On the other hand, কল্যানী বিশ্ববিদ্যালয় (*kalyani viswavidyalaya*) is translated to *Kalyani University* in which কল্যানী (*kalyani*) is transliterated to *Kalyani* and বিশ্ববিদ্যালয় (*viswavidyalaya*) is translated to *University*.

Two different bilingual training sets have been kept that contain entries mapping Bengali person names and location names to their respective English transliterations. To automatically analyze the bilingual training sets to acquire knowledge in order to map new Bengali person and location names to English, transliteration units (TUs) are extracted from the Bengali-English pairs of person and location names and Bengali TUs are associated with their English counterparts. After retrieving the TUs from a Bengali-English pair, it associates the Bengali TUs to the English transliteration units along with the TUs in context. But, in some cases, the number of TUs retrieved from the Bengali and English words may differ. The [ব্রজগোপাল (*brijgopal*) ↔ *brijgopal*] name pair yields 5 TUs in Bengali side and 4 TUs in English side [ ব্ | জ | গো | পা | ল ↔ *bri |*

jgo | pa | l]. In such cases, the system cannot align the TUs automatically and linguistic knowledge/feature is used to resolve the confusion. The hypothesis followed in the present work is that *the problem TU in the English side has always the maximum length*. If more than one English TU has the same length, then *system starts its analysis from the first one*. In the above example, the TUs *bri* and *jpo* have the same length. The system interacts with the linguistic knowledge and ascertains that *bri* is valid and *jpo* cannot be a valid TU in English since there is no corresponding conjunct representation in Bengali. So *jpo* is split up into 2 TUs *j* and *po*, and the system aligns the 5 TUs as [ব্ | জ | গো | পা | ল ↔ *bri* | *j* | *go* | *pa* | *l*]. Similarly, [কোলকাতা (*kolkata*) ↔ *kolkata*] is initially split as [কো | ল | কা | তা] ↔ *ko* | *lka* | *ta*], and then as [ko | l | ka | ta] since *lka* has the maximum length and it does not have any valid conjunct representation in Bengali.

In some cases, the knowledge of Bengali diphthong resolves the problem. In the following example, [সো | মা | লি | য়া (*somalia*) ↔ *so* | *ma* | *lia*], the number of TUs on both sides do not match. The English TU *lia* is chosen for analysis, as its length is the maximum among all the TUs. The vowel sequence *ia* corresponds to a diphthong in Bengali that has the valid representation <ইয়া>. Thus, the English vowel sequence *ia* is separated from the TU *lia* (*lia* → *l* | *ia*) and the intermediate form of the name pair appears to be [সো | মা | লি | য়া ↔ *so* | *ma* | *l* | *ia*]. Here, a *matra* is associated with the Bengali TU that corresponds to English TU *l* and so there must be a vowel attached with the TU *l*. TU *ia* is further splitted as *i* and *a* (*ia* → *i* | *a*) and the first one (i.e. *i*) is assimilated with the previous TU (i.e. *l*) and finally the name pair appears as: [সো | মা | লি | য়া (*somalia*) ↔ *so* | *ma* | *li* | *a*]. Similarly, [চে | ন্নাই (*chennai*) ↔ *che* | *nna* | *i*] and [রা | ই | মা (*raima*) ↔ *rai* | *ma*] can be solved with the help of diphthongs.

The number of TUs on both sides doesn't match for the examples, [শি | ব | রা | জ (*shivraj*) ↔ *shi* | *vra* | *j*], [খ | ড় | দ | হ (*khardah*) ↔ *kha* | *rda* | *h*]. It is observed that both *vr* and *kd* represent valid conjuncts in Bengali but these examples contain the constituent Bengali consonants in order and not the conjunct representation. During the training phase, if, for some conjuncts, examples with conjunct representation are outnumbered by examples with constituent consonants representation, the conjunct is removed from the linguistic knowledge base and training examples with such conjunct representation are moved to a *Direct example base* which contains the English words and their Bengali transliteration. The above two name pairs can then be realigned as: [শি | ব | রা | জ (*shivraj*) ↔ *shi* | *v* | *ra* | *j*], [খ | ড় | দ | হ (*khardah*) ↔ *kha* | *r* | *da* | *h*].

Otherwise, if such conjuncts are included in the linguistic knowledge base, training examples with constituent consonants representations are to be moved to the *Direct example base*. If the source TU is not found in the alignment, then source language symbols are replaced by the corresponding most probable symbol in the target language using the *baseline* model and the transliteration is obtained.

**Table 1: Results of the six - fold cross validation tests**

Test Set	Recall	Precision	F-Score
Set 1	88.10	77.90	82.69
Set 2	89.80	78.30	83.66
Set 3	90.43	79.40	84.56
Set 4	90.19	77.90	83.6
Set 5	90.80	78.40	83.9
Set 6	88.40	78.90	78.47
Average	89.62	78.47	83.63

**Table 2: Results of the Bengali to English Transliteration**

Model	Person name		Location name	
	WAR (in %)	TUAR (in %)	WAR (in %)	TUAR (in %)
Baseline	49.7	74.8	47.1	73.9
A	53.8	79.2	53.3	77.1
B	63.4	83.3	62.8	81.2
C	60.7	82.5	60.1	80.7
D	65.8	84.9	64.3	82.2
E	70.6	89.3	68.9	86.9
F	75.4	91.7	73.6	89.3

## 4. Experimental Results

A portion of the tagged (not NE tagged/POS tagged) news corpus, containing 62,280 wordforms, has been used to train the NER system. This training corpus is run through a HMM-based part of speech (POS) tagger to tag the training corpus with the 26 different POS tags, defined for the Indian languages<sup>1</sup>. This POS-tagged training set is searched for some specific POS tags (NNPC [compound proper noun], NNP [proper noun] and QFNUM [cardinals and ordinals numbers]) that represent NEs. These POS tags are replaced by the appropriate NE tags as defined earlier. The confusion matrix obtained from our POS tagger suggests that most ambiguities occur between the proper nouns and the common nouns. So, additionally the POS tags (e.g., NNC [compound common noun], NN [common noun]) representing common nouns are checked for the correctness and replaced by the appropriate NE tags in the training set, if necessary. The training set thus obtained is a corpus tagged with the sixteen NE tags (not NNE) and POS tags (not representing NEs). In the output, the POS tags are replaced by the NNE tag.

The training set is initially distributed into 6 subsets of equal size. In the cross validation test, one subset is withheld for testing while the remaining 5 subsets are used as the training sets. This process is repeated 6 times to yield an average result, which is called the 6-fold cross

<sup>1</sup>[http://shiva.iiit.ac.in/SPSAL2007/iiit\\_tagset\\_guidelines.pdf](http://shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines.pdf)

validation test. The experimental results of the 6-fold cross validation tests are reported in Table 1.

A close investigation to the experimental results reveals that the precision errors are mostly concerned with the organization names. A possible reason behind the fall in precision of organization names is that generally these do not contain any affixes that may be helpful in the identification of the unknown organization names. Experimental results of the 6-fold cross validation test yield an average F-Score value of 76.6% for the system [5] with the same training set. The NER models, reported in [6], demonstrated the average F-Score values of 71.27% (without linguistic knowledge) and 75.13% (with linguistic knowledge) with the six-fold cross validation test on the same training set. Thus the results reveal that the proposed NER model outperforms the existing NER models for Bengali defined in [5] [6]. To report the results of the transliteration models, the outputs of the Test Set 3 have been chosen as the test sets. The intuition behind this choice is that it produces the highest F-Score value for the NER system.

The performance of the transliteration system is evaluated in terms of Transliteration Unit Agreement Ratio (TUAR) and Word Agreement Ratio (WAR). Let, B is the input Bengali word, E be the English transliteration given by the user in open test and E' be the system-generated transliteration. TUAR is defined as,  $TUAR = (L - Err) / L$ , where L is the number of TUs in E, and Err is the number of wrongly transliterated TUs in E' generated by the system. WAR is defined as,  $WAR = (S - Err') / S$ , where S is the test sample size and Err' is the number of erroneous names generated by the system (when E' does not match with E). In order to develop two different training sets for the transliteration system, 7500 Indian person names and 6000 location names have been collected and their corresponding English transliterations have been stored manually.

The six different transliteration models along with the *baseline* model have been tested with the test sets of person and location names obtained from the output of the Test Set 3 of the NER system. The recognized NEs have been manually checked to discard the incorrectly identified NEs. The transliterations of person and location names are stored in the Gold standard test sets. The test sets of person and location names contain 335 and 197 location names.

The results of the tests in terms of evaluation metrics, WAR and TUAR, are presented in Table 2 for person and location names for Bengali to English (B2E) transliteration. The modified joint source-channel model (Model F) exhibits the best performance with a WAR of 75.4% and TUAR of 91.7% for person names. The same model has also demonstrated best for the transliteration of location names with a WAR of 73.6% and TUAR of 89.3%.

## 5. Conclusion and Future Works

This paper reports about a named entity recognition system in Bengali using a tagged Bengali news corpus based on statistical HMM and the subsequent transliterations of the recognized Bengali NEs into English. We have shown that NER system for Bengali has high recall and good F-Score values with HMM framework. A modified joint source-channel model has been presented to transliterate Bengali named entities into English and vice-versa.

Future works include investigating other methods to boost precision of the NER system. Building NER systems for Bengali using other statistical techniques like MEMM, CRF and analyzing the performance of these systems is another interesting task. The transliteration models are to be trained on additional person and location names.

## References

- [1] A. J. Viterbi. Error Bounds for Convolution Codes and an Asymptotically Optimum Decoding Algorithm. IEEE Transactions on Information Theory (13), 260-269. 1967.
- [2] A. McCallum and W. Li. Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons. In Proceedings of CoNLL-2003.
- [3] Andrew Bothwick. A Maximum Entropy Approach to Named Entity Recognition. Ph.D. Thesis. New York University, 1999.
- [4] Asif Ekbal and S. Bandyopadhyay. Pattern Based Bootstrapping Method for Named Entity Recognition. In Proceedings of ICAPR 2007, Kolkata, India, pp.349-355.
- [5] Asif Ekbal and S. Bandyopadhyay. Lexical Pattern Learning from Corpus Data for Named Entity Recognition. In Proceedings of ICON 2007, Hyderabad, India, pp.123-128.
- [6] A. Ekbal, S. Naskar and S. Bandyopadhyay. A Modified Joint Source Channel Model for Transliteration. In Proceedings of the COLING/ACL 2006, Australia, pp.191-198, 2006.
- [7] Daniel M. Bikel, R. Schwartz, Ralph M. Weischedel. An Algorithm that Learns What's in Name? Machine Learning (Special Issue on NLP), 1999.
- [8] H. Cunningham. GATE: A general architecture for text engineering. Comput. Humanit. (36), 223-254, 2001.
- [9] I. Goto, N. Kato, N. Uratani, and T. Ehara. Transliteration considering Context Information based on the Maximum Entropy Method. Proceeding of the MT-Summit IX, pp.125-132, 2003.
- [10] Li Haizhou, Zhang Min, Su Jian. A Joint Source-Channel Model for Machine Transliteration. 42nd Annual Meeting of the ACL, ACL 2004, Barcelona, Spain, pp. 159-166, 2004.
- [11] T. Brants. TnT: A statistical parts-of-speech tagger. In Proceedings of the sixth International Conference on ANLP, pp.224-231, 2000.

# Experiments with String Similarity Measures in the EBMT Framework

Natalia Elita, Monica Gavrilă, Cristina Vertan  
Faculty of Mathematics, Informatics and Natural Sciences,  
Department of Informatics, University of Hamburg  
{elita, gavrilă, vertan}@informatik.uni-hamburg.de

## Abstract

Measuring string similarity is a frequently used technique in various Language Technology (LT) applications, such as: Spell checkers, Translation Memories, Example-Based Machine Translation (EBMT) etc.

In this paper experimental results on string similarity measures are presented. The main goal of the experiments is to detect the most appropriate similarity measure which can be applied for retrieving candidate sentences for translation templates to be used in an EBMT system. The advantage of this approach is that it is based entirely on surface forms, therefore being independent from any linguistic resources. The results show that token-based measures are the most suitable for translation template extraction.

## Keywords

String Similarity Measures, EBMT, Overlap Coefficient

## 1 Motivation

Measuring string similarity is a frequently used technique in various Language Technology (LT) applications, such as: Spell checkers, Translation Memories, cognates extraction from bilingual texts, sentence and word alignment, Example-Based Machine Translation (EBMT) etc. In this section the motivation to use string similarity measures in the EBMT framework is addressed.

Machine Translation (MT) - translation from one natural language into another by means of a computerized system, (see [1, 6, 5] for more details) - is a task of Natural Language Processing (NLP) that is being continuously studied and many attempts have been made to improve the quality of its output.

There are several approaches to the MT (e.g. rule-based MT, statistical MT etc.). The current paper focuses on the EBMT approach, that was first inspired by Makoto Nagao ([8]). EBMT is an implementation of the translation by analogy principle, which states that humans translate by first decomposing a sentence into sub-phrases, translating these sub-phrases, which are then combined into a translation of a given sentence. A part of any EBMT system is a database of examples, that can be stored as: strings, annotated tree structures, generalized examples (templates), etc. In this paper the template-based EBMT is chosen as

a framework of the present research. In order to get a translation for a given string, three stages have to be performed. First, matching the input on the database of templates, then retrieving the corresponding target language (TL) parts and finally recombining the TL parts into a coherent translation (for details about EBMT in general, and template-based EBMT in particular refer to [4, 7, 10, 11]).

In EBMT similarity measures are used in the matching phase: given an input string in the source language (SL), similar sentences from the database of examples are retrieved, by means of a given similarity measure. In this paper similarity measures are used in the process of building the translation templates. This is realized by means of a Similarity Matrix (defined below), that uses the similarity measures in order to find good candidate sentences (see Example 1), which would later be generalized into templates.

The motivation for such a research comes from the problems encountered while generalizing pairs of sentences into templates, as outlined in [7]. The algorithm used, namely the principle of string co-occurrence and frequency thresholds, states: SL and TL strings that co-occur in two (or more) sentence pairs of a bilingual corpus are likely to be translations of each other.

**Example 1:** Given two entries of an English-German corpus

**1:** <en>Construction of research reactor at Garching underway</en> -> <de>Startschuss fuer Bau des Forschungsreaktors in Garching</de>

**2:** <en>Accompanied by protests , the first sod was turned today for the construction of the new nuclear research reactor .</en> -> <de>Begleitet von Protesten ist heute der Startschuss fuer den Bau des neuen Forschungsreaktors bei Muenchen gefallen .</de>

In the SL part the strings that co-occur are: *construction, of, research, reactor*; in the TL part: *Startschuss, fuer, Bau, des, Forschungsreaktors*.

Thus, the two sentences can be generalized into a template of the form:

[construction of research reactor **V1**] - [Startschuss fuer Bau des Forschungsreaktors **V11**], or  
[**V1** construction of **V2** research reactor **V3**] - [**V11** Startschuss fuer **V21** Bau des **V31** Forschungsreaktors **V41**], where **V<sub>i</sub>** corresponds to a variable in the template.

Hence, the two sentences are good candidates for templates: they are **similar enough** (see section 3). Similarity is calculated on surface-forms only, there-

fore the use of any linguistic resources is unnecessary.

**Definition:** For a monolingual corpus with  $N$  sentences, the **similarity matrix**  $S$  is defined as follows:

$$s(i, j) = -1, \text{ for } j < i, 1 \leq i, j \leq N;$$

$$s(i, i) = 1, \text{ for } 1 \leq i = N$$

$$s(i, j) = BSM(sentence_i, sentence_j), \text{ for } j > i, 1 \leq i, j \leq N;$$

where  $BSM = \text{Best Similarity Measure}$

As sentence similarity is a symmetric property, only values above the main diagonal are examined.

The advantage of using the similarity matrix is that only a sub-corpus, created from these sentences being above a certain threshold, is used as input for the template extraction engine, thus the search space for templates is considerably limited. The thresholds are experimentally determined, as shown in section 3.

Twenty-one similarity measures were analyzed and compared. Those ones performing best were used to build the similarity matrix for a given SL (cf. Figure 1).

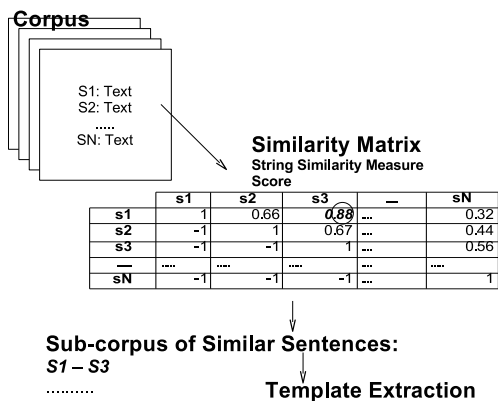


Fig. 1: Extracting similar sentences

The rest of the paper is organized as follows: in the next section, two modified similarity measures are described and the definitions of the measures used to create the similarity matrix are introduced. An account of the results obtained from a series of experiments made on string similarity measures is given in the third section. Finally the conclusion and further work are presented.

## 2 Similarity Measures

String similarity measures are divided in the literature into three categories: character-based, token-based, and hybrid. In the case of the first two, the similarity is calculated on character and token level respectively. In the case of the hybrid measures, the similarity is first calculated on the character level, then the obtained scores are used by a token-based metric. A good definition, purpose, and classification of similarity measures can be found in [3].

In the experiments, twenty-one similarity measures of all three types mentioned above were investigated. Eighteen of these measures are part of the SimMetrics

package (SimMetrics is an open source Java library of similarity measures. For more details refer to [9]). Additionally we modified and extended two of them (**Normalized Token Levenshtein Distance** and **Longest Common Subsequence Similarity**) and implemented one (**Common Words**), for the purpose of finding similar sentences.

### 2.1 Modified Similarity Measures

**Normalized Token Levenshtein Distance (NTLD)** is a modified version of the traditional character-based Levenshtein Distance, and it has the following formula:

$$NTLD(s1, s2) = 1 - \frac{TLD}{2 * \max(\text{Length}(s1), \text{Length}(s2))},$$

where  $TLD$  is the traditional Levenshtein Distance, but at token level, and  $\text{Length}(s)$  means the number of tokens of  $s$ .

The **Longest Common Subsequence Similarity (LCSS)** is based on the Longest Common Subsequence (LCS) character-based algorithm. More details on this algorithm can be found in [2]. The initial algorithm is transformed into a token-based one. This way the token-level LCS between two sentences is computed. Given two sentences  $s1$ , and  $s2$  the computation of the LCSS follows the steps below:

1. Calculation of the LCS at token level:

$$LCS_{TokenString}(s1, s2) = LCS\_String$$

2. Calculation of the LCSS at token level as:

$$LCSS_{Tokens}(s1, s2) = \frac{\text{Length}_{token}(LCS\_String)}{\max(\text{Length}_{token}(s1), \text{Length}_{token}(s2))}$$

3. Subtraction of a penalty of 0.1 for each word-distance, in case the words found in the  $LCS\_String$  are not one after another in the sentences  $s1, s2$ . In case of multiple results, the maximum value is considered. This score is  $LCSS_{Penalties}$ .

4. If the output of step 3. contains multiple results, the longest one (as number of characters), is considered as best the match. The computation is done according to a formula similar to the one in step 2:

$$LCSS_{Characters}(s1, s2) = \frac{\text{Length}_{characters}(LCS\_String)}{\max(\text{Length}_{characters}(s1), \text{Length}_{characters}(s2))}$$

LCSS takes values within  $[0, 1]$ . 0 indicates that the sentences are completely different, and 1 that the sentences are identical.

### 2.2 Other Similarity Measures

In this subsection the definitions of the measures used to build the similarity matrix are presented.

**Common Words (CW)** is a trivial similarity measure that counts the number of identical tokens

(words) for the two given strings. It does not take into account the word order.

**Overlap Coefficient (OC)** is a metric that determines to what degree one string is a substring of another. Its formula is given below:

$$OC(s1, s2) = \frac{|s1 \& s2|}{\min(|s1|, |s2|)},$$

where  $|s|$  is the number of tokens in  $s$ , and  $|s1 \& s2|$  the number of common tokens in  $s1$  and  $s2$ .

### 3 Experimental Results

In this section the experiments we made in order to find similar sentences for template extraction are described and some of their results are presented. Two parallel, sentence aligned corpora were used for the experiments:

- a technical corpus in three languages: German (Ge), Romanian (Ro), and English (En), of approximately 2300 sentences (cca. 25000 tokens for each language);
- a small news corpus of 100 sentences, in German and English

First, the thresholds for each similarity measure were experimentally determined. Then a decision was made on which of the considered similarity measures is more effective for the goal that was set.

For each similarity measure, the initial threshold was established after testing the measure on a small set of artificial examples. Observations were made on how the value changed when the compared sentences were of different length, when the word order was different etc. This value was adjusted afterwards, as a result of testing each measure on the real data, namely, 100 sentences of a corpus, so that the precision increases.

Measure	Threshold Value
Common Words (CW)	initial 5, modified 3
NLTD	0.7
Matching Coefficient	0.55
Block Distance	0.6
Jaccard Similarity	0.45
Overlap Coefficient (OC)	initial 0.66, modified 0.5
Q-Grams Distance	0.65

**Table 1:** *Token-based similarity measures with the established thresholds*

In Table 3 **CONC** means the Chapman Ordered Name Compound Similarity. More details on the measures can be found in [9].

A threshold is a minimal value calculated for two similar sentences. A pair of sentences in a SL is considered to be **similar enough**, when the sentences under consideration fulfill the following constraints:

1. have at least three words in common (**CW Threshold**);
2. the sequence of common elements consists of at least 50% content words (lexical words);

Measure	Threshold Value
TagLink Token	0.5
Euclidean Distance	0.5
Smith-Waterman ( <b>SW</b> )	0.6
Smith-Waterman-Gatoh	0.6
Jaro	0.7
Jaro Winkler	0.7
Needleman-Wunch	0.7
Levenshtein Distance	0.75,
Dice Similarity	0.75,
Cosine Similarity	0.75

**Table 2:** *Character-based similarity measures with the established thresholds*

Measure	Threshold Value
Monge-Elkan	0.9
CONC Similarity	0.75
TagLink	0.7

**Table 3:** *Hybrid similarity measures with the established thresholds*

3. one sentence is a sub-sentence of the other to the proportion of 50% (**OC Threshold**).

The closer the value to 1, the more similar the sentences are. The value of 0 indicates that the sentences are completely different, and the value of 1 indicates that the sentences are identical. In the tables 1, 2, 3 an overview of the similarity measures with the established thresholds is given.

In the first experiment, similar sentences were extracted from 100 sentences taken from the technical corpus. This small number of sentences was chosen for an easier interpretation of the results, and in order to make observations and assumptions. The results are reflected in Table 4.

The experiments were run on each of the three categories of measures mentioned in section 2. As a result, the same sentence pairs were extracted by several similarity measures of the same category. That is why the total number of sentences and the unique number are different.

The following observations and conclusions were drawn from the analysis of these data. From each group of similarity measures, the one that extracts the most similar sentence pairs that would be best candidates for the template extraction is chosen.

**Hybrid methods** seemed the most promising in theory, but proved to be not efficient in practice. From this group, TagLink measure, though it extracted a relatively small number of sentence pairs, was chosen as the best.

**Example of sentence-pair extracted:** - English.  
*TagLink: 0.76*

Writing and sending a multimedia message  
Reading and replying to a multimedia message

Although the **character-based measures** extract the biggest number of sentence pairs, they depend a lot on the length of the sentences. They generally are not suitable for the EBMT. A good example is given in [11]. They proved to be quite slow and ineffective for the goal that was set. The sentence-pairs they

Token-based	Ge	En	Ro
CW	4	11	11
Matching coefficient	12	10	9
Block Distance	13	12	13
Jaccard Similarity	12	10	9
OC	24	19	25
Q-Grams Distance	9	9	6
<b>Total</b>	74	71	73
<b>Unique pairs</b>	26	30	31
<b>Character-based</b>			
Character-based	Ge	En	Ro
Levenshtein Distance	1	3	2
Dice Similarity	5	4	3
Cosine Similarity	5	4	3
Euclidean Distance	5	4	3
Jaro	35	32	56
Jaro-Winkler	86	72	109
Needleman-Wunch	24	40	22
SW	83	82	49
SW-Gotoh	107	103	73
Tag Link Token	70	67	62
<b>Total</b>	421	411	382
<b>Hybrid</b>			
Hybrid	Ge	En	Ro
CONC	48	48	29
Tag Link	19	17	19
<b>Total</b>	67	65	48
<b>Unique pairs</b>	58	59	40

**Table 4:** Number of sentence pairs extracted by each similarity measure

extracted were not similar enough to be good candidates for translation templates. Some of the extracted sentence-pairs had in common only some characters. Smith-Waterman-Gotoh extracted the biggest number of sentence pairs in case of German and English, and Jaro-Winkler in case of Romanian.

**Example of sentence-pair extracted:** - German. *SmithWatermanGatoh: 0.6*  
 Kurzmitteilungen  
 Lesen und Beantworten einer Multimedia - Mitteilung

**Token-based similarity measures** proved to be the most effective for the goal.

**Example of sentence-pair extracted:** *CW+OC*  
German  
 Einstellungen fuer Kurzmitteilungen und E-mail - Mitteilungen  
 Einstellungen fuer Multimedia - Mitteilungen  
English  
 Settings for text and e-mail messages  
 Settings for the multimedia messages  
Romanian  
 Setari pentru mesaje text si e-mail  
 Setari pentru mesaje multimedia

The **OC** measure performs the best (highest number of **similar enough** sentences) of all for all the three languages considered. However, considering the type of the corpus, a disadvantage was noticed: **OC** extracts many sentence pairs, where only one or two tokens overlap. This way the length of a possible template is too short. It happens especially in the case when one of the two sentences to be compared is very short, and is totally contained in the other sentence.

This disadvantage can be easily overcome, if **CW**, with an established threshold is used on the set of sentence pairs extracted by the **OC**. When combined, the thresholds were set to 3 for **CW**, and 0.5 for **OC**.

The results of **OC** combined with **CW** (**OC+CW**) were compared with the outcome of the **NTLD** and of **LCSS** combined with **CW** (**LCSS+CW**). The threshold for the **LCSS** was established at 0.33 and for **NTLD** at 0.7.

Experiments were run on the same set of 100 sentences. The results are included in Table 5.

	German	English	Romanian
OC + CW	18	31	27
NTLD	16	39	32
LCSS + CW	14	34	23

**Table 5:** Sentence pairs above the established thresholds

It can be noticed that quantitatively the results are comparable, but qualitatively they differ a lot. The **NTLD** extracts many sentences, where only short sequences overlap. The quality of the sentence pairs extracted by **OC+CW** is higher. Thus these sentence pairs become better candidates for templates. The number of extracted sentence pairs in German is smaller. This can be conditioned by the structural peculiarity of the language, and by the fact that the algorithms are case-sensitive for this language.

**LCSS** extracts valid pairs if combined with **CW**, having the same threshold as in the case of **OC**: 3. Unlike **OC+CW**, **LCSS+CW** considers also the word order of the two compared sentences.

Further, the precision and the recall of the best similarity measures, namely **OC+CW** were computed.

The results are included in table 6.

	German	English	Romanian
Precision	1	0,7	0,96
Recall	1	1	1

**Table 6:** Precision and recall calculated on 100 sentences

The value of recall is always one, as the first and third constraints from the **similar enough** sentences definition (Section 3) were taken into account.

### 3.1 Other Experiments

In this subsection two new experiments are described: the first shows how the number of the extracted similar sentences is influenced by the language (language dependency), the second by the corpus type (corpus dependency).

1. The combination of **OC** with threshold set to 0.5 and **CW** set to 3 was used to build the similarity matrix as this combination proved to be the most effective for the goal. It was built for sets of different size in different languages for the technical corpus (cf. Figure 2). The chart shows us that a comparable number of similar sentence pairs is extracted for English and German, as for Romanian - a smaller number, compared to English and German. Two reasons can explain this outcome:

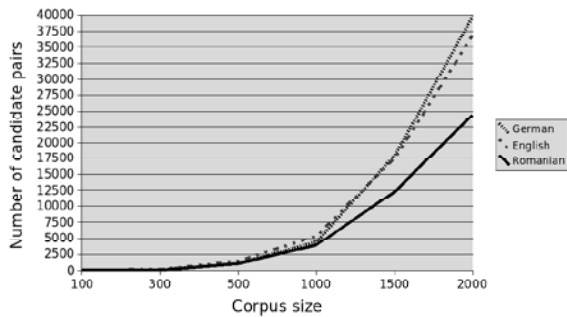


Fig. 2: Sentence pairs extracted - Technical Corpus

- German and English are both Germanic languages, while Romanian is a representative of the Romance languages;
- Compared to the other two languages, Romanian is a highly inflected language, especially in case of nouns and adjectives (e.g. the Romanian word 'lampa' - English 'the lamp' - has six (6) inflected forms).

2. An experiment with different type of corpora was made to check how corpus dependent the amount of extracted sentence pairs is. The results of the experiment run on 100 sentences corpora (technical and news) are shown in figures 3 and 4.

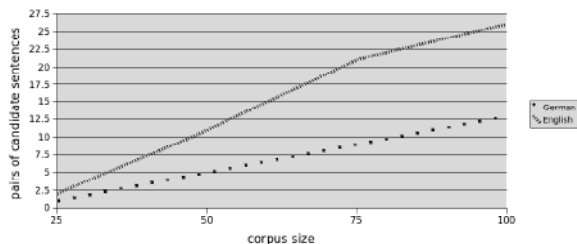


Fig. 3: Sentence pairs extracted - News Corpus

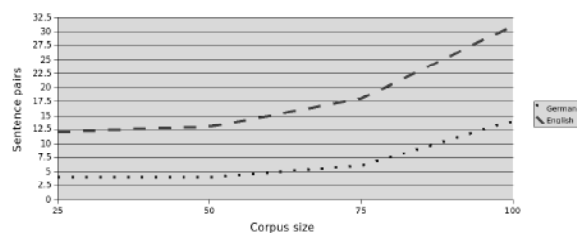


Fig. 4: Sentence pairs extracted - Technical Corpus

From these charts, one can see that the shape of the curves for the two languages is rather similar in the case of the technical corpus, and slightly different for the news corpus. A bigger number of sentence pairs is extracted for the technical corpus due to its restricted language.

A smaller number of sentence pairs is extracted for German in both cases. One of the reason is the value of the **CW** threshold, which is set to 3. A language specific characteristic for German is the composition of

words, which correspond to several words in English: e.g in English: 'the tax reform' reaches the threshold, but its correspondent in German: 'die Steuerreform' is below the threshold. This proves that, in order not to lose data, the thresholds should be language-sensitive.

## 4 Conclusions

In this paper a comparison of string similarity measures in the framework of EBMT is presented. A similarity matrix is defined and used to find similar sentence pairs, that become candidates for translation templates. Twenty-one string similarity measures were analysed, including two modified similarity measures. Experiments were run on two sets of data in three languages. When building the similarity matrix a combination of **CW** and **OC**, or of **LCSS** and **CW** proved to be the most efficient. The number of the similar sentences extracted is influenced by the language and corpus type.

We consider that the established thresholds for the extraction of similar sentences suit the aim that was set. The results obtained will further be used in the template extraction process.

## References

- [1] D. J. Arnold, L. Balkan, S. M. R. Humphreys, and L. Sadler. *Machine Translation: an Introductory Guide*. Blackwells-NCC, London, 1994.
- [2] L. Bergroth, H. Hakonen, and T. Raita. A survey of longest common subsequence algorithms. In *Proc. of the Seventh International Symposium on String Processing and Information Retrieval - SPIRE 2000*, pages 39–48, A Curuna, Spain, September 2000. ISBN: 0-7695-0746-8.
- [3] H. Camacho and A. Salhi. A string metric based on a one-to-one greedy matching algorithm. *Research in Computing Science*, 19:171–182, May 2006.
- [4] I. Cicekli and A. Guvenir. *Learning Translation Templates From Bilingual Translation Examples*, volume Recent advances in Example-based Machine Translation, pages 225–286. Kluwer Acad. Publ., 2003.
- [5] B. J. Dorr, P. W. Jordan, and J. W. Benoit. A survey of current paradigms in machine translation. *Advances in Computers*, 49:2–68, 1999.
- [6] J. W. Hutchins. *Machine Translation: A brief History*, volume Concise History of the Language Sciences. From the Sumerians to the Cognitivists, pages 431–445. Oxford: Elsevier Science Ltd., 1995.
- [7] K. McTait. *Translation Patterns, Linguistic Knowledge and Complexity in an Approach to EBMT*, volume Recent advances in Example-based Machine Translation, pages 307–338. Kluwer Acad. Publ., 2003.
- [8] M. Nagao. A framework of a mechanical translation between japanese and english by analogy principle. In *Proc. of the international NATO symposium on Artificial and human intelligence*, pages 173–180, New York, NY, USA, 1984. Elsevier North-Holland, Inc.
- [9] SimMetrics. <http://www.dcs.shef.ac.uk/sam/simmetrics.html>.
- [10] H. Somers. *An Overview of EBMT*, volume Recent advances in Example-based Machine Translation, pages 3–57. Kluwer Acad. Publ., 2003.
- [11] C. Vertan and V. E. Martin. Experiments with matching algorithms in example based machine translation. In *In Proceedings of the International workshop "Modern approaches in Translation Technologies", in conjunction with RANLP*, September 2005.



# The Importance of Named Entities in Cross-Lingual Question Answering Scenarios

Sergio Ferrández, Óscar Ferrández, Antonio Ferrández,  
Rafael Muñoz

Department of Software and Computing Systems  
University of Alicante, Spain  
{*sferrandez, ofe, antonio, rafael*}@*dlsi.ua.es*

## Abstract

This paper presents a study on the impact and the need for Named Entity Recognition (NER) in Cross-Lingual Question Answering (CL-QA) in order to overcome the errors that usually occur by referring to entities in different languages. The motivation behind introducing NER in a CL-QA scenario is detecting whether or not entities need to be translated. For this research, an English-Spanish Question Answering (QA) system and a NER tool are used. Moreover, we show a study on the need for translating and non-translating the cross-lingual references of named entities. The experimental evaluation on each question set employed in the official CLEF 2004, 2005 and 2006 evaluation campaigns proves that the current approach to CL-QA improves the overall accuracy of the initial CL-QA system, at the same time yielding better results than other current bilingual QA systems.

## Keywords

Cross-Lingual Question Answering, Named Entity Recognition, Multilingual environments.

## 1 Introduction

In the recent years, the exponential growth of digital information requires processes capable of searching, filtering, retrieving and classifying this information as pertinent from large volumes of texts. Moreover, the relevant information required by the users might be in different languages and from several sources. Obviously, this is one of the difficulties that impedes the right acquisition of information.

In order to achieve this purpose, applications like Information Extraction (IE), Information Retrieval (IR) and Question Answering (QA) are used. Besides, the need for IR tools that permit to accede to multilingual information is of interest to the research community.

IR is the science of searching for information in documents and, QA can be defined as the answering to precise or arbitrary questions formulated by users. Evidently, QA is not a simple task of IR. The aim of a QA system is to find the correct answer to a user question in a non-structured collection of documents. In Cross-Lingual (CL) environments, the question is formulated

in a different language from the one of the documents, which increases the difficulty. As it was revealed in the Cross-Language Evaluation Forum (CLEF) 2006 [9], multilingual IR and QA tasks have been recognized as an important issue in the information access.

The overall accuracy of CL-QA systems is directly affected by their ability to correctly analyze and translate the question that is received as input. An imperfect or fuzzy translation of the question causes a negative impact on the overall accuracy of the systems [4]. To overcome this, currently most of the implementations [11, 13, 14] are based on the use of on-line translation services, and some of them use Machine Translation (MT) techniques. However, MT systems generate errors such as translations of names that should be left untranslated. The impact of this kind of mistakes should be controlled and assessed.

In this paper, we present an approach for reducing the aforementioned mistakes within the CL-QA task. Our strategy combines a CL-QA system, which performs the references between words in different languages using the Inter Lingual Index (ILI) module of EuroWordNet (EWN) [15], with a Named Entity Recognition (NER) tool [2]. The original contributions of this research consist in knowing how the NER component lessens the errors committed for wrong references to ILI, even if a Machine Translation (MT) is used, and the need for translating and non-translating the cross-lingual references of named entities. Moreover, the empirical experimentations exposed throughout the paper prove the importance of knowing when the entities should be translated or not.

The rest of the paper is organized as follows: section 2 presents our approach to CL-QA system adding a NER tool. In the section 3, the experiments are shown and discussed. And finally, section 4 wraps up the paper with our conclusions and future work proposals.

## 2 Proposed method to Intertwine NER and CL-QA

Our approach is made up of two main components: the first one consists of a NER tool and the other one is an open domain CL-QA system. The former embodies various machine learning algorithms in order to detect and classify the entities in the texts, and the latter component is based on complex syntactic pat-

Original Question at CLEF 2006	Entities
0007 How many countries are members of <i>NATO</i> at the moment?	ORG( <i>NATO</i> )
0129 Which organization did <i>Shimon Peres</i> chair after <i>Yitzhak Rabin's</i> death?	PER( <i>Shimon Peres</i> ) PER( <i>Yitzhak Rabin</i> )
0179 Where is <i>Ystad</i> located?	LOC( <i>Ystad</i> )
0180 Who created the operating system <i>OS/2</i> ?	MISC( <i>OS/2</i> )

**Table 1:** *Entities at CLEF 2006.*

terms using Natural Language Processing (NLP) tools. In the following subsections, we describe in detail these components, the strategy to combine them and a detailed study on the need for translating Named Entities (NEs) in CL-QA.

## 2.1 The NER tool

Concerning the NER component, we have used the system architecture presented in [2]. This system combines three classifiers by means of a voting strategy and carries out the recognition of entities in two phases: detection of entities and classification of the detected entities. The three classifiers integrated are based on Hidden Markov Model, Maximum Entropy and Memory-based learner algorithm. The outputs of the classifiers are combined using a voting strategy. Each classifier has weight depending on its performance for each one of the categories. When at least two of the three classifiers agree, the category of the entity is the one with the highest number of votes. When the classifiers disagree, the class from the classifier which weight is the highest is selected.

The features used by this method are mainly lexical, orthographical, contextual, morphological and statistical [3]. However, this system also provides a set of language independent features, which makes this system easily portable to other languages.

This system is able to recognize four entity types, assigning to each detected entity one of the following categories: LOCation, ORGanization, PERson or MISCellaneous (miscellaneous category is assigned when the detected entity cannot be enclosed in any of the aforementioned categories). The system has been evaluated in [3] with a corpus provided by CoNLL-2002<sup>1</sup>, achieving overall results around 81%.

In Table 1, we present examples showing questions from the CLEF QA 2006 dataset and the detected and classified entities by the NER tool for each question.

In the case of this research, although this system was initially developed for Spanish, we have made an extension of this system in order to apply it to other languages such as English. Its architecture and the language independent feature sets used by the system made this extension easy and possible. Moreover, we have tailored the training phase by having created different training datasets from the question sets pro-

vided by last editions of CLEF. These datasets were annotated manually, and they were merged in order to obtain the final training sets. For instance, the training corpus used for applying this NER tool to the official English questions of CLEF 2006 was generated from the annotated questions belonging to the 2004 and 2005 editions.

Regarding this work, the system obtained around 60% in overall precision of each year's question sets, with the person category being the best classified. This category achieved results higher than 80% both in precision and recall. We would like to point out the small size of the training corpus; in order to evaluate the system for each question dataset, the training corpus is made of the question datasets belonging to the remained two years, i.e. 400 questions. We expect that increasing the training corpus size would imply obtaining higher precision and recall results.

## 2.2 The CL-QA system

The fundamental characteristic of our CL-QA system [5] is the strategy used for the question processing module in which the ILI Module of EWN is used with the aim of reducing the negative effect of question translation on the overall accuracy.

Our open domain CL-QA system is designed to answer English questions from Spanish documents. The system is based on complex syntactic pattern matching using NLP tools [1, 8, 12]. Also, a new proposal of Word Sense Disambiguation (WSD) for nouns (presented in [7]) is applied to improve the precision.

The system introduces two improvements: (1) the consideration of more than only one translation per word by means of using the different synsets of each word in the ILI module of EWN; (2) unlike the current bilingual English-Spanish QA systems, the question analysis is developed in the original language without any translation. The system develops two main tasks in the question analysis phase:

- The detection of the expected answer type. The system detects the type of information that the answer has to satisfy to be a candidate of an answer (proper name, quantity, date, ...).
- The identification of the main Syntactic Blocks (SB) of the question. The system extracts the SB that are necessary to find the answers.

In order to show the complete process, an example of a question at CLEF 2006 is provided:

- **Question 107 at CLEF 2006:** *How many soldiers does Spain have?*

- **SB:**

[Noun Phrase *soldier*]

[Verb Phrase *to have*]

[Noun Phrase *Spain*]

- **Type:** entity-amount
- **Keywords to be referenced with ILI:** soldier have Spain

- **soldier**  $\mapsto$  soldado

<sup>1</sup> <http://www.cnts.ua.ac.be/conll2002/>

Dataset	Questions					
		overall	PER	LOC	ORG	MISC
Questions CLEF 2006	with NEs	89%	31%	24.5%	22.5%	24%
	NEs should be translated	42.69%	3.2%	65.3%	40%	50%
Questions CLEF 2005	with NEs	93%	34%	25.5%	24%	13.5%
	NEs should be translated	36%	10.3%	50.9%	39.6%	55.5%
Questions CLEF 2004	with NEs	81%	23.5%	28%	15%	20.5%
	NEs should be translated	44.89%	2.1%	60.7%	56.7%	48.8%

**Table 2:** Percentage of questions containing NEs and percentage of NEs that should be translated.

- **have**  $\mapsto$  estar-enfermo tener padecer sufrir causar inducir hacer consumir tomar ingerir experimentar tener poseer tener recibir aceptar querer constar figurar existir

- **Spain**  $\mapsto$  España

In some of the cases, the system finds more than one Spanish equivalent for one English word. The current strategy employed to get the best translation consists of assigning a weight depending on the frequency of each word in ILI. The words that are not in EWN are translated into the rest of the languages using an online Spanish Dictionary<sup>2</sup>. Also, the system uses bilingual gazetteers of organizations and places in order to translate words that have not been linked using ILI. Moreover, in order to decrease the effect of incorrect translation of the proper names, the matches using these words in the search of the answer are realized using the translated word and the original word of the question.

The strategy followed by our CL-QA system has been ranked first in the English-Spanish QA task at CLEF 2006 [6, 9].

### 2.3 The need for translating NEs

This section presents a study on the need for translating NEs in CL-QA. The dataset used has been the official 600 English questions of CLEF 2004, 2005 and 2006. The aim of this study is to find out solutions in order to overcome the errors in the references of NEs between different languages. We provide results on how important is to translate NEs in CL-QA, taking into account most of the mistakes regarding wrong ILI references, and how they can be successfully translated. Normally, wrong ILI references are caused by trying to translate a person's name that should not be translated.

Table 2 presents the results on our study to find out the percentage of questions that contain NEs and the percentage of these NEs that need to be translated. The percentage of questions with NEs is quite high (81% for 2004, 93% for 2005 and 89% for 2006, i.e. 87.7% on average), and nearly half of these NEs should be translated (44.89% for 2004, 36% for 2005 and 42.69% for 2006, i.e. 41.2% on average). Considering each entity type individually, it can be seen that it is very important for CL-QA to translate locations, organizations and miscellaneous entities while the impact of not treating person entities would be low.

<sup>2</sup> <http://www.wordreference.com>

In a nutshell, the study has proved that it is important to translate NEs in CL-QA. It has also been revealed that a specialised treatment should be carried out depending on the entity type. Concretely, ILI's performance for person entities is very low. In fact, the CL-QA system obtains better results if person entities are not translated at all than if they are translated by ILI. These aspects will be discussed in detail in the following sections.

### 2.4 The addition of NER to the CL-QA system

The main goal to achieve would be to detect when a keyword within an entity should not be translated. This fact involves the need for intertwining NER and CL-QA in order to obtain this kind of knowledge. Therefore, the decision taken is to apply NER to recognize four types of entities (ORG, PER, LOC and MISC) and incorporate this knowledge into the CL-QA system. Then, the CL-QA system takes the entities and, if the type of the entity is PER, no reference to ILI will be done for the words within the entity. Besides, the empirical tests (shown in the next section) proves the importance of recognizing entities to improve the keywords translations. This strategy is proposed as our novel method to enrich the ILI translations and non-translations in CL-QA environments.

Tables 3 and 4 show two examples that detail how the CL-QA system chooses the keywords of the entities which are not translated. This fact results in an obvious improvement to the answer extraction phase.

For example, in Table 3, the proper name "Jan" is confused with the abbreviation of the month "January" by both the ILI module of EuroWordNet and the MT service. In this case, the need for some kind of treatment such as NER is clear, because without this kind of information, the CL-QA system is not able to answer the question.

Next, the example shown in Table 4 illustrates an error which happens when using ILI as the references to translate question terms. These situations generate errors, which modify completely the sense of the question and cause a considerable negative effect in the precision of the CL-QA system.

In the previous example, the proper name "John" is translated into the words "San Juan" by the ILI module of EWN. This situation causes a wrong sense of the question that does not permit to localize the correct solution in the answer extraction phase. However, by adding NER, the keyword "John" is classified

Language	Question 184 CLEF 2006
English Question	Who is Jan Tinbergen?
Spanish Question	¿Quién es Jan Tinbergen?
Translation	¿Quién es <b>enero</b> Tinbergen?
Keywords	
CL-QA + NE	Jan Tinbergen
CL-QA system	<b>enero</b> Tinbergen
MT	<b>enero</b> Tinbergen

Table 3: Question 184 CLEF 2006.

Language	Question 194 CLEF 2006
English Q.	How much was John Fashanu fined?
Spanish Q.	¿A cuánto ascendió la multa a John Fashanu?
Translation	¿Cuánto se multó John Fashanu?
Keywords	
CL-QA + NE	John Fashanu
CL-QA system	<b>San Juan</b> Fashanu
MT	John Fashanu

Table 4: Question 194 CLEF 2006.

as PER entity and it is not referenced producing a right translation.

On the other hand, the current strategy employed to reference the entities LOC and ORG consists of trying to link these entities using the ILI module. For instance, in the question 107 at CLEF 2006, *How many soldiers does Spain have?*, the LOC entity “Spain” is referenced to “España” using the ILI module. However, when there are entities that are not referenced in the ILI module, the system uses bilingual gazetteers of organizations and places in order to translate the entities of these questions.

## 3 Evaluation

### 3.1 Dataset

The experiments detailed in this section have been carried out using our CL-QA strategy (CL-QA system + NE recognizer), which has been compared to our monolingual Spanish QA system [10].

For making the evaluation, the CLEF 2004, 2005 and 2006 sets of 600 English and Spanish questions and the EFE 1994–1995 Spanish corpora are used. These corpora provide a suitable framework in order to check the CL-QA system precision.

The set of questions is composed of “*factoid questions*” and “*definition questions*”. The factoid questions are fact-based questions, asking for the name of a person, a location, the extent of something, the day on which something happened, etc. The definition questions have the structure: “What/Who is X?”.

Furthermore, with regard to the training corpora created for applying the NE recognizer, we have carried out the following strategy. We have annotated manually all question datasets (2004, 2005 and 2006) and in order to apply NER to the 2006 question set, we have used as a training corpora the question sets

belonging to 2004 and 2005 editions. For the 2005 question set, the 2004 and 2006 datasets were used as a training corpus, and finally, for the 2004 question set we have merged the 2005 and 2006 question sets in order to create the training corpus.

### 3.2 Results Analysis

The aim of these experiments is to check if our novel approach, adding a NE recognizer in order to control the references between languages, can improve the treatment of person entities.

Systems’ performance is shown in Table 5. The rows 1, 3 and 5 show the obtained precision<sup>3</sup> of our monolingual Spanish QA system for each dataset.

The remaining rows illustrate the experiments carried out using our CL+NER strategy presented in this paper (see rows 2, 4 and 6). Finally, the columns show the number of questions that contain person entities (column 4), the number of affected keywords which improve their translations because they are not referenced to ILI (column 5) and the number of questions that produce a gain in overall precision of the bilingual CL-QA task (column 6).

The results show the positive effect of the addition of NER in the question translation phase. For instance, in the 2006 dataset, 66 of 200 questions have at least one person entity. These detected entities will not be referenced to ILI and, therefore will not be translated. This fact improves the translation of 20 entities, by removing wrong ILI translations of entity words that should not be translated, and obtains better results. The reason why only for 20 entities (out of 66) there are changes in the translations is due to the fact that the remaining 46 are not present in ILI and therefore they would not be translate even in the absence of NER (e.g. the PER entity “*Iosif Kobzon*” in question 5 at CLEF 2006 (*Who is Iosif Kobzon?*)). However, from the 20 entities that change only four questions produced improvements in the final precision of the system. This is due to two reasons: 1) NIL<sup>4</sup> answers: some wrong translations take place in questions for which the answer is NIL. Therefore, in these cases improving the translation does not imply improving the results (e.g. question 38 at CLEF 2006 *To which organisation is Peter Anderson the alcohol adviser?*); and 2) Partial Translations: some entities are partially referenced in ILI and thus partially translated (e.g. Bill Clinton → ILI → *cuenta* Clinton). However, when applying NER, these entities are not translated at all (e.g Bill Clinton → NER → Bill Clinton). Even if the entity is incorrectly partially translated, the CL-QA system can retrieve a correct answer. Therefore, although it provides a better translation, it does not get a better result.

On the other hand, the experiments proves that our CL strategy obtains better results than other current bilingual QA systems. This affirmation can be corroborated checking the official results on the last edition of CLEF 2006 [9] where the precision on English-Spanish CL task was approximately 50% lower than the mono-

<sup>3</sup> It is also considered the inexact answer that contain more (or less) information than that required by the query

<sup>4</sup> NIL means that there is no answer in the corpus.

	Approach	Dataset	Prec.	Questions with PER	Improved Translations	Gain in QA
1	Spanish QA system	CLEF'06	50.5%	-	-	-
2	CL+NER Strategy	CLEF'06	44%	66	20	4
3	Spanish QA system	CLEF'05	51.5%	-	-	-
4	CL+NER Strategy	CLEF'05	42.5%	58	13	4
5	Spanish QA system	CLEF'04	41.5%	-	-	-
6	CL+NER Strategy	CLEF'04	33.5%	49	14	6

**Table 5:** Results regarding person entities and overall improvements.

lingual Spanish task (our method only around 20% lower).

## 4 Conclusion and Future Work

This paper presents a study on the official 600 English questions of CLEF 2004, 2005 and 2006 with the aim to valuate the impact and the need for intertwining NER and CL-QA. This study leads us to make a new proposal combining NER to CL-QA in order to overcome the existing problems within the references of entities between different languages. The main contribution of NER is to provide the indispensable knowledge to decide when the entities of the questions should be translated or not. The tests on the official CLEF set of English questions proves that the use of the NE recognizer improves the overall accuracy of our CL-QA system (4 questions at CLEF 2006, 4 questions at CLEF 2005 and 6 questions at CLEF 2006) and on the other hand, the system also obtains better results than other current bilingual QA systems [9].

Further work will study the possibility to take into account the multilingual knowledge extracted from Wikipedia<sup>5</sup>, in order to translate the LOC, ORG and MISC entities that have different names in English than in Spanish. Furthermore, there are situations where the PER entities should be translated. These situation will be studied and overcome, for instance, in the question 022 at CLEF 2006 “Which country was pope John Paul II born in?”, the PER entity “John Paul II” must be translated into “Juan Pablo II” in order to permit the system to find out the correct answer.

## Acknowledgments

This work has been performed by the QALL-ME consortium, which is a 6th Framework Research Programme of the European Union (EU), contract number: FP6-IST-033860. For more information about the QALL-ME consortium, please visit the consortium home page, <http://qallme.itc.it/>, and by the Spanish Government under the project CICYT number TIN2006-1526-C06-01.

<sup>5</sup> <http://www.wikipedia.org/> - A Web-based free-content multilingual encyclopedia project

## References

- [1] S. Acebo, A. Ageno, S. Climent, J. Farreres, L. Padró, R. Placer, H. Rodriguez, M. Taulé, and J. Turno. MACO: Morphological Analyzer Corpus-Oriented. *ESPRIT BRA-7315 Aquilex II, Working Paper 31*, 1994.
- [2] Ó. Ferrández, Z. Kozareva, A. Montoyo, and R. Muñoz. NERUA: sistema de detección y clasificación de entidades utilizando aprendizaje automático. In *Procesamiento del Lenguaje Natural*, volume 35, pages 37–44, 2005.
- [3] Ó. Ferrández, A. Toral, and R. Muñoz. Fine Tuning Features and Post-processing Rules to Improve Named Entity Recognition. In *Proceedings of the 11th International Conference on Applications of Natural Language to Information Systems, NLDB 2006*, pages 176–185, Klagenfurt, Austria, Junio 2006.
- [4] S. Ferrández and A. Ferrández. The Negative Effect of Machine Translation on Cross-Lingual Question Answering. *CI-CLing 2007. Lecture Notes in Computer Science*, 4394:494–505, 2007.
- [5] S. Ferrández, A. Ferrández, S. Roger, P. López-Moreno, and J. Peral. BRILI, an English-Spanish Question Answering System. *Proceedings of the International Multiconference on Computer Science and Information Technology. In 1st International Symposium Advances in Artificial Intelligences and Applications (AAIA '06)*, ISSN 1896-7094:23–29, November 2006.
- [6] S. Ferrández, P. López-Moreno, S. Roger, A. Ferrández, J. Peral, X. Alavaredo, E. Noguera, and F. Llopis. AliQAn and BRILI QA System at CLEF-2006. In *Workshop of Cross-Language Evaluation Forum (CLEF)*, September 2006.
- [7] S. Ferrández, S. Roger, A. Ferrández, A. Aguilar, and P. López-Moreno. A new proposal of word sense disambiguation for nouns on a question answering system. *Advances in Natural Language Processing. Research in Computing Science. ISSN: 1665-9899*, 18:83–92, February 2006.
- [8] F. Llopis and J. Vicedo. Ir-n, a passage retrieval system. In *Workshop of Cross-Language Evaluation Forum (CLEF)*, 2001.
- [9] B. Magnini, D. Giampiccolo, P. Forner, C. Ayache, V. Jijkoun, P. Osevana, A. Peñas, P. Rocha, B. Sacaleanu, and R. Sutcliffe. Overview of the CLEF 2006 Multilingual Question Answering Track. In *Workshop of Cross-Language Evaluation Forum (CLEF)*, September 2006.
- [10] S. Roger, S. Ferrández, A. Ferrández, J. Peral, F. Llopis, A. Aguilar, and D. Tomás. AliQAn, Spanish QA System at CLEF-2005. In *Workshop of Cross-Language Evaluation Forum (CLEF)*, 2005.
- [11] B. Sacaleanu and G. Neumann. DFKI-LT at the CLEF 2006 Multiple Language Question Answering Track. In *Workshop of Cross-Language Evaluation Forum (CLEF)*, September 2006.
- [12] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of NemLap-94*, pages 44–49, Manchester, England, 1994.
- [13] R. Sutcliffe, K. White, D. Slattery, I. Gabbay, and M. Mulcanhy. Cross-language French-English Question Answering using the DLT System at CLEF 2006. In *Workshop of Cross-Language Evaluation Forum (CLEF)*, September 2006.
- [14] D. Tomás, J. Vicedo, E. Bisbal, and L. Moreno. Experiments with LSA for Passage Re-Ranking in Question Answering. In *Workshop of Cross-Language Evaluation Forum (CLEF)*, September 2006.
- [15] P. Vossen. Introduction to eurowordnet. *Computers and the Humanities*, 32:73–89, 1998.

# Minimal Sets of Minimal Speech Acts

Debora Field  
Dept of Computer Science  
University of Liverpool, L69 3HX, UK  
*debora@csc.liv.ac.uk*

Allan Ramsay  
Dept of Computer Science  
University of Manchester, M60 1QD, UK  
*allan.ramsay@manchester.ac.uk*

## Abstract

Work on speech acts has generally involved the introduction of sets of different actions such as informing, reminding, bluffing, and lying, which have different preconditions and effects, and hence can be used to achieve a wide variety of different real-world goals. They tend to have indistinguishable surface forms, however. As such, it is extremely hard for the hearer to decide which action she thinks has been performed. It is therefore also extremely difficult for the speaker to be confident about how the hearer will respond.

We will show how to achieve complex goals on the basis of a very simple set of linguistic actions. These actions have clearly marked surface forms, and hence can easily be distinguished by a hearer. In order to do this, we have developed an epistemic planner with several interesting features, and with several optimisations that relate directly to aspects of the task at hand.

## Keywords

speech acts, linguistic actions, epistemic planner

## 1 Introduction

The idea that linguistic actions should be treated as similarly as possible to other actions has been widely discussed, from Searle's [27] development of Austin's [2] ideas through the combination of these ideas with classical AI planning theory [1, 8] to collections such as [7] and [6]. Most of this work introduces a variety of actions with radically different preconditions and effects, but with a very small number of surface markers. Simple declarative sentences, for instance, can be used for informing, or for reminding, or nagging, or lying, or bluffing, or . . .

Under these constraints, it is extremely difficult to see how hearers could decide which variety of action were being performed at any given moment. Hence a speaker could have little confidence that the hearer would behave as he expects. Several authors have therefore argued for a much simpler set of actions, with clearly distinguishable surface forms [5, 22]. This happily removes one burden from the participants in a conversation, namely identifying the action that is being performed. The cost is that they must do a considerable amount of inference.

In this paper we explore the use of an extremely sparse notion of 'linguistic act', one with no preconditions and a single effect of adding its own existence to the 'minutes' [19] of the conversation. The aim is to see what can be done with an absolutely minimal notion of speech act by reasoning from first principles. In everyday language, much of

the reasoning we describe below is 'frozen', but we would like to see if we can derive the consequences of more complex acts by reasoning about the effects of this very simple act in a variety of epistemic contexts.

We use a tightly integrated planner and epistemic inference engine to construct plans that achieve a set of epistemic goals, and a plan recogniser integrated with the same inference engine to determine the goals that lie behind a given utterance in a given context. We will show that you can achieve quite complex goals using a very constrained set of very simple speech acts that can be determined on the basis of surface form—claims, polar questions, WH-questions and commands—and that have extremely simple preconditions and effects. This is an extension of [11]'s suggestion that there is a single linguistic act, namely the act of saying something, with a more detailed examination of the differences between utterances whose surface forms mark them as belonging to one of the four classes above.

The core argument of this paper is that complex uses of the four basic acts can be derived by considering the inferences that can be drawn from them when they are used in specific epistemic contexts, with the planner and theorem prover being used to support this argument. The fact that some of these inference patterns are common enough to have become frozen—nagging, reminding, asking rhetorical questions, *etc.*—is not in doubt. We want to show how they emerged in the first place, and to cast them not as atomic actions that just happen to have identical surface forms, but as common inferencing patterns that may be recognised by a hearer.

## 2 Logical forms

The first stage in any attempt to build a system for manipulating natural language is to determine the relationship between surface form and content. Since surface form is the only thing the hearer H receives from the speaker S, it must contain an encoding of everything S wants to convey to H.<sup>1</sup> It may be that S also wants H to carry out some inference in order to flesh out the content of the current utterance by linking it to what has already been said, to the general context in which the utterance was produced, and to H's general background knowledge. Even so, the information that S wants H to use in order to carry out this inference must be encoded in the surface form, because there is nothing else.

In particular, we believe it is important to include information about the surface speech act in the logical form. In general, an utterance contains a partial description of a state

<sup>1</sup> For spoken language, the form will include the prosodic contour, and for situated language may also include visual clues. The principle is the same: the message is carried by perceptually distinguishable choices.

of affairs (the ‘propositional content’) + a report of S’s attitude to that state of affairs (the ‘mood’). In English, for instance, it is possible to classify utterances into statements, imperatives, polar questions and WH-questions purely on the basis of surface form. The reasons *why* someone might produce a given statement in a given situation are very varied, and cannot be determined just by looking at the surface form; but it is easy to see that it *looks* like a statement.

The simplest way of including the mood in the logical form is just to say the meaning of the utterance was that a sentence with the given propositional content and mood was uttered by S, as in Fig. 1:

(1) Have you seen John?

$$\begin{aligned} & \text{query}(\text{ref}(\lambda A(\text{speaker}(A))), \\ & \quad \text{ref}(\lambda B(\text{hearer}(B))), \\ & \quad \exists C::\{\text{past}(\text{now}, C)\} \\ & \quad \exists D::\{\text{aspect}(C, \text{perfect}, D)\} \\ & \quad \theta(D, \text{agent}, \text{ref}(\lambda E(\text{hearer}(E)))) \\ & \quad \&\theta(D, \text{object}, \text{ref}(\lambda F(\text{named}(F, \text{John}))) \\ & \quad \&\text{see}(D))) \end{aligned}$$

Fig. 1: Logical form for (1): mood as a simple wrapper

In general, logical forms like that in Fig. 1 need to be backed up by meaning postulates (MPs) that flesh out the truth conditions of the various terms. There is no point, for instance, in saying an event  $C$  is in the *simple* aspect with respect to speech time unless we can access a rule that spells out the consequences in terms of the relationship between speech time and the start and end of  $C$ .

The next move, then, is to see what kind of MPs are required for the mood markers. Note that we would have to do exactly this if we exploited a much larger set of actions such as nagging, reminding, bluffing, and so on. There is just as little point in saying S has performed a bluff without specifying the preconditions and effects of bluffing as in saying that he has performed a claim without specifying the preconditions and effects of claims.

As we will see shortly, the most basic reason for asking questions is because the information being queried is likely to help you in some way. We take it (Fig. 2)<sup>2</sup> that

$$\begin{aligned} & \forall S \forall H \forall P (\text{state}(S, H, P) \\ & \quad \rightarrow (\exists A (\text{know}(H, P) \rightarrow \text{poss}(H, \text{do}(A)))))) \\ & \forall S \forall P \forall H (\text{query}(S, H, P) \\ & \quad \rightarrow (\exists A (\text{know}(S, P) \rightarrow \text{poss}(S, \text{do}(A)))))) \\ & \forall S \forall H \forall P (\text{queryValue}(S, H, P) \\ & \quad \rightarrow \exists A \forall E (\text{know}(S, P, E) \rightarrow \text{poss}(S, \text{do}(A)))) \\ & \forall S \forall H \forall P (\text{command}(S, H, P) \\ & \quad \rightarrow \exists A \exists X (\neg \text{poss}(X, \text{do}(A)) \\ & \quad \& \text{do}(H, P) \rightarrow \text{poss}(X, \text{do}(A)))) \end{aligned}$$

Fig. 2: Meaning postulates for surface mood

the surface mood of an utterance says something about S’s goals—that for a statement there is some action  $A$  that H could do ( $\text{poss}(H, \text{do}(A))$ ) if H knew the propositional content  $P$  was true, for a query there is something  $S$  could do if he knew that  $P$  was true, and for an imperative there is something that someone (probably  $S$  or  $H$ , but not necessarily) could do if  $H$  carried out the action described by  $P$  (some imperatives have consequences that are beneficial someone other than  $S$ ).<sup>3</sup>

<sup>2</sup> The notation  $P.E$  in the third rule says  $P$  holds of  $E$ , or  $E$  satisfies  $P$ . The theorem prover (see §3.2) supports reasoning over intensional operators. Space precludes a discussion of this here: see [23] for details.

<sup>3</sup> Note that specifying the truth conditions of a term is not the same as

### 3 Epistemic inference and planning

The treatment of mood under discussion says that utterances explicitly mention S’s goals.  $S$  has a goal that he could achieve under certain circumstances. He constructs a sentence that tells  $H$  he has such a goal, and what this goal depends on.  $H$  is then normally expected to try to guess what  $S$ ’s goal is, and see if she can help with it. To embody this within a computational system, we need to be able to construct partial epistemic plans: on the account given above, linguistic acts arise when  $S$  constructs a plan to achieve a goal and realises it has a hole in it that can be filled by  $H$ . The goal may be a gap in  $S$ ’s knowledge, or it may be an action  $S$  cannot (or does not want to) carry out. We therefore need a planner that can construct plans, often involving reasoning about  $S$  and  $H$ ’s knowledge and belief, which include *hypothetical* actions that could be carried out under different circumstances, but that cannot be carried out as things stand. We also need to be able to recognise what the user’s plan was and to work out how to complete it. We therefore need a planner and an epistemic inference engine that are very tightly integrated.

#### 3.1 Actions with indirect consequences

In our domain, the effect of performing an action depends to a very large extent on the context in which it is executed, so you cannot simply retrieve appropriate actions by looking to see if their effects match your current goals. Instead, you must see if their effects *entail* your goals in the current situation. Furthermore, verifying the preconditions of an action can also require substantial amounts of inference.

Given the difficulties of using the plan-space approach in the current domain,<sup>4</sup> we choose to use a variant on state-space planning. State-space planners typically chain backwards from the preconditions of one action to the effects of another until they find a sequence of actions that can be performed starting in the current situation and leading to one where the system’s goals are true. The crucial differences from the basic STRIPS algorithm are underlined in Fig. 3.<sup>5</sup> Instead of finding a goal that is not a *member* of  $\text{WORLD0}$ , we must find one that is not *entailed* by it; and instead of finding an action that includes the current goal in its effects, we must find one whose effects, when combined with  $\text{WORLD0}$ , entail it.

```
plan(GOALS, PLAN0, PLAN2, WORLD0, WORLDN) :-
  % choose goal not currently provable
  member(GOAL, GOALS), \+WORLD0 |- GOAL,
  % choose action that would make it provable
  action(A, pre(PRE), add(ADD), delete(DELETE)),
  WORLD0+ADD-DELETE |- GOAL,
  plan(PRE, [], SUBPLAN, WORLD0, WORLD1),
  append(ADD, WORLD1, WORLD2),
  deleteAll(DELETE, WORLD2, WORLD3),
  append(PLAN0, SUBPLAN, PLAN1),
  plan(GOALS, [A|PLAN1], PLAN2, WORLD3, WORLDN) .
```

Fig. 3: State-space planner for actions with indirect effects

saying that every sentence containing it is true. This is obvious enough for terms like *simple* or *know* that appear inside the propositional content. It is less obvious for the truth conditions of *query* and *statement*, but it still holds. The truth conditions explain what the world would be like if the sentence were true. A large part of the flexibility of language arises from the ability of speakers to say things that are not true, and of hearers to recognise when this has happened and why.

<sup>4</sup> In plan-space planning, actions must have static effects.

<sup>5</sup> The planner in Fig. 3 is extremely simple, e.g., it does not address goal interaction ([26, 30, 31]...), however, it suffices for this discussion.

### 3.2 Inference engine

It is clear, then, that we need a notion of entailment—we need an inference engine that can not only carry out proofs, but that can also retrieve actions that *would* make some proposition true if they were performed. The inference engine also has to be able to reason over belief sets.

We start by following the observation underlying Satchmo [20] that very large parts of our everyday knowledge can be expressed just with Horn clauses, and hence in Prolog. We use an adaptation of Satchmo as our basic engine. The basic algorithm is given in Fig. 4.<sup>6</sup> (i) and (ii)

```
% (i) Can you prove it just using Prolog?
prove(P) :- P.

% (ii) Do you have a disjunction where
% each branch supports the required conclusion?
% (Davis-Putnam)
prove(P) :-
    (Q or R),
    (Q => P),
    (R => P).

% (iii) Constructive (relevant) implication
(P => Q) :-
    % \+ prove(Q), % for relevance logic
    assert(P),
    (prove(Q) ->
        retract(P);
        (retract(P), fail)).
```

**Fig. 4:** Basic constructive Satchmo

are essentially the Davis-Putnam procedure [9]. (iii) embodies the constructive view of implication, that to prove  $P \rightarrow Q$  you must show that assuming  $P$  is true will lead you inexorably to accept  $Q$ : the test that  $P$  is not already provable turns this into strict/relevant implication by establishing that  $P$  is essential to the proof of  $Q$  [3]. This is the key difference between constructive and classical logic.<sup>7</sup>

The outline algorithm in Fig. 4 depends on the assumption that any Horn clauses in the problem statement have been turned into Prolog, so that they can be exploited in (i) and in the first step of (ii). For our present purposes, this basic inference engine has to be extended in two ways: we must support reasoning about beliefs, and find actions whose effects would entail a given goal if they were performed in the current context.

#### Epistemic reasoning using contexts

We believe that Fig. 4(iii) is particularly appealing as an account of implication for epistemic reasoning, because we take the view that the best way to reason about what someone else believes is to see what conclusions you would draw if your view of the world matched theirs. Reasoning about other people’s beliefs involves ascribing some basic set of propositions to their belief set, and then working out what you would do if you had that information. The basic set is typically constructed from a mixture of sources: what the other person says and what other people say about them, direct observation of people’s perceptions, and general assumptions about communally shared beliefs.<sup>8</sup>

<sup>6</sup> As with Fig. 3, the actual implementation is somewhat more complex.

<sup>7</sup> Within the framework of constructive logic,  $\neg P$  is taken to be an abbreviation for  $P \rightarrow \perp$ . Since (iii) deals with reasoning about formulae of the general form  $P \rightarrow Q$ , it can be used for formulae where  $Q$  is  $\perp$ , so we do not need any special machinery for dealing with negation.

<sup>8</sup> None of this is reliable—people can lie, observations can be mistaken, and general assumptions can fail. To make matters worse, reasoning

If we assign only a basic set of beliefs, however, our picture of their view of the world will not be very rich. We must work out what we would do if we had their beliefs and inferential capabilities (usually assumed to be similar to ours). Because proofs are necessarily finite, and practical theorem provers are necessarily resource-bounded (hence incomplete), this approach, taken by [16], avoids some of the more unintuitive consequences (logical blindness, logical omniscience) of thinking about belief in terms of possible worlds, as introduced by [14] and very widely followed. By accepting that reasoning over belief sets, by people and by automatic theorem provers, is resource-bounded, we avoid assuming that belief sets are deductively closed.<sup>9</sup>

We incorporate this notion into our theorem prover by introducing the ‘context’ in which a proposition is available (Fig. 5). We write  $P :: C$  to say proposition  $P$  is available in

```
% (i) Can you prove it just using Prolog?
prove(P) :- P.

% (ii) Do you have a disjunction where
% each branch supports the required conclusion?
% (Davis-Putnam)
prove(P) :-
    (Q::CQ or R::CR)::C,
    (Q::(CQ+C) => P),
    (R::(CR+C) => P).

% (iii) Constructive (relevant) implication
(P => Q) :-
    % \+ prove(Q), % for relevance logic
    assert(P),
    (prove(Q) ->
        retract(P);
        (retract(P), fail)).
```

**Fig. 5:** Satchmo with contexts

context  $C$ , and we let belief statements introduce contexts. Nested beliefs are dealt with by representing belief contexts as lists, with the innermost believer as the head of the list.

The revised version of Satchmo continues to exploit the fact that Horn clauses can be converted to pure Prolog—the only place where the new version differs from the original is in the distribution of the context in which a disjunction is proved across the disjuncts (ii). For this to work, we must make use of several axioms during the normal forming process. In particular, we use the rules in Fig. 6.

$$\begin{aligned} \vdash \text{bel}(X, P \ \& \ Q) &\equiv \vdash \text{bel}(X, P) \ \& \ \text{bel}(X, Q) \\ \vdash \text{bel}(X, P \rightarrow Q) &\equiv \vdash \text{bel}(X, P) \rightarrow \text{bel}(X, Q) \end{aligned}$$

**Fig. 6:** Normal form rules

The first rule is uncontroversial. The second, crucial to the conversion of epistemic rules into Horn clauses, requires that each half of the equivalence is acceptable under constructive logic (argument omitted due to space constraints).

The discussion above assumes that  $\text{bel}(X, P)$  means something like ‘It is reasonable to assume  $X$  will carry out the inference required to derive  $P$  from his base beliefs’. It is not, of course, possible to know exactly what someone

over beliefs requires a treatment of *degrees* of belief, which remains an open problem. Nonetheless, all reasoning about other people’s beliefs must start from some such ascription of a basic belief set.

<sup>9</sup> It is, of course, implausible that the resource bounds on our theorem prover correspond exactly to the point where a person would cease to reason about some set of beliefs. Nonetheless, by taking belief to be a constructive/proof-theoretic notion, we build in the assumption that it is resource-bounded and hence not closed under deduction. The resource bounds are omitted from Fig. 4 for clarity’s sake.



will do with his beliefs, so you can never be sure how much inference he will be prepared to carry out. The most sensible thing is to assume he will do roughly the same amount *you* would do: if you can derive a conclusion from what you think he believes using a reasonable amount of effort, it is reasonable to assume he can *and will* do the same. Short of telepathy, there is nothing else you can do, even though the conclusions you draw may be inaccurate.

### Hypothetical reasoning

From Fig. 3 we need to be able to find an action A whose effects E would entail a goal G in the current situation S if the action were performed. To do this, we have to show that there is a proof of G from E + S, and to remember that this proof depended on A. We do this by transforming action descriptions into hypothetical rules. Consider Fig. 7. From

```
action(paint(X, B, G),
      pre(isPaint(P) & has(X, P) & colour(P, G)),
      effects(colour(B, G)))
```

**Fig. 7:** *If you paint something you will change its colour*

this we can obtain a rule that says B would be coloured G if you were to paint it with G-coloured paint:

```
hypothesis(action(paint(X, B, G))::context([])
           => colour(B, G)::context([]))
```

**Fig. 8:** *B would be coloured G if you painted it*

This rule can be used in a proof, just like any other rule. To use it, of course, we must be able to ‘prove’ the antecedent, which we do simply by noting the hypothesis that this action is required for the proof to go through. Note that we do not attempt to perform the action now, so its preconditions are ignored. Deciding which actions to actually perform, and in what order, is the job of the planner. The planner asks the theorem prover to try to prove the goals, possibly using hypothetical actions (Fig. 9). When

```
plan(GOALS, PLAN0, PLAN2, WORLD0, WORLDN) :-
  % check that the goals are proveable,
  % possibly with the aid of hypothetical
  % actions, and collect all the actions
  % that were required
  prove(GOALS),
  setof(H, hypothetical(action(H)), ACTIONS),
  % pick one of the hypothesised actions and
  % retrieve its full description
  member(A, ACTIONS),
  action(A, pre(PRE), add(ADD), delete(DELETE)),
  plan(PRE, [], SUBPLAN, WORLD0, WORLD1),
  append(ADD, WORLD1, WORLD2),
  deleteAll(DELETE, WORLD2, WORLD3),
  append(PLAN0, SUBPLAN, PLAN1),
  plan(GOALS, [A|PLAN1], PLAN2, WORLD3, ).
```

**Fig. 9:** *State-space planner with integrated inference engine*

the proof is complete, the (names of the) hypothesised actions are gathered up. These actions are what is required for the goals to be satisfied, so the planner switches its attention to the preconditions of one of these as usual, and the rest of the algorithm is unchanged.

The basic notion that the effects of actions should be treated as the consequents of hypothetical rules is the key to integrating the planner and the theorem prover, and in particular to indexing actions so that they are retrieved exactly when they are needed for a proof.

Consider Fig. 10, a slightly more complex example.

```
forall(B,
  forall(D,
    different(D, C)
    -> action(steal(D, C, E),
             pre(has(C, E)),
             effects(has(D, E) & ~(has(C, E))))))
& forall(B,
  forall(C,
    forall(D,
      bel(B,
        action(sell(C, D),
              pre(valuable(D) & has(C, D)),
              effects(rich(C) & ~(has(C, D))))))
    & forall(X,
      forall(Y,
        (bel(X, (watch(Y) & Rolex(Y)) -> valuable(Y)))
        & bel(john, exists(X, money(X) & has(bill, X)))
        & bel(martin, exists(X, money(X) & has(bill, X))))))
```

**Fig. 10:** *If you want something, you can steal it; selling valuable things will make you rich*

Fig. 10 says everyone believes if you steal something from someone, you will have it and he won’t, and that if you sell something valuable you will be rich. It then tells us about John and Martin’s beliefs about some sums of money. Given all the machinery above, we can prove that (i) *bel(john, exists(X, money(X) & has(john, X)))* would be true if John believed he stole some money from Bill, and (ii) *bel(martin, exists(X, money(X) & not(has(john, X)))* would be true if Martin believed John stole some money from Bill, then Martin stole it from John.<sup>10</sup>

Note that all the information in Fig. 10 is part of someone’s belief set. The rules describing what stealing, selling and Rolex watches are like are marked as common knowledge. John and Martin’s private beliefs are also explicitly marked.<sup>11</sup> It is the participants’ beliefs about the situation that matter—properties of the situation that the participants are unaware of cannot enter into their reasoning about it.

## 4 Precons and pure literal deletion

Suppose we want to be rich. We can use the rules in Fig. 10 to try to come up with a plan, but we will be blocked if we do not know of anyone who owns a Rolex watch. What if we were to add a dummy place-holder action that could be used to magically achieve any goal whatsoever (Fig. 11)?

```
forall(B,
  forall(D,
    action(dummyAction(B, D),
          pre(true),
          effects(know(B, D))))))
```

**Fig. 11:** *Place-holder action*

Given this action, our planner will come up with the plan in Fig. 12 as a way to become rich. The clear problem with this plan is that *dummyAction*, is not performable. The plan is, however, complete enough for us to talk about. We could say ‘*I have a plan I’d like to carry out, but I can’t do it, because I can’t carry out the bit where I find out who has a Rolex watch*’. This is exactly what we said the semantics

<sup>10</sup> Note that for (ii) Martin has to start by thinking of something John does have, in order to then steal it from him. We are not using the closed world assumption, so the fact that Martin thinks there is some money that he cannot prove belongs to John does not let us infer that there is any money that John does not have: for him to draw this conclusion without hypothesising any actions we would have to include  $\forall X (bel(X, \forall Y \forall Z \forall O (\neg(different(Y, Z) \& has(Y, O) \& has(Z, O)))))$ —that everyone believes that two different people cannot own the same thing.

<sup>11</sup> Clearly, in any practical situation, the participants will have more information than there is room to show in this paper. In particular, their beliefs about the (linguistic and extra-linguistic) situation in which they find themselves would have to be included.

*[dummyAction(self, watch(B)),  
dummyAction(self, Rolex(X)),  
dummyAction(self, has(D, B)),  
steal(self, D, B),  
sell(self, B)]*

**Fig. 12:** Plan with place-holder action

of questions is: when you utter a question, you tell H there is an action you want to perform, but that you lack vital information, *i.e.*, it makes sense in this situation to say (2).

(2) Who has a Rolex watch?

Given that S has said  $\exists A((\text{beliefs}[S] \ \& \ \text{bel}(S, P)) \rightarrow \text{poss}(S, \text{do}(A)))$  for some specific  $P$ , we ‘just’ have to be able to determine  $A$ . In principle, we must inspect every action we know of and check the following conditions: (i)  $\{\text{beliefs}[S], P\} \vdash \text{pre}(A)$  and (ii)  $\{\text{beliefs}[S]\} \not\vdash \text{pre}(A)$ .

We start by turning preconditions into rules that tell us about the circumstances under which an action is possible, as in Fig. 13. This means we can use the theorem

$$\forall B \forall C \forall D (\text{bel}(B, \text{has}(C, D)) \rightarrow \text{poss}(C, \text{do}(\text{action}(\text{steal}(B, C, D))))))$$

**Fig. 13:** Everyone believes that you can steal something from someone who has it

prover to look for appropriate actions simply by asking for proofs of  $\text{poss}(S, \text{do}(A))$  from  $\text{beliefs}[S] \ \& \ \text{bel}(S, P)$  and then checking that  $\text{poss}(S, \text{do}(A))$  does not follow from  $\text{beliefs}[S]$  alone. Finding such proofs may, however, require much effort, so inspecting every action in our vocabulary is impractical. We need to filter them to ensure only actions for which (i) and (ii) are likely to hold are considered.

We exploit a variation on [17]’s notion of ‘pure literals’. Kowalski noted that if you have a clause of the form  $(A_1 \ \& \ \dots \ \& \ A_n) \Rightarrow C$ , but no clauses with  $A_i$  as head for some  $i$ , this clause cannot contribute to any proofs and may as well be removed. A literal  $A$  that occurs only in the antecedent of a rule is called a ‘pure’ literal.<sup>12</sup> Purification is an iterative process: removing one clause because it has a pure literal may easily make literals in other clauses become pure, and so in many cases removable. We carry out purification when the rule set is first read in. The time complexity of purification is  $o((N \times L)^2)$  where  $N$  is the number of clauses and  $L$  is the maximum number of negative literals in a clause—costly, but not intractable. It can be carried out once for a given rule set, so that the cost is not incurred every time the rule set is used. The major difference is that when we spot a clause containing a pure literal, we archive it, rather than simply deleting it. Then when we add new local contextual facts to the rule set, we can *impurify* those rules that now become available again.

Purification is a sensible thing to do if you have a large body of background knowledge to be exploited in different contexts, since you can ensure you are always working with the relevant subset of the general knowledge. It is particularly valuable when we are looking for actions that are performable in a given context, since it means that ones whose preconditions are pure will not even be considered.

The final move is to merge steps (i) and (ii) above. The idea that we must find an action whose preconditions are

<sup>12</sup> Kowalski classed all literals that occurred only in antecedents (negative literals) or only in consequents (positive literals) as pure. We concentrate on negative pure literals here, dealing with positive ones by the techniques described in [21].

entailed by  $\text{beliefs}[S] \ \& \ \text{bel}(S, P)$  and then check they are *not* entailed by  $\text{beliefs}[S]$  is unappealing. We sidestep (ii) by keeping a record of the facts used in the proof of (i): if these include  $\text{bel}(S, P)$ , we assume that at the very least  $\text{bel}(S, P)$  is relevant, and we let this check replace (ii).

Suppose, for instance, S says (2) (see Fig. 14). We

*queryValue(ref( $\lambda A(\text{speaker}(A))$ ),  
ref( $\lambda B(\text{hearer}(B))$ ),  
 $\lambda C(\exists D::\{\text{watch}(D) \ \& \ \text{Rolex}(D)\}$   
 $\exists E::\{\text{aspect}(\text{now}, \text{simple}, E)\}$   
 $\theta(E, \text{agent}, C)$   
 $\ \& \ \theta(E, \text{object}, D)$   
 $\ \& \ \text{have}(E))$ )*

**Fig. 14:** ‘Who has a Rolex watch?’

start by assuming H believes S is telling the truth, and that H is feeling well-disposed towards S. Next we could simply look for an item that H believes satisfies  $\lambda C(\exists D::\{\text{watch}(D) \ \& \ \text{Rolex}(D)\} \exists E::\{\text{aspect}(\text{now}, \text{simple}, E)\} \theta(E, \text{agent}, C) \ \& \ \theta(E, \text{object}, D) \ \& \ \text{have}(E))$ . If we find such an item, H could just tell S about it.<sup>13</sup>

But perhaps there is nothing that H believes satisfies the given property, or H decides that simply providing S with a description of an individual is not helpful enough. Nevertheless, we may be able to find something that does make  $A$  performable, and provide that as an alternative to the question S asked. If we cannot, the next move is to see if anything would be possible if S performed  $A$ . We can do this by following the same strategy, looking for some  $A'$  such that  $\text{effects}(A) \rightarrow \text{poss}(S, \text{do}(A'))$ , then looking for something that would make  $A'$  performable.

Attempting to find actions that would become performable if certain information were true seems like a very open-ended task. The process of purification described above is the key to constraining it so that (a) it can be done in a manageable period of time and (b) we fairly often get unique solutions. The only rules we need to consider are ones for which the proofs that they are performable are impurified by the queried information.

## 5 Actions with no preconditions

Our planner differs from most others in the emphasis on allowing for preconditions that may require considerable amounts of inference to verify they are true in the current context, and effects that may have indirect consequences in the current context. However, we follow standard practice in assuming the preconditions *must* be true. This causes problems when we come to consider linguistic acts. The surface forms of English linguistic acts only distinguish between four types—statements, polar questions, WH-questions and imperatives. Although common parlance uses a wide variety of terms such as ‘*informing*’, ‘*reminding*’, ‘*nagging*’, ‘*bluffing*’ and ‘*lying*’, these are all names for the different consequences that the basic actions have when used in different contexts, rather than names for different actions. For H to realise S is informing her of

<sup>13</sup> H will have to find an appropriate way of describing it for S. Her internal name for it is likely to be some Skolem constant, but saying ‘*Yes, SK17 has one*’ is not very helpful. H will have to find a description that will enable S to identify this entity in his own view of the world: saying ‘*Yes, the person who has a Rolex watch has one*’ is not going to be helpful either. We will therefore have to constrain the information used by the generation algorithm to ensure it works for S [25].

a proposition P, not reminding her of it, she must do two things: (i) recognise S has produced a statement that encodes P (see §2); (ii) think about what S might gain by this.

Focusing now on (ii), suppose we encode the four basic actions in terms of preconditions and effects. Under what circumstances can you make a statement, and what effects is making a statement guaranteed to have?

Imagine a situation where I am giving a talk on semantics, when suddenly I say ‘*My father used to live at the bottom of the Atlantic Ocean*’. What will the consequences of saying this be? If I have just been discussing underspecified quantifier scope, they are likely to be that my audience will think I’ve gone mad. If, however, I’ve just been claiming that linguistic acts have no preconditions, it is possible that they will think I am illustrating my argument with a rhetorical example. The effects of a statement depend entirely context, and the same is true of all four utterance types.

Note that, although my utterance may seem bizarre in the context, there is nothing to stop me saying it—no preconditions that must hold before it is possible. But in the blocks world the robot *cannot* pick up a block if its hand is not empty. It is also impossible to steal a Rolex watch from me, because I don’t own one. But nothing makes it impossible for me to say anything at any time.

The most we can reliably say about linguistic actions, then, is shown in (Fig. 15). Fig. 15 says everyone knows

$$\forall B \text{bel}(B, \\ \text{action}(\text{say}(S, H, P), \\ \text{pre}(\text{aware}(S, P)), \\ \text{effects}(\text{minutes}([S, H], P))))$$

**Fig. 15:** *You can say anything you can think of at any time*

that a speaker S can produce an utterance that encodes a message P (which includes whether it was a statement, question, or imperative) for a hearer H whenever the idea occurs to him; and that the only reliable effect is that S and H will each put P in their copy of the minutes of the conversation. Neither party has to believe or disbelieve P before or after the action is performed. Notably, S can say P not believing it, and H need not believe P after S has said it.<sup>14</sup>

If neither party is committed to the truth of P after it has been put in the minutes, what *can* we say for sure? The most we can be confident of is that both are aware of it, *i.e.*, it is now available for them to inspect and think about.

What conclusions should H draw from the fact that both parties are now aware of an utterance UTT (remember: UTT is a proposition with a mood wrapper on it, so it is equivalent to a statement that certain information would make some action by either S or H possible)? It seems reasonable to suppose that S knows whether P is true, but in general H does not have direct access to this information.

Let us consider the case where UTT is *query* (‘*What’s the time?*’). If UTT is true, H knows that something could be achieved if S knew the time. H could now do several constructive things. She could try to find out what the time was, or to work out what S could do if he knew the time. H at least has a clue about what S wants to do and what information he needs to do it, so she can try to do something to help him. If, however, UTT is not true, H is stuck. All she knows is that knowing the time will not help S.

<sup>14</sup> Some forms of words have very closely associated ‘perlocutionary’ effects in certain situations, *e.g.*, ‘*I now pronounce you man and wife.*’ This, however, is a complex social action, rather than a linguistic act. We are not attempting to analyse such cases here.

Since both parties understand all this, there seems very little point in S putting UTT in the minutes unless he thinks H will believe he believes it, because if she doesn’t, there is very little she can do. So in a purely neutral context where neither party has any specific views on the reliability or cooperativeness of the other, it is nonetheless rational for S to produce utterances he believes and for H to believe this is what he is doing. Hence the default assumption that people are committed to what they say arises as a consequence of the assumption that linguistic actions are generally intended to help with underlying extra-linguistic plans. Our argument suggests this is more than a convention—that it is in fact the most sensible thing to do.

There are, of course, situations where people say things they are not committed to. Sometimes S believes H will know S is not committed to them; sometimes S hopes H will not spot it. These correspond to instances of ‘flouting’ and ‘violating’ Grice’s Maxim of Quality.

Suppose S says ‘*What’s the time?*’ in a situation where it is clear that both parties know the time, *e.g.*, S is pointing at a clock. H will add to her copy of the minutes that both parties are aware that S has claimed there is something he could do if he knew the time. Her first move will be to check whether  $\exists G(\neg \text{poss}(S, \text{do}(G))) \ \& \ (\text{knowRef}(S, \lambda X(\text{time}(X))) \rightarrow \text{poss}(S, \text{do}(G)))$  is consistent with everything she believes. There are only two ways for this to be inconsistent with her background knowledge. Either (i)  $\text{knowRef}(S, \lambda X(\text{time}(X)))$  is already true; or (ii)  $(\text{knowRef}(S, \lambda X(\text{time}(X))) \rightarrow \text{poss}(S, \text{do}(G)))$  is itself false.

Suppose H can prove  $\text{knowRef}(S, \lambda X(\text{time}(X)))$  is true.<sup>15</sup> S knows H will check the consistency of  $\neg \text{knowRef}(S, \lambda X(\text{time}(X)))$ , he knows she will spot that it is not consistent with everything else she believes, and he knows that she will make use of things they already mutually believe. What else can H assume, then, other than that S is drawing her attention to those things?

Suppose on the other hand that, as far as H can tell,  $\text{knowRef}(S, \lambda X(\text{time}(X)))$  is not true (*e.g.*, S is not pointing at a clock). She then has to consider whether  $\exists G(\text{knowRef}(S, \lambda X(\text{time}(X))) \rightarrow \text{poss}(S, \text{do}(G)))$  is consistent with what she believes. The only way for this to be inconsistent is if there is no G for which  $(\text{knowRef}(S, \lambda X(\text{time}(X))) \rightarrow \text{poss}(S, \text{do}(G)))$  is true. H will implicitly investigate this when she tries to recognise S’s plan, as outlined in §4, and hence there is no need for an explicit stage here in which she checks its consistency.

Similar arguments apply to cases where S makes a statement that is blatantly inconsistent with S and H’s mutual beliefs. If S and H already mutually believe P is false, being told that it is true will not lead H to look for a plan that depends on it. Again, the obvious outcome is that H’s attention is drawn to the evidence that contradicts P.

## 6 Conclusions

We have tried to show that the flexibility of language use can be explained by positing a small set of identifiable linguistic acts that can be used to achieve a range of goals

<sup>15</sup> Any such proof will draw upon specific beliefs that H ascribes to S. Any such beliefs will have a label saying something about their status: H might infer S believes something because she thinks he knows it, or because she thinks they mutually believe it, or just because she thinks he believes it, or ... H should inspect the provenance of the facts or rules she used in the proof of  $\text{knowRef}(S, \lambda X(\text{time}(X)))$ . Some will be general knowledge, but others will be things H explicitly ascribed to S.

in different contexts. This contrasts with having a much larger set of acts, each of which is used in a specific context to achieve a goal appropriate in that context. The contrast between these approaches roughly mirrors saying that the meanings of certain lexical items are underspecified versus that those items have multiple meanings. If a word is underspecified, the rules that link it to other concepts will be conditional, so that its significance in a context will emerge as a consequence of its interaction with other terms. If a word is ambiguous, you must choose between the various options before you can start reasoning with it.

The former approach has numerous advantages in lexical semantics—you do not have to make a choice between different ‘senses’ until you start using the utterance as part of some specific inference task, by which time you are likely to have the information you need for using it appropriately (and if you do not have it by then, you cannot make the choice anyway). Applying this notion to the analysis of speech acts brings two further benefits. (i) It enables us to trace the relationships between the various fine-grained illocutionary acts that can be manifested by each of the basic underlying acts. We are *not* arguing that people carry out all the inference steps that lead from S saying, for instance, ‘Do you know the time?’ in a context where it is clear that both parties do know the time to H apologising profusely for being late. Clearly, people can recognise patterns of inference they have carried out before, and it would make sense to equip our inference engine/planner with this ability. Nonetheless, demonstrating that it is at least possible to carry out this inference seems to us to be a worthwhile achievement. (ii) If you base your treatment of linguistic action on the notion that there is a fixed number of speech acts, each with its own specific preconditions and effects, you will be unable to cope with situations where a speaker uses some surface form in a way you have not anticipated. By putting much more emphasis on the participants’ ability to reason from first principles about how the underlying act interacts with other aspects of the epistemic context, we allow the possibility of responding to novel uses.

There are, of course, plenty of things left to do. We have not covered how discourse cues, in particular the use of different referential forms, can be used to keep track of the current focus of attention [4, 13, 15]. Our work exploits such a mechanism, which we use for guiding the search for referents for pronouns [28, 29] and for keeping track of the ‘question-under-discussion’ [12, 18]. We have also not discussed more complex plans. The examples in §5 indicate the kind of epistemic reasoning and planning that is involved in spotting whether a maxim has been flouted and reasoning about what S’s underlying goal might be. Elsewhere we have shown that our planner can solve blocks-world problems that cannot even be stated in standard STRIPS blocks-world actions [10, 24]. There are, however, interesting cases outside our scope, including phatic use of language, for example, which is extremely prevalent in everyday life.

There are other cases where the chain of inference is just too hard for our model to cope with, or where it requires more general knowledge than can easily be provided. However, such very complex cases will cause problems for any other implemented systems—they do not undermine the general thesis. Where surface locutionary acts produce complex illocutionary effects, there is always a chain of reasoning that connects the standard interpretation of the act and S and H’s current epistemic states to some goal of

one of the participants.

## 7 Acknowledgements

Part of this work was initially supported by an EPSRC grant, with recent developments partially funded under EU-grant FP6/IST No. 507019 (*PIPS: Personalised Information Platform for Health and Life Services*).

## References

- [1] J. F. Allen and C. R. Perrault. Analysing intention in utterances. *Artificial Intelligence*, 15:148–178, 1980.
- [2] J. Austin. *How to Do Things with Words*. 1962.
- [3] N. Belnap jr. A useful four-valued logic. In D. J.M. and E. G., editors, *Modern Uses of Multiple-Valued Logic*, pages 8–37, Dordrecht, 1977. D. Reidel.
- [4] S. E. Brennan, M. W. Friedman, and C. J. Pollard. A centering approach to pronouns. In *Proceedings of 25th meeting of the ACL*, 1987.
- [5] H. C. Bunt. Dialogue pragmatics and context specification. In H. C. Bunt and W. J. Black, editors, *Abduction, Beliefs and Context: Studies in Computational Pragmatics*, pages 81–151, Amsterdam/Philadelphia, 2000. John Benjamins.
- [6] H. C. Bunt and W. J. Black, editors. *Abduction, Beliefs and Context: Studies in Computational Pragmatics*. John Benjamins, Amsterdam/Philadelphia, 2000.
- [7] P. R. Cohen, J. Morgan, and M. E. Pollack. *Intentions in Communication*. Bradford Books, Cambridge, Mass., 1990.
- [8] P. R. Cohen and C. R. Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, 7(2):171–190, 1979.
- [9] M. Davis and H. Putnam. A computing procedure for quantification theory. *Journal of the Association for Computing Machinery*, 7(3):201–215, 1960.
- [10] D. G. Field and A. M. Ramsay. How to build towers of arbitrary heights. In *The 23rd Annual Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG 2004)*, University College Cork, 2004.
- [11] D. G. Field and A. M. Ramsay. Sarcasm, deception, and stating the obvious: Planning dialogue without speech acts. 22:149–171, 2004.
- [12] J. Ginzburg. Resolving questions I. *Linguistics and Philosophy*, 18:459–527, 1995.
- [13] B. J. Grosz, A. Joshi, and S. Weinstein. Centering: a framework for modeling the local coherence of discourse. 1995.
- [14] J. Hintikka. *Knowledge and Belief: an Introduction to the Two Notions*. Cornell University Press, New York, 1962.
- [15] A. Joshi and S. Weinstein. Formal systems for complexity and control of inference: A reprise and some hints. In *Centering Theory in Discourse*, pages 31–38.
- [16] K. Konolige. *A Deduction Model of Belief*. Pitman, London, 1986.
- [17] R. Kowalski. A proof procedure using connection graphs. *JACM*, 22(4):572–595, 1975.
- [18] S. Larsson. Questions under discussion and dialogue moves. In *Twendial’98*, 1998.
- [19] D. Lewis. Scorekeeping in a language game, 1979. *Journal of Philosophical Logic* 8: 339–59. Reprinted in D. Lewis *Philosophical papers Volume I*, 1983, pp. 233–249. New York and Oxford: Oxford University Press.
- [20] R. Manthey and F. Bry. Satchmo: a theorem prover in Prolog. In R. Lusk and R. Overbeek, editors, *Proceedings of the 9th International Conference on Automated Deduction (CADE-9)*, volume 310 of *Lecture Notes in Artificial Intelligence*, pages 415–434, Berlin, 1988. Springer-Verlag.
- [21] A. M. Ramsay. Generating relevant models. *Journal of Automated Reasoning*, 7:359–368, 1991.
- [22] A. M. Ramsay. Speech act theory and epistemic planning. In W. J. Black and H. C. Bunt, editors, *Abduction, Beliefs and Context in Dialogue: Studies in Computational Pragmatics*, pages 293–310, Amsterdam, 2000. John Benjamins.
- [23] A. M. Ramsay. Theorem proving for untyped constructive  $\lambda$ -calculus: implementation and application. *Logic Journal of the Interest Group in Pure and Applied Logics*, 9(1):89–106, 2001.
- [24] A. M. Ramsay and D. G. Field. Planning ramifications: when ramifications are the norm, not the problem. In *11th International Workshop on Non-monotonic Reasoning (NMR-06)*, pages 344–351, Ambleside, 2006. AAAI.
- [25] A. M. Ramsay and H. L. Gaylard. Relevant answers to WH-questions. *Journal of Language, Logic and Information*, 13(2):173–186, 2004.
- [26] E. D. Sacerdoti. Planning in a hierarchy of abstraction spaces. 1974.
- [27] J. R. Searle. *Speech Acts: an Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, 1969.
- [28] H. L. Seville. Experiments with discourse structure. In H. C. Bunt and E. G. C. Thijssse, editors, *3rd International Workshop on Computational Semantics*, pages 233–247, University of Tilburg, 1999.
- [29] H. L. Seville and A. M. Ramsay. Reference-based discourse structure for reference resolution. In *ACL Workshop on Discourse/Dialogue Structure and Reference*, pages 90–99, University of Maryland, June 1999.
- [30] G. J. Sussman. *A Computational Model of Skill Acquisition*. American Elsevier, New York, 1975.
- [31] A. Tate. Generating project networks. In *International Joint Conference on Artificial Intelligence*, volume 5, pages 888–893, 1977.

# Integrating Cross-Language Hierarchies by Text Classification

Fumiyo Fukumoto  
Interdisciplinary Graduate  
School of Medicine and Engineering  
Univ. of Yamanashi  
fukumoto@yamanashi.ac.jp

Yoshimi Suzuki  
Interdisciplinary Graduate  
School of Medicine and Engineering  
Univ. of Yamanashi  
ysuzuki@yamanashi.ac.jp

## Abstract

This paper presents a method for integrating cross-language(CL) category hierarchies by estimating category similarities. The method does not simply merge two different hierarchies into one large hierarchy, but instead extracts sets of similar categories, where each element of the sets is relevant to each other. It consists of three steps. First, we classify documents from one hierarchy into categories with another hierarchy using cross-language text classification(CLTC) technique, and extract category pairs of two hierarchies. Next, we apply  $\chi^2$  statistics to these pairs in order to obtain similar category pairs, and finally we apply the generating function of Apriori to the result of category pairs, and find sets of similar categories. The results show the effectiveness of the method.

## Keywords

Information integration, Integrating category hierarchies, Text classification

## 1 Introduction

With the exponential growth of information on the Internet, finding and organizing relevant materials on the Internet is becoming increasingly difficult. Internet directories such as Yahoo! and Google, which classify Web pages into pre-defined hierarchical categories, provide one solution to the problem. Categories in the hierarchical structures are carefully defined by human experts and documents are well-organized. However, a single hierarchy on some Internet is often insufficient in finding relevant documents for users. Because each hierarchy tends to have some bias in both defining hierarchical structure and classifying documents, e.g. coarse-grained hierarchies, while others represents a fine-grained classification. Moreover, existing Web search engines only support the retrieval of documents which are written in the same language as the query, while more and more languages are becoming to be used for Web documents, and it is now much easier to access documents written in foreign languages.

In this paper, we propose a method for integrating cross-language category hierarchies, Reuters'96 hierarchy and UDC code hierarchy of Mainichi Japanese newspaper documents by estimating category similarities. The method does not simply merge two different hierarchies into a large hierarchy, but instead extracts

sets of categories, where each category within the set is relevant to each other<sup>1</sup>. The method consists of three steps. First, we classify documents from one hierarchy into categories with another hierarchy using CLTC technique, and extracts category pairs of two hierarchies. Next, we apply  $\chi^2$  statistics to these pairs in order to estimate similar category pairs. Finally, we apply the generating function of Apriori[8] to the extracted pairs, and generate sets of similar categories.

## 2 Related Work

Integrating information on the Internet is crucial to provide intelligent Web services. One type of the information integration is ontology merging[6]. Ontologies have been established for knowledge sharing and are widely used as a means for conceptually structuring domains of interest. Fridman et al. proposed a method to combine two ontologies, which are represented in a hierarchical categorization[7]. Their method is based on the similarity between words with dictionaries. Stumme presented a method which uses the attributes of concepts to merge different ontologies[3]. It creates a new concept without regarding the original concepts in both ontologies. However, these methods require human interaction for merging process. Moreover, the evaluation of the result of ontology merging is an open issue in their methods.

Merging category hierarchies is another type of the information integration, and several efforts have been made to semi or full automatic integration of different hierarchies. Much of the previous work applies machine learning(ML) techniques[13] to classify each document into more than one categories. Naive Bayes(NB) is one of the ML techniques used for this type of document classification framework[1]. However, the performance is not more effective than SVMs and  $k$ NN[16], since NB selects poor weights for the decision boundary when one class has more training samples than another. Ichise et al. used Enhanced Naive Bayes(E-NB)[9] to integrate multiple Internet directories. They used the  $\kappa$ -statistics to find similar category pairs, and transferred the document categorization from a category in the source Internet directory to a similar category in the target Internet directory. They did not rely on words or word similar-

<sup>1</sup> The reason for extracting sets of categories is that each categorical hierarchy is defined by individual human experts, and different linguists often identify different number of categories in the same concepts. Therefore, it is impossible to handle *full* integration of hierarchies.

ity in a document, but instead relied on the category structure. They showed that the performance of their method was more than 10% better in accuracy than that of Agrawal and Doan’s methods[9], [1], [10]. However, their method is based on the existence of a large number of shared links. They reported that the performance was not better if there were fewer shared links. Moreover, the method finds only one-to-one mapping categories, while there exists one-to-many or many-to-many mappings in real-world Web directories.

Our work differs from earlier work in a couple of respects. First, we focused on the hierarchies from different languages, which has not previously been explored in the context of integrating hierarchies. Second, we use lexical information of documents through CLTC, which we believe is important to take into account the contents of documents, and hopefully more accurately than the method not using them. Third, we use a learning model, SVMs, that has widely used in TC, but it has not also been explored in the context of integrating hierarchies.

### 3 Integrating Hierarchies

The procedure consists of three steps: CLTC, estimating category correspondences and generating sets of similar categories.

#### 3.1 Cross-language text classification

The corpora we used is the Reuters’96 and the RWCP of Mainichi Japanese newspapers. We used Japanese-English and English-Japanese Machine Translation(MT) software for CLTC<sup>2</sup>. In the CLTC task, the system is trained using labeled documents in one language (e.g. English), and classifies labeled documents in another language (e.g. Japanese), and vice versa. We use a learning model, Support Vector Machines(SVMs)[14] to classify documents, as SVMs have been shown to be effective for classification[16, 12]. SVMs are basically introduced for solving binary classification, while TC is a multi-class, multi-label classification problem. Several methods which were intended for multi-class, multi-label data have been proposed[5]. We use *One-against-the-Rest* version of the SVMs model at each level of a hierarchy. We classify test documents using a hierarchy. We employ the hierarchy by learning separate classifiers at each internal node of the hierarchy. Similar to Dumais’ approach, we used a Boolean function  $P(L_1) \& \dots \& P(L_m)$ , where P is a decision threshold, and  $m$  is the number of hierarchical levels[12]. The process is repeated by greedily selecting sub-branches until it reaches a leaf.

We classified translated Mainichi Japanese document  $d^{M-MT}$  with UDC code category  $M$  into Reuters categories using SVMs classifiers. In a similar way, each translated Reuters document  $d^{R-MT}$  with category  $R$  is classified into Mainichi UDC categories. Fig. 1 illustrates Reuters and Mainichi documents classification. In Fig. 1, documents with UDC code cate-

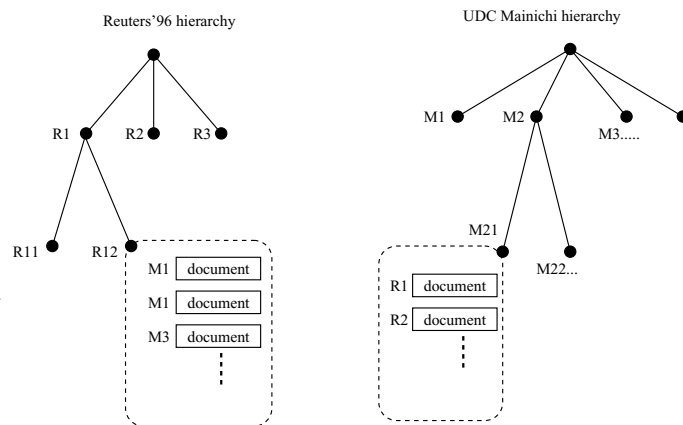


Fig. 1: Cross-language text classification

Table 1:  $M$  and  $R$  matrix

	$R$	$\neg R$
$M$	$\text{freq}(M, R) = a$	$\text{freq}(M, \neg R) = b$
$\neg M$	$\text{freq}(\neg M, R) = c$	$\text{freq}(\neg M, \neg R) = d$

gories ‘M1’ or ‘M3’ are classified into Reuters category ‘R12’, and documents with Reuters categories, ‘R1’ or ‘R2’ are classified into Mainichi UDC category, ‘M12’. As a result, we obtain Reuters and Mainichi category pairs from the documents which are assigned to the categories in each hierarchy.

#### 3.2 Estimating category correspondences

The second step to integrate hierarchies is to estimate Reuters and UDC category correspondences. We applied  $\chi^2$  statistics to the result of CLTC. Let us take a look at the Reuters’96 hierarchy. Suppose that the document  $d^{M-MT}$  with Mainichi UDC category  $M$  is assigned to Reuters category  $R$ . We can extract Reuters and Mainichi UDC category pairs. Then, based on the contingency table of co-occurrence frequencies of  $M$  and  $R$  which is shown in Table 1, we estimate category correspondences according to the  $\chi^2$  shown in Eq. (1). Here, co-occurrence frequencies of  $M$  and  $R$  is equal to the number of category  $M$  documents which are assigned to  $R$ . Similar to the Reuters hierarchy, we can also estimate category correspondences from Mainichi UDC hierarchy, and extract a pair  $(M, R)$  according to the  $\chi^2$  values. We note that the similarity obtained by each hierarchy does not have a fixed range. We thus apply the normalization strategy shown in Eq. (2) to the results obtained by each hierarchy to bring the similarity value into the range  $[0,1]$ .

$$\chi^2(M, R) = \frac{(ad - bc)^2}{(a + b)(a + c)(b + d)(c + d)} \quad (1)$$

$$\chi_{new}^2(M, R) = \frac{\chi_{old}^2(M, R) - \chi_{min}^2(M, R)}{\chi_{max}^2(M, R) - \chi_{min}^2(M, R)} \quad (2)$$

<sup>2</sup> We chose Japanese-English and English-Japanese MT Software Internet Honyaku-no-Ousama for Linux, Ver.5, IBM Corp.

```

0  Apriori(Set(m,r),min-sup){
1    C2 := Set(m,r) Set(mp,mc) Set(rp,rc);
2    k := 2;
3    while(Ck = 0){
4      Lk := {c | Ck | c.χ2 min-sup};
5      Ck+1 := Apriori-Gen(Lk);
6      forall c ∈ Ck+1 {
7        c.χ2 := Cal-χ2new(c.χ2);
8      }
9      k := k+1;
10   }
11   return k Lk ;
12 }

```

Fig. 2: Generating sets of similar categories

Let  $Set_r$  be a set of pairs obtained by Reuters hierarchy, and  $Set_m$  be a set of pairs by Mainichi hierarchy. We construct the set of  $M$  and  $R$  category pairs,  $Set_{(m,r)} = \{(M,R) \mid (M,R) \in Set_r \wedge Set_m\}$ , where each pair is sorted in descending order of  $\chi^2$  value. For each pair of  $Set_{(m,r)}$ , if the value of  $\chi^2$  is higher than a lower bound  $L_{\chi^2}$ , two categories,  $M$  and  $R$  are regarded as similar<sup>3</sup>.

### 3.3 Generating sets of similar categories

The last step to integrate different hierarchies is to generate sets of similar categories. We used generating function of Apriori algorithm developed by Agrawal[8]. Fig. 2 gives an algorithm for generating sets of similar categories.

The algorithm was presented as a heuristic for determining all frequent sets only, i.e. all sets with supports above a user-defined threshold, *minimum support*(*min-sup* in Fig. 2). We used normalized  $\chi^2$  values which are shown in Eq. (2) as minimum support. We note that  $C_2$  in line 1 consists of not only the extracted category correspondences  $Set_{(m,r)}$ , but also a set of Mainichi UDC code category  $m_p$  and its child category  $m_c$ ,  $Set_{(m_p,m_c)}$ , and that of Reuters,  $Set_{(r_p,r_c)}$ . This is because we utilize hierarchical structure to estimate sets of similar categories. The Apriori-Gen function in line 5 of Fig. 2 takes as argument  $L_k$ , where each element consists of the number of  $k$  categories, and returns a superset such that each element of the superset consists of  $k + 1$  categories. The number of the element of  $L_k$  does not have a fixed range. We thus use the normalization strategy which is shown in Eq. (2). Lines 6 – 8 of Fig. 2 shows that Eq. (2) is applied to each element of the set  $C_{k+1}$ .

The Apriori-Gen function is shown in Fig. 3. In line 2 – 13 of Fig. 3,  $p$  and  $q$  such that each differs the last  $k$ -th elements are concatenated and make a new set  $c$  with  $k+1$  categories. In line 7 – 11,  $\chi^2$  value of the new set  $c$  is calculated, and  $c$  is stored in the set  $C_{k+1}$  in line 12. Line 14 – 19 shows prune phase which deletes all candidates  $c \in C_{k+1}$  such that some

<sup>3</sup> We set  $\chi^2$  value of each element of  $Set_{(m,r)}$  to a higher value of either  $(M,R) \in Set_r$  or  $(M,R) \in Set_m$ .

```

0  Apriori-Gen(Lk){
1    Ck+1 = 0 ;
2    foreach p, q ∈ Lk such that
3      p.item1 = q.item1 ...
4      p.itemk-1 = q.itemk-1
5      p.itemk < q.itemk {
6      c := p ∪ q.itemk ;
7      if (p.χ2 > q.χ2) {
8        c.χ2 := p.χ2
9      } else {
10       c.χ2 := q.χ2
11     }
12     Ck+1 := Ck+1 ∪ {c};
13   }
14   foreach c ∈ Ck+1 {
15     foreach k-subsets s of c {
16       if (s ∩ Lk)
17         remove c from Ck+1;
18     }
19   }
20   return Ck+1 ;
21 }

```

Fig. 3: Apriori-Gen procedure

$k$ -subset of  $c$  is not in  $L_k$ .

## 4 Experiments

### 4.1 Data

We used Reuters'96 and UDC code hierarchies. The Reuters'96 corpus from 20th Aug. 1996 to 19th Aug. 1997 consists of 806,791 documents. These documents are organized into 126 categories with a four level hierarchy. The RWCP corpus labeled with UDC codes selected from 1994 Mainichi newspapers consists of 27,755 documents[11]. These documents are organized into 9,951 categories with a seven level hierarchy. We divided both Reuters'96(from 20th Aug. 1996 to 19th May 1997) and RWCP corpus into two equal sets: a training set to train SVMs classifiers, and a test set for TC in order to generate sets of similar categories. We divide a test set into two folds. The first fold is used to estimate thresholds, i.e. a decision threshold  $P$  which is used in CLTC, lower bound  $L_{\chi^2}$ , and minimum support. The second one is to generate sets of similar categories using these thresholds. We chose  $P = 0$  for each level of a hierarchy. The lower bound  $L_{\chi^2}$  and minimum support is .005 and 1%, respectively. These threshold values are determined as follows: we divided the first set of test data into three, and choose  $P$  and  $L_{\chi^2}$  values that maximized the average F-score among them, and minimum support that maximize the number of correct sets of similar categories among the topmost 2,000 sets<sup>4</sup>. We selected 109 categories from Reuters and 4,739 categories from Mainichi UDC

<sup>4</sup> It is preferable to choose an optimal minimum support using F-score as well as  $L_{\chi^2}$  value. However, it is difficult to make a correct data manually because of a large amount of categories (109 Reuters and 4,739 Mainichi UDC categories) and the number of sets with more than two categories.

**Table 2:** Performance of category correspondences ( $L_{\chi^2} = .005$ )

	Hierarchy			Flat		
	Prec	Rec	F	Prec	Rec	F
Mai & Reu	.503	.463	<b>.482</b>	.462	.389	<b>.422</b>
Reu	.342	.329	.335	.240	.296	.265
Mai	.157	.293	.204	.149	.277	.194

which have at least one document in each set. All Japanese documents were tagged by the morphological analysis Chasen[15]. English documents were tagged by a part-of-speech tagger[4]. We used noun words for both English and Japanese documents, and represented each document as a vector of noun words with frequency weights in the experiments.

## 4.2 Integrating Hierarchies

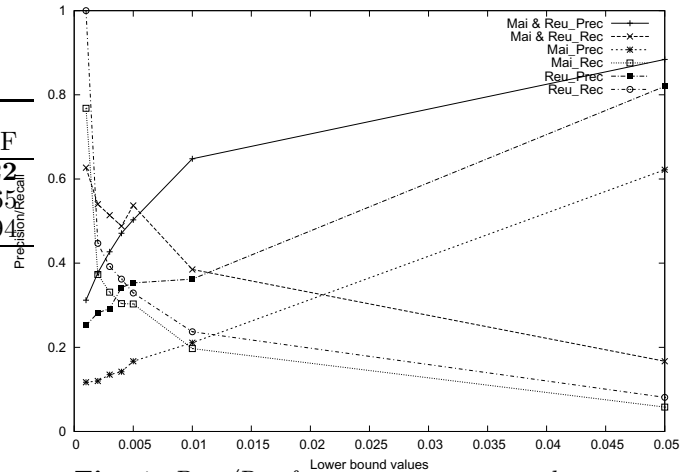
Table 2 shows F-score of category correspondences with  $L_{\chi^2} = .005$ . Here, F-score(F) is a measure that balances precision(Prec) and recall(Rec), i.e.  $F = \frac{2 * Prec * Rec}{(Prec + Rec)}$ . Let  $Cor$  be a set of correct category pairs within  $\pm 3$  days. The precise definitions of the precision and recall of the task are given below:

$$Prec = \frac{|\{(M, R) \mid (M, R) \in Cor, \chi^2(M, R) \leq L_{\chi^2}\}|}{|\{(M, R) \mid \chi^2(M, R) \leq L_{\chi^2}\}|}$$

$$Rec = \frac{|\{(M, R) \mid (M, R) \in Cor, \chi^2(M, R) \leq L_{\chi^2}\}|}{|\{(M, R) \mid (M, R) \in Cor\}|}$$

We compared the results by our method, i.e. hierarchical approach('Hierarchy' in Table 2) to the results by flat non-hierarchical approach('Flat' in Table 2). In the hierarchical approach, SVMs models were learned to distinguish each category from only those categories within the same level category, and that for the flat non-hierarchical approach, models were learned to distinguish each category from all other categories. Moreover, in the hierarchical approach, we applied Boolean function to each test document. 'Mai & Reu' in Table 2 shows the result by our method. 'Mai' and 'Reu' shows the result using only one hierarchy, UDC code, and Reuters, respectively. As can be seen from the results, integrating different hierarchies is more effective than only one hierarchy. Moreover, we found advantages in the F-score for the hierarchical approach, compared with a baseline flat non-hierarchical approach, as the former was .482 and the latter was .422. We thus report the result by a hierarchical approach in the following experiments.

We tested three methods with various  $L_{\chi^2}$  values yielding prec/rec curves for each method. Fig. 4 illustrates prec/rec by three methods against the changes of  $L_{\chi^2}$  value. The result obtained by integrating hierarchies shows better balance of recall and precision, and indicates that  $L_{\chi^2}$  values above .005 have precision of around 50% or more and that under .005 have recall of around 50% or more. Comparing the result 'Reu' with that of 'Mai', the result of 'Mai' was worse



**Fig. 4:** Prec/Rec for category correspondences

than that of 'Reu'. One reason is that the accuracy of TC. The micro-average F-score of TC for Reuters hierarchy was .815, while that of Mainichi hierarchy was .673, since Mainichi hierarchy consists of many categories, and the number of training data for each category is smaller than that of Reuters. McCallum et al. used a technique called 'shrinkage' which is especially useful for categories with small numbers of training documents[2]. This is a rich space for further exploration.

The rates of containing correct category pairs by three methods with  $L_{\chi^2} = .005$  are shown in Fig. 5. Fig. 5 shows that the result supports the usefulness of  $\chi^2$  statistics in each method, especially, the result by our method shows that 184 category pairs which are judged as correct are contained in the topmost 200 category pairs according to the  $\chi^2$  statistics. On the contrary, the result by our method was lower than the results by other two methods when the order of category pairs are larger than 2,001, since the total number of category pairs obtained by our method was 2,137 in all, while those obtained by using only Reuters hierarchy and RWCP hierarchy was large, i.e. 28,269 and 15,268, respectively, and some correct category pairs are contained among them. We used the result of category pairs with  $L_{\chi^2} = .005$  to generate sets of similar categories. Fig. 6 plots the rate of containing correct sets with more than two categories for the order of the sets sorted by normalized  $\chi^2$  value ( $min-sup=1\%$ ). The overall results for each method was better than those of category pairs, especially, our method shows that 193 out of the topmost 200 sets are judged to be correct. We recall that we used Apriori for generating set of similar categories. To examine how the ratio of minimum support of the Apriori affects the overall performance of our method, we used different minimum supports. Table 3 shows the total number of the sets with more than two categories for each ratio of minimum support, and Fig. 7 shows the result.

Fig. 7 shows the impact by our method that varying minimum support ratio has on the effectiveness for extracting sets with more than two categories. There is no difference among minimum support ratios, since the ratio of containing correct category sets are over 96% within the top 200, and 65% within the top 2,000 ac-



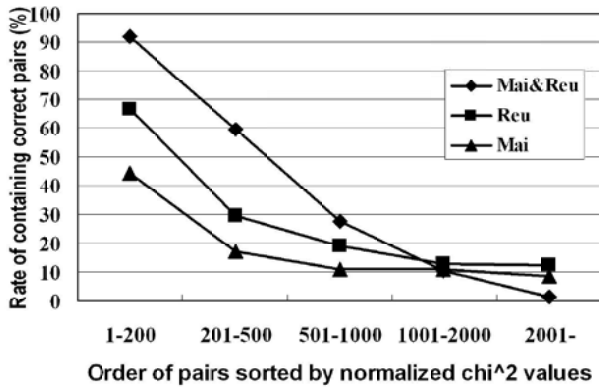


Fig. 5: Rates of containing correct pairs ( $L_{\chi^2} = .005$ )

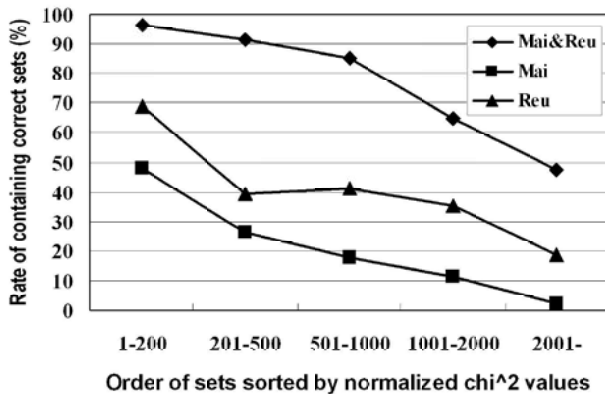


Fig. 6: Rates of containing correct sets of categories ( $min-sup = 1\%$ )

ording to the  $\chi^2$  values. On the other hand, it is clear from Fig. 7 that minimum support ratio is extremely sensitive to the ratio of correct sets of categories when the order of sets are larger than 2,001 sets.

## 5 Conclusion

We addressed the issue of a single hierarchy, and proposed a method for integrating hierarchies by estimating category similarities. The results were very encouraging, and the ratio of containing correct category sets are over 96% within the top 200, and 65% within the top 2,000 according to the  $\chi^2$  values. There are a number of interesting directions for future work. The results by our method depends on the performance of CLTC. We used a MT software for CLTC. Another option to do this is to utilize bilingual lexicon extracted from corpora, together with the bilingual

Table 3: Total # of sets against  $min-sup$

$min-sup(\%)$	Sets	$min-sup(\%)$	Sets
50	0	0.4	11,127
10	26	0.3	14,998
5	196	0.2	21,186
1	3,297	0.1	34,845
0.5	8,595		

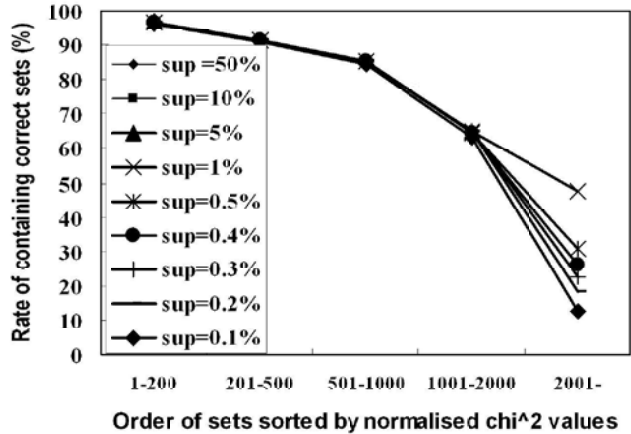


Fig. 7: Containing correct sets against  $min-sup$

dictionary. This is definitely worth trying with our method. The method should be expanded so that it can apply to more than three hierarchies. This is a rich space for further consideration. Moreover, applying the results to extract cross-lingually relevant documents, and comparing our method with the other existing methods are also included for future work.

## References

- [1] A.Doan, J.Madhavan, P.Domingos, and A.Halevy. Learning to map between ontologies on the semantic web. In *Proc. of the 11th International WWW Conference*, 2002.
- [2] A.McCallum, R.Rosenfeld, T.Mitchell, and A.Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proc. of the 15th ICML*, pages 359–367, 1998.
- [3] G.Stumme and A.Madche. Ontology merging for federated ontologies on the semantic web. In *Proceedings of the International Workshop on Foundations of Models for Information Integration (FMII'01)*, pages 413–418, 2001.
- [4] H.Schmid. Improvements in part-of-speech tagging with an application to German. In *Proc. of the EACL SIGDAT Workshop*, 1995.
- [5] J.Weston and C.Watkins. Multi-class support vector machines. In *Technical Report CSD-TR-98-04*, 1998.
- [6] N.Choi, I.Song, and H.Han. A survey on ontology mapping. In *Proceedings of SIGMOD, Record*, 35(3), pages 34–41, 2006.
- [7] N.F.Noy and M.A.Musen. Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI'00)*, pages 450–455, 2000.
- [8] R.Agrawal and R.Srikant. Fast algorithms for mining association rules. In *Proc. of Very Large Databases Conference*, pages 478–499, 1994.
- [9] R.Agrawal and R.Srikant. On integrating catalogs. In *Proc. of the 10th International WWW*, pages 603–612, 2001.
- [10] R.Ichise, H.Takeda, and S.Honiden. Integrating multiple internet directories by instance-based learning. In *Proc. of the 18th IJCAI*, pages 22–28, 2003.
- [11] RWCP. Rwc text database. 1998.
- [12] S.Dumais and H.Chen. Hierarchical classification of web content. In *Proc. of the 23rd SIGIR*, pages 256–263, 2000.
- [13] T.M.Mitchell. Machine learning. In *McGraw-Hill*, 1997.
- [14] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.
- [15] Y.Matsumoto. Japanese morphological analysis system Chasen manual. In *NAIST Technical Report*, 1997.
- [16] Y.Yang and Y.Lui. A re-examination of text categorization methods. In *Proc. of the 22nd SIGIR*, pages 42–49, 1999.

# Semantic Interpretation Supplementarily Using Syntactic Analysis

Kotaro Fuanakoshi Mikio Nakano Yuji Hasegawa Hiroshi Tsujino  
Honda Research Institute Japan Co., Ltd.  
8-1 Honcho, Wako-shi, Saitama, 351-0188, Japan  
{*funakoshi,nakano,yuji.hasegawa,tsujino*}@jp.honda-ri.com

## Abstract

This paper proposes a semantic interpretation method for domain-dependent language understanding systems. General-purpose parsers are used for syntactic analysis but tuning of them to domains is not necessary. Giving importance to domain knowledge and supplementarily using parse results deliver robust systems and rapid development of them with little linguistic expertise. An experiment using a dialogue system confirmed the effectiveness of the method.

## Keywords

Semantic Analysis, Syntactic Analysis, Rapid Prototyping

## 1 Introduction

Application developers who are not experts of speech and natural language processing expect frameworks with which they can easily build natural language understanding (NLU) systems that are robust against variations of expressions and noises or errors but still understand complex requests.

A key function of NLU is generating semantic representations from input expressions, often referred to as semantic interpretation. For semantic interpretation, there is an approach in which semantic representations are obtained by filling predefined templates with information extracted with keyword sets [4] or expression patterns [7]. This approach suites rapid development but is weak to handle complex requests. We will refer to this approach as template-based NLU (TLU).

There is another approach, which uses syntactic analysis. Syntactic structures, so called syntax trees or parse trees, are derived from input as the results of syntactic analysis using grammar rules or statistical data. Semantic representations are obtained by applying recursive compositional procedures to syntax trees. We will refer to this approach as syntactic-analysis-based NLU (SLU). Because SLU tackles the recursiveness of syntax and the compositionality of semantics, it can handle more complex expressions than TLU. However, building and maintaining SLU systems is not easy and those systems are generally less robust.

For rapid development, the semantic grammar approach (e.g., [9]) uses domain-independent parsers and grammar rules customizable according to domains. However this approach has a difficulty that developers must have in-depth knowledge of both the grammar and the domain [11]. Thus this approach is not adequate to non-experts. Approaches exploiting resources that require deep linguistic expertise such as [8] also suffer from the same problem.

To achieve robustness, several systems such as [2] perform TLU after SLU. However this approach enhances the

robustness of systems with respect only to simple requests that can be understood by keyword extraction.

In this paper, a novel approach for semantic interpretation will be presented, which does not cascade SLU and TLU but combines them. By giving importance to domain knowledge and supplementarily using syntax analysis, we pursue robust and powerful NLU systems and rapid development of them.

For syntactic analysis, ready-made general-purpose wide-coverage parsers are used. However cumbersome tuning of them to domains is not necessary. Therefore, system developments are accelerated and developers can quickly replace a parser with state-of-the-art ones.

Neither sentence patterns nor grammar rules are required. Hence, system maintainers do not have to bother about a lot of similar but mutually slightly different sentence patterns or a bunch of grammar rules whose scope of effects and side-effects are not intuitively recognizable even for experts. Unlike [9, 6, 1], our method does not require handcrafted mapping rules generating semantic representations from syntax trees, either.

The rest of this paper is organized as follows. Firstly, Section 2 describes the knowledge that our method requires. Secondly, Section 3 defines the semantic representation used as the output of our method. Then, the method is explained in Section 4 and evaluated in Section 5.

## 2 Domain knowledge

**Concept hierarchy:** The primary domain knowledge is an ontology that defines the domain. Fig. 1 shows a concept hierarchy for a hotel reservation system used in Section 5.

**Lexicon:** A lexicon consists of phrases with which concepts are expressed. The parenthetic phrases in Fig. 1 are the elements of a lexicon. These phrases are referred to as **concept expressions**.

**Onomasticon:** An onomasticon is provided by defining instances of concepts with unique symbols (used in task processing modules in back of NL interfaces) and describing proper names for each instance. Those unique symbols are referred to as **instance symbols**.

**Semantic frames:** Semantic frames define relations among concepts and are bone structures of the semantic representation explained in Section 3. Each concept has at least one semantic frame. Frame definitions can be inherited by sub-concepts. One can define multiple semantic frames for a concept. Fig. 2 shows two frames for concept *reserve* defined in the concept hierarchy in Fig. 1 and the slot definitions of the frames.

A slot definition specifies its name, the type of its value, verbal expressions to identify the slot (slot specifiers),

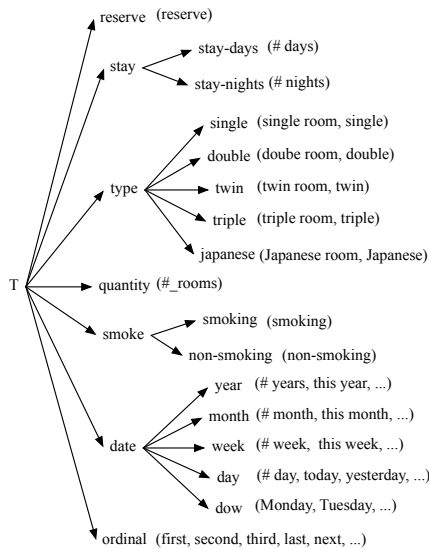


Fig. 1: A concept hierarchy

**Frame 1:**  $reserveF1(start, stay, type, quant, smoke)$   
 Example: “reserve one non-smoking single room from 23rd for 3 nights”

**Frame 2:**  $reserveF2(start, end, type, quant, smoke)$   
 Example: “reserve one non-smoking single room from 23rd to 26th”

Slot Name	Target Concept	Slot Specifier	Marker
<i>start</i>	date	“check-in”	“from”
<i>end</i>	date	“check-out”	“to, until”
<i>stay</i>	stay	“length of stay”	—
<i>type</i>	type	“room type”	—
<i>quant</i>	quantity	“number of rooms”	—
<i>smoke</i>	smoke	—	—

Fig. 2: Frames of concept *reserve* and slots

markers to identify the slot. Slot specifiers enable speakers to specify roles of information pieces with such an expression “25th check-in.” Markers have the same function with slot specifiers, but are used more grammatically. Both markers and slot specifiers are not requisites but will help accurate interpretation if they are included.

### 3 Semantic representation

We define **semantic trees** as the output of our proposed method explained in Section 4. A semantic tree represents a nesting structure of semantic frames as a tree structure. There are two types of nodes in semantic trees.

**Content node:** A content node retains a reference to a sub-expression that corresponds to the node in an input sentence. If the node indicates a named instance, it retains its instance symbol. Otherwise, the node retains a semantic frame.

**Value-group node:** A value-group node represents a parallel structure. A value-group node is an instance of one of two types: “enumerative” and “alternative”.

Fig. 3 shows a semantic tree corresponding to example (1). A *vg\_e* represents a value-group node of enumerative. Each content node is marked with its corresponding

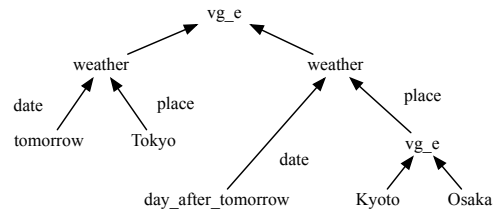


Fig. 3: A semantic tree

language expression. Here, the semantic frame of concept *weather* has two slots: *date* and *place*.

- (1) *asita no Tôkyô to asatte no Kyôto to Ôsaka no tenki* (the weather of Tokyo tomorrow and of Kyoto and Osaka the day after tomorrow)

## 4 Semantic interpretation

Our proposed method to generate semantic trees can be divided into four steps. The following four subsections explain each of them.

### 4.1 Lexical interpretation

Lexical interpretation performs pattern matching with regular expressions against an input utterance. Pattern matching of concept expressions, slot specifiers, and proper names are performed, and an interpretation of the maximum coverage without any overlaps between matched sequences is output. Fig. 4 shows the result of lexical interpretation of example (2). A matched sequence of characters is referred to as a **lexical match**. Fig. 4 contains nine lexical matches.

- (2) *8 gatu 23 niti chekkuin 25 niti chekkuauto de singuru to daburu wo hito heya dutu yoyaku* (Reserve a single room and double room with August 23rd check-in and 25th check-out.)

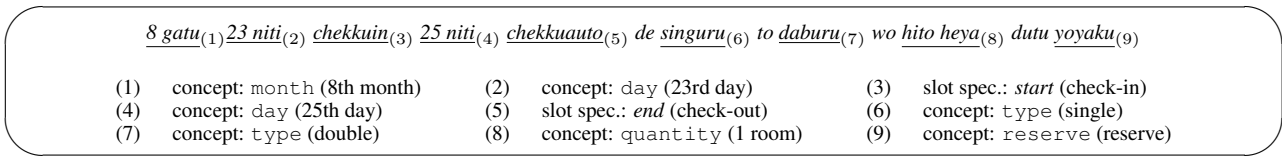
### 4.2 Access table generation

An **access table** represents a result of syntactic analysis in a matrix. It shows whether a path between two lexical matches found by lexical interpretation exists or not on a corresponding syntax tree.

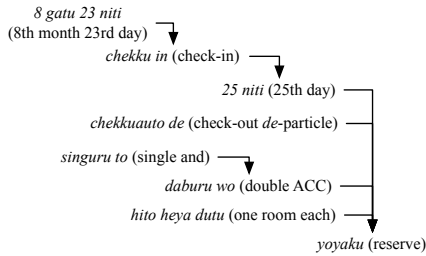
Our method uses an external parser. Any parser can be used only if a wrapper program to interpret the output format is provided. Our method presupposes dependency structure but it is easy to convert phrase structure to dependency structure. The current implementation uses CaboCha parser [5], which is statistics-based and outputs a parse result from any input even if the input is ungrammatical. Fig. 5 shows a parsing result of example (2).

From a syntactic analysis result, an access table is generated according to the result of lexical interpretation. As our method uses a given parser as it is, there can be segmentation inconsistencies between the results of lexical interpretation and syntactic analysis. Syntactic analysis results are conformed to lexical interpretation results because our method gives importance to domain knowledge.

Table 1 shows the access table generated from the lexical interpretation result shown in Fig. 4 and the parsing result shown in Fig. 5. A number *n* at the cross-point of row *x*



**Fig. 4:** A lexical interpretation result of example (2)



**Fig. 5:** A dependency parsing result of example (2) with literal translations (ACC is an accusative marker)

and column  $y$  shows that the lexical match shown on row  $x$  can access the lexical match shown on column  $y$  with  $n$  hops. Empty cells mean unreachability. For example, “double” can access “single” with 1 hop.

### 4.3 Frame interpretation

According to an access table, semantic frames of concepts and slot values of frames are decided.

#### 4.3.1 Frame combination

As explained in Section 2, a concept can have multiple frames. The frame which corresponds to a concept expression in an input sentence is not decidable until evaluating interpretations in terms of numbers of filled slots. Therefore, firstly all combinations of frames are generated. In the case of example (2), because concept *reserve* only has multiple frames in this domain, two combination sequences of frames are generated.

The rest of the frame interpretation process is done for each combination. A frame combination sequence induces a semantic tree.

#### 4.3.2 Bidding

With the use of a **slot assignment table** isomorphic to the given access table, slot values of frames are decided. Bidding is a procedure to claim slot values for each frame. Each frame can bid for lexical matches of concepts or named instances only if the lexical match of the frame is accessible to a target lexical match on the access table.

Table 2 shows the bidding result according to the access table shown in Table 1. Slot specifiers are omitted because they have no truck with bidding and some lexical matches are abbreviated for space. The third row of Table 2 represents the bidding state of the single semantic frame of concept *day* expressed as “23rd day”. The last row represents the bidding state of the second frame of concept *reserve*. The frame has two slots (*start* and *end*) that takes instances of concept *day* as their slot values. Thus both slots are put on “23rd day” and “25th day”.

If multiple frames claim the same lexical match, only the one which has the smallest hops to the lexical match can take it. This restriction is required to prevent inappropriate

	8th m.	23rd d.	25th d.	sgl	dbl	1 r.	rsrv
8th m.							
23rd d.	<i>month</i>						
25th d.							
single							
double							
1 room							
reserve		<i>start,</i> <i>end</i>	<i>start,</i> <i>end</i>	<i>type</i>	<i>type</i>	<i>quant</i>	

**Table 2:** A slot assignment table after bidding

bids. That is why the frame of “25th day” does not bid for “8th month” in the slot assignment table shown in Table 2. In the last result, “25th day” must take “8th month” as its slot value and this will be compensated in intra-sentence ellipsis resolution explained below.

#### 4.3.3 Slot conflict resolution

The proposed method in this paper assumes the uniqueness of semantic roles and a semantic frame cannot bid for a lexical match with more than one slot. Thus slot conflicts as in Table 2 (the third column and the fourth column of the last row) must be resolved. Slot conflict resolution follows the three criteria below.

**Slot specifiers:** If slot specifiers are found, they are used to resolve conflicts. In example (2), *chekkuin* (check-in) specifies the slot (*start*) that takes *23 niti* (23rd day) as its value, and *chekkuauto* (check-out) specifies the slot (*end*) that takes *25 niti* (25th day) as its value.

**Markers:** If markers defined in slot definitions are found in appropriate positions in expressions, corresponding slots are chosen. For example, in “reserve a room from the 10th to the 15th”, “from” and “to” are used as markers to decide slots taking “10th” and “15th” respectively.

**Slot definition orders:** If neither a slot identifier nor a marker is found, conflicts are resolved according to the definition orders of conflicting slots. This enables developers to encode word order tendencies.

#### 4.3.4 Intra-sentence ellipsis resolution

In parallel structures, ellipses happen frequently, because contents specified in first elements of parallel structures are often omitted in later elements. For example, in example (2), “8 *gatu* (8th month)” is given for “23 *niti* (23rd day)” but not for “25 *niti* (25th day)”.

If a parallel structure is found, intra-sentence ellipsis resolution is performed. Here, elements are recognized as in parallel when they satisfy one of the following conditions: (i) all elements are values of the same frame slot, or (ii) elements are values of two frame slots and the two slots make a relationship of beginning and ending.

“*singuru* (single)” and “*daburu* (double)” in example (2) is an instance of case (i) (see Table 2). “23 *niti* (23rd day)” and “25 *niti* (25th day)” in example (2) is an instance of

	8th month	23rd day	check-in	25th day	check-out	single	double	1 room	reserve
8th month									
23rd day	1								
check-in	2	1							
25th day	3	2	1						
check-out									
single									
double						1			
1 room									
reserve	4	3	2	1	1	2	1	1	

**Table 1:** An access table for example (2) (English literal translations only are presented)

	8th m.	23rd d.	25th d.	sgl	dbl	1 r.	rsrv
8th m.							
23rd d.	month						
25th d.	month						
single							
double							
1 room							
reserve	start	end	type	type	quant		

**Table 3:** A slot assignment table after intra-sentence ellipsis resolution

case (ii). In case (ii), only two elements are recognized as in parallel at a time.

If frame  $f$  is recognized as in parallel with frame  $g$  that is to the left of  $f$ , frame  $f$  can bid for a lexical match that frame  $g$  bids for with slot  $s$  as long as the concerned slot of  $f$  is  $s$  and  $s$  is empty. Table 3 shows the slot assignment table after intra-sentence ellipsis resolution.

### 4.3.5 Scoring

This is the end of frame interpretation. According to pre-defined rules, each frame combination is scored in terms of how many slots are filled well and to what extent frames match the context. Rules are not explained because of space limitations.

## 4.4 Semantic tree generation

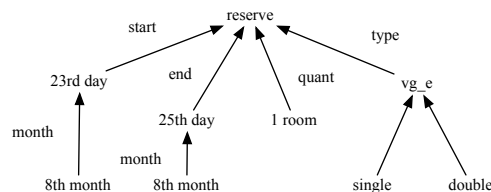
A semantic tree is generated from the best scored slot assignment table via the following three steps.

### 4.4.1 Slot value grouping

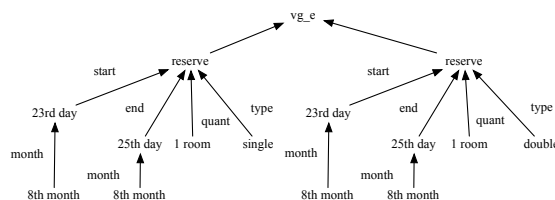
Firstly, repetition patterns of slot values are grouped and separated. Let's think about example (1). All the five concept expressions "asita (tomorrow)", "Tôkyô", "asatte (the day after tomorrow)", "Kyôto", "Ôsaka" in example (1) are slot values of concept weather expressed as "tenki". Among these five values, however, "asita" is related only to "Tôkyô". Likewise, "asatte" is related only to "Kyôto" and "Ôsaka". We have to represent this fact in a semantic tree to be generated. Therefore, such repetition patterns are detected and grouped.

### 4.4.2 Conversion

A semantic tree is generated from the given slot assignment table. Basically, a lexical match of a concept expression or a proper name is converted into a content node. However, if a semantic frame corresponding to a concept expression has grouped slot values, content nodes as many as the number of groups are generated. Fig. 6 shows a semantic tree generated from example (2). A semantic tree in Fig. 3 is also at this stage.



**Fig. 6:** A semantic tree for example (2)



**Fig. 7:** The factorization result of Fig. 6

### 4.4.3 Factorization

Factorization is a procedure to transform a semantic tree by removing value-group nodes. After factorization, a semantic tree contains only one value-group node as its root node. Fig. 7 shows the factorized result of the semantic tree in Fig. 6. Factorization makes the handling of semantic trees simpler in task processing modules which receive semantic trees as their input.

If some slots have multiple values, values of slots are combined. For example, a semantic tree of "asita to asatte no Tôkyô to Ôsaka no tenki (the weather of Tokyo and Osaka tomorrow and the day after tomorrow)" is factorized into four sub-trees representing the following structures: weather(Tokyo, tomorrow), weather(Tokyo, day\_after\_tomorrow), weather(Osaka, tomorrow), and weather(Osaka, day\_after\_tomorrow).<sup>1</sup>

## 5 Evaluation

With the proposed method, we implemented a framework with which one can build dialogue systems just by providing domain knowledge. The framework has a domain-independent slot-filling dialogue manager. We built a hotel reservation system with this framework. The domain knowledge was described in an XML file of 169 lines.

<sup>1</sup> On the other hand, a semantic tree of "asita no Tôkyô to asatte no Ôsaka no tenki (the weather of Tokyo tomorrow and of Osaka the day after tomorrow)" has two sub-trees representing weather(Tokyo, tomorrow) and weather(Osaka, day\_after\_tomorrow) owing to slot value grouping.

U1 : I want to reserve rooms for 7 persons from August 15th for 4 nights.  
 S2 : Which type of rooms would you like? We have single, double, twin, triple and Japanese rooms.  
 U3 : 1 smoking single, 1 non-smoking Japanese room and twin room.  
 S4 : Is your reservation correct as below?  
 Item 1: check-in:2006/8/15, check-out:2006/8/19 (4 nights) room-type:single quantity:1 smoking  
 Item 2: check-in:2006/8/15, check-out:2006/8/19 (4 nights) room-type:Japanese quantity:1 non-smoking  
 Item 3: check-in:2006/8/15, check-out:2006/8/19 (4 nights) room-type:twin quantity:1 non-smoking  
 U5 : Yes.

**Fig. 8:** An excerpt from an observed dialogue (originally in Japanese)

Nine subjects who were not familiar with the system were instructed to work on five different reservation tasks prepared in advance. Subjects performed keyboard dialogues with the system via web browsers. Subjects were not instructed anything about the system's competence except that the system understood Japanese. Fig. 8 shows an excerpt from an observed dialogue.

Utterance U1 contained a phrase "for 7 people", which the system did not know. However the system worked without any troubles just by ignoring them owing to its robust understanding mechanism. All parallel structures and intra-sentence ellipses in U3 were correctly understood. Note that anything about parallel structures was not considered when the domain knowledge was coded. S2 was generated by the system as a result of domain-independent slot-filling dialogue management. S4 was generated by a domain-dependent dialogue controller, which is not explained because it is out of the scope of this paper.

## 5.1 Task completion

The task completion rate was 65.0% (26 in 40 dialogues excluding 5 dialogues halted due to system malfunctions). This rate is not high but we confirmed that the system performed complex understanding as shown in Fig. 8 only with simple domain knowledge definitions.

One major reason of task incompletions was users' out-of-domain requests (OODRs). As subjects were not instructed anything detailed, they asked various OODRs to the system. Because the system had no ability to avoid misunderstanding of such OODRs, many dialogues broke-down. Half of the failed dialogues were due to this problem.

Another major reason was the lack of domain-independent quantity handling ability. With the given domain knowledge, the system handled quantities of rooms. However, it was just a frame slot of concept *reserve* and the system did not know the way to individually handle multiple rooms specified with a quantity expression. Subjects often specified multiple rooms with such an expression "Reserve two single rooms.", and after that they requested further options with such an expression "One room is smoking and the other is non-smoking." In this case, the system has a single representation for "two single rooms" and cannot set separate parameter values for each individual room. Most of the other half of the failed dialogues were due to this problem.

## 5.2 Semantic interpretation

We obtained 151 utterances out of 372 utterances collected in the experiment by leaving out out-of-domain requests

and utterances containing just one concept. The performance of the proposed method was examined on these 151 utterances.

In the 151 utterances, the proposed method recognized 453 pairs of two lexical matches that have a relationship of a frame and its slot value. 94.0% (426 pairs) of them were correct. This shows our method correctly interpreted most of concepts in in-domain utterances. Among those 426 pairs, only 221 pairs held valid dependency relationships on corresponding dependency structures. Thus, if we adhered to syntactic analysis results naively, only 48.8% of slot values would be interpreted correctly.

There were 77 parallel structures and 68 (88.3%) of them were correctly interpreted. We found 15 intra-sentence ellipses and 13 (86.7%) of them were correctly interpreted.

## 6 Concluding remarks

This paper described a method to generate semantic representations using syntactic analysis. Thus, it can handle more complex expressions than those not using syntactic analysis and is applicable to a wide variety of domains. In an NLU system K2 [10], users can command animated characters in a 3D virtual world to move objects. In this domain, since users' commands contain recursive structures in expressions referring to objects, TLU based frameworks such as [4] do not work well. We replaced the language understanding part of K2 by using our framework mentioned in Section 5 and found the framework reduced workload to build such an NLU system by about 90% in terms of the coding amount when compared to building the system from scratch. In future work, we would like to conduct more detailed evaluations.

## References

- [1] M. O. Dzikovska, J. F. Allen, and M. D. Swift. Integrating linguistic and domain knowledge for spoken dialogue systems in multiple domains. In *Proceedings of IJCAI-03 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 2003.
- [2] E. Jackson, D. Appelt, J. Bear, R. Moore, and A. Podlozny. A template matcher for robust NL interpretation. In *Proc. of Speech and Natural Language Workshop*, pages 190–194, 1991.
- [3] S. Knight, G. Gorrell, M. Rayner, D. Milward, R. Koeling, and I. Lewin. Comparing grammar-based and robust approaches to speech understanding: a case study. In *Proc. of EUROSPEECH 2001*, 2001.
- [4] T. Konashi, M. Suzuki, A. Ito, and S. Makino. A spoken dialog system based on automatic grammar generation and template-based weighting for autonomous mobile robots. In *Proc. of INTERSPEECH 2004*, 2004.
- [5] T. Kudo and Y. Matsumoto. Japanese dependency analysis using cascaded chunking. In *Proc. of CoNLL 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69, 2002.
- [6] D. Milward. Distributing representation for robust interpretation of dialogue utterances. In *Proc. of ACL 2002*, 2000.
- [7] M. Nakano, Y. Hasegawa, K. Nakadai, T. Nakamura, J. Takeuchi, T. Torii, H. Tsujino, N. Kanda, and H. G. Okuno. A two-layer model for behavior and dialogue planning in conversational service robots. In *Proceedings of IROS 2005*, 2005.
- [8] C. P. Rosé. A framework for robust semantic interpretation. In *Proc. of NAACL 2000*, 2000.
- [9] S. Seneff. TINA: A natural language system for spoken language applications. *Computational Linguistics*, 18(1), 1992.
- [10] T. Tokunaga, K. Funakoshi, and H. Tanaka. K2: Animated agents that understand speech commands and perform actions. In *Proc. of PRICAI 2004*, 2004.
- [11] Y.-Y. Wang and A. Acero. Rapid development of spoken language understanding grammars. *Speech Communication*, 48(3-4):390–416, 2006.

# A proposal of a morphological tagger for Spanish based on Cuban corpora

Lisette García-Moya  
Center of Pattern Recognition and  
Data Mining  
Universidad de Oriente  
Santiago de Cuba  
lisette@csd.uo.edu.cu

Aurora Pons-Porrata  
Center of Pattern Recognition and  
Data Mining  
Universidad de Oriente  
Santiago de Cuba  
aurora@csd.uo.edu.cu

Leonel Ruiz-Miyares  
Center for Applied Linguistic  
Santiago de Cuba  
leonel@lingapli.ciges.inf.cu

## Abstract

In this paper we describe a morphological tagger for Spanish based on Cuban corpora. The tagger combines Hidden Markov Models with some heuristics and dictionaries to provide the appropriate part-of-speech tag for each word in a text document, according to the context in which it appears.

Moreover, a morphological analyser that provides all possible morphological interpretations of words is used. It allows us to reduce possible grammatical tags and to obtain not only the appropriate part-of-speech tag, but also its morphological information. The proposed tagger achieves 97.76 % accuracy for a legal corpus.

## Keywords

Morphological tagger, Hidden Markov Model, Statistical Natural Language Processing.

## 1. Introduction

Part-Of-Speech (POS) tagging is an essential task for all Natural Language Processing activities, for example, Information Retrieval, which in turn helps managing the enormous amount of text documents available nowadays, such as: Web pages, news, scientific papers, emails, etc.

Many words in natural language are grammatically and semantically ambiguous. Grammatical disambiguation consists of assigning the appropriate part-of-speech tag to each word of a given textual document, according to the context in which word appears. This type of annotation is carried out by a morphological tagger.

Morphological taggers are classified into deductive systems based on knowledge, inductive systems based on machine learning approaches and hybrid systems.

In deductive systems –also known as linguistic approaches- the model is written by a linguist, generally in form of rules or constraints [14]. The linguistic models range from a few hundreds to several thousand rules, and they usually require years of hard work.

Inductive methods consider that linguistic knowledge may be inferred by experience. This experience is obtained by textual corpora. Inductive methods build a

computational model from a set of examples which may be annotated with linguistic information or not, using learning or statistical methods. These methods could be supervised or unsupervised, depending whether training data contains linguistic information or not, respectively. Many inductive techniques have been developed to solve the problem of grammatical disambiguation, such as:  $n$ -grams models [1], memory-based learning [5], transformation-based error-driven learning [3], Hidden Markov models (HMM) [11], maximum entropy [12] and decision trees [16]. Markov models combined with a good smoothing technique and with handling of unknown words perform at least as well as other current approaches [2,6].

Finally, hybrid models [10] combine statistical information with automatically extracted rule-based information trying to join the advantages of both approaches.

Most of the taggers have been developed for the English language. Nevertheless, several hybrid POS taggers for the Spanish language have been proposed, such as *Freeling* [4] and the Spanish version of *TreeTagger* [16]. *TreeTagger* is based on decision trees, whereas *FreeLing* is a trigram HMM tagger. Although these taggers achieve good results, they have some limitations.

*TreeTagger* tends to assign the proper noun tag to words beginning with a capital letter, even when that word is, in fact, a common noun, as in *Banco Popular de Ahorro*. The tagset of *TreeTagger* is very basic. As a consequence, a lot of potentially useful morphological information (including, for example, gender, number, verb person, etc.) is not included in the tags. Numbers were also problematic. They were generally treated as CARD (Cardinal), but in some cases they were tagged as CODE (Alphanumeric code). Verb forms with enclitic pronouns are tagged as verbs only, resulting in loss of information on such pronominal particles.

*Freeling* is unsuccessful when encountering certain words not present in its vocabulary, such as unknown place names (Azerbaiján and Tampere, tagged as a verb,

etc.). For unclear reasons, in some cases Freeling is not able to find the lemma of certain plural nouns and adjectives and left them in the plural [15].

Both TreeTagger and Freeling are neither able to recognise pronominal verbs that are reflexive of form, for example, *me abstengo*. Moreover, they do not recognise dates or times in short format, such as 25/12/2007, 25-12-2007 or 12:45.

On the other hand, there are some differences between the Spanish spoken in Cuba and the Spanish spoken in other Spanish-speaking countries, basically from a lexical point of view. The Spanish language together with Sub-Saharan African and Indocuban languages were three linguistic trends that strongly determined the own characteristics of the Spanish spoken in Cuba. Words of african origin such as *quimbombó*, *sambumbia* and *conga* and indigenous words (e.g. *hayaca*, *caguairán*, *fotuto*) enrich this language. At morphological level, there are no notable differences between the Spanish language spoken in Cuban and the Spanish of other countries. However, a morphological peculiarity could be mentioned: *vos* does not exist in Cuba; *tú* is used instead on informal environments and *usted* when the relation requires a polite form.

Ruiz-Miyares presented ETIPROCT [13], a morphological tagger for the Spanish spoken in Cuba,

which achieves satisfactory results. However, the tagset of this tagger is limited and it does not allow annotating the text with morphological information and lemmas.

In this paper, we propose a morphological tagger with a greater tagset and broader morphological information than ETIPROCT. It combines HMM with a morphological analyser, heuristics and dictionaries. This tagger is considered as a hybrid one. The morphological analyser we used is based on two-level morphology from Kimmo Koskenniemi [8]. This paper is focused on the morphological tagger.

## 2. The morphological tagger

The architecture of the proposed tagger is shown in Fig. 1.

The tokeniser divides the raw text into atomic items and identifies the sentence boundaries. It is able to recognise words, punctuation marks, symbols and identifiers. We understand as *identifier* any sequence of characters that is not a word in the language, such as: email addresses, URLs, expressions like:  $2+5*4=22$ , and others. Tokeniser also identifies acronyms, measurement units, abbreviations, phrases (nominal, adjectival, adverbial, prepositional, conjunctive and Latin phrases), dates, times and numbers by using several dictionaries and heuristics.

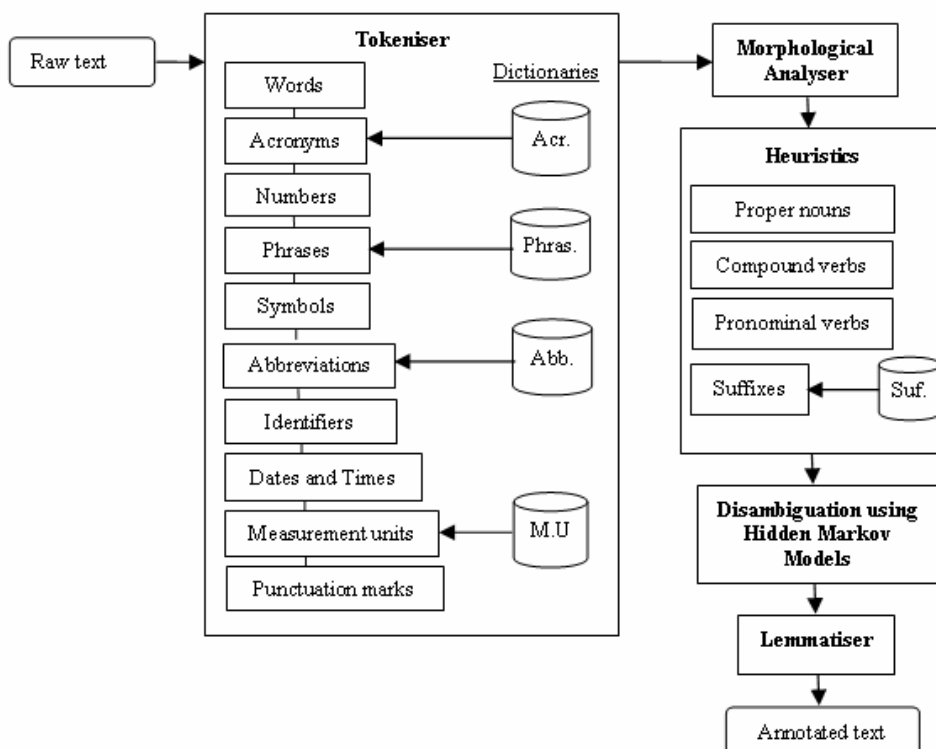
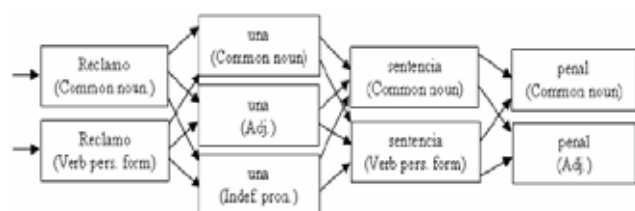


Figure 1. Architecture of the morphological tagger



The morphological analyser provides all possible interpretations of a given word and its morphological features. These features include gender, number, verb person, verb time, lemma information and so on. A *lemma* is defined as the canonical form of a word. A word could have different lemmas, for example, the lemma of the word *camino* is *camino* if it is a noun, and *caminar* if it is a verb.

Instead of regarding all possible tags for each word in the test data, we only consider the possible tags given by the morphological analyser. Thus, it allows us to constrain the set of possible grammatical categories of a given word and reduces the number of computations. For example, each word of the sentence *Reclamo una sentencia penal* has different possible tags given by the morphological analyser (see Figure 2).



**Figure 2. Possible tags are taken from morphological analyser**

The morphological analyser and dictionaries of abbreviations, acronyms and phrases include the proper characteristics of Spanish spoken in Cuba on the basis of the Cuban corpora.

The proposed tagger is also able to recognise proper nouns, compound verbs (e.g. *he votado*) and pronominal verbs (e.g. *me abstengo*) by using some heuristics such as capital letters, the presence of verb *haber* and certain pronouns, etc. The set of possible part-of-speech tags of each word is obtained as a result of tokenization process, morphological analysis and heuristics.

Disambiguation process is carried out by applying Hidden Markov Models from the set of possible part-of-speech tags obtained before. Finding the appropriate lemma of each word is trivial by using the part-of-speech tag obtained by HMM model and the information provided by the morphological analyser. The lemmatiser considers acronym meaning and the expanded form of abbreviations and measurement units as lemmas.

The tagset used in our proposal is shown in Table 1. It is important to mention that tags include not only information about the major parts of speech but also other morphological information, such as number and gender for nouns, tense for verbs, and superlative, diminutive and despective forms for adjectives.

Table 2 shows the morphological features for each POS tag provided by the morphological analyser.

**Table 1. Used tagset**

Proper noun	Adjectival phrase
Common noun	Verbal phrase
Personal pronoun	Adverbial phrase
Demonstrative pronoun	Prepositional phrase
Possessive pronoun	Conjunctive phrase
Indefinite pronoun	Latin phrase
Relative pronoun	Article
Interrogative and exclamative pronoun	Preposition
Verb in personal form	Conjunction
Verb infinitive	Interjection
Verb gerund	Contraction
Verb participle	Adjective
Verb in personal form with enclitic	Adverb
Verb infinitive with enclitic	Acronym
Verb gerund with enclitic	Number
Multiple numeral	Measurement unit
Cardinal numeral	Date and Time
Ordinal numeral	Identifier
Collective numeral	Symbol
Fractional numeral	Punctuation mark
Nominal phrase	

**Table 2. Morphological features for each POS tag**

POS tag	Morphological features
Common noun	gender, number, degree
Personal pronoun	gender, number, person, politeness
Demonstrative pronoun	gender, number
Possessive pronoun	gender, number, person, politeness
Indefinite pronoun	gender, number
Relative pronoun	gender, number
Interrogative and exclamative pron.	gender, number
Verb in personal form	transitivity, pronominality, mode, tense, number, person, politeness
Verb participle	gender, number
Verb in personal form with enclitic	transitivity, pronominality, mode, tense, number, person, politeness
Article	gender, number
Adjective	gender, number, degree

## 2.1 Hidden Markov Model

As we mentioned above, the proposed morphological tagger is based on Hidden Markov Models. HMM is a widely used probabilistic finite state machine having a set of states, an output alphabet, transition probabilities, observation probabilities and initial state probabilities. In

our HMM model, states correspond to part-of-speech tags and observations correspond to words.

The HMM will be used to assign the most probable tag to the words of an input sentence. As we use a bigram model, output probabilities only depend on the most recent category, that is,

$$\arg \max_{c_j \in \{T_1, \dots, T_n\}} \{P(w_k | c_i) \cdot P(c_i | c_j)\} \quad (1)$$

where  $w_k$  is the word to be disambiguated,  $\{T_1, \dots, T_n\}$  is the possible tagset for  $w_k$  and  $c_j$  is the tag assigned to the previous word. Transition and observation probabilities are estimated from a tagged corpus.

When  $w_k$  is at beginning of a sentence, the probability of  $c_i$  being the grammatical category of the first word in the sentence ( $\pi_i$ ) is estimated, instead of the transition probability  $P(c_i | c_j)$ , that is:

$$\arg \max_{c_i \in \{T_1, \dots, T_n\}} \{P(w_k | c_i) \cdot \pi_i\} \quad (2)$$

## 2.2 Handling unknown words

The words that were not seen during the training are known as *unknown words*. Currently, the method of handling unknown words that seems to work best for inflected languages is a suffix analysis.

As Spanish is an inflected language, we use this method to predict the possible tags of an unknown word. In order to do that, we built a dictionary of frequent suffixes and its possible POS tags. For example, the *-eria* suffix is an indicator that word could be a common noun (e.g. *extranjeria*) or a verb in personal form (e.g. *apareceria*).

In addition to the possible tags of the unknown word, an observation probability is required to applied equations (1) or (2).

To overcome data sparseness we apply the *Adding One* smoothing method, also known as Laplace's law [7], which adds one to all frequencies, thus avoiding zeroes and reducing the proportion between rare happening events. The observation probability is defined as follows:

$$P^{smoothing}(w_k | c_i) = \frac{f(w_k, c_i) + 1}{f(c_i) + |V|}$$

where  $V$  is the vocabulary in the training corpus,  $f(w_k, c_i)$  is the number of times that word  $w_k$  is tagged as  $c_i$  and  $f(c_i)$  is the number of words tagged as  $c_i$  in the training corpus. Then, the observation probability for unknown words is:

$$P^{smoothing}(w_k | c_i) = \frac{1}{f(c_i) + |V|}$$

If suffix analysis does not provide any possible grammatical category for the unknown word, Hidden Markov Model is applied assuming that unknown words may potentially have all tags, excluding those tags corresponding to closed categories (preposition, conjunction, article, etc.), which are considered to be all known. For unknown words, we consider as lemma the own word.

## 3. Experimental results

In order to evaluate our approach, a legal corpus containing 231634 words is built. This corpus was manually annotated by human experts.

We perform a 10-fold cross validation using 90% of the combined data set as training data and the remainder as test data. In the experiments, we use accuracy as our evaluation measure. It is defined as the ratio of the number of correctly tagged words to the total number of words.

The obtained results are shown in Table 3. Second and third columns contain the number of words in the training and test sets, respectively. As it can be appreciated, we obtained a similar accuracy to that of the current state-of-the-art taggers [2,4,9,12].

Table 3. 10-fold cross validation results

Subst	Training set	Test set	Accuracy (%)
1	208360	23274	98.02
2	209102	22532	97.71
3	208344	23290	97.70
4	209117	22517	97.69
5	207428	24206	97.70
6	209356	22278	97.94
7	207965	23669	97.69
8	208523	23111	97.77
9	208262	23372	97.69
10	208249	23385	97.73
<b>Average</b>			97.76

Table 4 summarizes the averaged accuracies obtained over different POS tags. As shown in the table, the tagger performs the worst in the verbs. The most common problems have been labelling as common noun words that should be infinitive verb, or labelling as adjective words that should be verb participle. In all other part-of-speech tags the accuracy values are similar. Thus, it seems that the effectiveness is not affected with different tags.

## 4. Conclusions

In this paper, a morphological tagger for texts written in Spanish with a particular emphasis on the Cuban variant has been presented. However, this tagger is able to process any text written in Spanish.

Table 4. Accuracy over different POS tags

POS tag	Averaged Accuracy
Proper nouns	100
Common nouns	97.69
Pronouns	98.01
Verbs	92.68
Numerals	98.97
Phrases	96.97
Article	98.23
Preposition	99.95
Conjunction	96.57
Contraction	99.98
Adjective	97.12
Adverb	98.06
Acronym	99.73
Number	98.09
Date and Time	100
Punctuation mark	99.96

The proposed tagger combines bigram-based HMM with a set of heuristics and dictionaries. Besides, it uses a morphological analyser which allows us to constrain the set of candidate grammatical categories to be considered for each word and provides richer morphological information.

In the experiments carried out on a legal corpus, we obtained a satisfactory accuracy (97.76%) that is similar to that of other taggers reported in the literature. The most common errors have been labelling words that should be verb infinitive as common noun, or words that should be verb participle as adjective. The proposed tagger becomes a high-quality tool for the annotation of Cuban corpora with part-of-speech information.

As future work, we plan to evaluate our morphological tagger on corpora from other knowledge domains. Also, we want to integrate it into other natural language processing tools, such as a named entity recogniser.

## 5. References

- [1] L. R. Bahl, F. Jelinek and L. R. Mercer. A Maximum-Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI, pp. 179-190, 1983.
- [2] T. Brants. TNT-A Statistical Part-of-Speech Tagger. In *Proc. of the 6th Applied Natural Language Processing Conference ANLP-2000*, Seattle, pp. 224-231, 2000.
- [3] E. Brill. A simple rule-based Part of Speech tagger. In *Proc. of the 3rd Conference on Applied Natural Language Processing of the Association for Computational Linguistics*, Trento, pp. 152-155, 1992.
- [4] X. Carreras, I. Chao, L. Padró and M. Padró. Freeling: an Open-source Suite of Language Analyzers. In *Proc. of the 4th International Conference on Language Resources and Evaluation*, Lisbon, pp. 239-242, 2004.
- [5] W. Daelemans, J. Zavrel, P. Berck and S. Gillis. MBT: A Memory-Based Part of Speech Tagger-Generator. In *Proc. of the 4th Workshop on Very Large Corpora*, Copenhagen, pp. 14-27, 1996.
- [6] S. Dandapat, S. Sarkar, and A. Basu. A Hybrid Model for Part-of-Speech Tagging and its Application to Bengali. *International Conference on Computational Intelligence*, pp. 169-172, 2004.
- [7] H. Jeffreys. *Theory of Probability*, Second Edition, Section 3.23, Oxford, Clarendon Press, 1948.
- [8] K. Koskenniemi. *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*. PhD Thesis. University of Helsinki, 1983.
- [9] L. Márquez. *Part-of-speech Tagging: A Machine Learning Approach based on Decision Trees*. PhD Thesis. Polytechnic University of Catalonia, Barcelona, 1999.
- [10] L. Márquez and L. Padró. A flexible POS tagger using an automatically acquired language model. In *Proc. of ACL-97*, Madrid, pp. 238-245, 1997.
- [11] A. Molina. *Disambiguation in Natural Language Processing by using machine learning techniques*. PhD Thesis. Polytechnic University of Valencia, (In Spanish) 2004.
- [12] A. Ratnaparkhi. *A Maximum Entropy Model for Part-of-Speech Tagging*. In *Proc. of the 1st Conference on Empirical Methods in Natural Language Processing*, EMNLP, Pennsylvania, 1996.
- [13] L. Ruiz-Miyares. *Development of a computational model based on tagging for processing textual corpora*. PhD Thesis. Universidad de Oriente, Santiago de Cuba - University of Twente, Holanda, (In Spanish) 2001.
- [14] C. Samuelsson and A. Voutilainen. Comparing a Linguistic and a Stochastic Tagger. In *Proc. of 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, ACL, Madrid, pp. 246-253, 1997.
- [15] A. Sandrelli and C. Bendazzoli. Tagging a Corpus of Interpreted Speeches: the European Parliament Interpreting Corpus (EPIC). In *Proc. of the 5th International Conference on LREC*, Genoa, pp. 647-652, 2006.
- [16] H. Schmid. Probabilistic Part-of-speech Tagging Using Decision Trees. In *Proc. of the Conference on New Methods in Language Processing*, Manchester, UK, pp. 44-49, 1994.

# A\* Parsing with Large Vocabularies

Guillem Gascó and Joan Andreu Sánchez  
Instituto Tecnológico de Informática  
Universidad Politécnica de Valencia  
Camino de Vera s/n, Valencia 46022 (Spain)  
{ggasco,jandreu}@dsic.upv.es

## Abstract

In this work, we study an A\* parsing algorithm to obtain exact Viterbi parse selection for a real task with a large vocabulary. We propose two different new bounds for parsing. The first one includes lexical information and guarantees the optimal solution with an important cost reduction. The second one offers a larger cost reduction but the optimality of the solution is not guaranteed. Nevertheless, in practice the optimal solution for this second bound is lost in very few cases. Experiments on the Penn Treebank corpus are reported in order to show the improvements obtained with the proposed bounds.

## Keywords

Parsing, Stochastic Context-Free Grammar, A\* Search

## 1 Introduction

Syntactic parsing is an important problem related to Natural Language Processing (NLP) that has proven to be useful for RNA Modelling [19], Language Modelling [8, 17, 2], and Machine Translation [21, 7], among others [14]. In stochastic syntactic parsing, a parse tree is obtained for an input sentence according to some criteria by using a stochastic model and a parsing algorithm.

Stochastic Context Free Grammars (SCFGs) are powerful formalisms that have been widely used for stochastic syntactic parsing [1, 20, 17, 9]. Some of the current syntactic parsing algorithms [9] are based on the classical Cocke-Younger-Kasami [11] (CYK) and Earley [10] algorithms. An important problem that is related to CYK and Earley algorithms is their cubic time complexity.

In recent years, other syntactic parsing algorithms have been considered for real NLP tasks. In these algorithms, other search strategies that are different from the Dynamic Programming scheme are used to compute a possible parse tree [5, 17, 12, 6]. Most of these algorithms use an agenda to store the items to be processed. The items are chosen according to some *figure of merit*. In some cases, the optimality of the parse tree is not guaranteed.

In [12], an A\* algorithm is presented to compute the exact, most probable parse tree of a string. Several bounds are considered for the A\* search and very good results are reported with delexicalized sentences from the Penn treebank.

In this work, we extend the work presented in [12] by exploring the application of the A\* search algorithm on real tasks with a large vocabulary. New bounds are introduced for lexicalized sentences.

In section 2, we briefly review agenda-based chart parsing algorithms. Then, in section 3, we discuss new bounds for the A\* parsing algorithm introduced in [12]. Finally, section 4 presents experimental results on the Penn treebank corpus.

## 2 Agenda-based Chart Parsing

The goal of stochastic syntactic parsing is to obtain a parse tree for a given input sentence. For this purpose, a stochastic grammatical model is used together with a parsing algorithm. SCFGs are grammatical models that are commonly used for stochastic parsing.

Some of the current syntactic parsing algorithms are based on the classical CYK [11] and Earley [10] algorithms. Both are dynamic programming algorithms. The cubic time complexity of these algorithms restricts their use when dealing with wide-coverage grammars and long sentences. Therefore, a method for accelerating parse selection must be considered.

One of these methods consists of using a beam-search strategy together with a greedy algorithm [17]. However, the nature of these algorithms sometimes implies the loss of the most probable parse tree because the global optimum is not necessarily optimal at an intermediate stage. Therefore, the optimum at this stage could be pruned from the list of hypotheses.

Another possible method to accelerate parse selection consists of using a chart together with an agenda. The agenda is used to store the items to be processed. The items are chosen from the agenda according to some *figure of merit*. If the *figure of merit* is appropriately chosen, the number of items that are processed before obtaining a possible parse tree is notably lower than the maximum number of items that should be processed in an exhaustive search.

In [4], a maximum-entropy-inspired parser is presented. First, a parser that uses a chart together with an agenda is used to generate possible candidate parses. The figure of merit that is used to choose the item from the agenda is defined by using a lexicalized SCFG [5]. Second, a probabilistic model that is based on the maximum entropy principle is used to evaluate the candidates parse trees introduced in the agenda in the first step. The parse tree obtained in this way is not guaranteed to be the exact, most probable parse

tree according to the SCFG. The experiments reported in [5] exhibited a very good performance on the Penn treebank corpus. Recent improvements in this parsing algorithm achieve about 92% f-measure by using semi-supervised learning [16].

In [12], an A\* algorithm is presented to compute the exact most probable parsing of a string. In this parser, the search is driven by a function that guarantees that the best parse string is not lost. In that work, several bounds are proposed for the A\* search, and experimental results are reported for delexicalized strings of the Penn treebank corpus. The space complexity is very large for some the bounds proposed in [12], which hampers their application in tasks with large vocabularies.

Given the large space complexity associated to some of the bounds proposed in [12], in the following section, we propose new bounds for real tasks with large vocabularies. We study experimentally which of them can be used in a real scenario. In addition, we propose using some bounds that can decrease the number of processed edges notably even though they do not guarantee the optimality of the solution.

### 3 Lexicalized Bounds for A\* Parsing

The main difficulty of obtaining the highest probability parse tree of a sentence by means of the probabilistic version of the CYK algorithm is its cubic cost. In [12], a A\* search procedure is applied to this task to reduce the time required while keeping the optimality. We summarise this procedure below.

An A\* search is a guided search across the problem solution space which is a special case of *Best-first search*. This solution space is composed by *edges*. An edge represents a non-terminal symbol of the grammar over a span. The search procedure uses a function  $f(e)$  in every edge  $e$  of the solution space in order to decide if it will be the next edge to be explored. The  $f(e)$  function is the combination of two functions:  $g(e)$  and  $h(e)$ . Function  $g(e)$  is the probability of the edge  $e$ , that is, the probability of the most probable parse tree that starts from the non-terminal symbol of the edge. The function  $h(e)$  is an estimation of the future probability of obtaining a goal edge (a solution of the problem) from  $e$ . The function  $h(e)$  is based on the outside part of the span of  $e$ , i.e., on its outside context. The closer that  $g(e)$  is to the real cost (probability), the more edges will be pruned. If the  $h(e)$  function does not overestimate the cost to reach the goal, the search will be complete and optimal [18].

This search method can be applied to the problem of finding the best parsing for a sentence, given a stochastic context-free grammar. The most promising edge at each moment is the one that is chosen edge to be expanded. Thus, an agenda with all the hypotheses (edges) ordered by their estimated cost (probability) is needed. From now on, we will assume that the probability of an edge is represented by its logarithm, so the product of probabilities is, in fact, the addition of their logarithms. The edge with the highest probability in the agenda is the most promising one.

In [12], some context summary estimates are proposed with good results. A summary of the context of an edge in a sentence is taken (for example, we only take into account the word on the left of  $e$  as its context). There are probably many sentences that fits the same context, so, we take the maximum probability of all the edges  $e'$  that fit that context, i.e., the minimum cost. Hence, the real cost of getting a goal from  $e$  is always lower or equal to this estimate. Context summary estimates are always admissible functions since they never overestimate the cost to reach the goal. Their value is the maximum probability over all the derivations that fit the context and the real probability cannot be larger. If the summary function is carefully chosen and the number of contexts is not excessively large, we can precompute them and access to each one in constant time per edge in parse time.

Some of these summary estimates are described below. The simplest of all is the **NULL** context estimate: all the possible contexts have the same probability. The **SX** context estimate takes into account the number of terminals on the left and on the right of the edge. **SXL** also takes into account the terminal that is on the left of the edge. Other more complex context summary estimates have been considered in that work. Finally, the non-practical context estimate is the real cost of the outside part of the edge, i.e., the **TRUE** estimate.

Most of the context summary estimates proposed cannot be computed when dealing with a big SCFG. For example, the space complexity for the **SXL** estimate (one of the simplest estimates) is  $\Theta(l^2 \cdot N \cdot T)$  where  $l$  is the maximum length of the sentences parsed,  $N$  is the number of non-terminal symbols, and  $T$  is the number of terminal symbols. For a 70-length sentence with the grammar proposed in Section 4 with 97 non-terminal symbols and more than 40,000 terminal symbols, there are more than  $1.9 \cdot 10^{10}$  possible contexts. Hence, assuming a space cost of 4B per context, more than 70GB is needed to store them.

Therefore, of the context summary estimates proposed in [12], the most informative one that can be used with grammars of this kind, is **SX**. This context estimate ignores the lexical context of the edge. For grammars with many of lexical rules, **SX** is too optimistic and prunes very few edges.

To solve this problem, a new estimate **SXLex** (a lexicalization of the **SX** estimate) can be used. This estimate uses more contextual information in order to improve the parsing process. **SXLex** can be divided into two parts. The first part is similar to the **SX** context summary estimate but does not take into account the probability of generating terminals from preterminal symbols, that is, the maximum probability of all the derivation trees that produce  $L$  preterminal symbols to the left of the edge symbol and  $R$  to the right. As with **SX**, this part can be precomputed. The second part of the **SXLex** estimate, the lexical part, must be computed one time for each sentence. The lexical part of a sentence  $S$ ,  $\text{Lex}(S, L, R)$ , taking into account  $L$  symbols to the left and  $R$  to the right, is the maximum probability over all the derivations from preterminal symbols to vocabulary words. That is:

$$Lex(S, L, R) = \sum_{r=1}^R \max_N P(N \rightarrow S_r) + \sum_{l=|S|-R}^{|S|} \max_N P(N \rightarrow S_l)$$

The SxLex estimate of an edge is the sum the two parts described above. The optimality of SxLex can be easily proved. The lexical part is the probability of an optimistic derivation from preterminals to the terminals of the string being parsed. The SX part is a context summary estimate for the preterminals before and after the edge. The sum of the parts is always greater or equal to the real future cost.

In order to reduce the number of edges to be processed even more, a new bound closer to the TRUE cost is proposed. This bound, SxLex2, is a combination of SX and the lexical part of the SxLex estimate. The main difference between SxLex and SxLex2 is that SxLex uses the SX estimate from the initial symbol of the grammar to the preterminals and SxLex2 uses the complete SX estimate from the initial symbol to the terminals (vocabulary words). It should be noted that this bound is not always optimistic. Hence, the most probable parsing is not guaranteed and is a 'non-optimal' bound. However, in spite of this fact, the derivation tree obtained is the optimal one in most of the cases.

Figure 1 shows an example of the value of the different estimates for a given edge in the parsing process of a sentence from Penn Treebank. If no bound is used (NULL estimate), the number of edges processed is 1732. The SX estimate only takes into account the non-terminal of the edge (NNP) and the number of terminals before and after it. As can be observed, SX is too optimistic, so the number of processed edges to parse the sentence is still large. Taking into account lexical information, SxLex, the number of edges decreases to 346. Finally, with the SxLex2 bound, the value obtained is closer to the TRUE value, and is still optimistic. The savings produced with this bound are considerable: only 85 edges are needed.

## 4 Experiments

For the experiments in this section, we used the Penn Treebank corpus [15] following [5, 12]. Sections 2-21 were used for training, and section 23 was used for testing. All words that had the POSTag CD (cardinal number [15]) were replaced by a special symbol that did not appear in the initial vocabulary. The training sections were used to obtain a SCFG.

Our version of the A\* parsing algorithm used a SCFG in CNF with unitary rules. Given that binarization process was not included in the A\* parsing algorithm, a binarization process was carried out on the original grammar obtained from the treebank. Several strategies for the binarization process could be used by considering a vertical and horizontal markovization [13]. This markovization process has proven to be very important in order for improving parsing performance. However, in this work, we did not use any

markovization. In order to obtain the SCFG from the treebank and to binarize it, we used the NLTK toolkit [3]. The obtained SCFG had 97 non-terminal symbols, 40,289 terminal symbols (the vocabulary of the task), and 51,495 rules (4,002 syntactic rules and 47,493 lexical rules). No pruning process was carried out on this SCFG.

Given that the training corpus did not contain all the vocabulary of the test set, we removed all the sentences of the test corpus that contained any unknown words. No smoothing process was carried out with the SCFG. In this way, the test corpus contained 1,582 sentences. Fig. 2 shows a histogram of the sentence length of the test set.

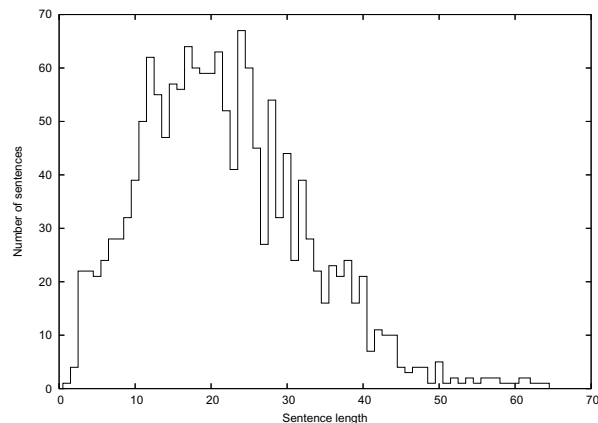


Fig. 2: Histogram of the sentence length of the test set.

In order to evaluate the bounds described in Section 3, we computed the average number of edges taken from the agenda before finding the best parse. The same evaluation is proposed in [5]. Fig. 3 shows the average savings with respect to the NULL estimate.

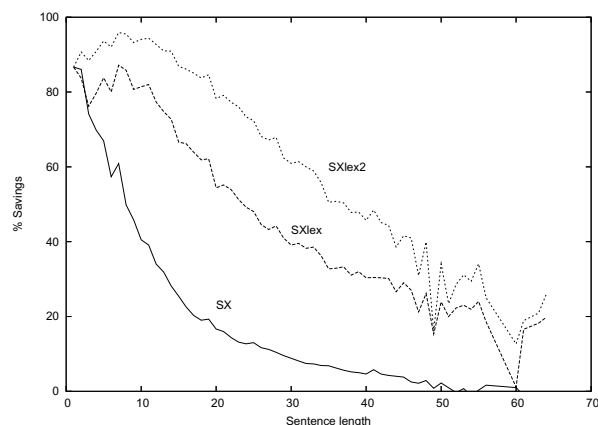


Fig. 3: Average savings with respect to the NULL bound as a function of the sentence lengths.

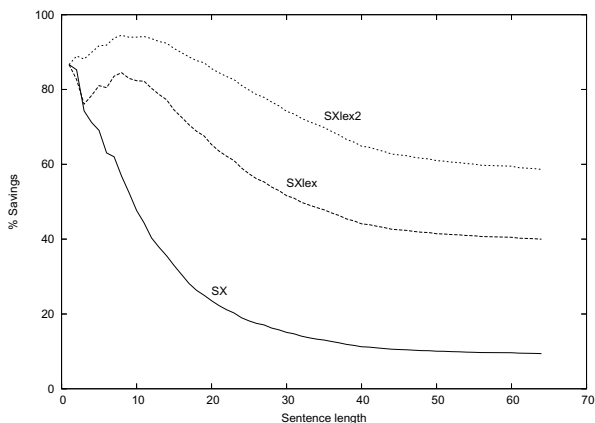
These results are not directly comparable with the ones reported in [12] because they have been carried out with dellexicalized strings and over a set of sentences of lengths between 18 and 26 words.

Note that the savings are remarkable for sentence lengths up to 40, which is a common sentence length

Estimate	a) SX	b) SXlex	c) SXlex2	d) TRUE
Context				
Score	-27.33	-32.83	-41.75	-49.16
Num. of edges needed	679	346	85	-

**Fig. 1:** Estimates for a given sentence: Context information needed, score obtained for each of the bounds and number of edges needed to finish the parsing.

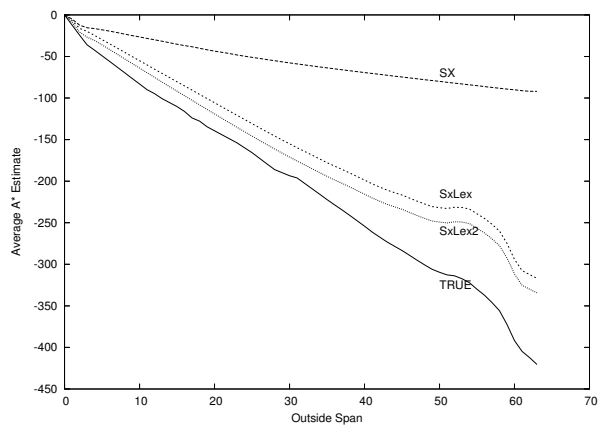
reported in other works. There was some variability for sentences longer than 40 words which may have been produced by the low number of sentences. Fig. 4 shows the cumulative average savings with respect to the NULL bound as a function of the sentence length. The performance of the SX bound decreases rapidly as the sentence length increases. The behaviour of the lexicalized bounds is considerably better. The cumulative average saving in sentences of length 18 to 26 is 15% for SX, 52% for SXLex, and 76% for SXLex2.



**Fig. 4:** Cumulative average savings with respect to the NULL bound as a function of the sentence length.

Note that the SX and SXlex estimates proposed in Section 3 are optimistic while SXlex2 could not guarantee the optimality of the solution. For the SXlex2 estimate, only 10% of the best parse tree obtained did not match the optimal solution.

Fig. 5 shows the average estimate for outside spans of different lengths. An analogous figure is presented in [12]. Note that the SX estimate increased the difference with regard to the TRUE value as the length of the span increased. The SXlex and SXlex2 estimates were close to the TRUE value, which justifies the savings shown in Fig. 3. Note that the SXlex2 estimate was greater than the TRUE value for all lengths. This is a positive aspect since the prune of the optimal solution was partially mitigated.



**Fig. 5:** Average estimate for different outside span lengths.

## 5 Conclusions

Lexicalized bounds for A\* parsing have been explored in this work. These new bounds consider lexical information in order to get a more realistic estimation of the future cost of the parsing. The use of lexical information has been proved to be useful in the search for the optimal solution. Other bounds that do not guarantee the optimality of the solution have also been proposed because they can be practical under the following conditions: if they produce more savings than the 'optimal' bounds and prune the most probable parsing in only few cases. These new proposed bounds are very useful for parsing with large vocabularies. In our experiments, the 'non-optimal' bound SXLex2 has been very effective.

However, the search for new bounds or the impact of the combination of these bounds is still an interesting task for further study. The behaviour of non-optimal but practical bounds must be studied. In addition, we propose to apply similar ideas to lexicalized parsing and the use of this search method to obtain the n-best parsings.

## Acknowledgement

This work has been supported by the EC (FEDER) and the Spanish MEC under grant TIN2006-15694-CO2-01.

## References

- [1] J. Baker. Trainable grammars for speech recognition. In Klatt and Wolf, editors, *Speech Communications for the 97th Meeting of the Acoustical Society of America*, pages 31–35. Acoustical Society of America, June 1979.
- [2] J. Benedí and J. Sánchez. Estimation of stochastic context-free grammars and their use as language models. *Computer Speech and Language*, 19(3):249–274, 2005.
- [3] S. Bird and E. Loper. Nltk: The natural language toolkit. In *Proceedings of the ACL demonstration session*, pages 214–217, Barcelona, July 2004.
- [4] E. Charniak. A maximum-entropy-inspired parser. In *Proc. of NAACL-2000*, pages 132–139, 2000.
- [5] E. Charniak, S. Goldwater, and M. Johnson. Edge-based best-first chart parsing. In *Sixth Workshop on Very Large Corpora*, pages 127–133, 1998.
- [6] E. Charniak and M. Johnson. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.
- [7] E. Charniak, K. Knight, and K. Yamada. Syntax-based language models for statistical machine translation. In *Proc. of MT Summit IX*, New Orleans, USA, September 2003.
- [8] C. Chelba and F. Jelinek. Structured language modeling. *Computer Speech and Language*, 14:283–332, 2000.
- [9] M. Collins. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637, 2003.
- [10] J. Earley. An efficient context-free parsing algorithm. *Communications of the ACM*, 8(6):451–455, 1970.
- [11] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.
- [12] D. Klein and C. D. Manning. A\* parsing: Fast exact viterbi parse selection. In *Proceedings of HLT-NAACL 03*, 2003.
- [13] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2003.
- [14] M. Lease, E. Charniak, M. Johnson, and D. McClosky. A look at parsing and its applications. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*, 16–20 July 2006.
- [15] M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.
- [16] D. McClosky, E. Charniak, and M. Johnson. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL'06)*, pages 337–344, Sydney, Australia, July 2006. Association for Computational Linguistics.
- [17] B. Roark. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249–276, 2001.
- [18] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, second edition, 2003.
- [19] I. Salvador and J. Benedí. Rna modeling by combining stochastic context-free grammars and n-gram models. *International Journal of Pattern Recognition and Artificial Intelligence*, 16(3):309–315, 2002.
- [20] A. Stolcke. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–200, 1995.
- [21] K. Yamada and K. Knight. A Decoder for Syntax-based Statistical MT. In *Meeting of the Association for Computational Linguistics*, 2002.



# Knowledge Acquisition through Error-Mining

Milagros Fernández Gavilanes      Eric Villemonte de la Clergerie  
Manuel Vilares Ferro

Computer Science Department, University of Vigo  
Campus As Lagoas s/n, 32004 Ourense, Spain

{mfgavilanes,vilares}@uvigo.es

Institut National de Recherche en Informatique et en Automatique  
Domaine de Voluceau, Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France  
Eric.De\_La\_Clergerie@inria.fr

## Abstract

We describe an approach to acquiring a knowledge representation applied on technical documents. We focus on corpus with a strong underlying structure, which allows us to follow a number of precise patterns of presentation. Our goal is to provide effectiveness by reducing both time and cost, as well as subjectivity.

## Keywords

Knowledge acquisition, parsing, term extraction.

## 1 Introduction

A number of proposals exploit parsing in order to permit semantic relations to emerge from text, by combining term extraction and term clustering facilities. The former acquire term candidates from tagged corpora through a shallow grammar. Term clustering groups and classifies these candidates in a graph reflecting the relations between them. So, some authors propose conflating candidates that are variants of each other through a self-indexing procedure [7], while others [5] post-process parse trees so as to emphasize the dependency relationships between the content words.

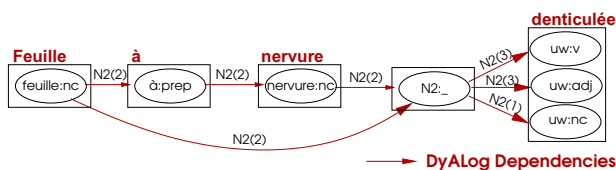


Fig. 1: Parsing shared-forest from DyALog

In our approach, the acquisition phase is performed from a *tree-adjoining grammar* (TAG) [8], generated from a source *meta-grammar* (MG) [4]. The clustering phase is performed on the basis of an iterative algorithm inspired by an error-mining strategy [10].

## 2 The running corpus

We introduce the strategy from a botanic corpus. We concentrate on the "Flore du Cameroun", which is

composed of about forty volumes in French, each one running to about 300 pages, organized as a sequence of sections, each one dedicated to one species and following a systematic structural schema. Sections include a descriptive part enumerating morphological aspects such as color, texture, size or form. This implies the presence of nominal phrases, adjectives and also adverbs to express frequency and intensity, and named entities to denote dimensions.

## 3 The parsing frame

We choose to work with TAGS [8], a grammatical formalism that has given rise to a lot of interest in the modeling of syntax in *natural language processing* (NLP) by combining properties such as the principle of extended domain of locality<sup>1</sup> and a polynomial time complexity, making it appropriate for practical purposes. Using DyALog [3] as parsing frame, we apply a tabular interpretation [1], which implies an efficient treatment of non-deterministic entries.

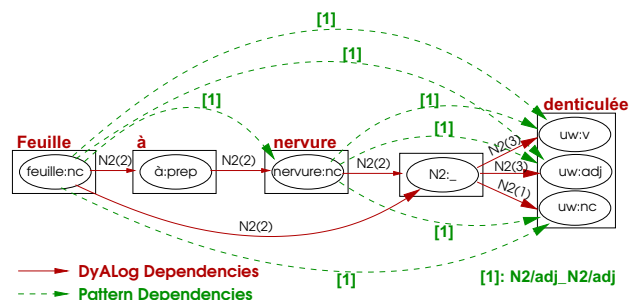


Fig. 2: Graph of dependencies from DyALog

The text is parsed on the basis of the MG concept [4], which permits the introduction of a high degree of abstraction in the design of NLP parsers by involving elementary constraints re-grouped in classes, these themselves inserted in a hierarchy of multiple heritage. This allows descriptions to be progressively refined, which is of particular interest when we are describing complex linguistic behavior. DyALog [3] returns

<sup>1</sup> it allows constraints to be defined at more than one level of the parse as compared to context-free rules and permits the use of atomic features.

total or partial parsing shared-forests from a possibly non-deterministic input on the basis of a TAG of large coverage for French, as we can see in Fig. 1 for the sentence "feuille à nervure denticulée", in future our running example. Arrows represent binary dependencies between words through some syntactic construction.

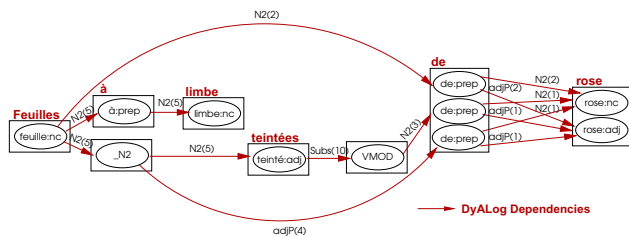


Fig. 3: Another parsing shared-forest from DyALog

From this shared-forest, we can extract a graph of dependencies of the type *governor/governed*, as is shown in Fig. 2 by using dotted-lines going from the governor term to the governed one. The probability of a dependency occurrence is labeled  $P(\text{word1:c1}, [\text{label}], \text{word2:c2})$ , being *word1* the governor word, *c1* the lexical category of the *word1*, *label* the tag of the dependence, *word2* the governed word and finally, *c2* the lexical category of the *word2*. Rectangular shapes represent *clusters*, that is, forms that refer to a position in the input string and all the possible lemmas with their corresponding lexical categories. We call the latter *nodes*, represented by ellipses. Lexical ambiguities correspond to clusters containing nodes with different lemmas, or the same lemma associated to different lexical categories.

### 3.1 Lexical ambiguities

The morpho-syntactic phase consists of a pipeline named Sxpipe [9], that concatenates a number of tasks such as chunking, entity recognition and tagging.

In spite of the strategy considered, tagging often becomes a non-deterministic and even incomplete task, especially in dealing with an encyclopedic corpus with a high degree of unknown words, as is shown in Fig. 1, where the word "denticulée" ("dentate") is initially labeled as unknown word (uw) with three possible associated lexical categories: verb (v), adjective (adj) and common noun (nc). These ambiguities cannot always be solved at lexical level and, in order to avoid prematurely discarding useful interpretations, all the available information should be translated to be considered at parsing time, which introduces an additional factor of syntactic ambiguity.

It is the case of "feuilles à limbe teintées de rose" that we could interpret as "rose's tinted laminar leaves", as "rose-tinted laminar leaves" or as "tinted laminar rose leaves". In the first case, the word "rose" would be a noun related to "teintées" ("tinted"), while in the other ones it is an adjective related to "feuilles" ("leaves"); as is shown in Fig. 3.

### 3.2 Syntactic ambiguities

Parsing in NLP is also an incomplete task because it deals with shallow/partial strategies focused on iden-

tifying dependencies between terms that are close in the text, as in the case of noun sentences involving:

1. Prepositional attachments, as in "feuille à nervure denticulée", that we could locally translate in two ways: "leaf with dentate vein" or "dentate leaf with vein". It becomes here impossible to establish if the word "denticulée" ("dentate") relates to "feuille" ("leaf") or to "nervure" ("vein"), as is shown in Fig. 1.
2. Coordination structures relating properties to a list of nouns, as [9] in "des sépales ovales-aigus, glabres ou éparsément hérissés" ("Sepals oval-pointed, smooth or scattered bristly"), where the property "hérissés" ("bristly") could be attached to "glabres" ("smooth") or to "éparsément" ("scattered").

both of them causing local non-determinism.

## 4 Knowledge acquisition

Once we recover the graph of dependencies, we extract the latent semantics in the document by compiling additional information from the corpus in order to eliminate useless dependencies. So, the lexical ambiguity in Fig. 3 should be decided in favor of the first alternative ("rose's tinted laminar leaves"), because we have the certainty that plants with rose colored leaves do not exist. Given that we are dealing with a corpus on botany, we should confirm that extreme by exploring it in-depth. That is, to solve the ambiguity we just need the information we are looking for; which leads us to consider an iterative learning process to attain our goal.

In similar terms we describe the syntactic disambiguation process for the example in Fig. 1, by selecting "dentate leaf with vein" as the correct interpretation. Also, we should associate "hérissés" ("bristly") to "éparsément" ("scattered") in the sentence "des sépales ovales-aigus, glabres ou éparsément hérissés" ("Sepals oval-pointed, smooth or scattered bristly"). So, term extraction is the starting point to formalize such a task.

### 4.1 Term extraction

We consider two principles. Firstly, the *distributional semantic* model [6] establishing that words whose meaning is close often appear in similar syntactic contexts. Also, we assume that terms shared by these contexts are usually nouns and adjectives [2], which means we have chosen to work with a nominal regime.

Term extraction is organized around the recognition of generic lexical and/or syntactic patterns. On the lexical side, we take advantage of linguistic marking information, focusing on conjunctions "X et X" ("X and Y"), interval definitions of type "de X à Y" ("from X to Y"); or relations involving more explicit physical information such as "en forme de X" ("in form of X") or "de couleur X" ("of color X"). The result serves to acquire simple concepts such as the value for color, form or domain properties; or to detect enumerations that can propagate some of these values.

1.	$P(\text{feuille:uc}, [\hat{a}-1], \text{nervure:uc})_{\text{local}(0)} = \frac{P_c(\text{feuille:uc})_{\text{local}} P_c(\text{nervure:uc})_{\text{local}}}{\sum_{X,Y} P_c(\text{feuille:X})_{\text{local}} P_c(\text{nervure:Y})_{\text{local}}}$
2.	$P(\text{feuille:uc}, [\hat{a}-1], \text{nervure:uc})_{\text{global}(n+1)} = \frac{\sum_{i=1}^n P(\text{feuille:uc}, [\hat{a}-1], \text{nervure:uc})_{\text{local}(i)}}{\#\text{dep}_{\text{local}(n)}}$
3.	$P(\text{feuille:uc}, [\hat{a}-1], \text{nervure:uc})_{\text{local}(n+1)} = \frac{P(\text{feuille:uc}, [\hat{a}-1], \text{nervure:uc})_{\text{local}(n)} P(\text{feuille:uc}, [\hat{a}-1], \text{nervure:uc})_{\text{global}(n+1)}}{\sum_{X,Y} \frac{P(\text{feuille:X}, [\hat{a}-1], \text{nervure:Y})_{\text{local}(n)}}{P(\text{feuille:X}, [\hat{a}-1], \text{nervure:Y})_{\text{global}(n+1)}}}$

**Table 1:** *Extraction of dependencies for "feuille à nervure denticulée"*

Syntactic patterns revolve around the following relations involving nouns and/or adjectives:

- Noun adjective: like "*feuilles elliptiques*" ("elliptical leaves").
- Noun sth noun: like "*fleur avec pétale*" ("flower with petal").
- Noun sth adjective: like "*pétale avec du rouge*" ("petal with red").
- Adjective adjective: like "*ovale elliptique*" ("elliptical oval").
- Adjective sth adjective: like "*rugueux ou poilu*" ("coarse or hairy").

while other ones, especially involving adverbs, will be considered as future work. So, the vocabulary is concentrated around these terms that from now on we call *pivot terms*.

## 4.2 Term clustering

We simplify the graph of dependencies in order to obtain the most pertinent ones. We look for these, which we baptize as *strong dependencies*, around pivot terms.

### 4.2.1 A simple syntactic constraint

We require a simple syntactic constraint establishing that a governed word can only have one governor. So, for example, in the sentence of Fig. 1, "*denticulée*" ("dentate") is governed by "*feuille*" ("leaf"), but also by "*nervure*" ("vein") and, in consequence, we should eliminate one of these dependencies. No other topological restrictions are considered. So, a governor word can have more than one governed one; as in the second interpretation of Fig. 1 ("*dentate leaf with vein*"), where "*feuille*" ("leaf") is the governor for "*nervure*" ("vein") and "*denticulée*" ("dentate"). Also, one word could be governor and governed at the same time, as is the case of "*nervure*" ("vein"), that is the governor for "*denticulée*" ("dentate"), but is also governed by "*feuille*" ("leaf").

Given that our graph of dependencies is a parse shared-forest, we have chosen to work with a term clustering technique that is inspired by an error-mining proposal originally designed to identify missing and

erroneous information in parsing systems [10]. Intuitively, we focus on detecting and later eliminating those dependencies that are found to be less probable in sentences including terms with a low frequency.

### 4.2.2 The iterative process

We combine two complementary iterative processes. For a given iteration, the first one computes the probability of each dependency; taking as starting point the statistical data provided by the original error-mining strategy and related to the lexical category of the pivot terms. The second process computes, from the former one, the most probable semantic class to be assigned to terms involved in the dependency. So, in each iteration, we look for both semantic and syntactic disambiguation, each one profiting from the other. A fixed point assures the convergence of the strategy [10].

We illustrate term clustering on our running example in Fig. 2, focusing on the dependency labeled  $[\hat{a}-1]$  relating "*feuille*" ("leaf") and "*nervure*" ("vein"); talking without distinction about weight, probability or preference to refer the same statistical concept. So, from Table 1, we have that:

1. To begin with, we compute the local probability of the dependency in each sentence, which depends on the weight of each word, this in turn depending on the word having the correct lexical category. To start the process, first category assumptions, denoted by  $P_c$ , are provided by the error-mining algorithm [10]. We take also into account the initial probability for the dependency considered,  $P_{\text{dep ini}}$ , a simple ratio on all possible derivations involving the lexical categories concerned. The normalization is given by the preferences for the possible lexical categories involving each one of the terms considered and here represented by variables  $X$  and  $Y$ .
2. We re-introduce the local probabilities into the whole corpus locally in the sentences, in order to re-compute the weights of all possible dependencies, estimating then globally the most probable ones. The normalization is given by the number of dependencies connecting the terms considered,  $\#\text{dep}$ .
3. The local value in the new iteration should take into account both the global preferences and the

4.		$P(\text{feuille:uc:org}, [\hat{a}-1], \text{nervure:uc:org})_{\text{local}(0)} = \frac{\frac{P(\text{feuille:uc}, [\hat{a}-1], \text{nervure:uc})_{\text{local}(0)}}{P(\text{feuille:uc:org})_{\text{local}(0)}}}{\sum_{X,Y} P(\text{feuille:uc:X})_{\text{local}(0)} P(\text{nervure:uc:Y})_{\text{local}(0)}}$
5.1		$P(\text{feuille:uc:org}, [\hat{a}-1], X)_{\text{global}(n+1)} = \frac{\sum_X P(\text{feuille:uc:org}, [\hat{a}-1], X)_{\text{local}(n)}}{\#\text{dep}_{\text{local}(n)}(\text{feuille})}$
5.2		$P(Y, [\hat{a}-1], \text{nervure:uc:org})_{\text{global}(n+1)} = \frac{\sum_Y P(Y, [\hat{a}-1], \text{nervure:uc:org})_{\text{local}(n)}}{\#\text{dep}_{\text{local}(n)}(\text{nervure})}$
5.3		$P(\text{feuille:uc:org}, [\hat{a}-1], \text{nervure:uc:org})_{\text{global}(n+1)} = \frac{P(\text{feuille:uc:org}, [\hat{a}-1], X)_{\text{global}(n+1)}}{P(Y, [\hat{a}-1], \text{nervure:uc:org})_{\text{global}(n+1)}}$
6.		$P(\text{feuille:uc:org}, [\hat{a}-1], \text{nervure:uc:org})_{\text{local}(n+1)} = \frac{\frac{P(\text{feuille:uc:org}, [\hat{a}-1], \text{nervure:uc:org})_{\text{local}(n)}}{P(\text{feuille:uc:org}, [\hat{a}-1], \text{nervure:uc:org})_{\text{global}(n+1)}}}{\sum_{X,Y} \frac{P(\text{feuille:uc:X}, [\hat{a}-1], \text{nervure:uc:Y})_{\text{local}(n)}}{P(\text{feuille:uc:X}, [\hat{a}-1], \text{nervure:uc:Y})_{\text{global}(n+1)}}}$

**Table 2:** *Extraction of classes for "feuille à nervure denticulée"*

local injection of these preferences in the sentences, re-inforcing the local probabilities. The normalization is given by previous local and global weights for the dependency involving all possible lexical categories associated to each one of the terms considered, and here represented by variables X and Y.

In dealing with semantic class assignment, the sequence of steps is shown in Table 2, illustrating the computation of the probability that "feuille" ("leaf") and "nervure" ("vein") are both organs, taking again the dependency labeled  $[\hat{a}-1]$  in Fig. 2:

4. In each sentence, we compute the local probability of this dependency if "feuille" ("leaf") and "nervure" ("vein") are both organs (org). We start from the local weight computed in Table 1, and also the initial preferences the terms involved corresponding to the classes considered<sup>2</sup>. The normalization is given by the probabilities for the possible classes involving each one of the terms considered, without specifying any particular class here represented by variables X and Y.
5. We calculate this preference at global level, by re-introducing it to the whole corpus locally in the sentences in order to re-compute the weights of all the possible classes in the sentence. We first compute the probability in the whole corpus (5.1 and 5.2) for each term and semantic class, disregarding the right and left context, represented by variables X and Y respectively. The probability (5.3) is a combination of the two previous ones.
6. After each iteration, we re-inject the previous global weight to obtain a new local one, by re-inforcing the local probabilities. The normalization is done by the addition of the preferences corresponding to the terms and classes involved in the dependency, for all the possible semantic classes considered.

<sup>2</sup> this is fixed by the user, in the case of the term being in a list associated to that class. Otherwise, this probability is obtained as a ratio of the total number of classes considered.

## 5 Experimental results

We describe some preliminary tests, using the running corpus as guideline. We consider two different quality references. The former, the number of learned elements. Secondly, the computational efficiency on a standard platform. Whatever is the case, these tests are performed in function of the number of iteration learning passes, once we have fixed three thresholds:

- First, the number of the occurrences of a term, that is the number of the governor/governed nodes in the graph of dependencies. This allows us to estimate the validity of the testing frame.
- Second, the percentage for success, showing possible existing relationships between computational loading and efficiency.
- Third, the probability of a dependency being non deterministic, looking to illustrate the impact of ambiguities on the learning task.

that we illustrate in Figs. 4, 5 and 6. As starting point, we take the information compiled for 6 organs, 10 properties and 10 markers.

More in detail, Fig. 4 reflects the execution time for the knowledge acquisition process, considering terms that appear more than 18 times, with a success index of over 90%. We consider here two tests, one related to dependencies whose probability is 1, and the other one focused on dependencies with a probability of over 0.2. The results seem to indicate a linear behavior in the first case and a polynomial complexity in the second one. Intuitively, this conclusion was expected given that knowledge acquisition should be more efficient in dealing with dependencies that are totally guaranteed.

In the same way, the number of learned elements seems to be greater when dealing with high confidence dependencies, as shown in Fig. 5, than when working with the weaker ones included in Fig. 6. Another interesting point is the behavior observed for the different classes learned in Figs. 5 and 6. So, properties, such as

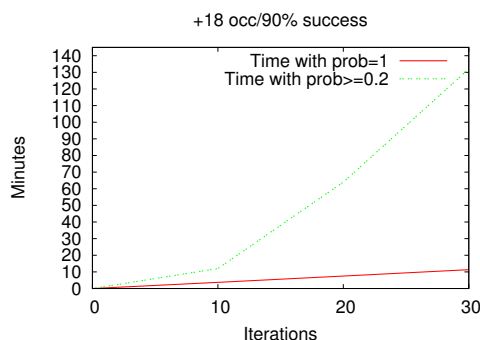


Fig. 4: Time complexity for the learning process

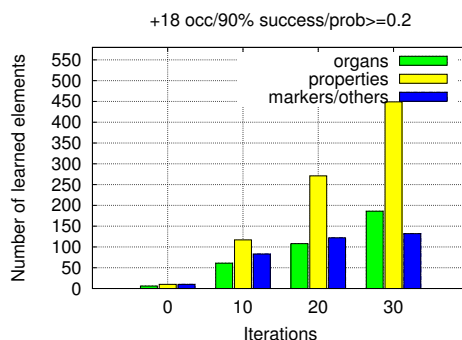


Fig. 6: Learning dependencies with poor probability

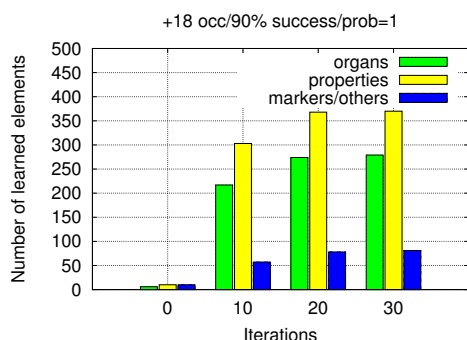


Fig. 5: Learning dependencies with high probability

form or color, are in both cases the classes on which knowledge acquisition runs with greater certainty.

This is also the case with the classes on which term extraction was already defined at lexical level, involving extremely precise linguistic information, as is the case of *organs*. In consequence, knowledge acquisition on these terms is relatively independent of the iterative process and, in particular, of the level of probability considered for dependencies. This is underlined by the asymptotic behavior, when the number of iterations grows and the process converges, showing a similar behavior in both cases.

In the same sense, the asymptotic behavior observed in Figs. 5 and 6 seems to indicate that *organs* reach a high degree of recognition, depending on the probability of the dependencies considered. As we have seen in our running examples, this is justified by the fact that term extraction on these classes cannot be defined at lexical level, but often relies on the disambiguation of non-deterministic syntactic structures, which concerns the iterative knowledge acquisition process described.

Other marginal categories less involved in term extraction due to the absence of relevant lexical and/or syntactic information, show a closed behavior regardless of the probability considered for dependencies in Figs. 5 and 6. This explains the poor evolution on the number of elements learned in comparison with the results previously obtained on *properties* and *organs*.

## 6 Conclusions

We have introduced knowledge acquisition with a maximum degree of unsupervised tasks. The identification of semantic classes is approached from the detection of similar syntactic contexts around pivot terms. Existing relations between semantic classes are approached from the lexical and/or syntactic patterns connecting them, by using an error-mining technique.

## Acknowledgments

This research was partially supported by the Spanish Government under project TIN2004-07246-C03-01, and the Autonomous Government of Galicia under project PGIDIT05PXIC30501PN and the Network for Language Processing and Information Retrieval.

## References

- [1] M. Alonso, D. Cabrero, E. de la Clergerie, and M. Vilares. Tabular algorithms for TAG parsing. In *Proc. of EACL'99*, pages 150–157, 1999.
- [2] J. Bouaud, B. Bachimont, J. Charlet, and P. Zweigenbaum. Methodological principles for structuring an ontology, 1995.
- [3] E. de la Clergerie. Dyalog: a tabular logic programming based environment for nlp. In *Proc. of 2nd Int. Workshop on Constraint Solving and Language Processing*, 2005.
- [4] E. de la Clergerie. From metagrammars to factorized TAG/TIG parsers. In *Proc. of IWPT'05*, pages 190–191, 2005.
- [5] B. Habert, E. Naulleau, and A. Nazarenko. Symbolic word clustering for medium-size corpora. In *COLING*, pages 490–495, 1996.
- [6] Z. Harris. *Mathematical Structures of Languages*. John Wiley & Sons, New York, U.S.A., 1968.
- [7] C. Jacquemin and D. Bourigault. Term extraction and automatic indexing. *Handbook of Computational Linguistics*, pages 599–615, 1999.
- [8] A. Joshi. An introduction to Tree Adjoining Grammar. In A. Manaster-Ramer, editor, *Mathematics of Language*, pages 87–114. John Benjamins Company, 1987.
- [9] G. Rousse and E. de la Clergerie. Analyse automatique de documents botaniques: le projet biotim. In *Proc. TIA'95*, pages 95–104, 2005.
- [10] B. Sagot and E. de la Clergerie. Error mining in parsing results. In *Proc. of the 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 329–336, 2006.



# A two-tiered approach to detecting English article usage: an application in scientific paper writing tools

Luiz Genoves Jr<sup>1</sup>, Richard Lizotte<sup>2</sup>, Ethel Schuster<sup>2</sup>, Carmen Dayrell<sup>1</sup> and Sandra Aluísio<sup>1</sup>

<sup>1</sup>University of São Paulo, ICMC-NILC, CP 668,13560-970, São Carlos/SP, Brazil

<sup>2</sup>Northern Essex Community College, USA

genoves@icmc.usp.br, rlizotte@necc.mass.edu, eschuster@necc.mass.edu, c\_dayrell@yahoo.com, sandra@icmc.usp.br

## Abstract

Most studies related to automatic correction of article errors have adopted a three-class approach, which decides between the use of the, a/an or zero article. This approach does not take into consideration that users may have different difficulties depending on their levels. In the specific case of Brazilian writers, the most difficult task is to decide whether or not an article is required in English, rather than which article to use. If users (writers?) are informed that the article is needed, they will most likely be able to restore it themselves. In this paper, we propose a two-tiered approach to automatically detect English article usage errors in scientific papers written in English by Brazilian speakers of Portuguese. Our approach is composed of two linked tasks viewed as binary classification choices: deciding whether or not a given noun phrase (NP) should contain an article and restoring missing articles. If there is an article where none is required, the system suggests the user remove it. Otherwise, it means that either a definite (the) or an indefinite article (a/an) is needed and a second classifier is applied to restore the missing article. We have used three WEKA algorithms (J48, NB, and JRIP) to perform these two tasks. The classifiers have been trained using a 10-fold cross-validation test on a corpus of published abstracts from specific research areas. The main rationale behind this decision is based on the assumption that scientific texts are made up of specific terminology and expressions, which are likely to affect the performance of taggers and parsers.

## Keywords

English article usage detection, academic writing tools, corpus-based language processing, English as a Foreign Language.

## 1. Introduction

Scientific writing poses a heavy burden on non-native English writers. They have to deal with both the natural complexity of the writing process and the conventions of scientific writing regarding the use of appropriate schematic structures and conventional expressions common to academic and scientific discourse. These problems are even more acute when the writer is a novice researcher and does not have full command of English grammar and usage at the sentence level. This paper builds on previous studies related to several types of writing tools [1], [2] and [6]. These tools are domain dependent since they work with a database of authentic papers to capture idiosyncrasies of the discourse community and to provide novice students with material

related to the one in which they need to write. The data used here were collected from five courses on academic writing offered to graduate students at two universities in Brazil. In these courses, these tools were used to assist the Brazilian researchers in producing scientific papers. We are currently working on the implementation of a rubric to analyze abstracts of scientific papers [19], [20]. When fully automated, this rubric should enable a writing tool to detect errors and offer suggestions for improvements. Our genre-based rubric includes seven dimensions: 1) organization and development of the text, 2) balance among the components of the schematic structure of the scientific genre, 3) coherence among components, 4) cohesive markers, 5) technical errors, 6) style and 7) presence of substantive material in certain components of the abstract instead of indicative content. The dimensions have two scale values each—high and low—which helps both annotate dimensions and achieve high consistency among the human judges. Dimensions 1 and 2 have already been implemented (<http://www.nilc.icmc.usp.br/azea-web>). The WEKA's SMO classifier was developed to cope with these dimensions and achieved 80.4% accuracy (kappa 0.73). The classifier automatically annotates sentences from abstracts with components of the abstract schematic structure used by Feltrim et al. (2005). Although this rubric is focused on the abstract, dimensions 5 and 6 can be used to other paper sections. Moreover, the evaluation with rubrics similar to the one developed can be replicated to other paper sections.

In this paper, we focus on dimension 5, related to technical errors. In order to explore the nature of the errors made by Brazilian writers and the question of how to help them correct these errors, an error analysis was conducted on 114 abstracts from five courses on academic writing offered to graduate students from the disciplines of pharmacology, chemistry, biology/genetics, physics and computer science at two universities in Brazil. The analysis of these abstracts has revealed that among article usage errors, the vast majority refer to either an article that was missing where it would be required in English or an article used where none was required. Cases in which users mix up the use of definite and indefinite articles are rather scarce. Therefore, we propose a two-tiered approach to automatically detect English article errors in scientific

papers written in English by Brazilian Portuguese speakers. The error analysis is presented in Section 2 and, as we shall see in Section 4, the abstracts from two of these groups (pharmacology and physics) make up our test corpus. In Section 3, we review related work. Section 5 shows both the results of the cross-validation tests using several WEKA algorithms as well as the results of the application of our two binary classifiers to a corpus of abstracts written by students and corrected by a native speaker of English.

## 2. Technical Error Analysis

All abstracts were tagged by a native speaker of English for 23 categories of error dealing with lexical use, syntactic accuracy, and mechanical correctness. Categories for syntactic accuracy included those dealing with article use (an article that was missing where one was required in English (ART-), an article used where none was required in English (ART+), one article needed to be substituted for another (ART)). In our data, six error categories stood out: in the following order: (1) the correct use of a word to express the intended meaning (WU), (2) the correct use of lexical items in idioms and common collocations (WUCol), (3) ART-, (4) ART+, (5) punctuation (P), (6) spelling (SP)) accounting for 66% of all error tokens where each category accounted for 5% or more of the total errors.

We believed at first that errors in article use would not play a significant role in our analysis. However, to our surprise, after WU, the most common category of error was in article use. Fully 19% of all errors were in misuse of the article of various kinds. 5% involved an extraneous article, ART+; 13% involved a missing article, ART- and 1% involved an incorrect article, ART. Surprisingly, the percentage of errors in article use was significantly greater than all verb use errors combined (9.5%), or all word order errors combined (7%), or all singular/plural errors combined (10%). The vast majority (82%) of abstract writers had at least one article error in their abstracts. It was clear from the data that a significant improvement in both the syntax and the comprehensibility of the abstracts could be achieved if either the identification and/or correction of article errors could be automated or heuristics could be provided to the abstract writers to help them improve their use of the articles.

## 3. Related work

Most research on the automatic detection of English article errors carried out so far has followed two basic approaches. One of these approaches is to provide the system with a set of heuristic rules, meaning that it resorts to additional knowledge sources in order to perform the task. By contrast, other studies have opted for generating rules automatically by applying methods to retrieve knowledge from a corpus which, as we shall see below, is the approach adopted by the present study. Na-Rae et al. [16] provide a good overview of various works within each perspective.

Bond et al. [3] and Heine [8] are good examples of the former. Bond et al. [3] generated articles in the translations of Japanese NPs into English and reached 77% accuracy. Heine [8] classified Japanese NPs as either definite or indefinite, reaching 98.9% accuracy in the restricted domain of appointment scheduling. As for machine learning systems, it is worth mentioning the pioneering work by Knight and Chander [11], which focused on the distinction between the use of the definite and the indefinite articles in English. Their overall accuracy was 78%. Minnen et al. [13] reached 82.6% accuracy when choosing between *the*, *a/an* and *zero* article. Izumi et al. [9] distinguishes between missing articles and erroneous use of articles in spoken English produced by Japanese speakers, reporting 30% recall and 50% precision. Precision rose to 80% by adding corrected sentences and artificially made errors to the training corpus. Izumi et al. [10] took a step further, and in addition to the two types of error considered in Izumi et al. (2003), they also tried to detect the inclusion of the article where it was not needed. Izumi et al. [10] reached 35% recall and 48% precision when the learner corpus was used in its original form, and 43% recall and 68% precision when including corrected sentences and artificially made errors in the corpus.

More closely related to the present study is Na-Rae et al. [16], which proposes a set of 12 features to detect three possible types of article usage, namely, *a/an*, *the* and *zero* article. With the exception of the head's countability, all features were established on the basis of the local context, that is to say, two words before the beginning of the NP (pre-pre-NP and pre-NP), the words within the NP, and one word after the NP (post-NP). The head's countability was assigned on the basis of frequency measures extracted from the corpus. Thus, if the plural form occurred in less than 3% of all instances of the noun, it was categorized as 'uncountable'. 'Pluralia tantum' was assigned to plural forms which account for more than 95% of all instances. If no occurrences of the noun were found in the corpus, it was categorized as 'unknown'. Nouns that did not fit in any of these categories were regarded as 'countable'. Na-Rae Han et al. [16] used a maximum entropy model and their classifier was trained on approximately 8 million NPs which were extracted from a corpus of 721 text files (approximately 31.5 million words). This corpus consisted of textbooks from the 10th to 12th grade reading levels selected from The Meta Metrics corpus. They reached 83% accuracy for published text and 85% agreement ( $\kappa = 0.48$ ) between the classifier and human annotators.

It is also worth mentioning recent contributions by Lee [12] and Nagata et al. [14], [15]. Lee's [12] primary aim was to restore missing articles. He proposed a set of 15 syntactic and semantic features which have been established on the basis of two different sources: the Collins parser and WordNet Version 2.0, and he used the log-linear model to

perform the task. For training, he used sections 00 to 21 of Penn Treebank-3, from which 260,000 NPs were extracted.

Nagata et al. [14] looked at three possible types of article errors in English texts produced by Japanese speakers: an article that was missing where it would be required, an article used where none was required in English, and wrong choice. The overall accuracy was 60%. Nagata et al. [15] built on this previous model and also took into consideration the preposition found in the surrounding context. Their best result was 80% accuracy.

An important point to stress here is that this paper differs from all previous studies in that the focus is on scientific texts. To the best of our knowledge, this type of study has not been done before. For all studies mentioned above, training and testing was based on corpora of newspaper texts and general written or spoken English. It should also be mentioned that, unlike most of the research mentioned above which focuses on English texts produced by native speakers of languages that do not have articles, Japanese in particular, our focus is on English abstracts which have been produced by native speakers of Brazilian Portuguese, which has articles.

## 4. Our approach

### 4.1 Corpora

Our training corpus is composed of 723 scientific abstracts from the most important journals of the disciplines of pharmacology (354) and physics (369), totaling 115,913 words. It contains 4886 sentences and each one has 6.54 NPs on average (standard deviation of 3.03).

Following the same procedures as adopted by Na-Rae et al [16], each abstract was tagged by the MXPOST (<http://www.cogsci.ed.ac.uk/~jamesc/taggers/MXPOST.htm>). A total of 31,960 NPs were identified by the chunker provided by Thomas Morton (<http://opennlp.sourceforge.net>). Each NP has 2.40 words on average (standard deviation of 1.25).

The most common article in the corpus was the *zero* article (65.7% of all NPs) followed by *the* (25.3%), and *a/an* (8.97%), presenting the same appearance order regarding quantity as that reported by Na-Rae et al [16].

A corpus of 78 student abstracts from the disciplines of pharmacology and physics was used to evaluate the performance of the two binary classifiers. This corpus contained 570 sentences (3585 NPs). Each sentence has 6.17 NPs on average (standard deviation of 3.43) and each NP has 2.31 words on average (standard deviation of 1.19). All abstracts were corrected by a native speaker of English.

### 4.2 Features

All features are extracted by considering the local context, that is, the NP and the words surrounding it. Local context comprises three main positions: 1) **NP tokens**: These refer to the first token inside the NP and the span of up to four tokens on the right and four tokens on the left of the head. If

the first token is an article, we consider the token in the immediate position on its right as the initial token; 2) **Head Noun**: the head of the noun phrase. It is specified using rules in Collins [5]; 3) **Outer tokens**: the tokens surrounding the NP, that is, two tokens before the NP (pre-pre-NP and pre-NP) and one after it (post-NP).

Six sets of features are proposed which, when subdivided, contain 39 features:

**Article**: It may be *the*, *a* (covering both *a* and *an*), or *null* (no article). This is our class feature.

**POS Tags**: the part of speech tag of all the 13 tokens regarded as local context.

**Word**: this feature checks whether the token under analysis is found within a list of the 35 most frequent words of the corpus, excluding articles. The threshold of 35 has been determined experimentally, since tests with more than 35 words have not shown any significant improvement in results. These 35 words are used as the possible values of this feature. Thus, whenever there is a match, the feature value is set to the word itself. Otherwise, the value is set to "unknown". This feature is applied to all the 13 tokens of the local context.

**Formulaic**: this feature is applied to the head noun and to the NP tokens as long as the token occurs at least 5 times in the corpus. If so, it checks whether it has co-occurred with one of the following categories: *the*, *a/an*, *zero* article. Tokens co-occurring with more than one category are assigned the "unknown" value. Otherwise, it receives the value of the respective category. Words with a frequency lower than 5 are discarded.

**Countability (head noun)**: This follows the procedures suggested by Na-Rae et al. [16] for deciding whether the head noun is countable or uncountable in the corpus. However, given that our corpus is much smaller than the one used by Na-Rae et al. [16], we have used the frequency list of our corpus as well as the BNC word frequency list.

**Discourse (head noun)**: This feature has two possible values: "new", if the head noun is used for the first time in the abstract under analysis, and "seen", if the head noun has appeared in one of the preceding sentences of the abstract in question.

### 4.3 Models

Although most studies on the detection of English article errors have adopted the maximum entropy model (see, for instance, [9], [10] and [16]), we opted for the WEKA environment [18]. This is mainly because it allows us to test different types of machine learning algorithms using the same input format. The following algorithms are used: (i) J48: the WEKA implementation of the C4.5 decision tree learning algorithm [17]; (ii) Naive Bayes (NB): Bayesian models are largely used in text mining problems; (iii) JRIP: a rule learning algorithm, which is an implementation of RIPPER [4] and generates few rules and hence can be



manually evaluated. To evaluate the induced classifiers, we used a 10-fold cross-validation test. For each classifier, we show its total accuracy and the Kappa statistic's value as well as the precision and recall for each class. Our baseline (BL) is 65.7%, which is the proportion of instances of the most frequent class, that is, the "zero article."

## 5. Experiments and Discussion

Before showing the results of our two binary classifiers, we present the overall accuracy of the three models mentioned in Section 4.3 when the decision involved the three types of article usage: *a/an*, *the*, zero article (Table 1). As can be seen in Table 1, the best result in this case was 77.4% by using J48. However, it is interesting to point out that, if the classifier is trained by discipline, accuracy rises to 81.3% on the pharmacology corpus and drops to 74.5% in the physics corpus. Our approach uses two independent binary classifiers. The first classifier (*HasArticle*) predicts whether a NP requires an article, be it definite or indefinite. In case of a positive answer, a second classifier (*DetermineArticle*) is used to determine whether a definite or an indefinite article is needed.

**Table 1. Overall accuracy (Acc) and Kappa value (K) for the three models when deciding between 3 classes. Precision (P) and recall (R) for each class are also presented**

Models	Acc	K	null		the		a/an	
			P	R	P	R	P	R
J48	<b>77.4</b>	<b>0.50</b>	83	92	64	55	49	28
NB	70.2	0.44	<b>87</b>	75	49	65	44	44
JRIP	73.8	0.34	74	97	<b>69</b>	35	<b>62</b>	10
BL	65.7	0	65	100	0	0	0	0

The WEKA environment enabled us to test these two binary classifiers with the three different algorithms (Table 2).

**Table 2. Accuracy and Kappa for the binary classifiers**

Models	<i>HasArticle</i>		<i>DetermineArticle</i>	
	Acc	K	Acc	K
J48	<b>83.71</b>	<b>0.63</b>	78.09	0.31
NB	78.57	0.49	<b>78.36</b>	<b>0.39</b>
JRIP	82.47	0.60	75.87	0.21

Table 3 shows the individual contribution of each set of features to the two binary classifiers. *PoS Tags* was the most predictive set of features for both (column *Only*). It is important to point out that head was the set of features which would inflict the greatest loss to the classifier (column *Without*). The full task classifier, which combines the *HasArticle* (J48) with *DetermineArticle* (Naive Bayes)

achieved 77.48% overall accuracy and a Kappa of 0.536, which is very similar to the results achieved by the ternary classifier.

**Table 3. Accuracy of sets of features**

Feature type	<i>HasArticle</i>		<i>DetArticle</i>	
	Only	Without	Only	Without
PoS Tags	<b>76.0</b>	75.4	<b>76.4</b>	75.1
Word	71.6	83.6	75.0	76.9
Formulaic	67.2	82.8	73.9	77.8
Discourse	65.7	82.2	73.8	77.9
Countability	65.7	80.4	73.8	77.7
Head	75.7	<b>74.5</b>	73.9	<b>73.8</b>

However, in the specific case of Brazilian writers, the binary approach has the additional advantage of focusing on their particular difficulty, that is, to decide whether or not an article is required in English. In doing so, *HasArticle* reached 83.7% accuracy (see Table 2). The Kappa value was high (0.63). If the article is needed, the users are most likely able to restore it themselves. *DetermineArticle* is only needed when the user opts for restoring the article automatically. To evaluate the performance of our two binary classifiers with student data, we used the corpus of 78 English abstracts from pharmacology and physics. Table 4 shows the results of these tests. As in the case of the training corpus, accuracy and Kappa rose when the two classifiers were used separately. J48 achieved 81% accuracy (Kappa = 0.56) for the task of deciding whether or not English NPs require an article (*HasArticle*). As for the second classifier (*DetermineArticle*), Naive Bayes was the best classifier (Kappa=0.41), although the accuracy of J48 was slightly higher (81%), but with a lower Kappa (0.35).

**Table 4. Accuracy and Kappa for the binary classifiers when applied to the students' abstracts**

	<i>HasArticle</i>		<i>DetermineArticle</i>	
	Acc	K	Acc	K
J48	<b>81%</b>	<b>0.56</b>	81%	0.35
NB	73%	0.43	<b>80%</b>	<b>0.41</b>
JRIP	81%	0.54	79%	0.26
BL	68%	0	78%	0

With the two binary classifiers combined, the overall accuracy was slightly reduced to 76.3% (Kappa=0.51). The statistics by classes are shown in Table 5. As explained earlier, all student abstracts were corrected by an English native speaker and hence all errors were marked manually. A total of 194 article usage errors were manually identified in the corpus. When the *HasArticle* classifier is applied to these 194 NPs, precision drops to 53% (Kappa=0). The class ART reaches 82% precision and 51% recall; the class NONE shows 26% precision and 60% recall. Most errors occur in NPs that require an article (81%, 74 of the 91 errors), which were not detected by the classifier. Although

the error set is very limited in size, this figure seems to indicate real errors are more difficult to detect.

**Table 5. Combining the two binary classifiers**

	Class	Precision	Recall
<i>HasArticle</i> (J48)	ART	71.7%	68.5%
	NONE	85.6%	87.4%
<i>DetArticle</i> (NB)	DEF	86.5%	89.1%
	INDEF	56.1%	50.2%

## 6. Final Remarks

In this paper, we proposed a two-tiered approach to automatically detect English article usage errors in scientific papers. The main advantage of this approach is its high accuracy when deciding whether or not a given NP should contain an article, which is the most frequent error of article usage made by Brazilians in English. It would therefore be very useful for improving students' writing. When tested on our training corpus, the classifier reached 83.7% accuracy. However, this performance rate fell to 53% for tests with our student abstracts. In future work, we intend to focus on improving the precision of the classifier for restoring articles by including features related to discourse and enlarging the corpus.

## 7. References

- [1] S. M. Aluísio; O.N. Oliveira Jr. 1995. A Case-Based Approach for Developing Writing Tools Aimed at Non-native English Users. In *Case-Based Reasoning Research and Development, Proceedings of the 1st International Conference on Case-Based Reasoning*. Lecture Notes in Computer Science 1010, p. 121-132.
- [2] S. M. Aluísio; I. Barcelos; J. Sampaio; O. N. Oliveira Jr. 2001. How to Learn the Many Unwritten 'Rules of the Game' of the Academic Discourse: A Hybrid Approach Based on Critiques and Cases to Support Scientific Writing. *IEEE International Conference on Advanced Learning Technologies*. Madison, Wisconsin, 1, p. 257-260.
- [3] F. Bond, K. Ogura and T. Kawaoka. 1995. Noun phrase reference in Japanese-to-English machine translation. *Proceedings of the 6th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI'95)*, p. 1-14.
- [4] W. W. Cohen. 1995. Fast effective rule induction. Armand Prieditis, Stuart J. Russell (Eds.): *Machine Learning*, Proceedings of the 12th International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, p. 115-123.
- [5] M. Collins. 1999. Head-driven statistical models for natural language parsing. Ph.D. Thesis, University of Pennsylvania, Philadelphia.
- [6] V. Feltrim; S. Teufel; M. G. V. Nunes; S. M. Aluísio. 2005. Argumentative Zoning Applied to Critiquing Novices' Scientific Abstracts. *Computing Attitude and Affect in Text: Theory and Applications*. 1st ed. Dordrecht, The Netherlands: Springer, 1: 159-170.
- [7] N. Fontana; S. M. (Caldeira), Aluísio; M.C.F. de Oliveira, and O.N. Oliveira Jr. 1993. Computer Assisted Writing Applications to English as a Foreign Language. *CALL*, 6 (2):145-161.
- [8] J. Heine. 1998. Definiteness predictions for Japanese noun phrases. 36th Annual Meeting of ACL and 17th International Conference on Computational Linguistics:COLING/ACL-98, p.519-525. Montreal, Canada.
- [9] F. Izumi, K. Uchimoto, T. Saiga, T. Supnithi and H. Isahara. 2003. Automatic error detection in the Japanese learners English Spoken data. *ACL-2003 Interactive Posters and Demonstrations: Companion Volume to the 41st Annual Meeting of ACL*. Sapporo, Japan.
- [10] F. Izumi, K. Uchimoto and H. Isahara. 2004. SST speech corpus of Japanese learners' English and automatic detection of learner's errors. *ICAME Journal* 28:31-48. Bergen, Norway.
- [11] K. Knight and I. Chander. 1994. Automated postediting of documents. *Proceedings of the 12th National Conference on Artificial Intelligence*. AAAI Press, Menlo Park, CA.
- [12] J. Lee. 2004. Automatic Article Restoration. *Proceedings of the Human Language Technology Conference of the North American Chapter of ACL*, p. 31-36. Boston, MA.
- [13] G. Minnen, F. Bond, and A. Copestake. 2000. Memory-based learning for article generation. *Proceedings of the 4th Workshop on Computational Natural Language Learning*, p. 43-48. Lisbon, Portugal.
- [14] R. Nagata, T. Iguchi, K. Wakidera, F. Masui, and A. Kawai. 2005. Recognizing Article Errors in the Writing of Japanese Learners of English. *System and Computers in Japan*, 36(7):54-63.
- [15] R. Nagata, T. Iguchi, K. Wakidera, F. Masui, A. Kawai and I. Naoki. 2006. Recognizing article errors using prepositional information. *System and Computers in Japan*, 37(12):17-26.
- [16] N. Han, M. Chodorow, and C. Leacock. 2006. Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12(2):115-129.
- [17] J. R. Quinlan. 1993. *C4.5: programs for machine learning*. São Mateo: Morgan Kaufmann.
- [18] I. H. Witten and E. Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. 2nd Edition, Morgan Kaufmann, San Francisco.
- [19] S. M. Aluísio; E. Schuster; V. D. Feltrim; A. P. Jr; O. N. Oliveira Jr. 2005. Evaluating scientific abstracts with a genre-specific rubric. In: *Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED 2005)*. Amsterdam: v.1, p. 738-740.
- [20] E. Schuster; S. Aluísio; V. Feltrim; A. Pessoa Jr.; O. N. Oliveira Jr. 2005. Enhancing the writing of scientific abstracts: a two-phased process using software tools and human evaluation. In: *Encontro Nacional de Inteligência Artificial (ENIA)*, 2005. v.1, p. 962-97

# Initial Requests in Institutional Calls: Corpus Study

Olga Gerassimenko

Riina Kasterpalu

Mare Koit

Andriela Rääbis

Krista Strandson

University of Tartu

2 J. Liivi Str.

Tartu, Estonia 50409

*name.surname@ut.ee*

## Abstract

Customers' initial requests are analysed in Estonian institutional calls. Direct and indirect requests (represented, respectively, by directives and questions) are used by customers differently in different dialogue types. The study shows that (1) indirect requests express both the speaker's uncertainty in its fulfillment and/or politeness, (2) certain linguistic patterns are used for representing requests. The differences in use of requests in different dialogue types should be taken into account when developing a dialogue system which interacts with a user in natural language and follows rules of human-human communication.

## Keywords

Dialogue system, dialogue corpus, dialogue acts, direct and indirect requests, linguistic patterns, automatic annotation.

## 1. Introduction

Many dialogue systems (DS) are built which provide information in a specified domain and interact with a user in spoken natural language [8, 9, 10]. Our goal is to build a DS which interacts with a user in Estonian following norms and rules of human-human communication. For that reason, we analyse actual human-human dialogues in order to find out how people communicate one with another, how they express their intentions and understand each other.

Requests can be formulated in various ways, e.g. *please tell me your address*, *could you give me your address*, *your address*, etc. These formulations differ by politeness while the content remains the same in every case. If people are asking the same things in different ways then one should be able to explain their expectations in every case. If we can find out the conditions which determine the form of a request then such a model can be implemented in a DS. However, an addressee can react differently to requests formed in different ways.

In this paper, we consider initial requests of customers who are calling an institution. The initial requests are the most important because they express the intention of a customer and will determine the course of the conversation.

The paper is organized as follows. In section 2 we give an overview of our empirical material. Section 3 considers the use of direct and indirect requests in the beginning of calls and possible continuations of a call. In section 4 we represent some linguistic cues and patterns found in requests. Section 5 discusses the use of directives and questions for expressing of initial requests in institutional calls. Finally, we will make conclusions.

## 2. Data

Our study is based on the Estonian dialogue corpus EDiC<sup>1</sup>. The corpus contains over 900 human-human spoken dialogues, among them over 800 phone calls. Dialogue acts are annotated in dialogue transcripts using a typology which originates from the conversation analysis (CA) [5]. This is a DAMSL-like dialogue act set with some differences. There are 126 dialogue acts in our typology. The acts are divided into two big groups – adjacency pair (AP) acts (the sub-groups are questions, directives, conventions, etc) and single (non-AP) acts (e.g. continuer, acknowledgement, etc)<sup>2</sup>. We make a difference between questions and directives in our typology. The main difference is formal – questions have explicit formal features in Estonian (interrogatives, intonation, word order) but directives do not have.

Questions are divided into five sub-groups: wh-questions, closed and open yes/no questions, alternative questions and questions that offer an answer. Closed and open yes/no questions have similar form but differ in the

---

<sup>1</sup> <http://math.ut.ee/~koit/Dialog/EDiC.html>

<sup>2</sup> Names of dialogue acts consist of two parts separated by a colon: the first two letters give an abbreviation of the name of an act group, e.g. QU – questions (and answers to them), VR – voluntary reactions. The third letter is used only for AP acts – the first (F) or the second (S) part of an AP act; 2) the full name of an act, e.g. QUF: WH-QUESTION, QUS: GIVING INFORMATION, VR: NEUTRAL CONTINUER. The act names are originally in Estonian.

expected answer. Asking an open yes/no question a speaker intends to get information – the expected second part of the AP is giving information (Ex<sup>3</sup> 1) but in the case of a closed yes/no question the answer yes is sufficient (Ex 2).

(1) *palun kas teil 'on: 'Vesseli kaupluse 'numbrit 'Elvas./ please do you have the number of Vessel shop in Elva*

QUF: OPEN YES/NO

(2) *ee kas teil=ee 'Lapimaa reisile 'on veel `vabu kohti./ do you have free places for the trip to Lapland*

QUF: CLOSED YES/NO

Other information-requests (and directive-actions in sense of DAMSL) are considered as directives (DIF: WISH) in our typology (Ex 3). Additionally to wish, the possible first parts of directive APs are proposal and offer.

(3) *palun 'öelge 'Hansa: 'Reiside: 'telefon. / please tell the phone of Hansa Travel*

DIF: WISH

We consider two kinds of customers' initial requests in this paper. The one kind is so-called direct requests which are represented as directives (DIF: WISH). The other kind is indirect ones – requests in form of questions (QUF: OPEN YES/NO). In both cases, the reactions can be giving/ missing information.

We divide the requests (both direct and indirect) into two groups on the basis of the reaction expected from the addressee. The first group is formed by information requests – a customer needs a certain information, e.g. a phone number. The other group are requests that expect an action by the addressee (e.g. to send a taxi by a dispatcher). Still, the action always is accompanied with giving information: the operator informs a customer if she is (un)able to perform the action and/or has performed it (Ex 4).

(4) *jaa, takso tuleb teile / yes, a taxi comes to you*

DIS: GIVING INFORMATION

For this paper, 144 calls (almost 20,000 tokens) were selected from EDiC. Four situational types are represented in the dialogues: directory inquiries, calls to travel agencies, to outpatients' offices and to a taxi service (Table 1). The selection was determined by the content of EDiC – these four types are the biggest.

In our subcorpus, customers' requests are information requests in directory inquiries and calls to travel agencies (Ex 5).

(5) *ma paluks filo'soofiateaduskonna 'dekanaadi 'numbrit. / I would like to get the faculty of philosophy dean office number*

DIF: WISH

The requests that expect an action occur in calls to

outpatients' offices or for a taxi (Ex 6).

(6) *ma palun `taksot `Ringtee `kuuskend kaheksa `bee. / I ask a taxi to Ringtee sixty eight B*

DIF: WISH

If a customer who calls an outpatients' office or a taxi service needs information then he usually asks a wh-question (these questions are not considered here).

**Table 1. Overview of the corpus**

Dialogue type	# dialogues	Customers' initial information acts (%)			
		requests (direct and indirect)	closed yes/no questions	wh-questions	other
1. Directory inquiries	60	80	-	20	-
2. Calls to travel agencies	36	70	9	20	1
3. Calls to outpatients' offices	26	85	4	4	7
4. Ordering a taxi	22	91	5	-	4
<i>Total</i>	<i>144</i>				

On the other hand, the requests can be divided into general and specific requests. General requests occur in calls to travel agencies. By using such a request the customer only indicates a problem domain (Ex 7).

(7) *sooviks odavalt 'Inglismaale sõita. / I'd like to travel to England cheaply*

DIF: WISH

In the remaining types of the analysed calls, the initial requests are specific (Ex 8, 9).

(8) *ma paluks 'Maarjamõisa 'kööki. / I would ask the kitchen of Maarjamõisa*

DIF: WISH

(9) *(.) me `palume taksot=e `Nõlvaku `viisteist. / we ask a taxi to Nõlvaku fifteen*

DIF: WISH

### 3. Direct and Indirect Requests

#### 3.1 Initial Requests

Why do we distinguish direct requests (wishes) from indirect ones (open yes/no questions)? An argument for differentiating between them is that it is harder for the addressee to refuse to perform a directive than answer *no* to a question. Anne Wichmann claims that the speaker uses a directive if he expects the addressee to fulfill it. By asking a question, the speaker mitigates his request [11].

Our analysis confirms the first claim. In case of **directory inquiries**, a customer uses a directive if he is sure that the requested information can be given (Ex 10).

(10) *ma sooviksin 'bussijaama telefoni'numbrit. / I would like the phone number of the bus terminal*

DIF: WISH

<sup>3</sup> Transcription of CA is used in examples.

On the contrary, asking a question can mean that the speaker **doubts** the ability of the addressee to fulfill the request (Ex 11).

(11) *.hh 'oskate=te mulle 'öelda ee kus asub 'firma 'Aa 'Pluss 'Farma 'Oo 'Üü / can you tell me where the company Aa Pluss Farma OY is located* QUF: OPEN YES/NO

**Table 2. Ratio of initial direct and indirect requests**

Dialogue type	% direct requests (wishes)	% indirect requests (open yes/no questions)
1. Directory inquiries	79	21
2. Calls to travel agencies	75	25
3. Calls to outpatients' offices	59	41
4. Ordering a taxi	85	15

Still, using a question instead of a directive can sometimes be considered as a mitigation of a directive, a polite way to express a request, which is not related to the speaker's uncertainty (Ex 12).

(12) *kas te 'oskaksite 'öelda mulle.: 'Liinavei 'Tartu: 'kogumispunkti telefoni'numbrit. / would you be able to tell me the number of Liinavei Tartu gathering place* QUF: OPEN YES/NO

The ratio of direct and indirect requests is 79:21 in directory inquiries (cf. Table 2).

In calls to **travel agencies**, all the initial requests are general and only determine a problem domain (Ex 13).

(13) *min:d uvitavad turismireisid 'Lõuna=Eestis. / I'm interested in tourist trips in Southern Estonia* DIF: WISH

An indirect request often is a fusion of a general wish and a yes/no question (Ex 14).

(14) *=sooviksin: sõita talvevaheajal 'Hollandisse aga ma ei ole valinud vel 'piirkonda, kas te oskaksite 'soovitada midagi. / I'd like to travel to Netherlands in winter holidays but I have not made a choice of a region, could you suggest something* QUF: OPEN YES/NO

Using a question instead of a directive (25% and 75%, respectively, cf. Table 2) can be considered as a specification of a (general) request.

The two remaining dialogue types (calls to outpatients' offices and to a taxi service) are different (Table 2). Firstly, an action of an operator is expected here in most cases (88% and 95% of dialogues, respectively). Secondly, there are differences in the general framework – rights of customers and obligations of operators (cf. [1]). If a

customer calls **an outpatients' office** with the aim to book a consultation with a doctor then his request sometimes can not be fulfilled, e.g. the doctor does not have a reception, or the patient does not belong to the doctor's list (Ex 15).

(15) *khm kas saaks uroloogi järjekorda panna? / would it be possible to be entered into a waiting-list of an urologist?* QUF: OPEN YES/NO

**Table 3. Typical uses of indirect requests at the beginning of call**

Dialogue type	Reason for using indirect requests
1. Directory inquiries	Uncertainty and/or politeness
2. Calls to travel agencies	specification of a general request
3. Calls to outpatients' offices	Uncertainty
4. Ordering a taxi	Uncertainty

A customer uses an open yes/no question if he doubts whether his request can be fulfilled, making a kind of checking pre-sequence. This explains why the percent of indirect requests is higher in these calls (41%). On the contrary, when a customer orders a **taxi** then he is sure that the operator will send a taxi to him (because he will pay for the service), and he uses mostly (85%) a directive. A question is used only if a customer is uncertain (Ex 16).

(16) *on teil 'busstaksot saata Iks='Essi ette / do you have a bus taxi to send to XS* QUF: OPEN YES/NO

Table 3 summarizes the typical uses of indirect requests in our analysed corpus.

### 3.2 Reactions to Requests

A hearer can differently react to requests represented in different ways. For example, Tine Larsen claims that if a patient called an ambulance and asked *could you send...* then the operator started to clarify the circumstances and the seriousness of the case. But if the patient requested directly *send...* then the operator asked the address in order to send out the ambulance immediately [7]. This example demonstrates that a model exists in human mind which suggests formulations of requests of certain kind in a certain way. Obviously, such a model has been evolved in the process of generalisation of norms and rules used in the society. If a DS does not take into account such norms then it can react in a wrong way.

Our analysis did not demonstrate the dependency of an operator's reaction on the form of a customer's request. Still, the behaviour of an operator is determined by the dialogue type.

There are three possible continuations to the dialogue after a customer's request: (1) the operator grants it immediately, (2) the operator initiates an information-

sharing sub-dialogue, (3) the customer initiates a subdialogue himself. The first continuation is typical in **directory inquiries** (Ex 17, C – customer, O – operator) and when **ordering a taxi** – information was given and the needed action was performed immediately in 60% of cases in both dialogue types.

(17) C: *mt=hh tere,/ good morning* RIS:  
GREETING  
*öelge=palun: `pensioniameti `telefoni (.).h `number (.)  
`Tartus. / please tell me the phone number of the pension  
department in Tartu* DIF: WISH  
(...)  
O: *ee `number on `seitse=neli=`neli? / the number is  
seven four four* DIS: GIVING  
INFORMATION

On the contrary, all the calls to **outpatients' offices** are of type (2) – the operator always initiates a subdialogue asking several bits of data about the patient (name, ID code, time, etc).

In the phone calls to **travel agencies**, a customer gets an answer immediately if his goal cannot be achieved (18% of cases). As mentioned before, it is typical to travel agency dialogues that a customer starts a conversation with a general request (*I would like to take a trip to England*). Then a question-answer subdialogue follows specifying his request. This way, his initial (too general) request will not be granted by the operator directly, however, a sequence of answers to his questions can be considered as a grant of the initial request. There are no examples in our corpus, where positive answer to the general request is given immediately. The operator has to ask adjusting questions in order to give an answer. There are four possibilities to do it:

1) the operator asks the customer to specify his request. In this case she reacts to a general request by using particles *jah, jaa* 'yes'. Such cases are the most frequent in our corpus (Ex 18).

(18) C: *hh e sooviks: sõita Tallinnast `Münchenisse  
lennukiga. / I'd like to travel from Tallinn to Munich by  
plane* DIF: WISH  
O: *jaa? / yes* VR: NEUTRAL CONTINUER  
H: *ee `üliõpilasele kui=palju `maksab. / How much does it  
cost for a student* QUF: WH

2) The operator asks adjusting questions herself.

3) The customer specifies his request himself in the same turn. Then the request is followed by a long pause which indicates that the inquirer is expecting the partner's reaction – how to continue.

4) The operator refuses to answer immediately but offers another way to give information (per e-mail, fax etc).

The operator started a subdialogue in 29% of travel

dialogues, and the customer himself did it in 24% of cases (after the operator's acknowledgement *jah?* 'yes' which signals that she is expecting a specification of the request).

#### 4. Linguistic Features of Requests

There are certain lexical and syntactic cues in Estonian which can be used for representing and automatic recognition of direct and indirect requests.

Direct requests (DIE: WISH) are expressed using

- certain verbs (*soovima* 'to wish', *tahtma* 'to want', *paluma* 'to ask', *üttelema* 'to tell', *vaja olema* 'to be needed', etc)
- certain (verb) forms – conditional and imperative [6].

The single exceptions are (1) *palun* '[I] ask, please' in the indicative whose meaning includes politeness, and (2) (*mind*) *huvitab/ (ma) olen huvitatud* 'I'm interested in' which emphasizes the speaker's interest.

Table 4. Typical linguistic patterns of initial requests

Dialogue type	Direct requests	Indirect requests
1. Directory inquiries	<i>(öelge) palun; paluks(in) / sooviks / tahaks teada</i>	<i>(palun) öelge kas; palun kas teil on; (äkki) oskate öelda</i>
2. Calls to travel agencies	<i>sooviks(in) sõita / küside / teada / informatsiooni; ole(ksi)n huvitatud /mind huvitavad/ huvitaksid; tahaks sõita</i>	<i>ma tahtsin küsida... kas; sooviksin sõita ... kas</i>
3. Calls to outpatients' offices	<i>sooviks(in) ... aega</i>	<i>kas saab</i>
4. Ordering a taxi	<i>palun/ sooviks(in) ... taksot</i>	<i>on teil/sul</i>

Indirect requests are represented as yes/no questions (QUF: OPEN YES/NO). Most of (both open and closed) yes/no questions begin with a question-particle *kas* 'whether' (which is not translated, Ex 12) [4]. Some additional linguistic features can be used in order to differentiate them. Open yes/no question can include (a) pronouns *mingi, mingisugune* 'any, a', *mõni* 'some' indicating indefiniteness in a sentence, (b) a plural partitive, frequently used with a word *mingi* 'any'.

Some typical patterns are used in direct and indirect requests in different dialogue types (Table 4).

Some experiments are made in automatic annotation of dialogue acts in Estonian dialogues using methods of machine learning [2, 3]. Still, the tested methods did not discover many linguistic patterns found in corpus analysis carried out in this paper. The reason is obviously the small size of each sub-corpus.

## 5. Discussion and Conclusion

A customer expresses his initial intention in form of a (direct or indirect) request in 70-91% of cases, less in calls to travel agencies and more when ordering a taxi. Therefore, requests are the most frequent dialogue acts in the beginning of institutional calls. Wh-questions are mostly used in the remaining cases.

When making a request, a customer can use a direct or indirect dialogue act. Both of them should have similar effect because the intention of a customer is to get information in both cases. Choosing an indirect request (an open yes/no question) a customer takes the risk that the operator interprets it as a closed yes/no question and answers *yes* instead of giving information. Why customers take the risk to be misinterpreted? There are two main reasons for using a question instead a directive which are connected one with another – a speaker's uncertainty and politeness.

Our analysis shows that use of indirect requests depends on the dialogue type. If a customer feels himself entitled to get a service (and is convinced in the ability of the operator to provide it) then he predominantly uses a directive like when ordering a taxi (85% of requests are directives). If he doubts then he tends to use a question like in calls to outpatients' offices (59% are directives and 41% questions). Still, using a question can express politeness in the same time. However, directives used in calls to outpatients' offices always contain a verb in conditional which is an additional mean of expressing politeness.

We have analysed Estonian human-human spoken dialogues with the aim to design a dialogue system. We have chosen 144 institutional dialogues (phone calls) from the Estonian dialogue corpus. Four situational groups are represented in the dialogues. Customers use directives or questions for expressing their initial requests (direct and indirect requests, respectively). The most part of initial requests are direct (59–85%, depending on the dialogue type). Using an indirect request can mean both the speaker's uncertainty in fulfilling the request, and a polite way to express his intention. Some typical linguistic patterns are used in requests depending on the dialogue type. Developing a dialogue system one should take into account the strategies of human-human conversation used in the domain which is chosen for the system, and linguistic form of expression of intentions. The following lessons are learnt from the study:

- a customer who is calling an institution can represent his initial request using a directive or a question
- using a question typically means that a customer doubts in the ability of the operator to fulfill the request
- certain linguistic patterns exist that are used by

customers for representing their initial directives and questions

- choice of a question instead of a directive depends on the dialogue type.

Our next aim is to concentrate on automatic recognition of dialogue acts trying to enrich the machine learning methods with the rules found in the corpus study.

## Acknowledgements

This work is supported by Estonian Science Foundation (grant No 5685).

## References

- [1] P. Drew. Mis-alignments between caller and doctor in 'out-of-hours' telephone calls to a British GP's practice. J. Heritage and D. Maynard (eds.) *Communication in Medical Care: Interaction between Physicians and Patients*. Cambridge University Press, Cambridge, 2006, 416-444.
- [2] M. Fišel and T. Kikas. Automatic recognition of dialogue acts. *Language and Computer. Papers of Chair of General Linguistics*, 6, 233-245. University of Tartu, Tartu, 2006.
- [3] M. Fišel. Dialogue act recognition techniques. *Estonian Papers in Applied Linguistics*, 3. Tallinn, 2007, 117-134.
- [4] T. Hennoste, O. Gerassimenko, R. Kasterpalu, M. Koit, A. Rääbis, K. Strandson, and M. Valdisoo. Questions in Estonian Information Dialogues: Form and Functions. V. Matousek, P. Mautner (eds). *Text, Speech and Dialogue. 6th International Conference TSD 2005. Proceedings*, 420-427. Springer, 2005.
- [5] I. Hutchby and R. Wooffitt. *Conversation Analysis. Principles, Practices and Applications*. Polity Press, 1998.
- [6] M. Koit, M. Valdisoo, O. Gerassimenko, T. Hennoste, R. Kasterpalu, A. Rääbis, and K. Strandson. Processing of Requests in Estonian Institutional dialogues: Corpus Analysis. P. Sojka, I. Kopecek, K. Pala (Eds.). *Text, Speech and Dialogue. 9th International Conference, TSD 2006, Proceedings*, 621-628. Brno Czech Republic. Springer, 2006.
- [7] T. Larsen. Requests and the Construction of Emergency Relevance. *Proc. of the International Conference on Conversation Analysis*. Helsinki, 2006, 196.
- [8] M. F. McTear. *Spoken Dialogue Technology: Toward the Conversational User Interface* Springer, London, 2004.
- [9] W. Minker and S. Bennacef. *Speech and Human-Machine Dialog*. Kluwer Academic Publishers, Boston/Dordrecht/London, 2004.
- [10] S. Möller. *Quality of Telephone-based Spoken Dialogue Systems*. Springer, 2004.
- [11] A. Wichmann. The intonation of please-requests: a corpus-based study. *Journal of Pragmatics* 36, 2004:1521-1549.

# Ontology-based Semantic Annotation of Product Catalogues

Emiliano Giovannetti, Simone Marchi, Simonetta Montemagni, Roberto Bartolini

Istituto di Linguistica Computazionale - CNR

via G. Moruzzi 1

56124, Pisa

{emiliano.giovannetti, simone.marchi, simonetta.montemagni, roberto.bartolini}@ilc.cnr.it

## Abstract

This paper describes a methodology for the semantic annotation of product catalogues. We propose a hybrid approach, combining pattern matching techniques to exploit the regular structure of product descriptions in catalogues, and Natural Language Processing techniques which are resorted to analyze natural language descriptions. It also includes the access to an application ontology, semi-automatically bootstrapped from collections of catalogues with an ontology learning tool, which is used to drive the semantic annotation process.

## Keywords

Semantic Annotation of texts – Ontology learning - Information Extraction for e-commerce

## 1. Introduction

Semantic annotation of product catalogues constitutes a poorly explored field of research, yet it seems to represent a promising answer to the growing industrial and commercial need of sophisticated and concept-based browsing tools. Semantic annotation of catalogues can be exploited for several applications, both for the companies and, in particular, for final customers, the latter having the possibility of browsing the catalogue not just by keywords but also by concepts, in the spirit of the Semantic Web.

Previous attempts in this research area have been carried out in the framework of the European project CROSSMARC [1] and of the Czech national project Rainbow [2]. The goal of CROSSMARC was the design and development of an e-retail product comparison multi-agent system carrying out information extraction from Web pages containing product descriptions in different languages (namely, Greek, English, Italian, and French). This goal was achieved by combining language technologies, machine learning and user modelling techniques. In the CROSSMARC architecture, semantic annotation was carried out by processing the Web pages containing product descriptions from retailers and by adding markup tags to relevant information elements. A domain ontology was used as a “semantic glue” to link together the various analysis modules dealing with different languages.

The Rainbow project tackled the information extraction task from product catalogues from a different perspective: it used statistical information extraction techniques (namely, Hidden Markov Models) without any

recourse to NLP techniques. As in the CROSSMARC case, an ontology has been used to group the semantic labels produced by automatic annotation into product instances.

In both projects heavy reliance was made on HTML tags to retrieve information of interest. Yet, it is often the case that product catalogues in company web sites are presented as PDF documents available for download. If the starting point is no longer an HTML page but a PDF document, extraction strategies may have to be revised: this is the case we have been dealing with in this context.

In this paper we present a methodology for the semantic annotation of product catalogues which has been tested on Italian catalogues belonging to the furniture domain. The methodology was developed in the framework of the European VIKEF project (Virtual Information and Knowledge Environment Framework, IST-2002-507173, <http://www.vikef.net/>), aimed at creating an advanced software framework for enabling the integrated development of semantic-based Information, Content, and Knowledge (ICK) management systems.

## 2. Semantic annotation of product catalogues in VIKEF: the strategy

Automatic extraction of knowledge from product catalogues appears to be a complex task due to the fact that target information is typically organized to be appealing and readable by human end-users and not to be handled by automatic extraction systems. Semantic annotation of product catalogues thus poses different challenges at different levels. First, in catalogues images and text both contribute to the relevant information, being combined in a sometimes indivisible informational unit: from this it follows that semantic annotation of product catalogues should rely on information coming from both images and text. Concerning the textual part, catalogues do not contain continuous and linguistically sound text, i.e. typical sentences are constituted by nominal descriptions: this fact often discourages the recourse to traditional Natural Language Processing (NLP) techniques [3]. On the other hand, product descriptions appear as semi-structured texts where product names, prices, and other features appear in a regular order: unfortunately, this is generally not the case. Semantic annotation of product catalogues appears therefore as a complex task requiring the combination of different types of evidence and techniques.



In this paper, we focus on the analysis of the textual part of catalogues for which a hybrid approach is proposed, which combines pattern matching techniques exploiting the almost regular structure of product descriptions in catalogues, and NLP techniques which are resorted to analyze natural language descriptions. For the semantic annotation of texts, however, a formal representation of a given domain, i.e. an ontology, with respect to which annotation is carried out, is required.

After initially exploring the idea of exploiting well-established lexico-semantic resources such as Wordnet, we decided to adopt an ontology learning approach, i.e. to bootstrap the required domain knowledge from texts. To this end, a component for the semi-automatic construction of a formal representation of the domain was developed starting from an existing ontology learning tool: T2K (Text-to-Knowledge), a hybrid system combining linguistic technologies and statistical techniques jointly developed by CNR-ILC and Pisa University [4]. The *application ontology* [5] built starting from the results of the ontology learning process is then used to drive the semantic annotation process.

Semantic annotation of catalogue texts is carried out as follows. First, pattern matching techniques are resorted to for isolating individual product descriptions within the textual flow and for identifying their basic building blocks (e.g. the product *name*, its *price* as well as its natural language description). For each identified product, the natural language description is then processed by a battery of NLP tools for the analysis of Italian texts (AnIta, [6]) in charge of identifying relevant entities (e.g. *colour*, *material*, *parts* of a given product) and the relations holding between them (which can be referred either to the product itself or to individual parts). The process of semantic annotation of product descriptions is driven by the application ontology bootstrapped from texts: in particular, ontological information is used for the recognition of semantically relevant terms occurring in the free text part of the product descriptions, and for the semantic interpretation of syntactic ambiguities emerged during the linguistic analysis process.

### 3. The System

The general architecture of the implemented system includes two main components, the Product catalogues Terminology Processor (henceforth, PTP) and the Product catalogue Italian Semantic Annotator (henceforth, PISA), both exploiting the battery of NLP modules.

#### 3.1 The PTP module

PTP was developed for bootstrapping terminological and ontological knowledge from catalogue collections. PTP carries out the ontology learning task in two different steps: 1) extraction of domain terminology, both single and multi-word terms, from the catalogues; 2) organization and structuring of the set of acquired terms into a) fragments of

taxonomical chains, and b) clusters of semantically related terms.

Domain terms need to be recognized whatever their linguistic form in the documents is: term extraction thus requires some level of linguistic pre-processing of texts. In this case, term extraction is carried out starting from syntactically chunked texts. Candidate terms may be one word terms or multi-word terms. The acquisition strategy differs in the two cases. Potential single terms are extracted from the syntactically chunked text, in particular from the nominal heads of different chunk types (typically, nominal and prepositional chunks). Candidate terms are purely identified on a frequency basis (after excluding stop-words). The acquisition of multi-word terms follows a two stage strategy: first, the chunked text is analysed on the basis of a mini-grammar for the extraction of potential complex terms; second, the list of acquired potential complex terms is ranked according to the log-likelihood ratio association measure [7], which assesses the strength of the association between the words heading the chunks covering the candidate complex term. The set of rules used for identifying potential terms covers the main types of modification observed in complex nominal terms (e.g. adjectival modification, prepositional modification, up to more complex cases combining different modification types). The final TermBank is built by setting thresholds for the selection of potential terms, which can be interactively selected by users on the basis of the size of the document collection on the one hand and of the typology and reliability of expected results on the other hand.

In the second step, proto-conceptual structures involving the terms in the TermBank are identified. Since this represents a more complex task, the starting point is no longer the chunked text, but rather a dependency-annotated text enriched with the multi-word terminology acquired at the term extraction stage. In particular, terms in the TermBank are first organized into fragments of taxonomical chains, which are reconstructed starting from their internal linguistic structure (i.e. from their sharing the head and possibly modifiers). For instance, *steel legs* and *adjustable legs* can be seen as hyponyms of a general single term *legs*. PTP also performs the identification of clusters of semantically related terms which is carried out with CLASS, an algorithm to build distributionally-based semantic similarity spaces illustrated in [8]. For each term, a set of semantically related terms is identified: e.g. to the term *nero* 'black' the following set of related terms has been associated: *rosso* 'red', *verde* 'green', *bianco* 'white', etc., thus identifying a set of different colours. Automatically acquired clusters of semantically related terms are then merged into classes corresponding to super-concepts (e.g. COLOUR) of concepts directly built upon terms (e.g. "Blue", "Green", etc), the latter further structured in taxonomical chains (e.g. "Light\_Blue" is\_a "Blue", "Dark\_Green" is\_a "Green", etc.). A fragment of

the ontology semi-automatically learned by PTP starting from a collection of different furniture catalogues is reported in Figure 1.

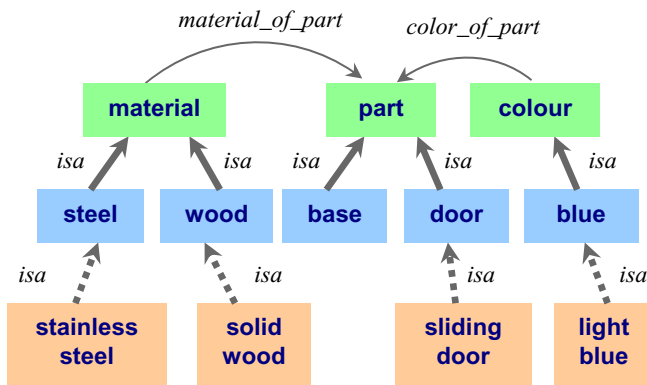


Figure 1. A fragment of the application ontology

The concepts of “Steel” and “Wood” (automatically clustered together by PTP), for example, have been manually set as sub-concepts of MATERIAL, as represented in the figure by solid arrows standing for *isa* relations. The same has been done for PARTS and COLOURS, two other top-level concepts. Dotted arrows, on the other hand, represent hierarchical relations between concepts automatically detected by PTP on the basis of the internal linguistic structure of the relative terms. Relations between top-level concepts (e.g. *material\_of\_part* and *color\_of\_part*) were added manually.

### 3.2 The PISA module

PISA has a two-module architecture composed by: the RegExp Manager component, performing pattern matching on the catalogue text to isolate individual product descriptions and to identify their basic building blocks, and the NLP Manager, in charge of the linguistic analysis of the free text descriptions. While the NLP Manager is catalogue-independent, the set of regular expressions interpreted by the RegExp Manager needs to be customised with respect to the specific typographic conventions adopted by a given company, since the structure of product descriptions typically vary from one catalogue type to another. In what follows, we report the results of experiments performed on two structurally different furniture catalogues: namely, IKEA and Zanotta. Currently, PISA is able to detect: 17 different entity types, ranging from *product*, *name*, *type*, *price*, *dimensions*, *product id* to *product\_part*, *material* and *colour*, and 15 relation types holding between recognised entities, e.g. *made\_of* holding between a *product* or *part\_of\_product* and its *material*.

#### 3.2.1 Pattern Matching for catalogue analysis

From a procedural point of view, individual product descriptions are firstly extracted through pattern matching

starting from a set of regular expressions, like the one reported in Figure 2. Once an individual description is identified, some of its parts can already be semantically classified and annotated: this is the case of entities like *name*, *type*, *price*, *dimensions* and *product id*, corresponding to subparts of the matching regular expression.

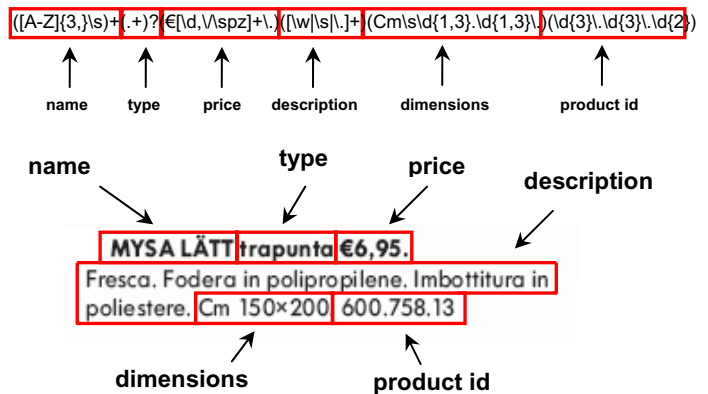


Figure 2. Example of regexp and a matching description

#### 3.2.2 Ontology-driven NLP Analysis

The linguistic analysis of product description is carried out by Anlta, a battery of NLP tools consisting in an “assembly line” whose main components include: tokenization of the input text, morphological analysis (including lemmatisation) and syntactic parsing, the latter articulated in two different stages, i.e. chunking (which also includes morpho-syntactic disambiguation) and dependency analysis. Semantic annotation of product catalogues is ontology-driven and operates starting from the output of dependency analysis. The ontology accessed by PISA, built with the help of the PTP (see section 3.1), is used to:

- detect and (semantically) annotate relevant entities inside the free text of the product descriptions;
- detect and annotate relations between annotated entities;
- resolve possible ambiguities found during the process of linguistic analysis.

Once a natural language product description has been syntactically analysed, the NLP Manager component carries out entity recognition and classification as follows. For each noun-headed chunk, PISA:

- 1) looks for a corresponding concept inside the ontology;
- 2) “climbs up” the ontology to identify the corresponding “root” concept (e.g. PART, MATERIAL or COLOUR);
- 3) produces the correct annotation of the detected entity (e.g. “Knob” is annotated as *Product\_Part*, “Oak” as *Material*, etc.).

Detection and classification of relations holding between entities is performed by combining ontological and syntactic constraints: this “hybrid constraint

satisfaction” strategy was already applied with encouraging results for the interpretation of queries in the domain of Natural Language Interfaces to Databases [8]. To give but one example, if an entity A has been detected in the NL product description and semantically classified according to the ontology as a *product\_part*, and if A is syntactically modified by a complement corresponding to another entity of type B classified as a *material*, a relation linking the concepts corresponding to A and B is looked for in the ontology: if such relation exists, entities A and B are linked with the same relation type, which is *material\_of\_part* in the case at hand. Let’s consider the following NL description of a chair in the IKEA catalogue: *Gambe in acciaio e schienale in plastica. Bianca.* ‘Steel legs and plastic back. White.’ Detected entities include:

- “Steel” and “Plastic” as *Material*;
- “Legs” and “Back” as *Product Part*;
- “White” as *Colour*.

Identified relations are: *part\_of\_product* holding respectively between “Legs” and “Back” and the product itself; *material\_of\_part* holding respectively between “Steel” and “Plastic” and the product parts “Legs” and “Back”; *colour\_of\_product* holding between “White” and the product.

Concerning syntactic disambiguation let’s consider the following nominal description appearing in a product description: *Tavolo in cristallo con bordi bisellati* ‘bevel edged plate glass table’ where a syntactic ambiguity occurs for what concerns the attachment of the prepositional phrase *con bordi bisellati*, which can be governed either by the nominal head *tavolo* or by *cristallo*. In situations like this one, the ontology can be usefully exploited to perform syntactic disambiguation. In the case at hand, the ontology asserts that: “tavolo” is a *Product*; “cristallo” is a *Material*; “bordo” is a *Part*. Since there is no property linking a *Material* to a *Part*, but there is one linking a *Product* to a *Part* (i.e. *part\_of\_product*), the correct interpretation is that “bordi bisellati” is a part of “tavolo”.

### 3.2.3 PISA at work

An example of semantic annotation is reported in Figure 3 relative to the product description: *SANELA cuscino €12,95. Fodera in cotone. Cm 40x60. 900.582.56.*

Through pattern matching it is possible to extract the product name, the type (“cushion”), the price, its dimensions and the product identifier, as well as the relations between this information and the product itself (i.e. *name\_of\_product*, *price\_of\_product*, etc). The natural language description identified at this stage is then passed to the NLP Manager which is in charge of acquiring, with the support of the ontology, further information about the product: in this example, the system detects a part (cover) and a material (cotton), as well as a relation holding between them (i.e. *material\_of\_part*). The final annotated product description is reported in Figure 3 where the

different detected entities and relations are listed, including the fact that the cover of the cushion is made out of cotton.

```

<entity data_id="25">
  <product>SANELA cuscino €12,95. Fodera
  in cotone. Cm 40x60. 900.582.56</product>
</entity>
<entity data_id="26">
  <name>SANELA</name>
</entity>
<entity data_id="33">
  <part>fodera</part>
</entity>
<entity data_id="34">
  <material>cotone</material>
</entity>
<relation data_id="34">
  <reltype>name_of_product</reltype>
  <subject>26</subject>
  <object>25</object>
</relation>
<relation data_id="35">
  <reltype>part_of_product</reltype>
  <subject>33</subject>
  <object>25</object>
</relation>
<relation data_id="36">
  <reltype>material_of_part</reltype>
  <subject>33</subject>
  <object>34</object>
</relation>

```

Fig. 3: Excerpt of product semantic annotation

## 4. Evaluation

An evaluation of both PTP and PISA components was carried out, though in different ways. Due to the lack of a golden standard furniture ontology, a task-based evaluation was carried out for what concerns the results of the ontology learning process, in terms of its role in driving the semantic annotation process within PISA. Evaluation of PISA was carried out with respect to the annotation results obtained for two different furniture catalogues, IKEA 2006 and Zanotta: in the IKEA case, 793 out of 984 products were annotated, and 136 out of 167 in the case of Zanotta.

For the evaluation we have distinguished between annotations obtained through pattern matching and annotations performed thanks to the ontology-driven linguistic analysis. We have created a “gold-standard” corpus of reference by randomly extracting and manually annotating 10% of the IKEA products and 20% from Zanotta. Evaluation was concerned with *name*, *type*, *dimensions*, *price*, and *id* extracted by pattern matching and *product material*, *product colour*, *product part*, *product part material*, and *product part colour* extracted by the ontology driven linguistic analysis.

Concerning evaluation metrics, we have calculated *precision* and *recall* on the basis of the following parameters:

- COR(rect): the number of annotations that are found to be correct after comparison with the gold-standard annotations for the same text span;

- INC(orrect): the number of annotations that are found to be incorrect;
- PAR(tially correct): the number of annotations that are partially correct after comparison with the gold-standard annotations (e.g. partial credit is given to the detection of “Birch” in relation to “Solid birch”);
- ACT(ual): the total number of annotations, calculated as COR + INC + PAR.

Using these parameters, we have calculated precision (PRE), which measures the system output’s accuracy, as:

$$PRE = \frac{COR + 0.5 \cdot PAR}{ACT}$$

To calculate recall, two additional parameters were considered:

- MIS(sing): the number of gold-standard annotations in the key that are not present in the system output;
- POS(sible): the total number of annotations in the gold-standard, computed as the sum of COR, PAR and MIS.

Recall was computed as follows:

$$REC = \frac{COR + 0.5 \cdot PAR}{POS}$$

Concerning IKEA, the system has scored a precision of 0,99 for annotations obtained through pattern matching and 0,89 for those obtained through ontology driven linguistic analysis, while, regarding recall, 0,94 for the former and 0,70 for the latter. With Zanotta, both precision and recall are equal to 1 for what concerns the pattern matching analysis (this follows from the regular structure of product descriptions), and respectively to 0,86 and 0,50 for ontology driven NLP analysis. To improve recall we are working on two different fronts:

- **ontology coverage:** the main cause for this relatively low recall is due to missing concepts in the application ontology and the consequent failure in detecting and annotating the relative entities and relations inside the free text description;
- **ontology-driven linguistic analysis:** another problem source turned out to be the adopted strategy for relation detection and annotation, which currently fails when facing unusual syntactic constructions.

## 5. Conclusions and Future Perspectives

In this paper a methodology for the semantic annotation of product catalogues has been introduced. The proposed approach combines Pattern Matching techniques and ontology-driven Natural Language Processing, the former to annotate product features appearing in fixed schemata, the latter to isolate, extract and annotate entities and relations found in the free text product description. The

exploited application ontology has been built semi-automatically, with the help of an existing ontology learning tool which was customised to deal with product catalogues, and starting from a corpus of linguistically analyzed product descriptions.

Among the possible applications fully exploiting the proposed methodology for semantic enrichment of product catalogues, the idea of “Intelligent Catalogue Browsing” has been developed, allowing the user to look for products on the basis of content fully exploiting the power of semantic annotation and thus overcoming the well-known limits of keyword-based searching. Concerning further directions of research we plan to augment the application ontology coverage to increase recall of annotations in the free text part, mostly by trying to automatise as much as possible the ontology construction process. Furthermore, we want to improve the ontology-driven linguistic analysis algorithm, in order to minimize annotation errors deriving from complex syntactic constructions and to extend the methodology to other domains.

## 6. References

- [1] M. T. Pazienza, A. Stellato, M. Vindigni. Cross-lingual multi-agent retail comparison. In: AWSS 2003 Workshop on Applications, Products and Services of Web-based Support Systems, 13 Oct. 2003.
- [2] M. Labsky, V. Svatek, P. Praks, O. Svab. Information extraction from HTML product catalogues: coupling quantitative and knowledge-based approaches. In: Dagstuhl Seminar on Machine Learning for the Semantic Web, 2005.
- [3] O. Šváb, M. Labský, V. Svátek. RDF-Based Retrieval of Information Extracted from Web. In: SIGIR’04 Semantic Web Workshop, Sheffield, 2004.
- [4] R. Bartolini, D. Giorgetti, A. Lenci, S. Montemagni, V. Pirrelli. Automatic Incremental Term Acquisition from Domain Corpora. In *Proceedings della 7th International conference on Terminology and Knowledge Engineering (TKE2005)*, 17-18 August 2005, Copenhagen, Denmark.
- [5] R. Studer, V. R. Benjamins, D. Fensel. Knowledge engineering: Principles, and methods. *IEEE Transactions on Data and Knowledge Engineering*, 25:161 – 197, 1998.
- [6] R. Bartolini, A. Lenci, S. Montemagni, V. Pirrelli. Grammar and Lexicon in the Robust Parsing of Italian. Towards a Non-Naïve Interplay. In: *Proceedings of Coling 2002-Workshop on Grammar Engineering and Evaluation*, Taipei.
- [7] T. Dunning. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1), 1993.
- [8] R. Bartolini, C. Caracciolo, E. Giovannetti, A. Lenci, S. Marchi, V. Pirrelli, C. Renso, L. Spinsanti. Creation and Use of Lexicons and Ontologies for Natural Language Interfaces to Databases. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC 2006)*.
- [9] P. Allegrini, S. Montemagni, V. Pirrelli. Example-based automatic induction of semantic classes through entropic scores. *Linguistica Computazionale*, XVI-XVII: 1-45, 2003.

# A Stable Statistical Constant Specific for Human Language Texts

Felix Golcher  
Humboldt-Universität zu Berlin  
Unter den Linden 6  
10099 Berlin, Germany  
felix.golcher@hu-berlin.de

## Abstract

A novel character-level statistical measure is described which quantifies the level of repetitions in a text. It behaves remarkably uniformly for texts from all 20 tested languages. In contrast to most other text-statistical quantities, the proposed measure is computed from the text as a whole, not from a tokenised text reduced to a frequency list. For growing text sizes, it converges rapidly to a constant value. This text length independent behaviour is an uncommon feature for text-statistical constants. The described phenomenon of constant repetitiveness has so far not been observed in any non-natural language text.

## Keywords

Text statistics; lexical constant; language universal; suffix trees

## 1 Introduction

Since George Kingsley Zipf first published the famous empirical law since named after him [13], a lot of text statistical regularities have been proposed, usually in the form of a formula with some constants in it. (See [2] for an overview.)

However, recent publications have raised doubts as to whether these laws hold and whether these constants are constant. In [11], it is shown for all alleged lexicostatistic constants known at the time that they systematically depend on text size. Additionally, Evert and Baroni [5] demonstrate the low predictive power of many of the laws that were proposed to cope with such text length dependencies.

The *theoretical* significance of Zipf's Law and its relatives is limited by three factors: firstly, they merely make propositions about word frequency lists instead of full human texts. Secondly, they apply in a very similar fashion to randomly produced pseudo text [9, 8] and thus are not a specific property of language. Thirdly, they are not easily interpreted theoretically: it's unclear what the validity of Zipf's Law actually tells us about the system of natural language and its properties.

This paper introduces a new text statistical measure  $V$  which quantifies the level of repetitiveness in a text. For natural language text,  $V$  converges rapidly towards a fixed value, as the text size grows, and

the convergence point is a good constant over texts from different languages. This was tested with 20 languages, from three distinct language families, written with three different classes of writing systems.

So far, this constant repetition rate has only been observed for natural language text. The possible establishing of this phenomenon of constant repetitiveness as a universal and exclusive feature of human text could have some impact on the theory of language: on the one hand, it would impose restrictions on every realistic language model, since such a model would have to reproduce this property in its output (see the discussion in Section 6). On the other hand, the phenomenon would bring up two new questions: if the level of repetitiveness is so amazingly constant, why is this so and what mechanism keeps it constant?

Section 2 gives the necessary conceptual background and defines  $V$ . Section 3 describes the experiments which survey  $V$  for texts from different languages. The results are shown in Section 4. Section 5 reports an experiment which gives more insight into the nature of the investigated quantity. Section 6 discusses comparable known text statistical measures. Section 7 gives an outlook.

## 2 The measure $V$

### 2.1 Defining $V$

In the context of this work, a text is simply a string of symbols. I define the repetitiveness  $V$  of a text  $T$  as  $k/t_0$ , where  $t_0$  is the length of  $T$ , and  $k$  is the number of its substrings which occur with more than one continuation in  $T$ . In other words,  $V$  is the number of *ended* repetitions divided by the text length.

Consider the example text<sup>1</sup>  $T = \text{xabcdecdeabcbx}$ . There are 7 substrings with more than one continuation: **abc**, **bc**, **c**, **cde**, **de**, **e**, and **b**. The text length  $t_0$  is 14, hence  $V = 7/14 = 1/2$ .

If there are no repetitions in the text,  $V$  is obviously 0. If the text consists of the same character repeating – except for the last character – then  $k = t_0 - 2$  and thus  $V = (t_0 - 2)/t_0$ , which approaches 1 as  $t_0$  grows.

Quantifying the repetitiveness of a text by defining  $V$  can be justified a priori by its conceptual simplicity and adequateness (it measures repetitions). Its

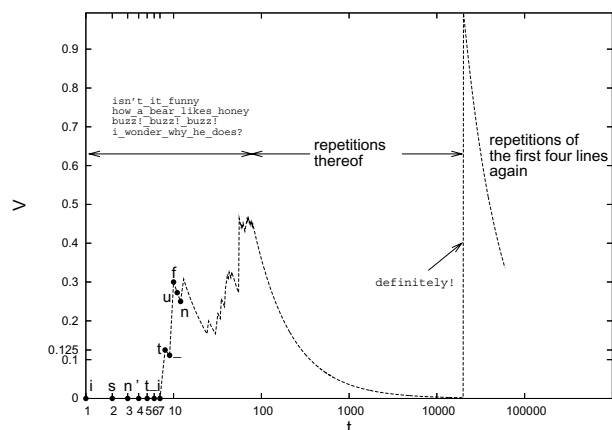
<sup>1</sup> Example text is written in `type writer font`.

remarkable properties will serve as an a posteriori justification.

The number of substrings of a text is  $t_0(t_0 + 1)/2$  where  $t_0$  denotes the text length. This expression quickly gets very large. The practical computation of  $V$  is carried out using the *suffix tree* of  $T$  which can be built in linear time and space complexity [12].  $k$  is then simply the number of nodes in this tree-like index structure [6].

The focus of this paper is not the value of  $V$  for the whole text, but how  $V$  develops if we read the text character by character and view  $V$  as a function of  $t$ , the length of the text part read so far.

## 2.2 Exemplifying $V(t)$



**Fig. 1:** The development of  $V$  for an example text set up to clarify the interpretation of  $V$  as a measure of the level of repetitions. The scale on the  $x$ -axis is logarithmic.

Fig. 1 shows the development of  $V$  for an artificial example text of 60,149 characters: the first 20,000 characters are repetitions of the following four lines:

```
isn't_it_funny
how_a_bear_likes_honey
buzz!_buzz!_buzz!
i_wonder_why_he_does?
```

After this, new text (**definitely!**) is introduced, before the bear song repeats again until the end of the text.

For the first six text characters (**isn't\_**) there is no repetition ( $V = 0$ ). The seventh one (**i**) is a repetition of the first one. But only after the eighth character is read (**t**), does it have two different continuations (**s** and **u**).  $V$  jumps to  $1/8$ . The **t** itself is a repetition but the next character (**\_**) is no new continuation and  $V$  drops to  $1/9$ . After the ninth character (**f**) is read, we have three substrings with different continuations (**i**, **t\_** and **\_**) and  $V = 3/10$ . The following **u** and **n** don't terminate any repetition, and  $V$  drops again. In this way,  $V$  follows a slow upward trend until the end of the four cited lines.

Nothing new is introduced now for nearly 20,000 characters. Since no repetition does ever terminate in this phase,  $V$  drops steadily.

When this very long repetition is ended by the sudden appearance of **definitely!**, the situation changes radically. All at once we have nearly 20,000 substrings with different continuations and  $V$  jumps to a value close to 1 accordingly. After this interruption, the text gets repetitious again and  $V$  drops for a second time.

## 3 Languages and corpora

$V(t)$  was compared for natural language texts from 20 languages. They belong to the three language families Indo-European, Dravidian, and Uralic. Their writing systems instantiate three different classes of writing systems.

### 3.1 The investigated languages

Regarding the genetic relations of the tested languages we refer to [1].

Fourteen Indo-European languages were investigated: The Slavic language Russian, the West Germanic languages English and German, the Romance language French, and the ten Indo-Iranic languages Assamese, Bengali, Gujarati, Hindi, Marathi, Oriya, Punjabi, Sinhala, Urdu, and Kashmiri (subclassified as Dardic).

Tamil, Kannada and Malayalam from the southern branch of the Dravidian language family were included, as was Telugu from the Telugu-Kui branch.

From the Finno-Ugric branch of the Uralic languages, the Finno-Saamic Finnish and the Ugric Hungarian were investigated.

### 3.2 The writing systems

The same text can come out completely different when written in different writing systems. Since the definition of  $V$  is based on repetitions on the character level, the writing system used can be expected to affect the value of this quantity. To investigate this effect, the experiments have been performed on texts written with different scripts.

We adopt the classification of writing systems proposed in [4]. The authors classify scripts "with respect to how symbols relate to the sounds of the language" [4, p. 4]. The resulting classification of the scripts overlaps only in part with the genetic relations cited above.

#### 3.2.1 Abugidas

"In an *abugida*, each character denotes a consonant accompanied by a specific vowel, and the other vowels are denoted by a consistent modification of the consonant symbols [...]" [4, p. 4].

Most languages spoken in the Indian language area use historically related abugidas. This applies to Assamese, Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Oriya, Sinhala, Tamil and Telugu.

### 3.2.2 Abjads

“In a consonantary, here called an *abjad* [...] the characters denote consonants (only) [...]” [4, p. 4]

Some of the languages spoken in India use scripts based upon the Arabic script, the world’s most widespread abjad. From the set of tested languages, this applies to Urdu, Kashmiri, and Punjabi. Urdu is an abjad following [4]. Regarding Kashmiri, see Section 3.2.3.

Punjabi is written in two different scripts: on the one hand in Gurmukhi (an abugida); on the other hand in the Perso-Arabic abjad. The corpus used for this investigation is written in Perso-Arabic.

### 3.2.3 Alphabets

“In an *alphabet*, the characters denote consonants and vowels” [4, p. 4].

German, English, Finnish, French and Hungarian use different variants of the Latin alphabet.

Russian is written with the Cyrillic alphabet.

The script used for Kashmiri is based on the Perso-Arabic abjad, but called an alphabet in [4]. I follow this classification.

## 3.3 The corpora

The corpora of the tested Indian languages are all part of the EMILLE corpus [18]. For each of these languages, I used between 2 and 20MB (that is approximately between 200,000 and 2 million tokens) of the written part of this corpus. Most of the data stems from various Indian dailies.

The German texts are taken from the online edition of the *Süddeutsche Zeitung* – a high quality German newspaper.

For English, a part of the Brown Corpus [17] was used.

For French [20], Russian [15], Finnish [14] and Hungarian [16], novels were used.

## 4 Experimental Results

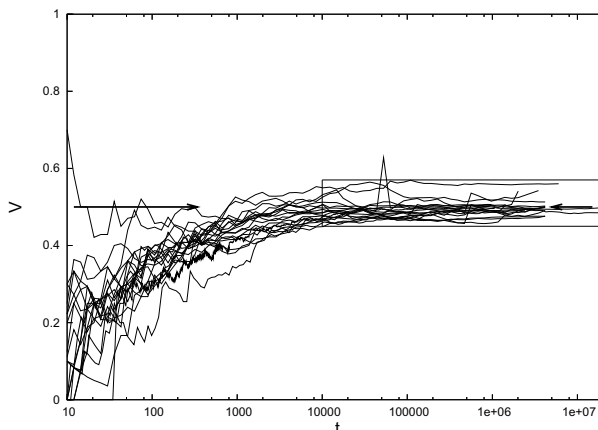
Intuitively, there seems to be no reason for a uniform behaviour of  $V(t)$  in different texts, let alone in different languages. On the contrast, it seems natural to expect changing levels of repetitiveness both within one text and between texts. The repetitiveness could probably depend on various factors such as subject, genre, author, the morphological structure of the language or the writing system.

Fig. 2 and Fig. 3 show the evolution of  $V(t)$  for growing text sizes  $t$ . Fig. 3 is an enlargement of the central part of Fig. 2.

We can draw a set of observations from these figures:

- O1** For all investigated corpora  $V$  converges towards a constant<sup>2</sup>.
- O2** This constant is reached after as few as about 10,000 characters, that is after approximately three pages of text.

<sup>2</sup> The obvious jumps in most of the curves are addressed below.



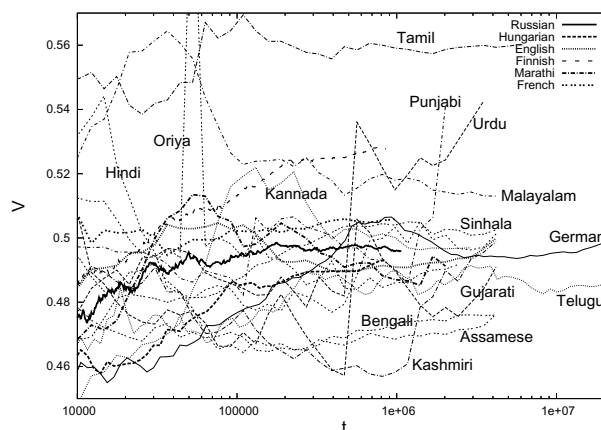
**Fig. 2:**  $V$  for all tested languages. Refer to the text for a list of languages. Fig. 3 enlarges the box in the middle and shows a label for each language. The characteristic value of  $1/2$  is clarified by the bold arrows. The scale on the x-axis is logarithmic.

**O3** For shorter text lengths, an average curve is easily discernible, although the convergence level is not yet reached.

**O4** The convergence level is compatible with  $1/2$ .

We will henceforth summarise the uniform behaviour of the  $V$ -curves as described by the observations **O1** through **O4** under the term *V-convergence*. So far,  $V$ -convergence has been found in all tested natural language texts.

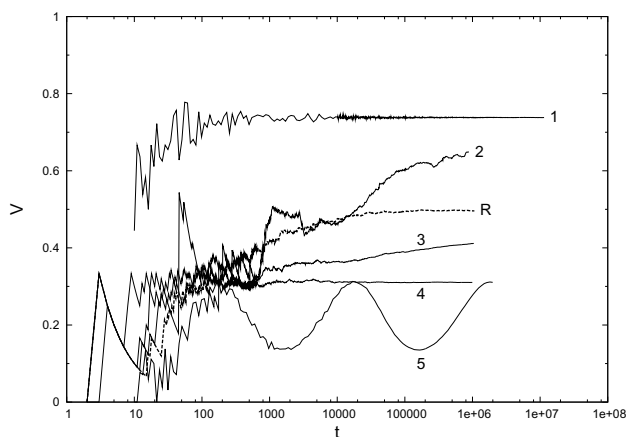
The  $V$ -curves of other texts show a much more diverse behaviour. A small set of examples of such texts is shown in Figure 4 (for comparison,  $V(t)$  for the Rus-



**Fig. 3:** Enlarged middle part of Fig. 2.

sian text is shown as curve R):

- 1 A uniformly distributed random text, i.e. each character has the same probability of occurrence at each text position. Alphabet size is 3.
- 2 c sources from the Linux 2.6.0 kernel. Generally,  $V(t)$  for source code runs above 0.5 and shows a rather unpredictable behaviour.
- 3 Random text generated as described in [3]. This elaborated language model was designed to emulate basic statistic characteristics of natural text such as mean word and sentence length.
- 4 A random text which simulates the English character distribution.
- 5 A uniformly distributed random text<sup>3</sup> with alphabet size 100. Compared with curve 1,  $V(t)$  looks rather different here. In general, for this class of texts, shape and height of  $V(T)$  depend heavily on the alphabet size. This contrasts with the behaviour of natural language texts: although the set of symbols of abugidas (Section 3.2.1) is usually twice as large as for alphabets (Section 3.2.3), there seems to be no immediate impact on  $V(t)$  (see Figure 3).



**Fig. 4:** The  $V$ -curves of different kinds of text. See the text for a detailed description. The scale on the  $x$ -axis is logarithmic.

On the grounds of the experiments reported here, the still highly speculative hypothesis can be formulated that  $V$ -convergence might be a universal and exclusive feature of natural language texts.

This hypothesis has to be thoroughly checked by testing many more texts from different languages, scripts, styles and epochs. So far, additional experiments with the Chinese LCMC corpus [19] were performed. This corpus exists in two different scripts, the traditional characters and the romanised transcript pinyin. The  $V$  of the character version converges towards  $0.27 \pm 0.02$ , while the pinyin version shows  $V$ -convergence with  $V$  approaching  $0.52 \pm 0.01$ . See also the discussion at the end of Section 5.

<sup>3</sup> The peculiar oscillations reflect the fact that first the bigrams repeat, then the trigrams, and so on.

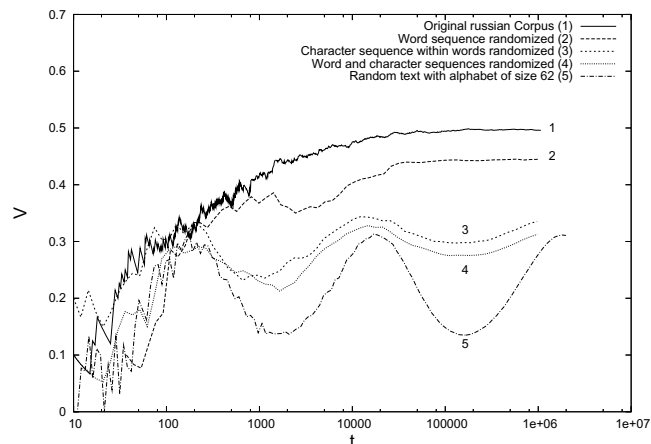
**A Remark:** the bumps that can be seen for many of the corpora in Fig. 3 are due to longer repetitions in these corpora as exemplified in Fig. 1. For web based corpora, such as the EMILLE corpus [18], longer repetitions are hard to avoid, since cut and paste can easily multiply chunks of text or whole texts, especially if an online edition of a newspaper is used as the source.

The fact that longer repetitions in the text are reflected as bumps in its  $V$ -curve could be converted into a method for detecting artificial repetitions in large corpora, provided one is able to cope with the heavy memory consumption of suffix trees. This index structure is used for the technical implementation of the computation of  $V(t)$  as mentioned in Section 2.1.

The novels and the German corpus don't show any bumps. For the novels, this smoothness can be expected, because long repetitions are naturally avoided. In the German corpus they occurred, but were carefully filtered out by a variety of ad-hoc heuristics.

## 5 The impact of randomisation

It is a special feature of the quantity  $V$  that it is based on character strings, not on words. Accordingly, it measures repetitions both below and above word level. It is a natural question, which of these two kinds of repetitions contribute more to the value of  $V$ . To address this question, I separately randomised the internal structure of words and the sequence of words.



**Fig. 5:** The impact of randomisation on  $V(t)$ . See the text for a detailed description.

The starting point of the investigation was the original Russian corpus. The result of the different randomisations of the corpus is shown in Fig. 5:

- 1  $V(t)$  for the original Russian corpus.
- 2 The inner structure of words is left untouched, but their sequence is scrambled.  $V(t)$  is considerably lowered.
- 3 The characters of the words are randomised, while the word order is left untouched. Each word is replaced by a random character string. Equal surface forms are replaced by equal random strings



drawn from a 59 character alphabet. Each character had the same probability.

- 4 combining the randomisation schemes for curves 2 and 3.
- 5 for comparison only:  $V(t)$  for random text, drawn from a uniformly distributed alphabet of size 62.

Clearly, randomisation always lowers  $V$ . This is to be expected since a random string of characters and words can only result in random repetitions. Since the system of human language prescribes the reoccurrence of certain structures, we expect that a deliberate destruction of these structures will diminish the level of repetitions.

Randomising the internal structure of the words affects  $V$  much more than only randomising word order. This shows that repetitions below word level contribute more to the value of  $V$  than repetitions on a larger scale. This corresponds to another observation:  $V$ -convergence occurs in all tested texts written with scripts in which graphemes and phonemes correspond. This includes the (invented) pinyin script, but does not apply to the traditional Chinese script for which no  $V$ -convergence was found. Together, these two observations could be interpreted as a first hint that this phenomenon is rooted in the phonemic level of language.

## 6 Other text statistical regularities and constants

This section discusses how  $V(t)$  and the phenomenon of  $V$ -convergence can be compared with other text statistical constants and regularities.

The best known such regularity is *Zipf's Law* [13]. It states that the most frequent word in any natural language text is twice as frequent as the second most frequent one and three times as frequent as the third most frequent one, and so on. Zipf's law roughly holds, except for the most frequent and the very infrequent words. But, as sketched in [9], and shown in more detail in [8], Zipf's Law is also valid for random text. This over-generality greatly reduces its significance: there's little value in knowing about a property which natural language text shares with noise. As pointed out in the discussion of Figure 4,  $V$ -convergence, on the other hand, could so far not be observed for random text, even if it simulates a natural character distribution or was designed to simulate the statistical features of natural language [3]. If it can be confirmed that  $V$ -convergence is a universal and exclusive feature of natural language text, we gain a strong tool to decide about the adequacy of statistical language models: if such a model is not able to reproduce  $V$ -convergence in its output, it cannot be said to mimic the structure of human language. This hurdle can be expected to be much higher for models which aim at modelling both words and their sequence. Models which reuse existing natural language words will have a lesser problem, as we know that the word sequence has a smaller impact on  $V$  than the inner structure of the words themselves (see Section 5).

Besides *Zipf's Law*, a lot of lexicostatic quantities were proposed to measure – for example – lexical richness or the productivity of word formation processes [2, 10]. Many of these text statistic quantities were proposed as constants, independent of text size. But it was shown that, in practice, these alleged constants tend to vary with text length [11]. Similarly, there is a class of models which try to capture these text length dependencies. Evert and Baroni [5], however, show that the predictive power of most of these models is low: the behaviour computed for small text sizes cannot be extrapolated to larger texts. In contrast to this,  $V(t)$  converges very rapidly towards a fixed value around which it fluctuates only a little.

As can be seen from Fig. 1 and 4, different kinds of text can produce qualitatively diverse  $V$ -curves. In contrast, most lexicon based text statistical measures have only a few degrees of freedom. Consider Zipf's law as an example: it is usually depicted in a *Zipf plot*: starting with an ordered frequency list, the place in this list is shown on the x-axis, while the frequency is shown on the y-axis. This will always yield a monotonously decreasing function. The potential variability in  $V(t)$  makes its uniformity in natural language text more surprising than the validity of Zipf's Law.

$V(t)$  is computed from the full character sequence of the text and is thus sensitive to structural changes on all levels. In contrast, the lexicostatic quantities discussed in this section are usually derived from summary statistics such as the number of Hapax Legomena or the vocabulary size. Thus, they lose, from the start, most of the information contained in the full text: they remain the same if the text is replaced by a random sequence of random tokens, as long as these tokens have the same frequency distribution as the tokens of the original text.

As a consequence, none of the randomisation methods applied in Section 5 would have any effect on these word frequency based measures, since the statistics of the lexicon is left untouched.

This striking difference between lexicostatic measures and constants, on the one hand, and  $V$ -convergence, on the other hand, effectively counters the argument that the latter might turn out to be an alternative manifestation of one of the former, for example of Zipf's Law.

All these features – its exclusive occurrence in natural language text, its higher sensitivity to structural changes of the text, its stable convergence and its richer structure – make  $V(t)$  and its convergence towards  $1/2$  much more informative and significant than any of the token frequency related models and constants.

## 7 Outlook

If  $V$ -convergence can be firmly established as a feature of natural language text, this would immediately raise two questions: why is the level of repetitions so very constant? It is clear that too many repetitions in language are bad: it's both boring and time consuming. On the other hand, if nothing ever repeats we have no chance of recognising known elements or

of regaining lost information: no understanding without repetition and no stable communication without redundancy. But why should repetitions be so evenly distributed?  $V = 1/2$  seems to be some kind of optimum, but what does it optimise? The other question that would be raised is: what keeps  $V$  this constant? What is the mechanism within the human language system that regulates repetitiveness?

But before all of these questions can gain real relevance, a second round of experiments is necessary:  $V(t)$  has to be investigated for more texts – natural and non-natural – being as diverse as possible.

In order to get a clearer picture of  $V$  and the phenomena surrounding it, the exact shape of this quantity will have to be measured carefully. One obvious question is whether there is a significant deviation of the convergence point of  $V(t)$  from  $1/2$  or not.

Another thrilling task ahead is to examine  $V(t)$  for spoken corpora, maybe in phonetic transcription. Is  $V$ -convergence a phenomenon of written language or does it also occur in spoken language?

A related project [7] investigates the impact of stylistic differences, like authorship, on similar data.

## Acknowledgements

I thank my doctoral advisor Prof. Dr. Anke Lüdeling for her indispensable support, Prof. Dr. Klaus Schulz for giving me the opportunity of carrying out fundamental research, and Karsten Tabelow, Verena Harpe, Adriana Hanulíková and Anna McNay for critical comments and proof reading. Last but not least, I thank the anonymous referees for their valuable comments.

## References

- [1] R. E. Asher and J. Simpson, editors. *The Encyclopedia of Language and Linguistics*. Pergamon Press, Oxford, New York, Seoul, Tokyo, 1994.
- [2] R. H. Baayen. *Word frequency distributions*. Kluwer, Dordrecht, 2001.
- [3] C. Biemann. A random text model for the generation of statistical language invariants. In *Proceedings of HLT-NAACL-07*, Rochester, NY, USA, 2007.
- [4] P. Daniels and W. Bright. *The World's Writing Systems*. Oxford University Press, 1996.
- [5] S. Evert and M. Baroni. Testing the extrapolation quality of word frequency models. In *Proceedings of Corpus Linguistics 2005*, 2006.
- [6] F. Golcher. Statistische Aspekte von Suffixbäumen natürlichsprachiger Texte, Feb. 2005. Thesis for postgraduate studies at the University Munich (in German<sup>4</sup>).
- [7] F. Golcher. A new text statistical measure and its application to stylometry. In *Proceedings of Corpus Linguistics 2007*, to appear.
- [8] R. F. i Cancho and R. V. Solé. Zipf's law and random texts. *Advances in Complex Systems*, 5:1–6, 2002.
- [9] W. Li. Random texts exhibit zipf's-law-like word frequency distribution. *ieeet*, 38(6):1842–1845, 1992.
- [10] A. Lüdeling and S. Evert. Linguistic experience and productivity: Corpus evidence for fine-grained distinctions. In *Proceedings of the 2003 Corpus Linguistics Conference*, Lancaster, 2003.
- [11] F. J. Tweedie and R. H. Baayen. How variable may a constant be? measures of lexical richness in perspective. *Computers and the Humanities*, 32:323–352, 1998.
- [12] E. Ukkonen. On-line construction of suffix-trees. *Algorithmica*, 14(3):249–260, 1995.
- [13] G. K. Zipf. *Human Behavior and The Principle of Least Effort*. Hafner Publishing Company, New York, London, 1949.
- [14] F. Bremer. Koti<sup>5</sup> [online]. July 2005 [cited 03/04/07]. Available from World Wide Web: <http://www.gutenberg.org/dirs/etext04/8phnm10.txt>. Finnish by Alma Suppainen, Original published in 1839.
- [15] F. Dostoevsky. Crime and Punishment (in Russian) [online]. 2004. Available from World Wide Web: <http://lib.ru>. Biblioteka Maksima Moshkova.
- [16] G. Gèza. Egri Csilagok. PDF of unknown origin, 1899.
- [17] H. Kučera and W. N. Francis. *Computational Analysis of Present-Day American English*. Brown University Press, Providence, RI, USA, 1967.
- [18] A. McEnery, P. Baker, R. Gaizauskas, and H. Cunningham. EMILLE: building a corpus of South Asian languages. *Vivek, A Quarterly in Artificial Intelligence*, 13(3):23–32, 2000.
- [19] A. McEnery, Z. Xiao, and L. Mo. Aspect marking in english and chinese: Using the lancaster corpus of mandarin chinese for contrastive language study. *Literary and Linguistic Computing*, 18(4):361–378, 2003.
- [20] M. Proust. Du Côté de Chez Swann [online]. 2001 [cited June 20 2006]. Available from World Wide Web: <http://www.gutenberg.org/etext/2650>. Projekt Gutenberg – plain text version.

<sup>4</sup> Translated title: *Statistical aspects of suffix trees of natural language texts*

<sup>5</sup> Home, Swedish original: *Hemmet*

# Prototyping Efficient Natural Language Parsers

Carlos Gómez-Rodríguez, Jesús Vilares and Miguel A. Alonso  
Universidade da Coruña  
Campus de Elviña s/n  
15071 La Coruña  
{cgomezr, jvilar, alonso}@udc.es

## Abstract

We present a technique for the construction of efficient prototypes for natural language parsing based on the compilation of parsing schemata to executable implementations of their corresponding algorithms. Taking a simple description of a schema as input, Java code for the corresponding parsing algorithm is generated, including schema-specific indexing code in order to attain efficiency.

Key words: parsing schemata, context-free grammars, tree-adjointing grammars

## 1 Introduction

The process of parsing, by which we obtain the structure of a sentence as a result of the application of grammatical rules, is a highly relevant step in the automatic analysis of natural language sentences. In the last decades, various parsing algorithms have been developed to accomplish this task. Although all of these algorithms essentially share the common goal of generating a tree structure describing the input sentence by means of a grammar, the approaches used to attain this result vary greatly between algorithms, so that different parsing algorithms are best suited to different situations.

Parsing schemata, described in [16], provide a formal, simple and uniform way to describe, analyze and compare different parsing algorithms. The notion of a parsing schema comes from considering parsing as a deduction process which generates intermediate results called *items*. An initial set of items is directly obtained from the input sentence, and the parsing process consists of the application of inference rules which produce new items from existing ones. Each item contains a piece of information about the sentence's structure, and a successful parsing process will produce at least one *final item* containing a full parse tree for the sentence or guaranteeing its existence.

In this paper, we will give a brief insight into the concept of parsing schemata by introducing a concrete example: a parsing schema for Earley's algorithm [5]. Given a context-free grammar  $G = (N, \Sigma, P, S)^1$  and a sentence of length  $n$  which we denote by  $a_1 a_2 \dots a_n$ , the schema describing Earley's algorithm is as

<sup>1</sup> Where  $N$  denotes the set of nonterminal symbols,  $\Sigma$  the set of terminal symbols,  $P$  the production rules and  $S$  the axiom.

follows<sup>2</sup>:

*Item set:*

$$\{[A \rightarrow \alpha.\beta, i, j] \mid A \rightarrow \alpha\beta \in P \wedge 0 \leq i < j\}$$

*Initial items (hypotheses):*

$$\{[a_i, i - 1, i] \mid 0 < i \leq n\}$$

*Deductive steps:*

$$\text{EARLEY INITTER: } \frac{}{[S \rightarrow .\alpha, 0, 0]} S \rightarrow \alpha \in P$$

$$\text{EARLEY SCANNER: } \frac{[A \rightarrow \alpha.a\beta, i, j] \quad [a, j, j + 1]}{[A \rightarrow \alpha a.\beta, i, j + 1]}$$

$$\text{EARLEY PREDICTOR: } \frac{[A \rightarrow \alpha.B\beta, i, j]}{[B \rightarrow .\gamma, j, j]} B \rightarrow \gamma \in P$$

$$\text{EARLEY COMPLETER: } \frac{\frac{[A \rightarrow \alpha.B\beta, i, j]}{[B \rightarrow \gamma., j, k]}}{[A \rightarrow \alpha B.\beta, i, k]}$$

*Final items:*

$$\{[S \rightarrow \gamma., 0, n]\}$$

Items in the Earley algorithm are of the form  $[A \rightarrow \alpha.\beta, i, j]$ , where  $A \rightarrow \alpha.\beta$  is a grammar rule with a special symbol (dot) added at some position in its right-hand side, and  $i, j$  are integer numbers denoting positions in the input string. The meaning of such an item can be interpreted as: "There exists a valid parse tree with root  $A$ , such that the direct children of  $A$  are the symbols in the string  $\alpha\beta$ , and the leaf nodes of the subtrees rooted at the symbols in  $\alpha$  form the substring  $a_{i+1} \dots a_j$  of the input string".

The algorithm will produce a valid parse for the input sentence if an item of the form  $[S \rightarrow \gamma., 0, n]$  is generated: according to the aforesaid interpretation, this item guarantees the existence of a parse tree with root  $S$  whose leaves are  $a_1 \dots a_n$ , that is, a complete parse tree for the sentence.

A deductive step  $\frac{\eta_1 \dots \eta_m}{\xi} \Phi$  allows us to infer the item specified by its consequent  $\xi$  from those in its antecedents  $\eta_1 \dots \eta_m$ . *Side conditions* ( $\Phi$ ) specify the valid values for the variables appearing in the antecedents and consequent, and may refer to grammar

<sup>2</sup> From now on, we will follow the usual conventions by which nonterminal symbols are represented by uppercase letters ( $A, B \dots$ ), terminals by lowercase letters ( $a, b \dots$ ) and strings of symbols (both terminals and nonterminals) by Greek letters ( $\alpha, \beta \dots$ ).

rules as in this example or specify other constraints that must be verified in order to infer the consequent.

## 2 Motivation

Parsing schemata are located at a higher abstraction level than algorithms. As can be seen in the example, a schema specifies the steps that must be executed and the intermediate results that must be obtained in order to parse a given string, but it makes no claim about the order in which to execute the steps or the data structures to use for storing the results.

Their abstraction of low-level details makes parsing schemata very useful, allowing us to define parsers in a simple and straightforward way. Comparing parsers, or considering aspects such as their correction and completeness or their computational complexity, also becomes easier if we think in terms of schemata. However, when we want to actually test a parser by running it on a computer and checking its results, we need to implement it in a programming language, so we have to abandon the high level of abstraction and worry about implementation details that were irrelevant at the schema level.

The technique presented in this paper automates this task, by compiling parsing schemata to Java language implementations of their corresponding parsers. The input to the compiler is a simple and declarative representation of a parsing schema, which is practically equal to the formal notation that we used previously. For example, a valid schema file describing the Earley parser is:

```
@goal [ S -> alpha . , 0 , length ]

@step EarleyInitter
----- S -> alpha
[ S -> . alpha , 0 , 0 ]

@step EarleyScanner
[ A -> alpha . a beta , i , j ]
[ a , j , j+1 ]
-----
[ A -> alpha a . beta , i , j+1 ]

@step EarleyCompleter
[ A -> alpha . B beta , i , j ]
[ B -> gamma . , j , k ]
-----
[ A -> alpha B . beta , i , k ]

@step EarleyPredictor
[ A -> alpha . B beta , i , j ]
----- B -> gamma
[ B -> . gamma , j , j ]
```

## 3 Compiling Parsing Schemata

The compilation process, which transforms a declarative description of a parsing schema into a Java implementation of its corresponding parser, proceeds according to the following principles:

- A class is generated for each deductive step in the schema.
- The generated implementation will create an instance of this class for each possible set of values satisfying the side conditions that refer to production rules. For example, a distinct instance of the Earley PREDICTOR step will be created

for each grammar rule of the form  $B \rightarrow \gamma \in P$ , which is specified in the step's side condition.

- The classes representing deductive steps have an `apply` method which tries to apply the deductive step to a given item. If the step is in fact applicable to the item (as determined by checking if the given item matches any of the step's antecedents), the method returns the new items obtained from the inference once all combinations of previously-generated items that satisfy the rest of the antecedents have been found.
- In order for our implementations to maintain the theoretical complexity of parsing algorithms, two distinct kind of indexes are generated for each schema: *existence indexes*, used to check whether an item exists in the item set, and *search indexes*, used to search for items conforming to a given specification. Apart from items, deductive steps are also indexed in *deductive step indexes*. These indexes are used to restrict the set of “applicable deductive steps” for a given item, discarding those known not to match it. Deductive step indexes usually have no influence on computational complexity with respect to input string size, but they do have an influence on complexity with respect to the size of the grammar, since the number of deductive step instances depends on grammar size when production rules are used as side conditions. All the generated indexing code is placed into two classes (the *item handler* and the *deductive step handler*) whose function is to provide efficient access to items and deductive steps, responding to queries issued by the deductive parsing engine. The indexing mechanism is explained in detail in [9].
- The execution of deductive steps in the generated code is coordinated by a *deductive parsing engine* [15] as described by the pseudocode in Figure 1. This is a schema-independent algorithm, and therefore its implementation is the same for any schema. It works with the set of all items that have been generated (either as initial hypotheses or as a result of the application of deductive steps) and an *agenda*, implemented as a queue, which contains the items we have not yet tried to trigger new deductions with. When the agenda is emptied, all possible items will have been generated, and the presence or absence of final items in the item set at this point indicates whether or not the input sentence belongs to the language defined by the grammar.

## 4 Parsing Context-Free Grammars

We have used our technique to generate implementations of three popular parsing algorithms for context-free grammars: CYK [11, 18], Earley and Left-Corner [12].

The schemata we have used describe recognizers, and therefore their generated implementation only

```

steps = {deductive step instances};
items = {initial items};
agenda = [initial items];
for each deductive step with an empty antecedent (s) in steps {
  result = s.apply([]);
  items.add(result);
  agenda.enqueue(result);
  steps.remove(s);
}
while agenda not empty {
  curItem = agenda.removeFirst();
  for each deductive step applicable to curItem (p) in steps {
    result = p.apply(curItem);
    items.add(result);
    agenda.enqueue(result);
  }
}
return items;

```

**Fig. 1:** Pseudocode of the deductive parsing engine

checks sentences for grammaticality by launching the deductive engine and testing for the presence of final items in the item set. However, these schemata can easily be modified to produce a parse forest as output [3]. If we want to use a probabilistic grammar in order to modify the schema so that it produces the most probable parse tree, this requires slight modifications of the deductive engine, since it should only choose the item with the highest probability when several items are available to match an antecedent.

The three algorithms have been tested with sentences from three different natural language grammars: the English grammar from the Susanne corpus [13], the Alvey grammar [4] (which is also an English-language grammar) and the Deltra grammar [14], which generates a fragment of Dutch. The Alvey and Deltra grammars were converted to plain context-free grammars by removing their arguments and feature structures. The test sentences were randomly generated by starting with the axiom and randomly selecting nonterminals and rules to perform expansions, until valid sentences consisting only of terminals were produced. Note that, as we are interested in measuring and comparing the performance of the parsers, not the coverage of the grammars; randomly-generated sentences are a good input in this case: by generating several sentences of a given length, parsing them and averaging the resulting runtimes, we get a good idea of the performance of the parsers for sentences of that length.

For Earley's algorithm, we have used the schema file described earlier. For the CYK algorithm, grammars were converted to Chomsky normal form (CNF), since this is a precondition of the algorithm. In the case of the Deltra grammar, which is the only one of our test grammars containing epsilon rules, we have used a weak variant of CNF allowing epsilon rules. For the Left-Corner parser, the schema used is the *sLC* variant described in [16].

The experiments are described in detail in [8]. The following conclusions can be drawn from them:

- The empirical computational complexity of the three algorithms is below their theoretical worst-case complexity of  $O(n^3)$ , where  $n$  denotes the length of the input string. In the case of the Susanne grammar, the measurements we obtain are close to being linear with respect to string size. In the other two grammars, the measurements grow faster with string size, but are still far below the cubic worst-case bound.
- CYK is the fastest algorithm in all cases, and it generates less items than the other ones. This may come as a surprise at first, as CYK is generally considered slower than Earley-type algorithms, particularly than Left-Corner. However, these considerations are based on time complexity relative to string size, and do not take into account complexity relative to grammar size. In this aspect, CYK is better than Earley-type algorithms, providing linear -  $O(|P|)$  - worst-case complexity with respect to grammar size, while Earley is  $O(|P|^2)$ .<sup>3</sup> Therefore, the fact that CYK outperforms the other algorithms in our tests is not so surprising, as the grammars we have used have a large number of productions. The greatest difference between CYK and the other two algorithms in terms of the amount of items generated appears with the Susanne grammar, which has the largest number of productions. It is also worth noting that the relative difference in terms of items generated tends to decrease when string length increases, at least for Alvey and Deltra, suggesting that CYK could generate more items than the other algorithms for larger values of  $n$ .

<sup>3</sup> It is possible to reduce the computational complexity of Earley's parser to linear with respect to the grammar size by defining a new set of intermediate items and transforming accordingly prediction and completion deduction steps. Even in this case, CYK performs better than Earley's algorithm due to the lower number of items generated:  $O(|N \cup \Sigma| n^2)$  for CYK vs.  $O(|G| n^2)$  for Earley's algorithm, where  $|G|$  denotes the size of the grammar measured as  $|P|$  plus the summation of the lengths of all productions.

- Left-Corner is notably faster than Earley in all cases, except for some short sentences when using the Deltra grammar. The Left-Corner parser always generates fewer items than the Earley parser, since it avoids unnecessary predictions by using information about left-corner relationships. The Susanne grammar seems to be very well suited for Left-Corner parsing, since the number of items generated decreases by an order of magnitude with respect to Earley. On the other hand, the Deltra grammar's left-corner relationships seem to contribute less useful information than the others', since the difference between Left-Corner and Earley in terms of items generated is small when using this grammar. In some of the cases, Left-Corner's runtimes are a bit slower than Earley's because this small difference in items is not enough to compensate for the extra time required to process each item due to the extra steps in the schema, which make Left-Corner's matching and indexing code more complex than Earley's.
- The parsing of the sentences generated using the Alvey and Deltra grammars tends to require more time, and the generation of more items, than that of the Susanne sentences. This happens in spite of the fact that the Susanne grammar has more rules. The probable reason is that the Alvey and Deltra grammars have more ambiguity, since they are designed to be used with their arguments and feature structures, and information has been lost when these features were removed from them. On the other hand, the Susanne grammar is designed as a plain context-free grammar and therefore its symbols contain more information.

## 5 Parsing Tree-Adjoining Grammars

Although all the examples we have seen so far correspond to context-free parsing, our compilation technique is not limited to working with context-free grammars, since parsing schemata can be used to represent parsers for other grammar formalisms as well. All grammars in the Chomsky hierarchy can be handled in the same way as context-free grammars, and other formalisms can be added by defining element classes for their rules using the extensibility mechanism included in the system for defining new kinds of objects to use in schemata. The code generator can deal with these user-defined objects as long as some simple and well-defined guidelines are followed in their specification.

In particular, we have also used our system to generate parsers for tree-adjoining grammars [10]. A tree-adjoining grammar (TAG) includes a set of *elementary trees* of arbitrary depth which can be combined by using the *substitution* and *adjunction* operations. The substitution operation is used to substitute an elementary tree for a leaf node (which must be labelled as a *substitution node*) in another elementary tree. The adjunction operation allows us to insert an *auxiliary*

*tree* (an elementary tree with a distinguished frontier node, called the *foot node* and labelled with the same nonterminal as its root) into another elementary tree.

The possibility of using elementary trees of arbitrary depth and the adjunction operation provide an extended domain of locality with respect to context-free grammars, and the set of languages which can be recognized with TAG is a strict superset of context-free languages. This makes TAG an interesting formalism for natural language parsing, since some phenomena present in natural languages cannot be represented by context-free grammars.

We have used our compiler to generate implementations for some of the most popular parsers for tree adjoining grammars [1, 2]: a CYK-based algorithm, two extensions of Earley's algorithm with and without the valid prefix property, and Nederhof's parsing algorithm. These implementations were tested both with artificially-generated grammars and a real-life, wide-coverage Feature-Based Tree Adjoining Grammar: the XTAG English grammar [17].

The TAG parsing schemata can be written in a format readable by our compiler in the same way as the context-free parsing schemata seen in the previous sections. Although the main constituents of TAG's are elementary trees instead of productions, each elementary tree may be expressed as a set of productions which can be used as side conditions for deductive steps. In order for the steps to be able to check whether the adjunction or substitution operation is allowed at a given node, we define boolean expressions that query the grammar for this information. In the case of the XTAG, we also need to include feature structures inside items and add unification operations to the deductive steps.

The performance results obtained from TAG parsers show that both string length and grammar size can be important factors in performance, and the interactions between them sometimes make their influence hard to quantify. The influence of string length in practical cases is usually below the theoretical worst-case bounds (we found the empirical complexity to be around  $O(n^3)$ , while the worst-case bound for these TAG parsers is  $O(n^6)$ ). Grammar size becomes the dominating factor in large TAG's such as the XTAG, making tree filtering techniques advisable in order to achieve faster execution times.

By comparing performance of TAG and CFG parsers on artificially-generated grammars generating the same languages, we could see that using TAG's to parse context-free languages causes a significant overhead both in practical computational complexity and in constant factors, increasing execution times by several orders of magnitude with respect to CFG parsers.

A detailed explanation of the performance results obtained by applying our compilation technique to TAG parsers can be found at [6, 7].

## 6 Conclusions and future work

The construction of efficient prototypes directly from parsing schemata is very useful for the design, analysis and comparison of parsing algorithms, as it allows us to test them and check their results and performance

without having to implement them in a programming language. As we have seen by comparing the performance of several well-known parsers for natural language grammars (context-free grammars and tree-adjoining grammars), not all algorithms are equally suitable for all grammars. In this work we provide a quick way to evaluate several parsing algorithms in order to find the best one for a particular application.

Currently, we are applying our compilation technique to automatically derive robust, error-correcting parsers from standard parsers for context-free grammars and tree adjoining grammars.

## Acknowledgments

Supported in part by Ministerio de Educación y Ciencia (MEC) and FEDER (TIN2004-07246-C03-01, TIN2004-07246-C03-02), Xunta de Galicia (PGIDIT05PXIC30501PN, PGIDIT05PXIC10501PN, Rede Galega de Procesamento da Linguaxe e Recuperación de Información) and Programa de Becas FPU (MEC).

## References

- [1] M.A. Alonso, D. Cabrero, E. de la Clergerie, and M. Vilares. Tabular algorithms for TAG parsing. In *Proc. of EACL'99*, pages 150–157, Bergen, Norway, 1999.
- [2] M. A. Alonso, E. de la Clergerie, V. J. Díaz and M. Vilares. Relating tabular parsing algorithms for LIG and TAG. In H. Bunt, John Carroll and G. Satta (eds.), *New Developments in Parsing Technology*, pages 157-184, Kluwer Academic Publishers, Dordrecht-Boston-London, 2004.
- [3] S. Billot and B. Lang. The structure of shared forest in ambiguous parsing. In *Proc. of ACL'89*, pages 143–151, Vancouver, British Columbia, Canada, 1989.
- [4] J.A. Carroll. Practical unification-based parsing of natural language. Technical Report no. 314, University of Cambridge, Computer Laboratory, England. PhD Thesis., 1993.
- [5] J. Earley. An efficient context-free parsing algorithm. *Communications of the ACM*, 13(2):94–102, 1970.
- [6] C. Gómez-Rodríguez, M.A. Alonso, and M. Vilares. On theoretical and practical complexity of TAG parsers. In P. Monachesi, G. Penn, G. Satta and S. Wintner (eds.), *FG 2006: The 11th conference on Formal Grammar. Malaga, Spain, July 29-30, 2006*, chapter 5, pp. 61-75, CSLI, Stanford, 2006.
- [7] C. Gómez-Rodríguez, M.A. Alonso, and M. Vilares. Generating XTAG parsers from algebraic specifications. In *Proceedings of the 8th International Workshop on Tree Adjoining Grammar and Related Formalisms. Sydney, July 2006*, pp. 103-108, Association for Computational Linguistics, East Stroudsburg, PA, 2006.
- [8] C. Gómez-Rodríguez, J. Vilares and M. A. Alonso. Compiling Declarative Specifications of Parsing Algorithms. In *Database and Expert Systems Applications*, volume of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin-Heidelberg-New York, 2007.
- [9] C. Gómez-Rodríguez, M.A. Alonso, and M. Vilares. Generation of indexes for compiling efficient parsers from formal specifications. In R. Moreno-Díaz, F. Pichler, and A. Quesada-Arencibia (eds.), *Computer Aided Systems Theory*, volume of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin-Heidelberg-New York, 2007.
- [10] A.K. Joshi and Y. Schabes. Tree-adjoining grammars. In G. Rozenberg and A. Salomaa, eds, *Handbook of Formal Languages. Vol 3: Beyond Words*, pages 69–123. Springer-Verlag, Berlin/Heidelberg/New York, 1997.
- [11] T. Kasami. An efficient recognition and syntax algorithm for context-free languages. Scientific Report AFCRL-65-758, Air Force Cambridge Research Lab., Bedford, Massachusetts, 1965.
- [12] D. J. Rosenkrantz and P. M. Lewis II. Deterministic Left Corner parsing. In *Conference Record of 1970 Eleventh Annual Meeting on Switching and Automata Theory*, pages 139–152, Santa Monica, CA, USA, 1970.
- [13] G. Sampson. The Susanne corpus, Release 3, 1994.
- [14] J. J. Schoorl and S. Belder. Computational linguistics at Delft: A status report, Report WT-M/TT 90–09, 1990.
- [15] S.M. Shieber, Y. Schabes, and F.C.N. Pereira. Principles and implementation of deductive parsing. *Journal of Logic Programming*, 24(1–2):3–36, 1995.
- [16] K. Sikkell. *Parsing Schemata — A Framework for Specification and Analysis of Parsing Algorithms*. Springer-Verlag, Berlin/Heidelberg/New York, 1997.
- [17] XTAG Research Group. A lexicalized tree adjoining grammar for English. Technical Report IRCS-01-03, IRCS, Univ. of Pennsylvania, 2001.
- [18] D. H. Younger. Recognition and parsing of context-free languages in time  $n^3$ . *Information and Control*, 10(2):189–208, 1967.

# Evaluating Wrapped Progressive Sampling for Automatic Algorithmic Parameter Optimisation

Hendrik J. Groenewald, Gerhard B. van Huyssteen and Martin J. Puttkammer  
Centre for Text Technology (CTeXt)  
North-West University  
Potchefstroom 2531, South Africa  
{handre.groenewald, gerhard.vanhuyssteen, martin.puttkammer}@nwu.ac.za

## Abstract

Determining the algorithmic parameter combinations that deliver the best performance in applications using machine learning algorithms is a very important part in the development process. Exhaustive searches are slow and computationally expensive, which motivates the investigation of more efficient methods of automatic algorithmic parameter optimisation. Wrapped progressive sampling is one such a method and is utilised in a tool named *Paramsearch*. An alternative method for determining the sizes of the progressive datasets used in the wrapped progressive sampling procedure is proposed and implemented as *PSearch*. *PSearch* and *Paramsearch* are evaluated and compared to an exhaustive search on the tasks of lemmatisation and hyphenation in Afrikaans. Results indicate that both *PSearch* and *Paramsearch* are generally more efficient in terms of execution time and computational resources than an exhaustive search. It is also shown that *PSearch* delivers more accurate results than *Paramsearch* on the tasks of lemmatisation and hyphenation in Afrikaans.

## Keywords

optimisation, machine learning, parameters, lemmatisation, hyphenation.

## 1. Introduction

It is a well-known fact that changing the parameter settings of machine learning algorithms causes large fluctuations in generalisation accuracy [1]. The default parameter settings for machine learning algorithms are not guaranteed to deliver the best performance, while estimating for forecasting the best performing parameters is also generally very hard, due to the complexity of parameter interactions.

One way of finding the best algorithmic parameter settings is to perform an exhaustive search throughout all of the possible combinations of parameter settings. However, this approach is time-consuming and computationally intensive, because the system has to be retrained for every possible parameter setting of the involved algorithm. For instance, the Tilburg Memory-Based Learner [2], a machine learning system, has five algorithms, six distance metrics, five feature weighting possibilities and three class voting weights. The number of nearest neighbours to consider can also be defined for some algorithms. This will amount to circa 4,500 ( $5 \times 6 \times 5 \times 3 \times 10$ ) different combinations of parameter settings, which implies

that the system has to be retrained 4,500<sup>1</sup> times to evaluate the performance of every combination. Such an exhaustive search can be expected to be a lengthy operation; for example, an exhaustive search throughout all of the valid combinations of algorithmic parameters for one machine learning algorithm on the training data of the Afrikaans lemmatiser took 176 hours and 28 minutes to complete.

The purpose of this study is to investigate alternative, more efficient ways of parameter optimisation than exhaustive searches, in order to obtain the best performing algorithmic parameter combinations for two Afrikaans core technologies (viz. a lemmatiser and a hyphenator), when trained on two machine learning algorithms.

The next section focuses on wrapped progressive sampling as a method for parameter optimisation. *Paramsearch* and *PSearch*, two tools for automatic parameter optimisation are also introduced. In Section 3, various aspects of the performance of *Paramsearch* and *PSearch* are evaluated and compared in terms of classifier ranking, accuracy and execution time. The article ends with some general concluding remarks and suggestions for future work in Section 4.

## 2. Wrapped Progressive Sampling

### 2.1 Paramsearch

As an alternative approach to an exhaustive search, Van den Bosch [3] developed a tool (*Paramsearch*) that produces combinations of algorithmic parameters that are estimated to deliver best results. *Paramsearch* implements wrapped progressive sampling (WPS), a combination of classifier wrapping [4] and progressive sampling [5], for data sets containing more than a 1,000 instances. WPS is currently not widely used in the field of natural language processing, but it could be of great advantage if implemented for applications involving the machine learning of natural language.

The general idea behind *Paramsearch* is to determine the best performing algorithmic parameter settings through

---

<sup>1</sup> This number of experiments is only for purposes of illustration as some of the parameter settings can only be used in conjunction with certain algorithms.



competitions among all the possible combinations of parameter settings, based on their performance evaluated on smaller subsets of the original dataset. The process starts by randomising and dividing the original dataset into training sets (80% the size of the original dataset) and evaluation sets (20% the size of the original data set). The system is then trained with all of the valid parameter combinations on the first subset (containing 500 instances) of the original dataset. The worst performing parameter combinations are discarded after this step, and the size of the training data set is increased. This process of discarding the worst performing parameter combinations and increasing the size of the training data set is continued until only one setting is left, or until the largest available data set has been used for training.

The sizes of the data sets are generated according to the following three-step procedure [6]:

**Step 1:** Let  $n$  be the number of instances in the data set used for training (80% of the original data set). A quadratic sequence of 20 data sets is created by using a factor  $f$  as in Equation 1:

$$f = \sqrt[20]{n} \quad (1)$$

**Step 2:** A sequence of  $i = \{1 \dots d\}$  data sets is generated containing  $size_i$  number of instances each. For  $i=1$ ,  $size_1 = 1$  and then for every  $i > 1$ , the number of instances contained is defined as:

$$size_i = size_{i-1} * f \quad (2)$$

**Step 3:** The data sets are then limited to those containing more than 500 instances. A data set of 500 instances is used as the first set. An evaluation set, 20% the size of every generated data set, is also generated for every data set. The evaluation sets are extracted from the 20% of the original data set used for evaluation.

Daelemans and Van den Bosch [6] have evaluated *Paramsearch* on a number of machine learning algorithms and NLP tasks, such as named entity recognition and Dutch morphological analysis. Their main finding was that *Paramsearch* does not produce much effect with algorithms that have little variation in their parameters.

Experimenting with *Paramsearch* on the training data of the Afrikaans lemmatiser indicated that 99% of the parameter settings were discarded based on only 2% of the available training data (see Table 1 below). This raised suspicion about the ability of *Paramsearch* to deliver the best performing algorithmic parameter combinations.

**Table 1. Relation between data set size and number of parameter settings evaluated by *Paramsearch***

Data set #	Size (# Instances)	# Evaluated Parameter Settings
1	500	925
2	720	232
3	1,245	13
4	2,154	3
5	3,727	3
6	6,448	1

The problem here is that the small differences between the sizes of the progressive data sets generated at the start of the procedure cause a large number of parameter settings to be filtered out at the start of the process, when the sizes of the subsets are still relatively small in comparison to the original data set. Some of the parameter settings that are filtered out in the beginning could possibly include settings that could have performed very well later on in the process when more data is used for training purposes.

## 2.2 PSearch

In order to overcome the problem relating to the small differences in the progressive data sets and the fact that *Paramsearch* is only available for two of the five TiMBL classification algorithms, we created our own implementation of *Paramsearch*, which we call *PSearch*. *PSearch* operates on the same principles as the original *Paramsearch*, with the major differences being the way that the sizes of the training and evaluation sets are generated and the number of algorithmic parameter combinations that are discarded after each step in the WPS process. In addition, *PSearch* supports all of the TiMBL classification algorithms, as well as C4.5 [7].

The approach followed by *PSearch* is to discard equations 1 and 2 above for the generation of the progressive data set sizes and instead define the sizes of the progressive data sets as percentages of the size of the available data set. In this way, the 80% of the original data set (i.e. the training set) is divided into 8 subsets, which are calculated<sup>2</sup> as follows:

- $Size_{(Subset\ 1)} = 0.01 \times Size_{(Original\ data\ set)}$
- $Size_{(Subset\ 2)} = 0.02 \times Size_{(Original\ data\ set)}$
- $Size_{(Subset\ 3)} = 0.04 \times Size_{(Original\ data\ set)}$
- $Size_{(Subset\ 4)} = 0.05 \times Size_{(Original\ data\ set)}$

<sup>2</sup> The percentages used in calculating the new sizes of the datasets were iteratively determined on the training data of the Afrikaans lemmatiser.

- $\text{Size}_{(\text{Subset } 5)} = 0.3 \times \text{Size}_{(\text{Original data set})}$
- $\text{Size}_{(\text{Subset } 6)} = 0.65 \times \text{Size}_{(\text{Original data set})}$
- $\text{Size}_{(\text{Subset } 7)} = 0.85 \times \text{Size}_{(\text{Original data set})}$
- $\text{Size}_{(\text{Subset } 8)} = 1 \times \text{Size}_{(\text{Original data set})}$

Another difference between *PSearch* and *Paramsearch* is that *PSearch* limits the number of algorithmic parameter setting combinations that are discarded after each iteration when the data set size is enlarged. The number of parameter settings is limited to prevent any excessive decreases occurring in a single iteration. If the number of "surviving" parameter combinations is less than 20% of the number of parameter setting combinations in the previous iteration, the parameter settings in the preceding bin are also included. This process of including more bins of algorithmic parameter settings is continued until the number of "surviving" parameter settings is larger than, or equal to 20% of the number of parameter setting combinations evaluated in the previous round. The advantage of this process is that it prevents *PSearch* from discarding large numbers of parameter settings based on their performance during the early stages of the WPS procedure when the data set sizes are still relatively small.

Table 2 displays the new sizes of the progressive data sets, as well the number of parameter settings evaluated by *PSearch*.

**Table 2. Relation between data set size and number of parameter settings evaluated by *PSearch***

Data set #	Size (# Instances)	# Evaluated Parameter Settings
1	800	925
2	1,500	630
3	2,000	238
4	4,000	135
5	15,000	114
6	30,000	72
7	40,000	33
8	57,781	6

Compared to Table 1, Table 2 shows a more gradual decrease in the number of evaluated parameter settings and therefore represents a much more desirable situation, as the chances of good performing parameter settings being eliminated at the beginning of the process are decreased.

In the next section, performance of *Paramsearch* and *PSearch* are compared in terms of ranking of best classifiers, accuracy of best-ranked parameter settings compared to default settings, and execution time.

### 3. Comparing *Paramsearch* and *PSearch*

The performance of *Paramsearch* and *PSearch* is compared in this section on a lemmatisation (section 3.1) and a hyphenation task in Afrikaans (section 3.2). Both *Paramsearch* and *PSearch* were used to generate combinations of algorithmic parameter settings that are expected to do well on these tasks. These combinations of algorithmic parameters are compared to the results obtained from exhaustive searches. The exhaustive searches were performed throughout all the valid parameter combinations, as tested by *Paramsearch* and *PSearch* for IB1 (the default *k*-Nearest Neighbour algorithm in TiMBL [2]) and C4.5 [7]. 925 different combinations of parameter settings were tested in the IB1 exhaustive search, with 180 combinations for C4.5.

#### 3.1 Lemmatisation

The training data for the lemmatisation task consist of 72,226 instances, with 20 features each. Every instance represents a lemmatised word and the features are made up of letter sequences. The data contains 278 classes, containing information for transforming the inflected word-form to its linguistically correct lemma.

#### Ranking

Table 3 shows a comparison of the rankings produced by *PSearch* and *Paramsearch* for the IB1 and C4.5 algorithms, evaluated on the training data of the Afrikaans lemmatiser. The ranks as calculated in the exhaustive searches are displayed in Table 3 for the five best combinational settings as predicted by *Paramsearch* and *PSearch*. The exhaustive search ranking signifies a position out of 925 for IB1 and a position out of 180 for C4.5. Thus, a ranking of 1 signifies the best combinational setting out of 925 (or 180 in the case of C4.5), while a ranking of 925 signifies the combinational setting with the lowest accuracy score. Consider for example the combinational setting ranked by *Paramsearch* as the best performing setting for IB1, which in fact achieved a ranking of 271 out of 925 (see Table 3).

**Table 3. Comparison of the rankings produced by *PSearch* and *Paramsearch***

Predicted Ranking	Exhaustive Search Rankings ( <i>Paramsearch</i> )		Exhaustive Search Rankings ( <i>PSearch</i> )	
	IB1	C4.5	IB1	C4.5
1	271	88	1	55
2	149	126	2	56
3	154	135	7	57
4	88	153	11	58
5	89	169	12	59

The results in Table 3 indicate that *PSearch* delivers combinational settings with higher rankings than *Paramsearch*. *PSearch* even delivered the combinational settings ranked 1 and 2 in the case of IB1, which shows that *PSearch* can be assumed to be more suitable than *Paramsearch* for producing good performing algorithmic parameter settings for the lemmatisation task in Afrikaans, especially when used for IB1. A reason for the good performance of *PSearch* is that the larger datasets used at the start of the WPS procedure reduce the chance of discarding settings that performs badly on small amounts of training data, but performs better on larger datasets.

#### Best setting compared to default setting

Table 4 shows a comparison based on accuracy between the best settings predicted by *Paramsearch* and *PSearch* and the default setting of the involved algorithm. The percentage error reduction is measured as the percentage of error that was saved by *PSearch* or *Paramsearch*. The percentage error reduction may be negative if the predicted settings perform worse than the default setting.

**Table 4. Predicted best settings compared to default setting (Lemmatization)**

	Best Setting Accuracy	Default Setting Accuracy	% Error Reduction
<i>Paramsearch</i> (IB1)	92.40%	91.36%	12.03
<i>PSearch</i> (IB1)	92.80%	91.36%	16.66
<i>Paramsearch</i> (C4.5)	88.41%	90.81%	-26.09
<i>PSearch</i> (C4.5)	91.21%	90.81%	4.35

The results in Table 4 indicate that the best settings predicted by *Paramsearch* and *PSearch* deliver better results than the default settings, except in the case of C4.5 where the error percentage was in fact increased by the setting produced by *Paramsearch*.

#### Execution Time

**Table 5. Comparison of *PSearch* and *Paramsearch* execution times to an exhaustive search (Lemmatization)**

	Execution Time (min)	
	IB1	C4.5
<i>Paramsearch</i>	8.75	0.05
<i>PSearch</i>	29.5	0.78
Exhaustive Search	10 588.00	41.1

Table 5 illustrates the significant advantage in terms of execution times that *PSearch* and *Paramsearch* have over an exhaustive search. *Paramsearch* is also much faster than *PSearch*, but this difference seems to be of less importance

when considering the fact that *PSearch* is more likely to produce better results than *Paramsearch* as far as the lemmatisation task is concerned. The difference in execution times of *PSearch* and *Paramsearch* can be accredited to the larger subsets utilised by *PSearch* and the fact the *PSearch* limits the number of settings that can be discarded after each iteration in the WPS procedure.

### 3.2 Hyphenation

The comparisons of the previous section are repeated for an Afrikaans hyphenation task. The hyphenation training data consist of 100,000 instances. A context window of 6 characters was used, resulting in 12 features. There are only two classes in the training data, indicating whether the word should be hyphenated at a certain position or not.

#### Ranking

**Table 6. Comparison of the rankings produced by *PSearch* and *Paramsearch* (Hyphenation)**

Position as Ranked by <i>PSearch</i> and <i>Paramsearch</i>	Exhaustive Search rankings for settings produced by <i>Paramsearch</i>		Exhaustive Search rankings for settings produced by <i>PSearch</i>	
	IB1	C4.5	IB1	C4.5
1	8	50	1	19
2	9	80	3	22
3	5	74	2	21
4	21	109	4	15
5	25	137	13	28

Table 6 shows that *PSearch* performed better than *Paramsearch* at the hyphenation task for both algorithms. *PSearch* was less successful at predicting the best performing algorithmic combinations for C4.5 when compared to the good results obtained for IB1, but nevertheless still outperformed *Paramsearch*.

#### Best setting compared to default setting

Table 7 indicates that the default parameter settings of IB1 and C4.5 perform better than the settings predicted as the best by *Paramsearch*. The reason for this is that the default settings performed badly on the first data sets generated at the start of the WPS procedure and was accordingly discarded at an early stage of the procedure by *Paramsearch*. Table 7 further shows that the *PSearch* approach of enlarging the sizes of the datasets used early on in the WPS process and limiting the number of parameter setting combinations that are discarded after each iteration have a positive effect on the results, since *PSearch* was able to predict parameter settings that perform better than the default settings.

**Table 7. Predicted best settings compared to default setting (Hyphenation)**

	Best Setting Accuracy	Default Setting Accuracy	% Error Reduction
<i>Paramsearch</i> (IB1)	98.2%	98.32%	-7.14
<i>PSearch</i> (IB1)	98.39%	98.32%	4.56
<i>Paramsearch</i> (C4.5)	96.59%	97.21%	-25
<i>PSearch</i> (C4.5)	97.91%	97.21%	25

#### Execution Time

**Table 8. Comparison of the rankings produced by *PSearch* and *Paramsearch* execution times to an exhaustive search (Hyphenation)**

	Execution Time (min)	
	IB1	C4.5
<i>Paramsearch</i>	2.51	0.05
<i>PSearch</i>	38.23	1.57
Exhaustive Search	952.1	41.60

The results in Table 8 show the same trend as the results in Table 5. The execution times of *Paramsearch* are generally much faster than that of *PSearch*, but these differences in execution time seem to be very small in comparison to the execution times of an exhaustive search.

## 4. Conclusion

Results indicate that both *PSearch* and *Paramsearch* can be used to predict good performing algorithmic parameter combinations and that both these two methods are more efficient in terms of execution time than an exhaustive search. *PSearch* did however perform better than *Paramsearch* on both the lemmatisation and hyphenation tasks. An important result is that although the sizes of the progressive datasets as utilised by *PSearch* were customised with the lemmatisation task in mind, good results were obtained when applying *PSearch* to the hyphenation task.

The most important advantage of WPS remains the significant reduction in execution time when compared to an exhaustive search. In this sense *PSearch* has the disadvantage that it has a longer execution time than *Paramsearch*. The difference in execution time can be attributed to the larger sizes of the progressive datasets employed by *PSearch*, as well as the fact that *PSearch* limits the number of combinations of parameter settings than are discarded after every step in the WPS procedure.

The execution time of *PSearch* can however be considered relatively small when compared to the execution time of an exhaustive search.

The performance of *PSearch* and *Paramsearch* seems to be dependent on the machine learning algorithm of choice, the sizes of the progressive datasets, the interactions between these variables and also on the structure of the training data (i.e. features, number of classes etc.). Controlling and predicting these interactions are a difficult task and provides further motivation for experimenting with the sizes of the progressive datasets in the WPS procedure when different machine learning algorithms and tasks are involved.

Determining the best performing parameter combinations in an effective manner remains an important part of the development process of applications using machine learning algorithms. Future work is necessary to determine the relations between the sizes of the progressive training sets employed by WPS and the other variables that may affect the performance of the algorithms. This can be done by extending *PSearch* to more machine learning algorithms and evaluating the performance of *PSearch* on alternative classification tasks than lemmatisation and hyphenation.

## 5. Acknowledgements

The authors would like to acknowledge the inputs and assistance received from Antal van den Bosch.

## 6. References

- [1] A. van den Bosch. Wrapped Progressive Sampling Search for Optimizing Learning Algorithm Parameters. Proceedings of the 16<sup>th</sup> Belgian-Dutch Conference on Artificial Intelligence, R. Verbrugge, N. Taatgen and L. Schomaker, Eds. 2004.
- [2] W. Daelemans, A. van den Bosch, J. Zavrel and K. van der Sloot. TiMBL: Tilburg Memory Based Learner, Version 5.1, Reference Guide, ILK Technical Report 04-02. Tilburg, 2004.
- [3] A. van den Bosch. Paramsearch 1.0 Beta Patch 24. Available: <http://ilk.uvt.nl/software.html#paramsearch>
- [4] R. Kohavi and G.H. John. Wrappers for Feature Subset Selection. Artificial Intelligence Journal, 1997.
- [5] F. Provost, D. Jensen and T. Oates. Efficient Progressive Sampling. Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, 1999.
- [6] W. Daelemans and A. van den Bosch. Memory-based Language Processing. Cambridge University Press, Cambridge, 2005.
- [7] J.R. Quinlan. Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA, 1993.

# Exploiting glossaries for automatic terminology processing

Le An Ha

Research Group in Computational Linguistics,  
Research Institute in Information and Language Processing  
University of Wolverhampton, Stafford Street, Wolverhampton  
WV1 1SB, UK  
L.A.Ha@wlv.ac.uk

## Abstract

This paper analyses a valuable but forgotten resource in automatic terminology processing (ATP): glossaries. It argues that glossaries are widely available, especially on the Internet, and that they contain valuable terminological knowledge which can be exploited by automatic procedures. The empirical analysis of a set of glossaries collected from the Internet substantiates these arguments. The paper also presents a method to extract knowledge patterns from glossaries. An evaluation is then performed showing the usefulness of the extracted patterns in ATP. In two experimented domains, the improvements are 5% and 16% over f-measures respectively. The paper concludes that glossaries should be further studied and exploited.

## Keywords

Automatic terminology processing, automatic term extraction, pattern extraction, pattern heuristics.

## 1. Introduction

Automatic terminology processing (ATP) concerns the deployment of automatic or semi-automatic procedures to build, maintain, and exploit terminologies. Although generic ATP methods have been proposed ([10]), adapting them to different domains remains a challenge. To address this problem, it has been argued that glossaries (lists of the most important terms relating to a specific domain, together with their brief definitions or explanations) can be used. This is because glossaries are widely available and contain domain-specific terminological knowledge that, if extractable, can be used by ATP engines.

This paper discusses which terminological knowledge can be extracted from a glossary, and how. We first discuss what glossaries are, their features, and a sample set of 7 glossaries collected from the Internet (Section 2). The types of terminological knowledge which glossaries contain and how they can be retrieved (Section 3) are then discussed. An evaluation will show the extent to which the extracted knowledge is useful for ATP (Section 4). Conclusion will be found in Section 5.

## 2. Glossaries and their features

### 2.1 Definitions and usages

In contrast to other terminological resources, there is a lack of studies on glossaries in terminology processing literature. More often, authors discuss dictionaries or encyclopaediae

([15], [16], and [19]), possibly because they consider glossaries to be similar to them.

Definitions of a glossary generally agree that it is a list of technical terms along with their brief explanations, and that glossaries can be used for alternative purposes such as a reference point of a book, a common terminology for internal communication of a company, or a place where explanations of jargon used on a website can be found.

### 2.2 Glossaries in the information era

In this information and knowledge era, the general public constantly exploits increasingly available resources for their own needs using the Internet and the World Wide Web (WWW). Just as with books, there may be several terms on a website with which some readers are not familiar, and a short explanation is needed. Recognising this need, website authors put glossaries onto their websites and enhance them with features provided by the Internet and WWW such as hyperlinks, multimedia presentations, and search facilities. Search engines (such as Google) have developed search features which exploit available glossaries to allow users to find definitions of words and phrases; this confirms that there are both supply of and demand for glossaries.

### 2.3 Collecting glossaries from the Internet

To confirm the hypothesis that glossaries contain valuable terminological knowledge which can be extracted and used in ATP, we collected a set of glossaries from various domains from the Internet to be used for empirical analysis and to design algorithms to extract terminological knowledge. To do this, firstly we searched Google for the keyword "glossary", and obtained the glossaries from the first 100 results. In this study, we discuss a set of 7 glossaries covering 7 different domains. Domains and descriptions of the selected ones are shown in Table 1.

Glossaries on the WWW are presented in different ways and formats, varying from plain text only to those that use every available hypertext features. The hypertext markup language ([21]) does provide a set of html tags to be used to mark terms and definitions; they are <DL>, <DT>, and <DD>. When this tagset is used, terms are usually highlighted in bold and their definitions indented. However, not all compilers of Internet glossaries are aware of, or want to use, this tagset. In our sample set, three glossaries use other html tags to identify terms and explanations.

**Table 1: Domains and descriptions of the collected glossaries**

Glossary	Description	Domain
JAVA	Java Reference	Java
WEATHER	National Weather Service Glossary	Weather
CANCER	CancerhelpWebsite glossary	Cancer
UNICODE	The Unicode Standard, Version 4.0 glossary	Unicode Standard
CITIZEN	Glossary and Acronym of the U.S. Citizenship and Immigration services	US Citizenship
CHEMISTRY	General Chemistry Glossary	Chemistry
WATER	Water Science Glossary of Terms	Water science

## 2.4 Features of glossaries

Empirical analysis shows that glossaries' features can be divided into two categories: i) essential features: required in order for something to be considered a glossary; ii) supplementary features: used to enrich glossaries. Essential features of a glossary include: i) a list of terms; ii) a short description attached to each term (which will be referred to as a gloss<sup>1</sup>); iii) a method to quickly search for entries. The list of terms contain entries which the authors think is important, or worthy of inclusion; the numbers of entries in our set vary from 121 (CITIZEN) to 2641 (WEATHER). Each entry in a glossary is followed by a gloss (short description). Entries in glossary are sorted alphabetically help searching. Supplementary features of glossaries include: i) cross references: provide reader with references to other relevant terms, and ii) multimedia presentation: used to present information using audio and animated visual effects. In our sample set of glossaries, all but WEATHER provide hyperlinked cross references to other relevant terms. Only CHEMISTRY has multimedia presentation.

The glosses in a glossary constitute its most important parts. A gloss (of an entry) provides description, explanation, or any information the author thinks may help readers to understand the entry quickly. The following extract is an example of glosses.

**absolute temperature.** Temperature measured on a scale that sets absolute zero as zero. In the SI system, the Kelvin scale is used to measure absolute temperature. (CHEMISTRY)

Generally speaking, a gloss is different from a definition found in a dictionary or encyclopaedia. Written by domain experts rather than lexicographers, they tend to be more informal. A gloss can also be considered a summary of information that would provide readers with concise knowledge of the term ([8]). A summary of various word statistics on glosses can be found in Table 2. Average numbers of sentences per gloss vary from 2.4 to 3.3; average numbers of words per gloss: 26.8 - 66.6.

<sup>1</sup> We borrow the term "gloss" from WordNet ([6]) to describe the information attached to a term in a glossary. Originally, glossary meant a collection of glosses, and glosses were notes made in the margins or between the lines of a book ([14]).

**Table 2: Number (#) of sentences and words in glosses**

Glossary	Total # of sentences	Aver # sentences/gloss	Number of words	Aver # words/gloss
JAVA	684	2.67	6860	<b>26.80</b>
WEATHER	6353	<b>2.40</b>	78827	29.85
CANCER	3515	2.70	34229	54.19
UNICODE	858	2.47	10095	29.09
CITIZEN	393	<b>3.25</b>	8053	<b>66.55</b>
CHEMISTRY	3010	2.88	35159	33.68
WATER	458	2.99	6190	40.46

## 3. Exploiting glosses

Given that glosses are used to explain the meaning of their entries and to provide important information about them, we first discuss several studies of definitions in the field of terminology processing. We then discuss methods to extract useful terminological knowledge from glosses.

### 3.1 The study of definitions in terminology processing literature

According to [2], [9], [15], and [20], the classic formula for a definition is  $X = Y + \text{distinguishing characteristics (differentia)}$ , in which  $X$  is the entry, and  $Y$  is a genus<sup>2</sup> term superordinating  $X$ . The differentia differentiates  $X$  from other concepts in the domain. Swales ([20]) has argued that the definition formula is often realised using a set of linguistic patterns; most of these patterns occur in our selected glossaries.

### 3.2 Pre-processing glosses

Parsing technologies allow us to analyse glosses quickly without a great deal of errors. Parsers such as that of [3] provide a reasonably accurate shallow syntactical analysis of a sentence. Using the output of [3], we can analyse the collected glosses in terms of sentence structure as well as the head words of these structures.

We use the parser to process the selected glossaries. The parser's outputs have proved to be sufficient, apart from some consistent errors fixed by post-processing rules. Analysing glosses using parser's outputs also provides an indication of parser's performance in ATP tasks.

Using the parser output, the genus terms (in the definition formula) can be retrieved. The genus terms are located in the 'first sentence' of a gloss. If the 'first sentence' of a gloss is an NP, the genus is its head (e.g. *Temperature measured on a scale that sets absolute zero as zero*: temperature). If the 'first sentence' is a complete sentence, the genus is often the head of the argument of the copular verb (*Visible light is electromagnetic radiation with a wavelength between 400 and 750 nm*: radiation). Table 3 presents the ten most used genus terms extracted from selected glossaries.

<sup>2</sup> The terms "genus" and "differentia" are borrowed from [2].

**Table 3: Ten most popular genus terms extracted from glossaries**

Glossary	Genus terms
JAVA	keyword, protocol, method, item, system, class, language, unit, type, definition
WEATHER	system, model, time, area, wind, term, cloud, center, product, instrument
CANCER	cancer, operation, treatment, cell, lymphoma, drug, doctor, tube, substance, disease
UNICODE	character, acronym, synonym, standard, sequence, name, system, script, set, collection
CITIZEN	alien, category, limit, child, provision, number, immigrant, person, public, law
CHEMISTRY	substance, compound, reaction, example, unit, element, change, prefix, acid, process
WATER	water, process, substance, rock, term, unit, measure, amount, feature, system

### 3.3 Differentiae, use of verbs, and knowledge patterns in glosses

In glosses, often there is no differentia element (as in the classic formula of definition). Rather, there is an explanation why the term is important. The following example illustrates this observation.

**Alpha-Fetoprotein (AFP).** Substance found in the bloodstream of some men with testicular cancer. The level rises when the cancer is growing and falls when the cancer is shrinking. ... (CANCER)

In this example, it can be argued that there are many substances which can be found in the bloodstream of men with testicular cancer. Thus, the first ‘sentence’ explains why *AFP* is important in Cancer rather than trying to distinguish it from other substances. It provides also a connection from *AFP* to other important concepts in the domain (i.e. *testicular cancer*). Consider another example:

**Xenylamine.** Chemical which has been found to cause bladder cancer. (CANCER)

In this example, the most important fact about *Xenylamine* is that it is found to cause bladder cancer: its connection with the domain. Following examples reinforce the view that the gloss of an entry justifies its inclusion, often by establishing the entry’s connection with the domain.

**Case.** A Java keyword that defines a group of statements to begin executing if a value specified matches the value defined by a preceding switch keyword. (JAVA)

Here, the gloss is also a true definition stating the differences between *case* and other Java keywords.

**Aqueduct.** a pipe, conduit, or channel designed to transport water from a remote source, usually by gravity. (WATER)

The connection between *aqueduct* and the WATER domain is that an aqueduct transports water.

**Certificate of Citizenship.** Identity document proving U.S. citizenship. Certificates of citizenship are issued to derivative citizens and to persons who acquired U.S. citizenship (see

definitions for Acquired and Derivative Citizenship). (CITIZENSHIP)

The connection between *Certificate of Citizenship* and the US Citizenship domain is that Certificate of Citizenship is an identity document, Certificate of Citizenship proves U.S citizenship, and Certificate of Citizenship is issued to derivative and acquired citizens.

In the majority of cases, relations between the entry and the domain are explicitly stated using verbs such as “contain” (CHEMISTRY), “cause” (CANCER), and “define” (JAVA). Empirical observation suggests that such verbs, whilst varying across different domains, are used repeatedly within a glossary, and thus retrievable [6].

It can be argued that these significant verbs are, in fact, the central parts of the knowledge patterns which signal the important knowledge in a field, such as “A CONTAIN B” in the domain of Chemistry:

**acid:** a compound containing detachable hydrogen ions;

**alloy:** a mixture containing mostly metals;

or “A STOP B” in the domain of Cancer:

**Anaesthetic:** Drug which stops feeling, especially pain;

**Aminoglutethamide:** Drug used to treat breast cancer which stops the Adrenal Gland from making sex hormones....

The notion of *knowledge patterns* in ATP has already been discussed in various studies ([1], [5], [11], [15]), although different terms may be used instead of *knowledge patterns*. In this study, the term *knowledge pattern* assumes a general meaning: a linguistic pattern which expresses important knowledge in the domain. We shall focus on patterns whose anchors are verbs, for example “X IS\_A compound”, “X CONTAIN ring”, “X CONTAIN Y”, and “X TREAT Y”.

### 3.4 Extracting and scoring knowledge patterns from glossaries

#### 3.4.1 Pattern extraction in NLP

Pattern extraction is an interesting topic in NLP, as patterns are a means to extract further information. A pattern extraction method often has two components: a pattern heuristic and a pattern scoring method. A pattern heuristic is needed in order to identify pattern candidates. Once identified, pattern candidates are assigned scores so that significant patterns have a greater effect on the intended tasks. Relevant works that propose pattern heuristics and scoring methods include [12], [13], [17], and [18].

#### 3.4.2 The proposed pattern heuristic

Similar to [17] and [18], we concentrate on subject–verb–argument patterns which often express important relations. As subjects (the entries being described) in sentences in glossaries are often omitted, it is safe to concentrate on the verb–argument parts of knowledge patterns. We propose a pattern heuristic that will capture patterns from glossaries at three levels of detail as follows:

VERB + NP (the verb is followed by an NP)

VERB + TERM (the verb is followed by a TERM found in the glossary)

VERB + head (the verb is followed by a specific head word of an NP)

The three levels are intended to capture patterns at different levels of detail, leaving the pattern scoring method to assess their significance. The first two pattern heuristics are similar to those proposed in the literature. The additional pattern heuristic (VERB + NP's head) is intended to capture patterns of general verbs (be, have, etc.) which may otherwise be overlooked by other heuristics. To illustrate this ability, consider the following context: "is a compound", from which other pattern heuristics may suggest only two pattern candidates: "be NP" and "be <TERM>", both considered too general for the domain of Chemistry. In this case, the third level is used to suggest the pattern candidate: "be compound". The output of the parser is used to identify VPs and the verb's arguments. Following is an example of how the pattern heuristic works:

**Burkitt's Lymphoma.** Burkitt's Lymphoma is a rare and special type of lymphoma that is usually treated with combination chemotherapy.

From this sentence, the parser returns:

```
(S1 (S (NP (NP (NNP Burkitt) (POS 's)) (NNP Lymphoma))
      (VP (AUX is)
           (NP (NP (DT a) (ADJP (JJ rare) (CC and) (JJ special)) (NN
type)) (PP (IN of) (NP (NN lymphoma)))
           (SBAR (WHNP (WDT that))
                 (S (VP (AUX is) (ADVP (RB usually))
                       (VP (VBN treated) (PP (IN with)
                                         (NP (NN combination) (NN chemotherapy))))))))) (.))
```

which contains three VPs. For the first VP ("is a rare type of ... treated with combination chemotherapy"), the algorithm suggests two patterns: "BE *lymphoma*" and "BE NP". This is because *lymphoma* is identified as the head of the NP "a rare and special type of lymphoma that is usually ...". The second VP ("is usually treated with combination chemotherapy") does not produce any pattern candidate. For the third VP ("treated with combination chemotherapy"), the algorithm discovers three more patterns: "TREATED WITH *chemotherapy*", "TREATED WITH <TERM>", and "TREATED WITH NP".

We call patterns which have <TERM> as their arguments (such as CONTAIN <TERM>) "binary patterns". Patterns which have specific words as their arguments (such as "BE *lymphoma*") are called "unary patterns". It can be said that two types of patterns reflect two types of relations: relations between two individual terms and relations between individual terms and the whole terminology (the domain).

### 3.4.3 Assigning scores to pattern candidates

In ATP, it is important to assign scores to pattern candidates so that significant patterns have higher scores and, as a result, stronger influence on ATP than

insignificant patterns. Several scoring methods have been experimented with and among them, the following formula has proved to be the best way to score patterns:

$$SRa(p_i) = \frac{F(p_i)}{Fr(p_i)} \log(F(p_i))$$

in this formula,  $F(p_i)$  denotes the frequency of the pattern  $p_i$  in the glossary and  $Fr(p_i)$ : the frequency of  $p_i$  in the reference corpus. This scoring method rewards both patterns which occur frequently in the glossary, and patterns which occur frequently in the glossary in comparison to in a reference corpus. This scoring method can be considered similar to those of [13] and [18]. Examples of high-scoring patterns include: "FIGHT <TERM>", "INCREASE *risk*" (from the glossary CANCER); "CONTAIN *ring*", "DISSOLVED IN <TERM>" (CHEMISTRY).

## 4. Knowledge patterns and ATP

### 4.1 Incorporating knowledge patterns

The extracted knowledge patterns can be considered as semantic information, which has already been used in ATP. A generalised way to incorporate semantic information into the termhood<sup>3</sup> function is to add semantic information scores to it:  $FK(t) = F(t) + \alpha_1 K_1(t) + \alpha_2 K_2(t)$

In this formula,  $F(t)$  is the original termhood function (such as frequency and *C-value*),  $K_1(t)$ : the score of the semantic information contexts of the term candidate  $t$  independent of other term candidates,  $K_2(t)$ : the score of the semantic information contexts which also involve other term candidates, and  $\alpha_1, \alpha_2$ : the weights of these scores. In our case,  $K_1(t)$  is calculated as:

$$K_1(t) = \sum_{p_i \in C_u(t)} S(p_i)$$

Here,  $C_u(t)$ : the set of all instances where a unary knowledge pattern  $p_i$  suggests a relation between the term candidate  $t$  and the pattern's right argument;  $S(p_i)$ : the score of the pattern  $p_i$  of the instance (the calculation of this score has been discussed in the previous section); and  $K_2(t)$ :

$$K_2(t) = \sum_{p_i \in C_b(t)} S(p_i) F(t_i)$$

Here,  $C_b(t)$  is the set of all instances where a relation between the term candidate  $t$  and another term candidate  $t_i$  is suggested by a binary pattern  $p_i$ , and  $F(t_i)$  is the termhood score of the term  $t_i$ , which is the right argument of the pattern  $p_i$  in the instance.  $K_2(t)$  can also be calculated recursively.  $\alpha_1$  and  $\alpha_2$  values are assigned by experiments.

<sup>3</sup> A termhood score is a score which indicates how likely a term candidate is a term. A termhood function assigns termhood score to a term candidate.



## 4.2 Evaluation

We choose to evaluate our proposed methodology over two domains: Cancer and Chemistry, whose knowledge patterns were extracted from their glossaries (see Sections 2 and 3). Texts for the two chosen domains were collected from the Internet. The Cancer domain corpus (CanCor) contains 1248 documents (750,000 words); the Chemistry corpus (ChemCor) contains 300 documents (380,000 words).

### 4.2.1 Gold standard and evaluation metrics

To evaluate the quality of the extracted terms, we compare the outputs (the term candidate lists) provided by different termhood functions. For each domain, we combine three different glossaries to form a final list of terms. The number of glossaries in which a term appears is used as its weight. String matching is used to estimate the total number of terms, their total weights, and the average weight of a term which can be extracted from these two corpora.

For a list of top N term candidates proposed by a termhood function, precision is calculated as the total weight of correct terms (weighted hits) divided by the average weight of N terms in the corpus, recall is the weighted hits divided by the total weight of terms identified using string matching. F-measure is calculated as usual.

### 4.2.2 Results

The results show that the average improvements over the baseline termhood function (frequency) of f-measures over six values of N from Cancor and ChemCor are 5% and 16% (statistically significant at  $p=0.05$ ) respectively. It is observed that knowledge patterns have a greater effect on shorter lists of term candidates than longer ones.

## 5. Conclusions

This paper discusses a forgotten resource in ATP: glossaries. We have argued that glossaries are increasingly available and used; and that they contain valuable terminological knowledge. A method to extract one type of terminological knowledge presented in a glossary - knowledge patterns - is discussed. The extracted knowledge patterns are then shown to be useful in ATP: it helps increase the performance of automatic term extraction (in term of weighted f-measures) by 5% and 16% respectively over two corpora. These extracted knowledge patterns can also be used in other ATP tasks ([8], [11]).

## References

- [1] Ahmad, K., and H. Fulford. 1992. Knowledge Processing: 4. Semantic Relations and their Use in Elaborating Terminology. CS-92-07. University of Surrey.
- [2] Amsler, R. A. 1980. The Structure of the Merriam-Webster Pocker Dictionary. PhD Thesis. Austin: University of Texas.
- [3] Charniak, E. 2000. A Maximum-Entropy-Inspired Parser. In Proceedings of *NAACL-2000*, pp. 132-139. Seattle, WA.
- [4] Collins, M. 1999. Head-Driven Statistical Models for Natural Language Parsing. PhD Thesis. University of Pennsylvania.
- [5] Davidson, L., J. Kavanagh, K. Mackintosh, I. Meyer, and D. Skuce. 1998. Semi-automatic Extraction of Knowledge-Rich Contexts from Corpora. In Proceedings of *Computerm'98*. pp. 50-56. Montreal, Canada.
- [6] Fellbaum, C. (ed.) 1998. *WordNet: An Electronic Lexical Database*. Cambridge, Massachusetts: MIT Press.
- [7] Ha, L. A. 2003. Extracting important domain-specific concepts and relations from a glossary. In Proceedings of the *6th CLUK Colloquium*, pp. 49-56. Edinburgh, UK.
- [8] Ha, L. A., and C. Orasan. 2005. Concept-centred summarisation: producing glossary entries for terms using summarisation methods. In Proceedings of *RANLP 2005*, pp. 219-225. Borovets, Bulgaria.
- [9] ISO. 1990. *ISO 1087 Vocabulary of terminology*. Geneva: ISO.
- [10] Jacquemin, C., and D. Bourigault. 2003. Term Extraction and Automatic Indexing. In R. Mitkov (ed.). *Handbook of Computational Linguistics*, pp. 599-615. Oxford: Oxford University Press.
- [11] Meyer, I. 2001. Extracting knowledge-rich contexts for terminography: A conceptual and methodological framework. In D. Bourigault, C. Jacquemin, and M.-C. L'Homme (ed.). *Recent Advances in Computational Terminology*, pp. 279-302. Amsterdam: John Benjamins.
- [12] Nenadic, G., and Ananiadou, S. 2006. Mining semantically related terms from biomedical Literature. *ACM Transactions on Asian Language Information Processing* 5(1): 22-43.
- [13] Oakes, P. M., and C. D. Paice. 1999. The automatic generation of templates for automatic abstracting. In Proceedings of *British Computer Society: Information Retrieval Specialist Group Colloquium*, pp. 72-79. University of Strathclyde, UK.
- [14] OED. 1989. *The Oxford English Dictionary*. Oxford: Oxford University Press.
- [15] Pearson, J. 1998. *Terms in context*. Amsterdam: John Benjamins.
- [16] Picht, H., and J. Draskau. 1985. *Terminology: an introduction*. Guildford: University of Surrey.
- [17] Riloff, E. 1993. Automatically constructing a dictionary for information extraction tasks. In Proceedings of the *Eleventh National Conference on Artificial Intelligence*, pp. 811-816. Washington DC.
- [18] Riloff, E. 1996. Automatically Generating Extraction Patterns from Untagged Text. In Proceedings *AAAI-96*, pp. 1044-1049. Portland, Oregon.
- [19] Sager, J. C. 1990. *A practical course in terminology processing*. Amsterdam: John Benjamins.
- [20] Swales. 1971. *Writing Scientific English*. Lagos: Nelson Publishing Company.
- [21] W3C. 1999. HTML 4.01 Specification. <http://www.w3.org/TR/html>

# ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge

Catherine Havasi  
Laboratory for Linguistics and Computation  
Brandeis University  
415 South Street  
Waltham, MA 02454  
*havasi@cs.brandeis.edu*

Robert Speer  
Common Sense Computing Group  
MIT Media Lab  
20 Ames Street  
Cambridge, MA 02139  
*rspeer@mit.edu*

Jason B. Alonso  
Tangible Media Group  
MIT Media Lab  
20 Ames Street  
Cambridge, MA 02139  
*jalonso@media.mit.edu*

## Abstract

The Open Mind Common Sense project has been collecting common-sense knowledge from volunteers on the Internet since 2000. This knowledge is represented in a machine-interpretable semantic network called ConceptNet.

We present ConceptNet 3, which improves the acquisition of new knowledge in ConceptNet and facilitates turning edges of the network back into natural language. We show how its modular design helps it adapt to different data sets and languages. Finally, we evaluate the content of ConceptNet 3, showing that the information it contains is comparable with WordNet and the Brandeis Semantic Ontology.

## Keywords

Knowledge representation, common-sense reasoning, natural language processing, information extraction

## 1 Introduction

Understanding language in any form requires understanding connections among words, concepts, phrases and thoughts. Many of the problems we face today in artificial intelligence depend in some way on understanding this network of relationships which represent the facts that each of us knows about the world. Researchers have looked for ways to automatically discover such relationships, but automatic methods can miss many basic relationships that are rarely stated directly in corpora. When people communicate with each other, their conversation relies on many basic, unspoken assumptions, and they often learn the basis behind these assumptions long before they can write at all, much less write the text found in corpora.

Grice's theory of pragmatics [5] states that when communicating, people tend not to provide information which is obvious or extraneous. If someone says

"I bought groceries", he is unlikely to add that he used money to do so, unless the context made this fact surprising or in question. This means that it is difficult to automatically extract common-sense statements from text, and the results tend to be unreliable and need to be checked by a human. In fact, large portions of current lexical resources, such as WordNet, FrameNet, PropBank, Cyc, SIMPLE and the BSO, are not collected automatically, but are created by trained knowledge engineers. This sort of resource creation is labor intensive and time consuming.

In 2000, the Open Mind Common Sense project began to collect statements from untrained volunteers on the Internet. Since then, it has amassed over 700,000 pieces of information both from free and structured text entry. This data has been used to automatically build a semantic network of over 150,000 nodes, called ConceptNet. In this paper we introduce ConceptNet 3, its newest version. We then compare information in ConceptNet to two primarily hand-created lexical resources: the Generative Lexicon-inspired Brandeis Semantic Ontology project [13] and WordNet [4].

## 2 The Open Mind Common Sense Project

The Open Mind Common Sense (OMCS) project serves as a distributed solution to the problem of common sense acquisition, by enabling the general public to enter common sense into the system with no special training or knowledge of computer science. The project currently has 14,000 registered English language contributors.

OMCS collects data by interacting with its contributors in activities which elicit different types of common sense knowledge. Some of the data is entered free-form, and some was collected using semi-structured frames where contributors were given sentences and would fill in a word or phrase that completed the sen-

tence. For example, given the frame “\_\_\_\_\_ can be used to \_\_\_\_\_”, one could fill in “a pen” and “write”, or more complex phrases such as “take the dog for a walk” and “get exercise”.

Open Mind Commons [15] is a new interface for collecting knowledge from volunteers, built on top of ConceptNet 3, which allows its contributors to participate in the process of refining knowledge. Contributors can see the statements that have previously been entered on a given topic, and give them ratings to indicate whether they are helpful, correct knowledge or not. Also, Commons uses the existing knowledge on a topic to ask relevant questions. These questions help the system fill in gaps in its knowledge, and also help to show users what the system is learning from the knowledge they enter.

Each interface to OMCS presents knowledge to its users in natural language, and collects new knowledge in natural language as well. In order to use this knowledge computationally, it has to be transformed into a more structured representation.

## 2.1 The Birth of ConceptNet

ConceptNet is a representation of the Open Mind Common Sense corpus that is easy for a variety of applications to process. From the semi-structured English sentences in OMCS, we are able to extract knowledge into more computable representations. When the OMCS project began using the data set to improve intelligent user interfaces, we began employing extraction rules to mine the knowledge into a semantic network. The evolution of this process has brought us to ConceptNet 3.

In this version of ConceptNet, we focus on the usefulness of the data in the OMCS project to natural language processing and artificial intelligence as a whole. We have aimed to make ConceptNet modular in a way which enables us to quickly and easily make ConceptNets for other data sets such as the Brazilian Open Mind. To support this change of focus, improvements such as higher-order predicates, polarity and improved weighting metrics have been introduced.

ConceptNet and OMCS are useful in a wide variety of applications where undisambiguated text is used. One example of this is improving the accuracy of speech recognition [8]. ConceptNet can also be used to help an intelligent user interface understand the user’s goals and views of the world [9]. For use of ConceptNet 3 as an evaluative tool please see [6]. An extensive summary of applications using the ConceptNet framework can be found in [10].

## 2.2 Multilingual Knowledge Collection

In 2005, a sister project to Open Mind Common Sense was established at the Universidade Federal de São Carlos, in order to collect common sense knowledge in Portuguese [2]. The *Open Mind Commonsense no Brasil* project has now collected over 160,000 statements from its contributors. GlobalMind [3], a project to collect similar knowledge in Korean, Japanese, and Chinese and to encourage users to translate knowledge among these languages and English, was launched in 2006. These projects expand the population that can

contribute to Open Mind, and give us the potential to build connections between the knowledge bases of the different languages and study the cultural differences that emerge.

ConceptNet 3 is flexible enough with its natural language tools that it can build ConceptNets for multiple languages and synthesize them into the same database. We have now done so with the Portuguese corpus, which is the most mature of OMCS’ sister projects.

## 2.3 OMCS and Other Resources

### 2.3.1 Cyc

The Cyc project [7] is another attempt to collect common sense knowledge. Started by Doug Lenat in 1984, this project utilizes knowledge engineers who handcraft assertions and place them in Cyc’s logical frameworks, using a logical representation called CycL. To use Cyc for natural language tasks, one must translate text into CycL through a complex and difficult process, as natural language is ambiguous while CycL is logical and unambiguous.

### 2.3.2 WordNet

Princeton University’s WordNet [4] is one of the most widely used natural language processing resources today. WordNet is a collection of words arranged into a hierarchy, with each word carefully divided into distinct “senses” with pointers to related words, such as antonyms, *is-a* superclasses, and words connected by other relations such as *part-of*. WordNet’s popularity may be explained by the ease a researcher has in interfacing it with a new application or system. We have endeavored to accomplish this flexibility of integration with ConceptNet.

### 2.3.3 BSO

Currently being developed, the Brandeis Semantic Ontology (BSO) [13] is a large lexical resource based in James Pustejovsky’s Generative Lexicon (GL) [12], a theory of semantics that focuses on the distributed nature of compositionality in natural language. Unlike ConceptNet, however, the BSO focuses on the type structure and argument structure as well as on relationships between words.

An important part of GL is its network of qualia relations that characterize the relationships between words in the lexicon, and this structure is significantly similar to the set of ConceptNet relations. There are four types of qualia relations: *formal*, the basic type distinguishing the meaning of a word; *constitutive*, the relation between an object and its parts; *telic*, the purpose or function of the object; and *agentive*, the factors involved in the object’s origins [12].

We’ve noticed that these qualia relations line up well with ConceptNet 3 relations. *IsA* maps well to the formal qualia, *PartOf* to the constitutive, *Used-For* to the telic. The closest relation in ConceptNet 2 to the agentive relation was the *CapableOfReceiving-Action* relation, but this is too general, as it describes many things that can happen to an object besides how

it comes into being. In order to further this GL compatibility, we've added the *CreatedBy* relation and implemented targeted elicitation frames to collect statements that correspond with the agentive qualia.

### 3 The Design of ConceptNet 3

In developing ConceptNet 3, we drew on our experience with working with ConceptNet as users and observed what improvements would make it easier to work with. The new architecture of ConceptNet is more suitable to being incrementally updated, being populated from different data sources, and searching in complex queries such as those that are necessary to discover common-sense analogies. We believe that these improvements make ConceptNet more accessible to a variety of developers of artificial intelligence applications.

#### 3.1 Concepts

The basic nodes of ConceptNet are *concepts*, which are aspects of the world that people would talk about in natural language. Concepts correspond to selected constituents of the common-sense statements that users have entered; they can represent noun phrases, verb phrases, adjective phrases, or prepositional phrases (when describing locations). They tend to represent verbs only in complete verb phrases, so "go to the store" and "go home" are more typical concepts than the bare verb "go".

Although they are derived from constituents, concepts are not literal strings of text; a concept can represent many related phrases, through the normalization process described later.

#### 3.2 Predicates

In a semantic network where concepts are the nodes, the edges are *predicates*, which express relationships between two concepts. Predicates are extracted from the natural language statements that contributors enter, and express types of relationships such as *IsA*, *PartOf*, *LocationOf*, and *UsedFor*. Our 21 basic relation types are not a closed class, and we plan to add more in the future.

In addition to these specific relation types, there are also some underspecified relation types such as *ConceptuallyRelatedTo*, which says that a relationship exists between two concepts, but we can't determine from the sentence what it is. Though they are vague, these connections can help to provide information about the context around a concept, and they provide a fallback for cases where the parser is unable to parse a sentence. They are also used in several current applications [10].

Predicates maintain a connection to natural language by keeping a reference to the original sentence that generated them, as well as the substrings of the sentence that produced each of their concepts. This way, if the computer generates a new predicate without human input, like when it forms a hypothesis based on other knowledge, it can follow the example of other

Relation	Example sentence pattern
IsA	<i>NP</i> is a kind of <i>NP</i> .
MadeOf	<i>NP</i> is made of <i>NP</i> .
UsedFor	<i>NP</i> is used for <i>VP</i> .
CapableOf	<i>NP</i> can <i>VP</i> .
DesireOf	<i>NP</i> wants to <i>VP</i> .
CreatedBy	You make <i>NP</i> by <i>VP</i> .
InstanceOf	An example of <i>NP</i> is <i>NP</i> .
PartOf	<i>NP</i> is part of <i>NP</i> .
PropertyOf	<i>NP</i> is <i>AP</i> .
EffectOf	The effect of <i>VP</i> is <i>NP VP</i> .

**Table 1:** Some of the specific relation types in ConceptNet 3, along with an example of a sentence pattern that produces each type

predicates to express this new predicate in natural language.

#### 3.3 Modular Structure

ConceptNet 3 is built on top of the Common Sense Application Model of Architecture (CSAMOA) [1], a four-layer software design pattern intended to ease the development of common sense applications. By dividing components of common sense reasoning along consistent lines, CSAMOA encourages the development of reusable and interchangeable software components.

The layers of CSAMOA, in order, are the Corpus layer, which preserves original, human representation of common sense knowledge; the Representation layer, which abstracts the knowledge into a machine-interpretable form; the Realm layer, which helps navigate or performs generic computations on the structure of the machine-interpretable representation; and the Application layer, which is devoted to processing all user interactions and performing all other operations pursuant to the particular application. ConceptNet 3 was developed as a Representation layer for use with OMCS as the Corpus layer.

The use of CSAMOA and its emphasis on modularity represents a major change in the design choices underlying ConceptNet. In particular, we want the various components of ConceptNet, such as the parsing or reasoning components, to be customizable for different applications. For instance, the parsing patterns can be changed to handle different forms of natural language input, and the NLP procedures themselves can be replaced in order to generate a ConceptNet in a language besides English.

The most notable improvements CSAMOA has brought to ConceptNet are in its processing-oriented architecture. ConceptNet's data, data models, and processing code are now clearly separated, which permitted many advances in adding multiple language capabilities and improving the extraction of knowledge from unparsed text. ConceptNet's role in larger applications is also more clearly defined, allowing for the simplification of the code base.

## 4 Creating ConceptNet

### 4.1 Pattern Matching

Predicates in ConceptNet are created by a pattern-matching process, as they have been in previous versions [10]. We compare each sentence we have collected with an ordered list of patterns, which are regular expressions that can also include additional constraints on phrase types based on the output of a natural language tagger and chunker. These patterns represent sentence structures that are commonly used to express the various relation types in ConceptNet. Table 1 shows some examples of patterns that express different relations. The phrases that fill the slots in a pattern are the phrases that will be turned into concepts.

Many of these patterns correspond to elicitation frames that were presented on the OMCS website for users to fill in; the fact that so many sentences were elicited with predictable sentence structures means that these sentences can be reliably turned into predicates.

Other patterns, such as “*NP* is a *NP*”, represent sentence structures that contributors commonly used when entering knowledge as free text. For these patterns, the constraints on phrase types (such as *NP*) imposed by the chunker are particularly important to prevent false matches.

Before a sentence goes through the pattern-matching process, common typographical errors and spelling mistakes are corrected using a simple replacement dictionary. If the sentence is a complex sentence with multiple clauses, we use patterns to extract simpler sentences out of it to run through the process.

### 4.2 Normalization

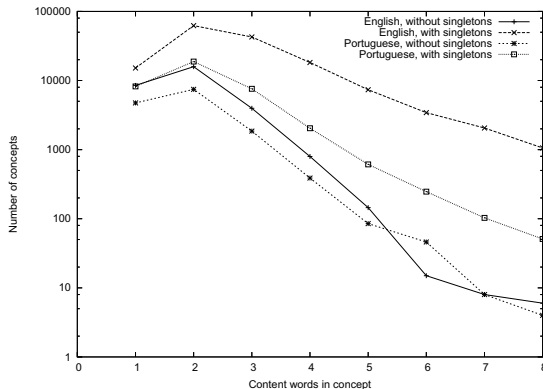
When a sentence is matched against a pattern, the result is a “raw predicate” that relates two strings of text. The *normalization* process determines which two concepts these strings correspond to, turning the raw predicate into a true edge of ConceptNet.

The following steps are used to normalize a string:

1. Remove punctuation.
2. Remove stop words.
3. Run each remaining word through a stemmer. We currently use Porter’s Snowball stemmer, in both its English<sup>1</sup> and Portuguese versions [11].
4. Alphabetize the remaining stems, so that the order of content words in the phrase doesn’t matter.

A concept, then, encompasses all phrases that normalize to the same text. As normalization often results in unreadable phrases such as “endang plant speci” (from “an endangered species of plant”), the normalized text is only used to group phrases into concepts, never as an external representation. This grouping intentionally lumps together many phrases, even ones

<sup>1</sup> For compatibility with previous work, we use the original version of the English Snowball stemmer (the one commonly called “the Porter stemmer”), not the revised version.



**Fig. 1:** The number of words in the texts of concepts after normalization. The “without singletons” lines leave out sporadic concepts that only appear in one predicate, discarding many phrases that are too long to be useful concepts

that are only related by accidents of orthography, because we have found this to be an appropriate level of granularity for reasoning about undisambiguated natural language text collected from people.

### 4.3 Open Mind Commons

ConceptNet would be nothing without the ability to collect knowledge from contributors on the Internet. The statements that currently comprise ConceptNet were collected from the Open Mind Common Sense web site, which used prompts such as “What is one reason that you would **ride a bicycle**?” to collect statements of common sense from its users.

Open Mind Commons [15] is an update of the original knowledge-collection website, OMCS 1, built on top of ConceptNet 3. The interface now includes activities that help refine its existing knowledge, by giving feedback to its users about what it already knows and what gaps seem to exist in its knowledge.

This feedback arises from a process that discovers analogies among the existing knowledge in ConceptNet. If concept *X* and concept *Y* appear in corresponding places in many equivalent predicates, they are considered to be similar concepts. Then, if concept *X* appears in a predicate that is not known about concept *Y*, Open Mind Commons can hypothesize that the same predicate is true for *Y*, and it can make this inference stronger by finding other similar concepts that lead to the same hypothesis. By following the links to natural language that are maintained in ConceptNet, it can turn the hypothesized predicate into a natural language question, which it asks to a user of the site.

Another kind of question that Open Mind Commons will ask based on analogy is a “fill in the blank” question: if it determines that it doesn’t know enough predicates of a certain type about a concept, compared to what it knows about similar concepts, it will ask the

## Knowledge about ocean

Similar objects to **ocean**: sea, water, beaches, aquarium, lake

### An inquiring mind wants to know...

Is on the ocean somewhere that people can be? Yes / No / Doesn't make sense / Why do you ask?	You would find _____ near the ocean. <input type="button" value="Teach OpenMind"/>
Is on the ocean somewhere that coral reefs can be? Yes / No / Doesn't make sense / Why do you ask?	ocean is a kind of _____ <input type="button" value="Teach OpenMind"/>
Would you find an ocean in a pool? Yes / No / Doesn't make sense / Why do you ask?	an ocean is used for _____ <input type="button" value="Teach OpenMind"/>
Is on the ocean somewhere that seagulls can be? Yes / No / Doesn't make sense / Why do you ask?	ocean can be _____ <input type="button" value="Teach OpenMind"/>

**Fig. 2:** *Open Mind Commons asks questions to fill gaps in its knowledge*

user to fill in that predicate. Figure 2 shows Commons asking both kinds of questions about the topic *ocean*.

Asking questions based on analogies serves to make the database's knowledge more strongly connected, as it eliminates gaps where simply no one had thought to say a certain fact; it also helps to confirm to contributors that the system is understanding and learning from the data it acquires.

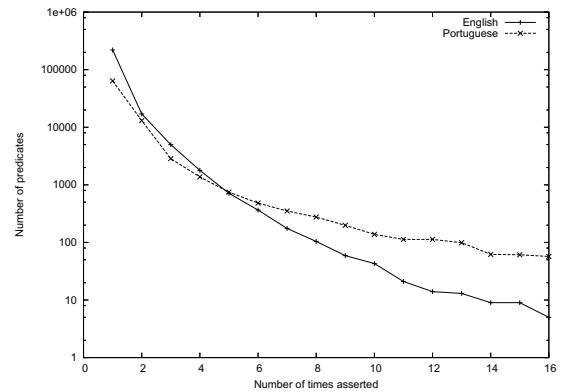
## 4.4 Reliability of Assertions

In ConceptNet 3, each predicate has a score that represents its reliability. This score comes from two sources so far. A user on Open Mind Commons can evaluate an existing statement and increase or decrease its score by one point. The score can also be implicitly increased when multiple users independently enter sentences that map to the same predicate, and this is where the majority of scores come from so far.

The default score for a statement is 1—it is supported by one person: the person who entered it. Statements with zero or negative scores (because a user has decreased their score) are considered unreliable, and are not used for analogies in Open Mind Commons. Statements with positive scores contribute to analogies with a weight that scales logarithmically with their score.

In general, a significant number of predicates were asserted multiple times; Figure 3 shows the distribution of scores among all these predicates. Surprisingly, although the Portuguese corpus has been around for a shorter time and has fewer predicates, its predicates tend to have higher scores. The fact that all Portuguese statements were entered through structured templates, not through free text, may have caused them to coincide more often.

The highest-scored predicate in the English OMCS is “Dogs are a kind of animal”, asserted independently by 101 different users. The highest-scored predicate in Portuguese is “Pessoas dormem quando elas estão com sono” (“People sleep when they are tired”), asserted independently by 318 users.



**Fig. 3:** *The distribution of scores among predicates extracted from OpenMind*

## 4.5 Polarity

In ConceptNet 3, we have introduced the ability to represent negative assertions. This capability allows us to develop interfaces that may ask a question of a user and draw reasonable conclusions when the answer is “no.” The pattern matching process includes additional patterns, which match sentences expressing the negation of one of our relation types.

To this end, we added a *polarity* parameter to our predicate models that can assume the values 1 and  $-1$ , and we introduced a collection of extraction patterns that mirror most of our other extraction pattern but detect negation. About 1.8% of the English predicates and 4.4% of the Portuguese predicates currently in ConceptNet 3 have a negative polarity.

Importantly, score and polarity are independent quantities. A predicate with a negative polarity can have a high, positive score, indicating that multiple users have attested the negative statement (an example is “People don’t want to be hurt”). Predicates with a zero or negative score, meanwhile, are usually unhelpful or nonsensical statements such as “Joe is a cat” or “A garage is for asdfghjkl”, not statements that are “false” in any meaningful way.

## 5 Evaluation

The quality of the data collected by OMCS was measured in a 2002 study [14]. Human judges evaluated a random sample of the corpus and gave positive results, judging three quarters of the assertions to be “largely true”, over four fifths to be “largely objective and sensible”, and 84% “common enough to be known by someone by high school”.

Here, we evaluate ConceptNet 3 in a different way: by testing how often its assertions align with assertions in similar lexical resources. The structure of Cyc is not readily aligned with ConceptNet, but WordNet and the BSO both contain information that is comparable to a subset of ConceptNet. In particular, certain

ConceptNet relations correspond to WordNet’s pointers and the BSO’s qualia, as follows:

ConceptNet	WordNet	BSO
IsA	Hypernym	Formal
PartOf	Meronym	Constitutive
UsedFor	<i>none</i>	Telic

BSO’s fourth qualia type, Agentive, corresponds to the ConceptNet relation CreatedBy, but this relation is new in ConceptNet 3 and we have not yet collected examples of it from the public.

In this evaluation, we examine IsA, PartOf, and UsedFor predicates in ConceptNet, and check whether an equivalent relationship holds between equivalent entries in WordNet and the BSO. The test set consists of all predicates of these types where both concepts normalize to a single word (that is, they each contain one non-stopword), as these are the concepts that are most likely to have counterparts in other resources. Such predicates make up 11.1% of the UsedFor relations, 21.0% of IsA, and 31.2% of the PartOf relations in ConceptNet.

For each predicate, we determine whether there exists a connection between two entries in WordNet or the BSO that have the same normalized form (stem) and the appropriate part of speech (generally nouns, except that the second argument of UsedFor is a verb). This allows us to make comparisons between the different resources despite the different granularities of their entries. If such a connection exists, we classify the predicate as a “hit”; if no such connection exists between the corresponding entries, we classify it as a “miss”; and if no match is possible because a resource has no entries with one of the given stems, we classify it as “no comparison”.

The criterion for determining whether “a connection exists” does not require the connection to be expressed by a single pointer or qualia. For example, the only direct hypernym of the first sense of “dog” in WordNet is “canine”, but we want to be able to match more general statements such as “a dog is an animal”. So instead, we check whether the target database contains the appropriate relation from the first concept to the second concept *or* to any ancestor of the second concept under the IsA relation (that is, the hypernym relation or the formal qualia). Under this criterion, ConceptNet’s (IsA “dog” “anim”) matches against WordNet, as “anim” is the Porter stem of “animal”, WordNet contains a noun sense of “dog” that has a hypernym pointer to “canine”, and a series of hypernym pointers can be followed from “canine” to reach a sense of “animal”.

There are two major classes of “misses”. Sometimes, a ConceptNet predicate does not hold in another resource because the ConceptNet predicate is unreliable, vague, or misparsed; on the other hand, sometimes the ConceptNet predicate is correct, and the difference is simply a difference in coverage. We have assessed a sample of 10 misses between ConceptNet and WordNet in Table 3, and between ConceptNet and the BSO in Table 4.

We ran this evaluation independently for IsA, UsedFor, and PartOf predicates, against each of WordNet and the BSO (except that it is not possible to evaluate

Resource	Type	Hit	Miss	No comparison
WordNet	IsA	2530	3065	1267
WordNet	PartOf	653	1344	319
WordNet	Random	245	5272	1268
BSO	IsA	1813	2545	2044
BSO	PartOf	26	49	2241
BSO	UsedFor	382	1584	3177
BSO	Random	188	4456	2142

**Table 2:** The results of the evaluation. A “hit” is when the appropriate concepts exist in the target database and the correct relationship holds between them, a “miss” is when the concepts exist but the relationship does not hold, and “no comparison” is when one or both concepts do not exist in the target database

Missed predicate	Reason for difference
Swordfish is a novel.	unreliable
Bill is a name.	WordNet coverage
Sam is a guy.	vague
(offensive statement)	unreliable
A gymnasium is a hall.	vague
Babies are fun.	misparsed
Newsprint is a commodity.	WordNet coverage
Biking is a sport.	WordNet coverage
Cats are predators.	WordNet coverage
Seeds are food.	WordNet coverage

**Table 3:** A sample of ConceptNet predicates that do not hold in WordNet, with an assessment of whether the difference comes from unreliable/vague information in ConceptNet or a difference in coverage

UsedFor against WordNet). As a control to show that not too many hits arose from random noise, we also tested “randomized IsA predicates”. These predicates were created by making random IsA predicates out of the shuffled arguments of the IsA predicates we tested, so that these predicates would express nonsense statements such as “soy is a kind of peninsula”. Indeed, few of these predicates were hits compared to real ConceptNet predicates, even though IsA predicates are the most likely to match by chance. Table 2 presents the results, and Figure 4 charts the success rates for each trial (the ratios of hits to hits plus misses).

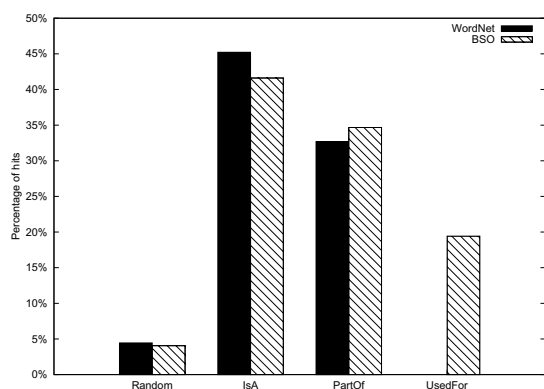
A Pearson’s chi-square test of independence showed that the difference in the hit vs. miss distribution between the real predicates and the randomly-generated ones is statistically very significant, with  $p < 0.001$  ( $df = 1$ ) for each relation type. WordNet has  $\chi^2 = 2465.3$  for IsA predicates and  $\chi^2 = 1112.7$  for PartOf predicates compared to random predicates; the BSO has  $\chi^2 = 1834.0$  for IsA,  $\chi^2 = 159.8$  for PartOf, and  $\chi^2 = 414.7$  for UsedFor compared to random predicates.

## 6 Discussion

As a resource, ConceptNet differs from most available corpora in the nature and structure of its content. Unlike free text corpora, each sentence of OMCS was entered by a goal-directed user hoping to contribute common sense, resulting in a wealth of statements that focus on simple, real-world concepts that often go unstated.

ConceptNet predicate	Reason for difference
A contest is a game.	BSO coverage
A spiral is a curve.	BSO coverage
A robot is a worker.	vague
A cookie is a biscuit.	BSO coverage; regional
An umbrella is waterproof.	misparsed
A peanut is a legume.	BSO coverage
A hunter is a camper.	BSO coverage
A clone is a copy.	BSO coverage
The president is a liar.	unreliable
People are hairdressers.	unreliable

**Table 4:** A sample of ConceptNet predicates that do not hold in the BSO, with an assessment of whether the difference is due to unreliable information in ConceptNet or a difference in coverage



**Fig. 4:** When ConceptNet predicates can be mapped onto relations between WordNet and BSO entries, they match a significant percentage of the time

Our evaluation has shown that our information frequently overlaps with two expert-created resources, WordNet and the Brandeis Semantic Ontology, on the types of predicates where they are comparable. The goal of ConceptNet is not just to emulate these other resources, though; it also contains useful information beyond what is found in WordNet or the BSO. For example, many “misses” in our evaluation are useful statements in ConceptNet that simply do not appear in the other resources we evaluated it against, such as “sauce is a part of pizza”, “a son is part of a family”, and “weekends are used for recovery”.

In addition, ConceptNet expresses many important types of relations that we did not evaluate here, such as CapableOf (“fire can burn you”, “birds can fly”), LocationOf (“you would find a stapler on a desk”, “you would find books at a library”), and EffectOf (“an effect of opening a gift is surprise”, “an effect of exercise is sweating”). All of these kinds of information are important in giving AI applications the ability to reason about the real world.

## References

- [1] J. Alonso. CSAMOA: A common sense application model of architecture. *Proceedings of the Workshop on Common Sense and Intelligent User Interfaces*, 2007.
- [2] J. Anacleto, H. Lieberman, M. Tsutsumi, V. Neris, A. Carvalho, J. Espinosa, and S. Zem-Mascarenhas. Can common sense uncover cultural differences in computer applications? In *Proceedings of IFIP World Computer Conference*, Santiago, Chile, 2006.
- [3] H. Chung. *GlobalMind — bridging the gap between different cultures and languages with common-sense computing*. PhD thesis, MIT Media Lab, 2006.
- [4] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.
- [5] P. Grice. Logic and conversation. In *Speech Acts*. Academic Press, 1975.
- [6] C. Havasi. An evaluation of the brandeis semantic ontology. *To Appear In Proceedings of the Fourth International Workshop on Generative Approaches to the Lexicon*, 2007.
- [7] D. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 11:33–38, 1995.
- [8] H. Lieberman, A. Faaborg, W. Daher, and J. Espinosa. How to wreck a nice beach you sing calm incense. *Proceedings of the 10th international conference on Intelligent user interfaces*, 2005.
- [9] H. Lieberman, H. Liu, P. Singh, and B. Barry. Beating some common sense into interactive applications. *AI Magazine*, 25(4):63–76, 2004.
- [10] H. Liu and P. Singh. ConceptNet: a practical commonsense reasoning toolkit. *BT Technology Journal*, 2004.
- [11] M. F. Porter. Snowball: a language for stemming algorithms. Snowball web site, 2001. <http://snowball.tartarus.org/texts/introduction.html> – accessed Jan. 31, 2007.
- [12] J. Pustejovsky. *The Generative Lexicon*. MIT Press, Cambridge, MA, 1998.
- [13] J. Pustejovsky, C. Havasi, R. Sauri, P. Hanks, and A. Rumshisky. Towards a generative lexical resource: The Brandeis Semantic Ontology. *Proceedings of the Fifth Language Resource and Evaluation Conference*, 2006.
- [14] P. Singh, T. Lin, E. T. Mueller, G. Lim, T. Perkins, and W. L. Zhu. Open Mind Common Sense: Knowledge acquisition from the general public. In *Proceedings of First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems*, Irvine, California, 2002.
- [15] R. Speer. Open Mind Commons: An inquisitive approach to learning common sense. *Proceedings of the Workshop on Common Sense and Interactive Applications*, 2007.



# Improving Word Alignment Based on Extended Inversion Transduction Grammar

Chung-Chi Huang†  
CLCLP, TIGP, Academia Sinica, Taiwan  
u901571@gmail.com

Wei-Teh Chen  
ISA, NTHU, Taiwan  
weitehchen@gmail.com

Jason S. Chang  
CS, NTHU, Taiwan  
jason.jschang@gmail.com

## Abstract

We propose a fusion of Inversion Transduction Grammar model with IBM-style notation of fertility to improve word-aligning performance. In our approach, binary context-free grammar rules on the source language, accompanied with orientation preferences on the target, and fertilities of words are leveraged to construct a syntax-based statistical translation model. Our model, inherently possessing the characteristic of ITG restrictions and allowing for many consecutive words aligned to one and vice versa, outperforms original ITG model and GIZA++ not only in alignment error rate (23% and 14% error reduction) but in consistent phrase error rate (13% and 9% error reduction) as well. Better performance in these two evaluation metrics will lead to better phrase-based machine translation with higher possibility.

## Keywords

Word alignment, inversion transduction grammar, IBM models, alignment error rate, parsing, and GIZA++.

## 1. Introduction

Statistical translation model is a model which, given a string pair, estimates the likelihood of being translations of each other whether relied on lexical information or syntactic aspects of languages involved. In spite of the fact that the methodologies varies, the intention is clear—trying to obtain better word alignment results since a better translation model implies better performance in various linguistic applications. Among them are phrase-based machine translation (Och and Ney, 2004; David Chiang, 2005; Liu et al., 2006) and inference of syntactic translation rules (Galley et al., 2004; Galley et al., 2006).

Since the pioneering work of (Brown et al., 1988), there have been a myriad of subsequent researches related to statistical translation model. They could mainly be classified into two categories: one paying little attention to the grammars of the languages (Vogel et al., 1996; Och and Ney, 2000; Toutanova et al., 2002) and the other explicitly utilizing languages' structural or syntactic information (Wu, 1997; Yamada and Knight, 2001; Cherry and Lin, 2003; Gildea, 2004; Zhang and Gildea, 2005). With more and more accurate syntactic analyzers (such as part-of-speech tagger and Stanford parser) being developed and in view of the deficiency in modeling grammatical facets of languages IBM-like models experience, latter researches have received increasing attention.

To incorporate syntax of involved languages, Yamada and Knight (2001) accepted source-language (SL, such as English) parse trees as input and made use of reordering, inserting and translating operations to transform the input parse trees into counterpart target-language (TL, such as French) strings. In contrast to flattening the input parse trees to do the transformation (reordering, inserting and translating) for every node, Wu's ITG (1997) attempted to associate each production rule commonly shared by two languages with word orientation. Besides, instead of accepting parse trees produced by a monolingual parser, Wu's approach makes possible constructing bilingual parse trees synchronously.

The strengths of two models are discussed in (Zhang and Gildea, 2004), which also found data-oriented bilingual parsing turned out to outperform tree-to-string model for word-level alignment. Nonetheless, in (Wu, 1997), constituent categories are not differentiated and the probabilities of the *straight* or *inverted* orientation of binary production rules, rather than trained on real-life cases, are all assigned constant.

Inspired by (Zhang et al. 2006), which suggests binarization of synchronous rules improves both speed and accuracy of a syntax-based machine translation system, in this paper, to capture the systematic differences in languages' grammars, such as SVO (English or Chinese), SOV (Japanese) and VSO (Arabic) word orders, we attach the information of identical or dissimilar orientation of languages' counterparts onto binary SL CFG rules, resulting in grammatical rewrite rules biased on SL side, or more specifically, biased ITG rules, *bITG* for short. For instance, the similar VO construct in both English and Chinese can be observed from the high probability of the *bITG* rule  $VP \rightarrow [VP NP]$  where square bracket indicates the same ordering (*straight*) of the two right-hand-side constituents in both languages when expanding the left-hand-side symbol. On the contrary, the different VO construct in English and Japanese can be modeled using high *inverted* probability of *bITG* rule  $VP \rightarrow \langle VP NP \rangle$  where pointed bracket denotes we expand the left-hand-side label into two right-hand-side symbols in reverse orientation in two languages. However, both *bITG* rules are inferred from the same binary CFG rule ( $VP \rightarrow VP NP$ ) of the source language, English, only with different order preferences on the target end.

Furthermore, in our model fusing bITG model with IBM-style fertilities, many contiguous words on the source can be aligned to one word on the target and vice versa based on fertility probabilities of words. Originally, Wu's ITG (1997) only allowed for a word in one language to be aligned to, at most, one word in another, which may decrease the accuracy of the bilingual parse trees and, in turn, the performance on word alignments. This one-to-one restriction on word-aligning is especially not suitable for language pair like English and Chinese since the tokenization work of Chinese sentences prior to word alignment would introduce many many-to-one or one-to-many links in that the resulting segments in Chinese sentences are independent of words on English side. That is, the segmentations in Chinese can be under- or over-segmented for the corresponding words in English. As a result, the translation model accommodating more than one-to-one correspondences is of great importance, especially for such language pair.

Section 2 and 3 describe our model in detail. Section 4 shows experimental results and section 5 concludes this paper.

## 2. The Model

### 2.1 An Example

First, an example of how bITG rules are exploited to assist in word-aligning sentence pairs is introduced. A more formal description of our model will be discussed in sequent sections.

We assume a parallel sentence pair and POS information of the SL sentence are fed into our model and it, using not only lexical translation rules but the binary SL CFG rules accompanied with orientation preferences of counterparts on the TL, synchronously parses the bilingual sentence pair and yields the word alignments at the leaf level of the bilingual parse tree.

The model assigns probabilities to substring pairs of the bilingual sentences after each of them is associated with possible syntactic labels on the source side. Take the sentence pair and its parse in Figure 1, where spaces in the Chinese sentence are used to distinguish the boundaries of segments and \* denotes the *inverted* orientation of the node's children on the target, for example. The substring pair (positive role, 積極作用) associated with constituent category *NP* will be assigned a probability. In this particular parse, the best probability of parsing (positive role, 積極作用) is the product of probabilities of *straight* bITG rule,  $NP \rightarrow [JJ NN]$ , and lexical translation rules,  $JJ \rightarrow \text{positive/積極}$  and  $NN \rightarrow \text{role/作用}$  where / denotes word correspondence in both languages. The higher probability of the rule  $NP \rightarrow [JJ NN]$  than that of the *inverted* rule  $NP \rightarrow \langle JJ NN \rangle$  not just instructs the model

to align the right-hand-side counterparts of two languages in a *straight* fashion more, but implies the similar word orientation for the syntactic structure in English and Chinese.

On the other hand, we would notice that the beginning half "*These factors will continue to play a positive role*" is translated into the back of the Chinese sentence whereas the ending half "*after its return*" is translated into the beginning. This phenomenon is very common while translating one language into another. The *inverted* word order rules trained on parallel corpus, like  $S \rightarrow \langle S PP \rangle$ , are devised to capture the systematic differences of the languages' grammars.

In the end, taking into account both the probabilities of lexical and grammatical rewrite rules and fertilities of words in languages, the model endeavors to find the best parse that applies more appropriate production rules to match the similarities and dissimilarities of two languages, which, in turn, yields better word alignment results. As for this example parse, the sentence pair associated with the syntactic label *S* results in best bilingual parse tree whose probability is estimated by the product of probabilities of the bITG rules,  $S \rightarrow \langle S PP \rangle$ , and root's two children, (*These factors will continue to play a positive role*, 這些條件將會繼續發揮積極作用)<sub>S</sub> and (*after its return*, 香港回歸後)<sub>PP</sub>.

We actually obtain probabilities of bITG rules, consisting of lexical rules and binary SL CFG rules with word orientation preferences on the target, and fertilities of words from a parallel corpus and SL CFG. Section 3 describes the training algorithm.

### 2.2 Runtime Parsing

In this section, we extend Wu's ITG (1997) such that our model incorporates the grammatical constituents on the source language and accommodates the cases of many contiguous words on the source aligned to one on the target and vice versa.

The English-French notation is used throughout this paper. *E* and *F* denote the source and target language respectively and  $e_i$  stands for the *i*-th word in sentence *e* in language *E* and  $f_j$  for the *j*-th word in sentence *f* in *F*.

As mentioned in (Wu, 1997; Zens and Ney, 2003), the ITG constraint allows for a polynomial-time parsing algorithm, based on a recursion equation that can be resolved by a CYK-style parser. During a parse of a sentence pair in our model, a table of  $\delta_{p,s,t,u,v}$ , which represents the *best* probability of parsing substring pair  $(e_{s+1} \cdots e_t, f_{u+1} \cdots f_v)$

English sentence: These factors will continue to play a positive role after its return

Chinese sentence: 香港回歸後這些條件將會繼續發揮積極作用

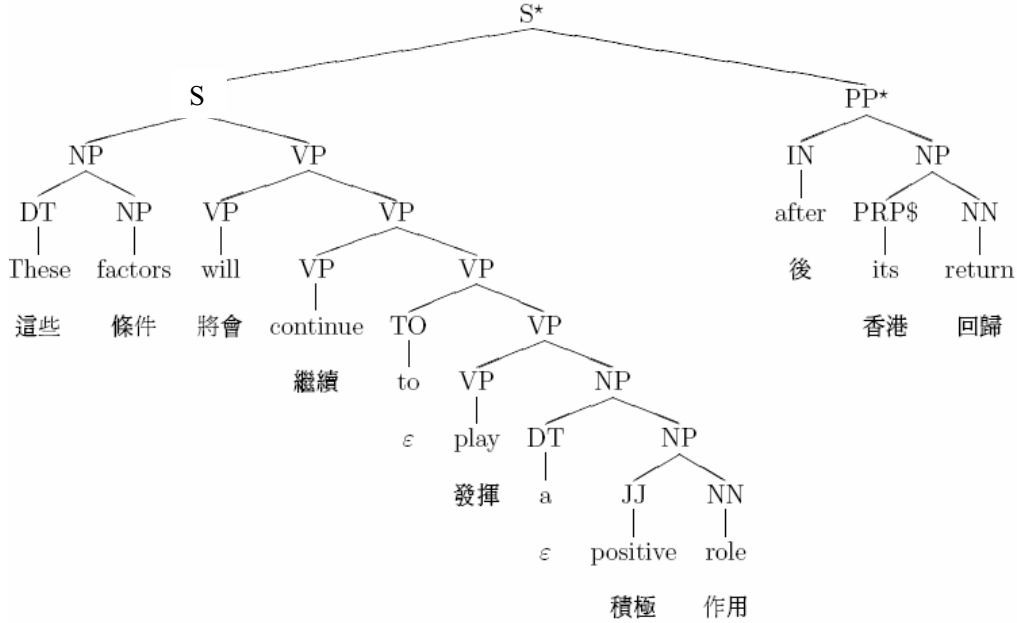


Figure 1. An example sentence pair and its bilingual parse tree

related to a syntactic label  $p$  on the  $E$  side, is constructed. We initialize this table with probabilities of one-to-one, one-to-zero and zero-to-one word correspondences limited on the scope of the sentence pair. Afterwards, relied on the work done previously, many-to-many word correspondences and parsing results of longer substring pairs would unveil themselves in a bottom-top manner. Meanwhile, integration of fertilities of words into the model further boosts the word-aligning performance.

Following is the CYK parsing algorithm in our model, where we parse a sentence pair  $(e, f)$ ,  $(e_1 \cdots e_m, f_1 \cdots f_n)$ , and the POS tag sequence of  $e$  is  $(t_1, \cdots, t_m)$ . In the algorithm,  $P(L \rightarrow t)$  denotes probability of a lexical rule and  $t$  could be  $e_i / f_j$ ,  $e_i / \varepsilon$  and  $\varepsilon / f_j$  where  $\varepsilon$  stands for NULL, while  $P(L \rightarrow [R_1 R_2])$  and  $P(L \rightarrow \langle R_1 R_2 \rangle)$  denote probabilities of binary bITG rules where  $R_1$  and  $R_2$  indicate the right-hand-side syntactic constituents of the CFG rules in  $E$ . Furthermore,  $\Pr(\Phi_{e_i} = x)$  and  $\Pr(\Phi_{f_j} = x)$  represent the probabilities of fertilities of  $e_i$  and  $f_j$  being associated with  $x$ , respectively.

### Parsing Algorithm

#### 1. Initial Step

For  $1 \leq i \leq m, 1 \leq j \leq n$

$$\delta_{t_i, i-1, i, j-1, j} = P(t_i \rightarrow e_i / f_j) \times \Pr(\Phi_{e_i} = 1) \times \Pr(\Phi_{f_j} = 1)$$

For every  $L \rightarrow t_i \in \text{grammar rules in } E$

$$\delta_{L, i-1, i, j-1, j} = P(L \rightarrow e_i / f_j) \times \Pr(\Phi_{e_i} = 1) \times \Pr(\Phi_{f_j} = 1)$$

For  $1 \leq i \leq m, 0 \leq j \leq n$

$$\delta_{t_i, i-1, i, j, j} = P(t_i \rightarrow e_i / \varepsilon) \times \Pr(\Phi_{e_i} = 0)$$

For every  $L \rightarrow t_i \in \text{grammar rules in } E$

$$\delta_{L, i-1, i, j, j} = P(L \rightarrow e_i / \varepsilon) \times \Pr(\Phi_{e_i} = 0)$$

For  $0 \leq i \leq m, 1 \leq j \leq n, L \in \text{syntactic labels in } E$

$$\delta_{L, i, i, j-1, j} = P(L \rightarrow \varepsilon / f_j) \times \Pr(\Phi_{f_j} = 0)$$

#### 2. Recurrent Step

$$\delta_{p, s, t, u, v} = \max_{\substack{q, r \in \text{syntax labels on } E \\ s \leq s' \leq t \\ u \leq u' \leq v}} \left\{ \begin{array}{l} P(p \rightarrow [q r]) \times \delta_{q, s, s', u, u'} \times \delta_{r, s', t, u', v} \\ P(p \rightarrow \langle q r \rangle) \times \delta_{q, s, s', u', v} \times \delta_{r, s', t, u, u'} \end{array} \right\}$$

However, for  $\delta_{p, s, t, u-1, u}$ , the possible choice to parsing the substring pair also includes  $\Pr(\Phi_{f_u} = (t-s)) \times$

$$\max_{\substack{q, r \in \text{syntax labels on } E}} \left\{ P(p \rightarrow [q r]) \times \frac{\delta_{q, s, s'+1, u-1, u}}{\Pr(\Phi_{f_u} = 1)} \times \frac{\delta_{r, s'+1, t, u-1, u}}{\Pr(\Phi_{f_u} = (t-s-1))} \right\}.$$

Similar principle applies for  $\delta_{p,s-1,s,u,v}$ .

### 2.3 Pruning

Although the complexity of described algorithm is polynomial-time, the execution time grows rapidly with the increase in the variety of syntactic labels, from three structural labels in (Wu, 1997) to the syntactic categories of the source language's grammar. As a result, pruning techniques are essential to reduce the time spent on parsing.

We adopt pruning in following two manners. The idea of the first pruning technique is to only keep parse trees whose probabilities fall within the best  $N \times \alpha$ , where  $N$  is the number of possible parses for SL substring  $e_{s+1} \cdots e_t$  and a constant length of the TL substring, and  $\alpha$  is a real number between 0 and 1. In other words, we remove less probable parse trees that are not in the best  $N \times \alpha$  ones.

The second pruning technique is related to the ratio of the length of SL and TL substring.  $\delta_{p,s,t,u,v}$  will be removed, or not calculated, if  $(t-s)/(v-u)$  is smaller than  $\theta_{ratio}$  or larger than  $1/\theta_{ratio}$  where  $0 \leq \theta_{ratio} \leq 1$ , since few words will be aligned to more than  $1/\theta_{ratio}$  words in another language. Applying these pruning techniques affects little in the word alignment quality with computational overhead reduced significantly.

## 3. Probability Estimation

In the first stage of our probabilistic estimation process, a word-aligning strategy is applied to acquire the initial word alignments from a sentence-aligned corpus. Thereafter, for every substring pair of each bilingual sentence pair, the SL substring will be related to some possible binary SL CFG rules and, based on initial word alignments, right-hand-side constituents of these rules will be associated with an orientation on the target end. Ultimately, we exploit occurrence of detected BITG rules to estimate probabilities.

### 3.1 Representation

By applying any existing word-level alignment method, the initial word alignment set  $\mathbf{A}$  for parallel corpus  $\mathbf{C}$  is obtained.  $\mathbf{A}$  is comprised of elements of the form  $(r, e_{i_1}^{i_2}, f_{j_1}^{j_2}, L, rhs, rel)$ , which represents substring pair  $(e_{i_1} \cdots e_{i_2}, f_{j_1} \cdots f_{j_2})$  in sentence pair  $r$  has  $L \rightarrow rhs$  as the derivation leading to the bilingual structure in the parse tree and  $rel$ , either *straight* or *inverted*, as the cross-language word order relations of constituents of  $rhs$ , denoting either a sequence of syntactic labels or a single terminating bilingual word pair.

Take the parse in Figure 1 for example, (after its return, 香港 回歸 後)<sub>pp</sub> would be represented by the 6-tuple

$(193, e_{10}^{12}, f_1^3, PP, IN NP, Inverted)$  where 193 is the sentence number of this pair, in the word alignment set  $\mathbf{A}$ .

### 3.2 Training Algorithm

The algorithm starts with a set  $\mathbf{H}$  initialized with the initial word alignment set  $\mathbf{A}$ . Then recursively select two elements, which have not yet been paired up, from  $\mathbf{H}$ . If these two elements have contiguous word sequence on  $e$  side and exhibit *straight* or *inverted* relation between  $e$  and  $f$  based on word alignments, a new tuple representing these two will be added into  $\mathbf{H}$ . At last, we utilize the occurrence in  $\mathbf{H}$  to infer probabilities of BITG rules,  $P(L \rightarrow [R_1 R_2])$ ,  $P(L \rightarrow \langle R_1 R_2 \rangle)$  and  $P(L \rightarrow t)$ . Besides, fertility probabilities related to words in both languages are calculated in this algorithm as well.

In the following algorithm,  $\mathbf{G}$  stands for the set of the binary SL CFG rules,  $|\mathbf{W}|$  for the number of entries in set  $\mathbf{W}$ ,  $\text{count}(p; \mathbf{Q})$  for the occurrence of  $p$  in set  $\mathbf{Q}$ ,  $\delta$ , a positive integer, for the tolerance of cross-language *straight/inverted* word order phenomenon, and  $\Phi_{e_i}$  and  $\Phi_{f_j}$  for fertility of the word  $e_i$  and  $f_j$ , respectively.

#### Algorithm for Probabilistic Estimation

$\mathbf{H} = \mathbf{A}$

For  $(r, e_{i_1}^{i_2}, f_{j_1}^{j_2}, L, rhs, rel) \in \mathbf{H}$ ,  $(r, \bar{e}_{i_1}^{\bar{i}_2}, \bar{f}_{j_1}^{\bar{j}_2}, \bar{L}, \bar{rhs}, \bar{rel}) \in \mathbf{H}$  have not yet been considered

If  $(\bar{i}_2 = \bar{i}_1 - 1)$

for every  $L' \rightarrow L \bar{L} \in \mathbf{G}$

If  $(j_2 + 1 \leq \bar{j}_1 \leq j_2 + \delta)$

$\mathbf{H} = \mathbf{H} \cup \{(r, e_{i_1}^{i_2}, f_{j_1}^{j_2}, L', L \bar{L}, \text{Straight})\}$

If  $(\bar{j}_2 + 1 \leq j_1 \leq \bar{j}_2 + \delta)$

$\mathbf{H} = \mathbf{H} \cup \{(r, e_{i_1}^{i_2}, f_{j_1}^{j_2}, L', L \bar{L}, \text{Inverted})\}$

Same principle applies when  $\bar{i}_2 = i_1 - 1$

Incorporate words aligned to null, each of which is denoted using 6-tuple representation, in both languages into  $\mathbf{H}$

For  $(r, e_{i_1}^{i_2}, f_{j_1}^{j_2}, L, rhs, rel) \in \mathbf{H}$

If  $(rhs \neq t)$

$$P(L \rightarrow [R_1 R_2]) = \frac{\text{count}((*, *, *, L, R_1 R_2, \text{Straight}); \mathbf{H})}{|\mathbf{H}|}$$

$$P(L \rightarrow \langle R_1 R_2 \rangle) = \frac{\text{count}((*, *, *, L, R_1 R_2, \text{Inverted}); \mathbf{H})}{|\mathbf{H}|}$$

Else

$$P(L \rightarrow t) = \frac{\text{count}((*, *, *, L, t, *); \mathbf{H})}{|\mathbf{H}|}$$

Based on **A** and **C**, Calculate  $\Pr(\Phi_{e_i})$  and  $\Pr(\Phi_{f_j})$   
using relative frequency

## 4. Experiments

To experiment, we trained our model on a large English-Chinese parallel corpus. To evaluate performance, we examined alignments produced by the proposed model using the evaluation metrics proposed by Och and Ney (2000) and by Ayan and Dorr (2006). For comparison, we also trained GIZA++, a state-of-the-art word-aligning system, on the same corpus.

### 4.1 Training

We used the news portion of Hong Kong Parallel Text (Hong Kong news) distributed by Linguistic Data Consortium (LDC) as our sentence-aligned corpus **C**. The corpus consists of 739,919 English-Chinese sentence pairs. English sentences are considered to be the source while Chinese sentences are the target. SL sentences are tagged and TL sentences are segmented before fed into any word alignment strategy or existing system. The average sentence length is 24.4 words for English and 21.5 words for Chinese. On the other hand, PTB section 23<sup>1</sup> production rules distributed by Andrew B. Clegg made up of our binary SL CFG **G**.

### 4.2 Evaluation

To evaluate our statistical translation model, 114 sentence pairs were chosen randomly from Hong Kong news as our testing data set. For the sake of execution time, we only selected sentence pairs whose length of English and Chinese sentences does not exceed 15, which cover approximately 40% of sentence pairs in the whole Hong Kong news corpus and where better word-aligning results can be obtained using GIZA++. We used the metrics of alignment error rate (AER) proposed by Och and Ney (2000), in which the quality of a word alignment result **A** done by an automatic system is evaluated using

$$\text{precision} = \frac{|\mathbf{A} \cap \mathbf{P}|}{|\mathbf{A}|}, \text{ recall} = \frac{|\mathbf{A} \cap \mathbf{S}|}{|\mathbf{S}|} \text{ and}$$

$$\text{AER}(\mathbf{S}, \mathbf{P}; \mathbf{A}) = 1 - \frac{|\mathbf{A} \cap \mathbf{S}| + |\mathbf{A} \cap \mathbf{P}|}{|\mathbf{A}| + |\mathbf{S}|}, \text{ where } \mathbf{S} \text{ (sure) is the set}$$

whose alignments are not ambiguous and **P** (possible) is the set consisting of alignments that might or might not

exist ( $\mathbf{S} \subseteq \mathbf{P}$ ). Thus, the human-annotated alignments may contain many-to-one and one-to-many relations.

In the experiment, we used an existing system, GIZA++, as our word-aligning strategy in training procedure. In other words, the initial word alignment set was produced by GIZA++ with default settings. Following table illustrates the experimental results of GIZA++, original ITG model in (Wu, 1997), and our extended ITG biased on English side.

**Table 1. Results of test data of different systems**

	P	R	AER	F
E to F	<b>.891</b>	.385	.459	.537
F to E	.882	.533	.333	.664
Refined	.879	.635	.261	.737
ITG	.844	.610	.290	.708
Our model w/o fertility	.866	.638	.263	.735
Our model w/ fertility	.878	<b>.692</b>	<b>.224</b>	<b>.774</b>

In this table<sup>2</sup>, P, R and F stand for precision, recall and F-measure<sup>3</sup> respectively. The performance of E to F (E stands for English and F for Chinese), F to E and refinement of both directions, proposed by Och and Ney (2000), of GIZA++, are shown, and so is that of original ITG, which also trained on the lexical output of GIZA++. The results of our translation model without or with the capability of making many-to-one/one-to-many links are listed in the last two rows.

Compared with ITG model that does not distinguish the constituent categories, our model without fertility probability, allowing for at most one-to-one alignment as the original ITG does, achieved 9% reduction in the alignment error rate. It follows the binary SL CFG rules accompanied with ordering preference of the counterparts on the TL trained on parallel corpus do capture the systematic differences of languages' grammars and impose a more realistic and suitable reordering constraints on word aligning for the languages pair.

On the other hand, compared to the refined alignments of both directions GIZA++ produced, our model with fertility, which is quite similar to the refined method that accommodates many-to-many alignment relations, increased the recall by 9% while maintaining high precision and overall achieved 14% alignment error reduction (increased F-measure by 5%).

<sup>2</sup>  $|\mathbf{S}|/|\mathbf{P}|$  is 85.56%.

<sup>3</sup> Calculated using the formula  $2 \times P \times R / (P + R)$ .

<sup>1</sup> <http://textmining.cryst.bbk.ac.uk/acl05/>

### 4.3 Consistent Phrase Error Rate

To evaluate the possibility of leading to better translation performance of a phrase-based MT model if provided the output of our model, on the other hand, we adopted the recently-proposed metric, consistent phrase error rate (CPER) by (Ayan and Dorr, 2006).

According to this research, the intrinsic evaluation metric of AER examines only the quality of word-level alignments but correlates poorly with MT community-standard metric—BLEU score. Consequently, we exploited CPER, correlating better with BLEU, to evaluate alignments in the context of phrase-based MT. Precision, recall and CPER are computed as

$$P = \frac{|P_A \cap P_G|}{|P_A|}, R = \frac{|P_A \cap P_G|}{|P_G|}, \text{ and } CPER = 1 - \frac{2 \times P \times R}{P + R}$$

the sets of phrases,  $P_A$  and  $P_G$ , generated by an alignment  $A$  and manual alignment  $G$  respectively, are known.

From Table 2, we notice proposed bITG model with fertility yielded lowest CPER, with great chance contributing to higher BLEU if a phrase-based MT system accepts the word alignments of our model.

**Table 2. Reports on CPER**

	P	R	CPER
E to F	.479	.383	.574
F to E	.544	.518	.470
Refined	.573	.606	.411
ITG	.569	.569	.431
Our model w/o fertility	.598	.597	.402
Our model w/ fertility	<b>.624</b>	<b>.626</b>	<b>.375</b>

## 5. Conclusion

A thought-provoking fusion of IBM-style fertility notation with syntax-based ITG model is described to capture the strengths of competing models. In our method, *straight/inverted* binary bITG rules, bypassing the problem that commonly-shared grammatical rules of two languages are difficult and time-consuming to design manually, are statistically modeled and devised to boost the word alignment quality. The proposed bITG model with fertilities reduced AER by 14% to 23% and CPER by 9% to 13% in comparison to GIZA++ and Wu’s ITG (1997), and lower CPER suggests better translation performance if a phrase-based MT is chained after our word-level alignment output. In this paper, the performance of ITG models trained on large-scale parallel corpus is shown for the first time and the result is inspiring.

## 6. References

[1] P. Brown, J. Cocke, S. D. Pietra, V. D. Pietra, F. Jelinek, R. Mercer, P. Roossin. 1988. A statistical approach to language

translation. In *Proceedings of the 12<sup>th</sup> conference on Computational Linguistics*, pages 71-76.

[2] C. Cherry and D. Lin. 2003. A probability model to improve word alignment. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 88-95.

[3] D. Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43<sup>rd</sup> Annual Meeting of the Association for Computational Linguistics*, pages 263-270.

[4] M. Galley, M. Hopkins, K. Knight and D. Marcu. 2004. What’s in a translation rule? In *Proceedings of HLT/NAACL-04*.

[5] M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang and I. Thayer. 2006. In *Proceedings of the 44<sup>th</sup> Annual Conference of the Association for Computational Linguistics*, pages 961-968.

[6] Y. Liu, Q. Liu and S. Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 44<sup>th</sup> Annual Conference of the Association for Computational Linguistics*, pages 609-616.

[7] F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38<sup>th</sup> Annual Conference of the Association for Computational Linguistics (ACL-00)*, pages 440-447.

[8] F. J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417-449.

[9] K. Toutanova, H. T. Ilhan and C. D. Manning. 2002. Extensions to HMM-based statistical word alignment models. In *Proceedings of the Conference on Empirical Methods in Natural Processing Language*.

[10] S. Vogel, H. Ney, and C. Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics*, volume 2, pages 836-841.

[11] D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377-403.

[12] K. Yamada and K. Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39<sup>th</sup> Annual Conference of the Association for Computational Linguistics (ACL-01)*.

[13] R. Zens and H. Ney. 2003. A comparative study on reordering constraints in statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 144-151.

[14] H. Zhang and D. Gildea. 2005. Stochastic lexicalized inversion transduction grammar for alignment. In *Proceedings of the 43<sup>rd</sup> Annual Meeting of the ACL*, pages 475-482.

[15] H. Zhang, L. Huang, D. Gildea and K. Knight. 2006. Synchronous binarization for machine translation. In *Proceedings of the NAACL-HLT*.

# Fast Training Methods of Boosting Algorithms for Text Analysis

Tomoya Iwakura and Seishi Okamoto  
Fujitsu Laboratories Ltd.

1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki 211-8588, Japan  
{iwakura.tomoya,seishi}@jp.fujitsu.com

## Abstract

This paper proposes two techniques to improve training speeds of boosting algorithms for learning rules represented by feature conjunctions that contribute to a significant improvement in accuracy on Natural Language Processing. The first one is generating candidate rules suited for pruning. The other is limiting search space by distributing features to buckets and repeatedly select a bucket and find a feature conjunction containing a feature in the selected bucket. The experimental results of English syntactic chunking show that our algorithm reduces training times by 2-3 orders of magnitude while keeping accuracy comparable with results for boosting algorithms without our techniques.

## 1 Introduction

Several boosting based learning algorithms have been successfully applied to Natural Language Processing (NLP) problems. These include text categorization [17], Named Entity Recognition [3], Natural Language Parsing [5], English syntactic chunking [13], sentence classification [12], anaphora resolution [8], and so on. Furthermore, the boosting based classifiers have shown good performance in classification speed in addition to classification accuracy [12, 13]

However, boosting based algorithms require long training times. One of the reasons is that boosting is a method for improving the classification accuracy of a given learning algorithm by combining several hypotheses or rules created with the learning algorithm. Furthermore, training speed of boosting based algorithms becomes more of a problem when considering feature conjunctions that contribute to a significant improvement in accuracy on NLP.

This paper proposes fast training methods of boosting algorithms for learning rules represented by feature conjunctions from large-scale training data for NLP.

In section 2, we present the boosting algorithms for learning rules. In section 3, we present the following methods to improve training speed of the boosting algorithms: 1) Generating candidate rules suited for pruning with a theoretical threshold. 2) Limiting search spaces of rules by distributing features to several buckets and repeatedly select a bucket and find a feature conjunction containing a feature in the selected bucket.

We present a task of English syntactic chunking for evaluating our methods in section 4, and we report experimental results in section 5. We present related works in section 6, and conclude this paper in section 7.

## 2 Boosting Algorithms

We consider the problem of classifying samples represented as a set of features. The problem is defined as fol-

lows: Let  $\mathcal{X}$  be an instance space. The goal is to induce a mapping  $class : \mathcal{X} \rightarrow \{\pm 1\}$  from given training samples  $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ , where  $\mathbf{x}_i \in \mathcal{X}$  is a set of features (which we call feature-set) and  $y_i \in \{\pm 1\}$  is a class label. We focus on the problem of binary classification.

We call the number of features in a feature-set  $\mathbf{x}_i$  its size and denote it by  $|\mathbf{x}_i|$  ( $0 < |\mathbf{x}_i|$ ). We call a feature-set of size  $k$  a  $k$ -feature-set.

Let  $\mathcal{F} = \{f_1, f_2, \dots, f_L\}$  be  $L$  kinds of features and  $x_{i,j} \in \mathcal{F}$  be a feature  $1 \leq j \leq |\mathbf{x}_i|$  included in  $\mathbf{x}_i$ . For example, feature-sets for NLP tasks consist of words, part-of-speech tags, character types of words, and so on. We denote a feature-set consisting of a feature  $f \in \mathcal{F}$  as  $\{f\}$ .

To construct classifiers for feature-sets, we apply boosting algorithms. Firstly, we define a weak-hypothesis for classifying feature-sets. Secondly, we describe the application of the boosting algorithms to classification of feature-sets.

### 2.1 Weak-hypothesis

We apply the idea of real-valued predictions and abstaining (RVPA, for short) used in BoosTexter [17] for the weak-hypothesis classifying feature-sets. The RVPA idea is to force each weak-hypothesis to output a confidence value of zero for feature-sets which do not satisfy the given condition. Let  $\mathbf{a}$  and  $\mathbf{b}$  be feature-sets, then we denote  $\mathbf{a} \subseteq \mathbf{b}$  if  $\mathbf{b}$  contains  $\mathbf{a}$ .

**Definition 1** *Weak-hypothesis for classifying feature-sets*  
Let  $\mathbf{f}$  and  $\mathbf{x}$  be feature-sets, and  $c$  be a real number, called a confidence value, then a classifier for feature-sets is defined as

$$h_{(\mathbf{f},c)}(\mathbf{x}) = \begin{cases} c & \mathbf{f} \subseteq \mathbf{x} \\ 0 & \text{otherwise} \end{cases}$$

We select  $T$  sets of  $\mathbf{f}$  and  $c$  on  $T$  times boosting iteration, where  $T > 0$ . We regard a pair of  $(\mathbf{f}, c)$  as a rule of the weak-hypothesis.

### 2.2 Applying Boosting

We apply boosting algorithms to construct classifiers for classifying feature-sets. Boosting selects  $T$  hypotheses to produce a final hypothesis  $class$  by using a given weak learner.  $class$  is defined as follows:

$$class(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^T h_{t(\mathbf{f}_t, c_t)}(\mathbf{x})\right)$$

The weak learner for inducing weak-hypotheses accepts  $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$  as input training samples with weights over samples  $\{w_{t,1}, \dots, w_{t,m}\}$ .  $w_{t,i}$  is the weight of sample number  $i$  on boosting iteration  $t$ , where  $0 < w_{t,i}$  for  $1 \leq i \leq m$  and  $1 \leq t \leq T$ . Given such input, the weak learner computes a weak-hypothesis  $h_{t(\mathbf{f}_t, c_t)}$  on boosting iteration  $t$ , where  $\mathbf{f}_t \in \mathcal{X}$  and  $c_t$  is a confidence value.

We examine two types of AdaBoost algorithms for constructing classifiers [16, 7]. Let  $\pi$  be a proposition, and  $\llbracket \pi \rrbracket$

be 1 if  $\pi$  holds and 0 otherwise. The AdaBoost with RVPA indicates to find a feature-set  $\mathbf{f}$  minimizing the following equation [16].

$$\sum_{y \in \{-1, +1\}} W_{t,y}(\mathbf{f}) * \exp(-y * h_{(\mathbf{f},c)}) + W_t(-\mathbf{f}) \quad (1)$$

where  $W_{t,y}(\mathbf{f}) = \sum_{i=1}^m w_{t,i} [[\mathbf{f} \subseteq \mathbf{x}_i \wedge y_i = y]]$  and  $W_t(-\mathbf{f}) = \sum_{i=1}^m w_{t,i} - W_{t,+1}(\mathbf{f}) - W_{t,-1}(\mathbf{f})$ . It can be shown [16] that the Eq.(1) is minimized for a particular  $\mathbf{f}$  by choosing:

$$c = \frac{1}{2} \log\left(\frac{W_{t,+1}(\mathbf{f})}{W_{t,-1}(\mathbf{f})}\right) \quad (2)$$

Then, by plugging Eq. (2) into Eq. (1), Eq.(1) is rewritten as follows:

$$\sum_{i=1}^m w_{t,i} - (\sqrt{W_{t,+1}(\mathbf{f})} - \sqrt{W_{t,-1}(\mathbf{f})})^2 \quad (3)$$

From Eq. (3), we see that minimizing Eq. (3) is equivalent to selecting a feature-set  $\mathbf{f}$  maximizing *gain*, which is defined as follows:

$$gain(\mathbf{f}) \stackrel{\text{def}}{=} |\sqrt{W_{t,+1}(\mathbf{f})} - \sqrt{W_{t,-1}(\mathbf{f})}| \quad (4)$$

We select a  $\mathbf{f}$  maximizing *gain* as  $t$ -th weak-hypothesis  $\mathbf{f}_t$ . After obtaining  $\mathbf{f}_t$  and  $c_t$ , we update the weights over samples by the following equation.

$$w_{t+1,i} = w_{t,i} \exp(-y_i h_{t(\mathbf{f}_t, c_t)}) \quad (5)$$

where  $1 \leq i \leq m$ . Two types of AdaBoost algorithms what we examine are the followings. The first is an AdaBoost for a real-valued prediction originally proposed by [16], which we call *AdaBoost-normalized* in this paper. The second is slightly modified version of *AdaBoost-normalized* used in ADTrees learning algorithm [7], which we call *AdaBoost-unnormalized* in this paper.

*AdaBoost-unnormalized* differs from *AdaBoost-normalized* in two ways. First one is initial weights over samples. Initial weights over samples in *AdaBoost-normalized* are  $w_{1,i} = 1/m$  ( $1 \leq i \leq m$ ), where  $m$  is the size of training sample set  $S$ . Compared with *AdaBoost-normalized*, initial weights over samples in *AdaBoost-unnormalized* are  $w_{1,i} = 1$  ( $1 \leq i \leq m$ ).

The other difference is that weight sum is normalized after updating weighs with Eq. (5) (i.e.  $\sum_{i=1}^m w_{t,i} = 1$ ) in *AdaBoost-normalized* at each round  $t$ .

### 2.3 Confidence Values and Pre-adjustment for Imbalanced Class Distribution

To apply boosting algorithms to NLP problems, we have to consider imbalanced class distribution and sparse feature distribution. In fact, it may well happen that  $W_{t,+1}(\mathbf{f})$  or  $W_{t,-1}(\mathbf{f})$  is very small or even zero. To avoid it, we use the smoothed values  $\varepsilon$  presented in [16]. We use the following Eq. (6) for the confidence values and weights updating.

$$c = \frac{1}{2} \log\left(\frac{W_{t,+1}(\mathbf{f}) + \varepsilon}{W_{t,-1}(\mathbf{f}) + \varepsilon}\right) \quad (6)$$

We set  $\varepsilon$  for *AdaBoost-normalized* to  $1/m$  and  $\varepsilon$  for *AdaBoost-unnormalized* to 1.

Furthermore, to reflect the imbalanced class distribution, we use the default rule and pre-adjustment presented in [15]. The default rule is  $\frac{1}{2} \log\left(\frac{W_{+1}}{W_{-1}}\right)$ , where  $W_y = \sum_{i=1}^m [[y_i = y]]$  for  $y \in \{\pm 1\}$ . Initial weights over samples are updated by the default rule before starting training.

## 3 Fast Training Methods

In this section, we present our algorithm called AdaBoost.DF for fast learning rules represented by feature conjunctions. First, we present three techniques for pruning candidates used in past research. Then, we present methods for generating candidate rules for efficient pruning. Finally, we present techniques for fast rule selection.

### 3.1 Candidate Pruning

We use the following pruning techniques that are used in the other boosting based algorithms [14, 12, 13].

- **Frequency constraint:** We employ a frequency threshold  $\xi$ , and examine candidate rules seen on at least  $\xi$  different examples.
- **Size constraint:** We employ a size threshold  $\zeta$ , and examine candidates whose size is no greater than  $\zeta$ .
- **Upper bound of gain:** We use upper bound of gain presented in [14]. This is defined as follows.

$$u(\mathbf{f}) \stackrel{\text{def}}{=} \max(\sqrt{W_{t,+1}(\mathbf{f})}, \sqrt{W_{t,-1}(\mathbf{f})})$$

For any feature-set,  $\mathbf{f}'$ , which contains  $\mathbf{f}$  (i.e.  $\mathbf{f} \subseteq \mathbf{f}'$ ), the *gain*( $\mathbf{f}'$ ) is bounded under  $u(\mathbf{f})$ , since  $0 \leq W_{t,+1}(\mathbf{f}') \leq W_{t,+1}(\mathbf{f})$  and  $0 \leq W_{t,-1}(\mathbf{f}') \leq W_{t,-1}(\mathbf{f})$ . Thus, if  $u(\mathbf{f})$  is less than  $\tau$ , which is *gain* of the current optimal rule, then candidate rules containing  $\mathbf{f}$  are safely pruned. Please refer to [14] for a detailed explanation.

### 3.2 Generating Candidate Rules

When learning rules represented by feature conjunctions, the way we generate candidate rules affects the training time. For example, a feature-set  $\{a, b\} \in \mathcal{X}$  can be generated from  $\{a\}$  and  $\{b\}$ . Thus, to improve training time, we consider a method for generating candidate rules suited for pruning.

We denote  $\mathbf{f}' = \mathbf{f} + f$  as the creation of  $k + 1$ -feature-set  $\mathbf{f}'$  by adding a feature  $f$  to a  $k$ -feature-set  $\mathbf{f}$ . We denote  $ID(f)$  as a function to return *id* that is an integer corresponding to  $f$ . Then, we define *gen* to create a new candidate as follows.

$$gen(\mathbf{f}, f) = \begin{cases} \mathbf{f} + f & \text{if } (ID(f) > ID(f')) \text{ for } \forall f' \in \mathbf{f} \\ \{\} & \text{otherwise} \end{cases}$$

where  $\{\}$  is 0-feature-set.

We consider methods to control the order of generating candidate rules by defining different *ID* function. For example, consider the following situation: We try to find a 2-feature-set maximizing *gain* from candidates consisting of features  $a, b, c \in \mathcal{F}$  with the relation of  $u(\{c\}) < \tau = gain(\{a\}) < u(\{b\})$ .

If  $ID(a) \langle ID(b) \langle ID(c)$  and we generate candidates, we obtain  $\{a, b\} = gen(\{a\}, b)$ ,  $\{a, c\} = gen(\{a\}, c)$ , and  $\{b, c\} = gen(\{b\}, c)$  as candidate 2-feature-sets, and no candidate is generated from  $gen(\{b\}, a)$ ,  $gen(\{c\}, a)$ , and  $gen(\{c\}, b)$ . When generating candidate feature-sets with in the case of  $ID(c) \langle ID(b) \langle ID(a)$ ,  $\{a, b\} = gen(\{b\}, a)$  is only generated as a candidate 2-feature-set because candidates derived from  $\{c\}$  are pruned by the relation of  $u(\{\{c\}\}) < \tau$  before examining *gen*.

To prune candidate feature-sets efficiently, we consider a method to generate as many infrequent candidate feature-sets as possible. We think that infrequent candidates can be pruned by the following techniques.

- Pruning by frequency constraint  $\xi$ , because their frequencies are low
- Pruning by upper bound of gain  $\tau$ , because *gain* values of infrequent feature-sets tend to be low



```

##  $S = \{(x_i, y_i)\}_{i=1}^m : x_i \in \mathcal{X} \text{ and } y_i \in \{\pm 1\}$ 
##  $f$  : a feature ( $f \in \mathcal{F} = \{f_1, \dots, f_L\}$ )
##  $f_q(f, y) : \sum_{i=1}^m [\{f\} \subseteq x_i \wedge y_i = y]$ 
##  $f_q(f) : f_q(f, +1) + f_q(f, -1)$ 
##  $ent[f]$ : Entropy of  $f$ 
##  $sortByEnt(\mathcal{F}, ent)$ : Sort features  $x \in \mathcal{F}$ 
## in ascending order based on their entropies.
##  $(a \% b)$ : Return the remainder of  $(a \div b)$ .
procedure distributeFtToBuckets( $S, n$ )
begin
## Prepare  $N$ -buckets
 $B = \{B[1], \dots, B[N]\}$ 
## Calculate entropy of each feature
For ( $f \in \mathcal{F}$ )
begin
 $p_{(+)} = f_q(f, +1) / f_q(f)$ 
 $p_{(-)} = f_q(f, -1) / f_q(f)$ 
 $ent[f] = -p_{(+)} \log_2 p_{(+)} - p_{(-)} \log_2 p_{(-)}$ 
end
## Sort features based on their entropy values
## and insert the results into  $F_s$ 
 $F_s \leftarrow sortByEnt(\mathcal{F}, ent)$ 
## Insert features into buckets
For  $i=1 \dots L : B[(i \% N)] \leftarrow F_s[i]$ 
return  $B$ 
end

```

**Fig. 1:** Distribute features to buckets based on entropies

To control generation of candidates, we assign small integer to infrequent feature as *id* for generating infrequent candidates based on *gen*. As a result, many candidates including infrequent features are generated.

Let  $f_q(\mathbf{f})$  be the number of samples including a feature-set  $\mathbf{f}$  in  $S$ , which is defined as  $f_q(\mathbf{f}) = \sum_{i=1}^m [\{\mathbf{f}\} \subseteq x_i]$ . When assigning an *id* to each feature, we use the following conditions.

- if  $(f_q(\{a\}) < f_q(\{b\}))$  then  $(ID(a) < ID(b))$
  - if  $(f_q(\{a\}) = f_q(\{b\}) \ \& \ lexo(a, b) = 1)$  then  $(ID(a) < ID(b))$
- where  $a, b \in \mathcal{F}$  are features and  $lexo(a, b)$  is a function to compare  $a$  and  $b$  with lexicographic order: if  $(a < b)$  then return 1, otherwise 0. We call the *id* assigning method **freq-numbering**.

For example, when assigning an *id* to each feature in  $\{apple, orange, peach\}$  in the case of  $\{f_q(\{apple\}) = 2, f_q(\{orange\}) = 2, f_q(\{peach\}) = 1\}$ , the result is  $\{ID(apple) = 2, ID(orange) = 1, ID(peach) = 3\}$ .

In addition to freq-numbering, we examine assigning *id* to each feature based on the order of their appearance in the training corpus, which we call **app-numbering**. In app-numbering, when we observe a new feature, we assign a unique integer  $I$  ( $0 \leq I$ ) as *id* to it. After that, if we observe a next new feature, we assign a unique integer  $I + 1$  to it.

### 3.3 Training with Distributed Features

Several boosting algorithms examine all features on every round for selecting an optimal rule satisfied with a criterion [7, 17, 5]. However, such methods are very time-consuming because a weak learner evaluates all features.

To improve the training speed of boosting algorithms, we consider methods to limit search space by distributing features to buckets and selecting a rule containing a feature belonging to a chosen bucket at each round. When selecting a rule by using that bucket, our method selects a feature-set maximizing *gain* in candidates generated from features in the selected bucket. As a result, training speed

```

##  $F_k$  : A set of  $k$ -feature-sets
##  $\tau$  : The current optimal gain
##  $f$  : A feature ( $f \in \mathcal{F} = \{f_1, \dots, f_L\}$ )
##  $\mathbf{f}_t$  : Optimal feature-set at round  $t$ 
procedure findConj( $F_k$ )
begin
For  $\mathbf{f} \in F_k$ 
begin
if ( $f_q(\mathbf{f}) < \xi$ ) continue
if ( $\tau < gain(\mathbf{f})$ )
begin
 $\tau = gain(\mathbf{f}), \ \mathbf{f}_t = \mathbf{f}$ 
end
if ( $u(\mathbf{f}) < \tau$ ) continue
For ( $f \in \mathcal{F}$ )
begin
## Generating candidate rules - see Sec. 3.2
 $F_{k+1} \leftarrow gen(\mathbf{f}^t, f)$ 
end
end
if ( $k < \zeta$ ) findConj( $F_{k+1}$ )
else return  $\mathbf{f}_t$ 
end

```

**Fig. 2:** Find an optimal feature-set at boosting round  $t$

is much faster. (We only examine a subset of all candidate rules at each boosting round.) We consider the following methods for distributing features to buckets.

- Entropy based distribution (*E-dist*): *E-dist* distributes features to buckets in ascending order based on their entropies. In this distribution, features are distributed to buckets while keeping average entropies in each bucket roughly the same.
- Frequency based distribution (*F-dist*): *F-dist* distributes features to buckets in ascending order based on their frequencies. In this distribution, features are distributed to buckets while keeping average frequencies in each bucket roughly the same.
- Random Distribution (*R-dist*): *R-dist* creates  $N$  buckets by randomly selecting features to be distributed to each bucket.

Fig. 1 describes the distribution of features based on *E-dist*. In addition to these distribution methods, we use a random selection of features to be examined, as proposed in [6]. This method selects  $L/N$  types of features that are examined at each iteration. Compared with bucketing approaches of *E-dist* and *F-dist* that we propose, all features are not ensured to be examined in the method based on random selection.

Fig. 2 describes an algorithm for learning a rule. At each iteration, one of  $N$ -buckets is given an initial 1-feature-sets  $F_1$ , and finds a feature-set maximizing *gain* from  $F_1$  and candidate rules generated from features in  $F_1$  (up to  $\zeta$ ). Fig. 3 describes an overview of our algorithm, which we call AdaBoost for Distributed Features (AdaBoost.DF, for short). If we set bucket size  $N$  to 1, then AdaBoost.DF examines all features on every round like [7, 17, 5].

## 4 English Syntactic Chunking

We use a task of English Syntactic Chunking (ESC) for our evaluation. We use the data set prepared for CoNLL-2000 shared tasks<sup>1</sup>. In this data set, the total of 10 base phrase, NP, VP, PP, ADJP, ADVP, CONJP, INITJ, LST, PTR,

<sup>1</sup> <http://lcg-www.uia.ac.be/conll2000/chunking/>.

```

##  $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m : \mathbf{x}_i \in \mathcal{X}, y_i \in \{\pm 1\}$ 
##  $\varepsilon$  : A smooting value.
##  $\varepsilon = 1/m$  for AdaBoost-normalized
##  $\varepsilon = 1$  for AdaBoost-unnormalized
##  $N$  : Bucket size
##  $Z_t$  : A normalization factor chosen so that  $w_{t+1,i}$ 
## will be a distribution at  $t + 1$  iteration.
procedure AdaBoost.DF()
begin
## Initializing weights:
 $c_0 = \frac{1}{2} \log(\frac{W_{+1}}{W_{-1}})$ 
For  $i = 1, \dots, m$ :
begin
if (AdaBoost-normalized) then
 $w_{1,i} = \exp(c_0) / (m * Z_0)$ 
else if (AdaBoost-unnormalized) then
 $w_{1,i} = \exp(c_0)$ 
end
## Distributing features into
## buckets  $B = \{B[1], \dots, B[N]\}$ 
 $B = \text{distributeFtToBuckets}(S, N)$ ;
## Training
For  $t = 1, \dots, T$ :
begin
##(1) Selfselect a feature-set  $\mathbf{f}_t$ 
 $\mathbf{f}_t = \text{findConj}(B[t \% N])$ 
##(2) Calculate prediction values of  $\mathbf{f}_t$  :
 $c_t = \frac{1}{2} \log(\frac{W_{t,+1}(\mathbf{f}_t) + \varepsilon}{W_{t,-1}(\mathbf{f}_t) + \varepsilon})$ 
##(3) Update weights:
For  $i = 1, \dots, m$ :
begin
if (AdaBoost-normalized) then
 $w_{t+1,i} = w_{t,i} \exp(-y_i h_{t(\mathbf{f}_t, c_t)}(\mathbf{x}_i)) / Z_t$ 
else if (AdaBoost-unnormalized) then
 $w_{t+1,i} = w_{t,i} \exp(-y_i h_{t(\mathbf{f}_t, c_t)}(\mathbf{x}_i))$ 
end
end
## Output: the final hypothesis
return  $\text{class}(\mathbf{x}) = \text{sign}(c_0 + \sum_{t=1}^T h_{t(\mathbf{f}_t, c_t)}(\mathbf{x}))$ 
end

```

**Fig. 3:** An overview of *AdaBoost.DF* based on *AdaBoost-normalized* [16] and *AdaBoost-unnormalized* [7] and SBAR are annotated. This task aims to identify these 10 types of chunks.

This data set consists of 4 sections (15-18) of the WSJ part of the Penn Treebank for the training data, and one section (20) for the test data. The training data and the test data consist of 211,727 and 47,377 tokens, respectively.

Each base phrase consists of one word or more. To identify word chunks becoming phrases, we use **IOE2** representation to represent word chunks because the ESC parser based on IOE2 representation trained with Support Vector Machines (SVMs) has shown good performance [11].

IOE2 expresses chunk state by using three tags: E, I and O. An E tag is given for every token existing at the end of a chunk. An I tag is given for those inside of a chunk. An O tag is given for outside of a chunk.

For instance, “[He] (NP) [reckons] (VP) [the current account deficit] (NP)...” is represented by IOE2 as follows. “He/E-NP reckons/E-VP the/I-NP current/I-NP account/I-NP deficit/E-NP”.

E-NP and E-VP are the end of an NP and VP. I-NP is insides of an NP. Two types of tags are created for each class

by IOE2 representation, and 21 ( $= 10 * 2 + 1$ ) types of tags are created for this task setting.

When identifying a tag for each word in a sentence consisting of  $n$  words  $\{w_1, \dots, w_n\}$ , we classify each word with the following features.

- Words and part-of-speech (POS) tags within a 5-token window: In addition to the current word  $w_j$  ( $1 \leq j \leq n$ ) and its POS tag  $p_j$ , we use  $\{w_{j-2}, w_{j-1}, w_{j+1}, w_{j+2}, p_{j-2}, p_{j-1}, p_{j+1}, p_{j+2}\}$ , which are words and POS tags within two left and two right.
- Predicted tags of the two words on the right: We classify words in a sentence into classes from right to left (from the end of the sentence to the beginning). We use tags  $t_{j+1}$  and  $t_{j+2}$  assigned to  $w_{j+1}$  and  $w_{j+2}$  with highest scores as features.

We use the one-vs-the-rest method to extend the binary classifier to multi-class in all the experiments. Each ESC parser consists of 21 classifiers in this extension. To identify proper chunks, we use the Viterbi search. We map confidence value of each classifier into the range of 0 to 1 with sigmoid function<sup>2</sup>, and select a tag sequence which maximizes the sum of those log values by Viterbi search.

## 5 Experimental Results

Fig. 4 shows the average training times obtained with ESC parsers from the perspective of bucketing methods, bucketing size, and size constraint. The boosting-iteration number  $T$  was experimentally set to 10,000. We set  $\xi$  to  $\{0, 5, 10\}$ ,  $\zeta$  to  $\{1, 2, 3\}$ ,  $N$  to  $\{1, 10, 100, 1000\}$ , respectively<sup>3</sup>.

From Fig. 4, trainings with **freq-numbering** are faster than trainings with **app-numbering** for all parameter settings. These results have shown that generating candidate rules based on freq-numbering contributes to improved training times. Furthermore, trainings with bucket size  $N = 10, 100$  and 1000 show 2-3 orders improvements compared to trainings with  $N = 1$ .

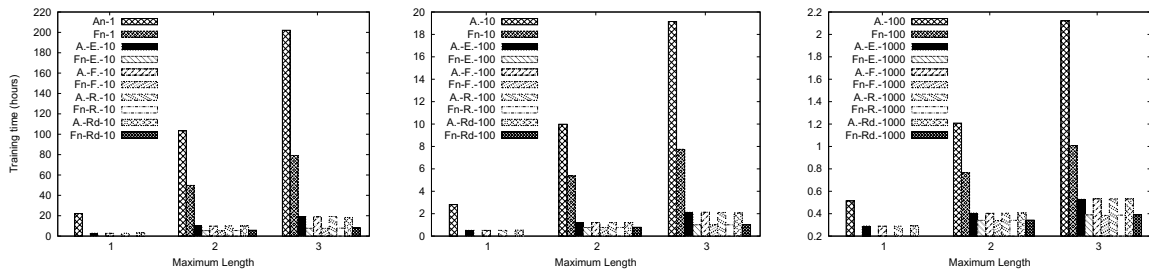
Fig. 5 shows average accuracies obtained with the ESC parsers. We used the standard measures for evaluation: precision, recall and their harmonic mean  $F_{\beta=1}$ . All the average accuracies obtained with ESC parsers trained with  $\zeta = \{2, 3\}$  have shown better performance than all the average accuracies obtained with ESC parsers trained with  $\zeta = 1$ . These results have shown that bucketing with finding feature conjunctions contribute to improved accuracies.

Table 1 lists average F-measures obtained with two types of boosting algorithms. These results have shown that freq-numbering contributes to improved training speeds while keeping accuracy. ESC parsers based on *AdaBoost-unnormalized* have shown slightly better performances than *AdaBoost-normalized*.

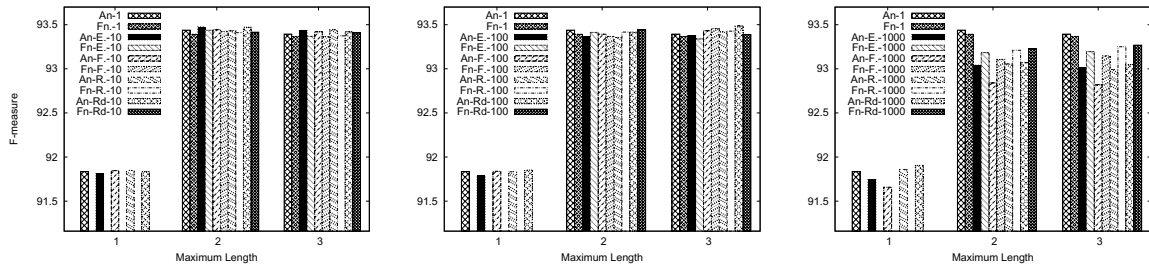
To examine the effect of the boosting iterations, we trained ESC parsers with parameters of  $T = 10,000$ ,  $N = \{1, 10, 100, 1000\}$ ,  $\xi = 0$  and  $\zeta = 2$ . Table 2 lists the best accuracies obtained with the ESC parsers over test data. The results have shown that ESC parsers based on *AdaBoost.DF* perform better by increasing the number of boosting iteration.

<sup>2</sup>  $s(X) = 1/(1 + \exp(-\beta X))$ , where  $X$  is a output of a classifier ( $X = c_0 + \sum_{t=1}^T h_{t(\mathbf{f}_t, c_t)}$ ) created by a boosting algorithm for a class. We set the  $\beta$  to 5 in this experiment.

<sup>3</sup> We conducted all the experiments with  $T = 10,000$  under Linux using 3.0 Ghz Xeon processor and 6 Gbyte of main memory. All systems are implemented in C++.



**Fig. 4:** Comparison of average training times after 10,000 iteration with two types of generating candidate rules, bucketing sizes (1,10,100,1000), frequencies constraint (0,5,10) and size constraints (1,2,3). “An”, “Fn”, “E.”, “F.”, “R.”, “Rd” and each number mean app-numbering, freq-numbering, E-dist, F-dist, R-dist, Random selection and bucket size, respectively. Graphs placed on leftmost, center and rightmost show average training times obtained with bucket size 0 and 10, E-dist of bucket size 100 and 1000, respectively. Training times for each bar is an average training times obtained with three kinds of frequency constraint ( $\xi=0,5,10$ ).



**Fig. 5:** Comparison of average accuracy ( $F_{\beta=1}$ ) after 10,000 iteration with two types of generating candidate rules, bucketing sizes (1,10,100,1000), frequencies constraint (0,5,10) and size constraints (1,2,3). Notations for bars are the same as those of Fig. 4. Graphs placed on leftmost, center and rightmost show average accuracies obtained with bucket size 0 and 10, bucket size 0 and 100, and bucket size 0 and 1000, respectively. Accuracies for each bar is an average accuracy of three kinds of frequency constraint.

**Table 1:** Comparison of two types of boosting algorithms. We lists the average F-measures obtained with the same parsers used in Figure 4 and 5.

Bucket sizes \ Numbering	AdaBoost-unnormalized		AdaBoost-normalized	
	app.	freq.	app.	freq.
$N = 1$	92.68	92.67	93.09	93.05
$N = 10$	92.71	92.67	93.09	93.08
$N = 100$	92.63	92.63	93.13	93.10
$N = 1000$	92.23	92.44	92.93	93.02

**Table 2:** The best results for various bucketing methods and bucket sizes on test data. We list the results of ESC parsers trained with parameters of  $T = 100,000$ ,  $N = \{1, 10, 100, 1000\}$ ,  $\xi = 0$ ,  $\zeta = 2$  and freq-numbering.

AdaBoost-normalized				
	E-dist	F-dist	R-dist	Rand
$N=1$	93.58			
$N=10$	93.67	93.68	93.59	93.71
$N=100$	93.65	93.68	93.62	93.71
$N=1000$	93.60	93.40	93.38	93.57
AdaBoost-unnormalized				
	E-dist	F-dist	R-dist	Rand
$N=1$	93.66			
$N=10$	93.75	93.67	93.69	93.69
$N=100$	93.72	93.68	93.68	93.72
$N=1000$	93.71	93.76	93.70	93.75

Fig. 6 shows F-measure obtained with different boosting iteration numbers over test data. Around boosting iteration 4,000, the ESC parsers trained by AdaBoost.DF with  $N = 1,000$  have shown lower performance than the ESC parser trained with size of  $N = 1$ . However, after iteration 4,000, the ESC parsers trained with AdaBoost.DF have shown better performance.

Table 3 lists comparison of the previous best results. ESC parsers based on AdaBoost.DF with parameters of  $N = 1,000$ ,  $\zeta = 2$  and *AdaBoost-unnormalized* have shown competitive performances to the previous results. Furthermore, the creation of these ESC parsers took only times with parameters of  $N = 1,000$ ,  $T = 10,000$ , freq-numbering and *AdaBoost-unnormalized* less than 1 hour. These results have shown that our approach can create ESC parsers to be competitive with the state-of-the-art results in less time.

The classifier based on random selection has shown better performance in our ESC parsers. However, there is a drawback: Recreation of the same classifier is not ensured by random selection because different features are selected, even if the same training sets are given.

## 6 Related Work

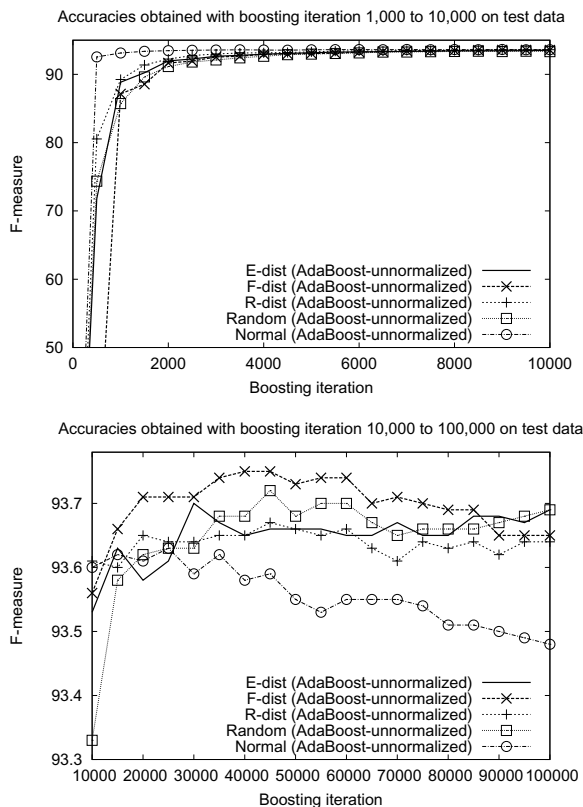
Pfahringner et al. proposed methods for finding feature conjunctions by selecting a path to be examined by following criteria, such as, a path including the current optimal rule, a path having the heaviest sum of weights, and so on [15].

Kudo et al. proposed to perform several pseudo iterations, in which the optimal feature is selected from the cache that maintains the features explored in the previous iterations [13].

Collins and Koo proposed to avoid unnecessary calculations by only updating *gain* of features co-occurring with a rule feature on examples at each round [5].

AdaBoost.MH<sup>RK</sup> takes the k most important weak-hypothesis into account [18].

LazyBoosting randomly selects a small proportion of features and selects a rule represented by a feature from the features at each iteration of the boosting algorithm [6].



**Fig. 6:** Accuracies ( $F_{\beta=1}$ ) obtained with ESC parsers trained with AdaBoost-unnormalized and freq-numbering over test data. Parameters for training with AdaBoost.DF are  $N = 1000$ ,  $\zeta = 2$ ,  $\xi = 0$ . “Normal” denotes ESC parsers trained with  $N = 1$ ,  $\zeta = 2$ ,  $\xi = 0$ .

**Table 3:** Comparison with previous best results. We list best results obtained by ESC parsers trained with  $\xi = 0$ ,  $\zeta = 2$  and AdaBoost-unnormalized.  $N = 1000$  means bucket size for creating classifiers.

MODEL	$F_{\beta=1}$
SVMs based on IOB2 rep. [10]	93.48
Regularized Winnow (RW) [20]	93.57
RW+ linguistic features[20]	94.17
SVMs based on IOE2 rep. [11]	93.85
SVMs-voting[11]	93.91
Perception in two layers [4]	93.74
Alternating Structure Optimization (ASO) [1]	93.60
ASO + unlabeled data [1]	94.39
CRF[13]	93.76
CRF+Reranking[13]	94.12
ME based a bidirectional inference[19]	93.70
<b>Boosting without bucketing</b>	93.66
<b>Boosting with E-dist</b>	93.71
<b>Boosting with F-dist</b>	93.76
<b>Boosting with R-dist</b>	93.70
<b>Boosting with Random selection</b>	93.75

Bagging crates multiple classifiers by making bootstrap replicates of the learning set and using these as new learning sets [2].

Our approach has the following characteristics: First one is that AdaBoost.DF generates candidate rules suited for pruning with upper bound of gain and frequency constraint. The other is that AdaBoost.DF limits search spaces by distributing features to several buckets and repeatedly selecting a bucket and finding a feature conjunction generated from features in the selected bucket. Furthermore, com-

pared with the feature selection based on random selection [6], our bucketing approaches based on E-dist and F-dist enable us to recreate the same classifiers.

## 7 Conclusion

We have proposed fast training methods for boosting algorithms learning rules represented by feature conjunctions, which we call AdaBoost.DF. AdaBoost.DF distributes features to several buckets, and induces a feature conjunction as a rule from limited search space by a bucket at each round. The experimental results of English Syntactic Chunking have shown improvement of 2-3 orders of magnitude for training speed without loss in accuracy.

We think that AdaBoost.DF approaches can accept weak learners such as decision trees [3], sub-tree based decision stump [12], and so on. The future work should evaluate AdaBoost.DF approach by using those weak learners with different NLP tasks. Future work should also examine AdaBoost.DF approach on other boosting algorithms, such as WeightBoost [9].

## References

- [1] R. Ando and T. Zhang. A high-performance semi-supervised learning method for text chunking. In *Proc. of the 43rd ACL*, pages 1–9, June 2005.
- [2] L. Breiman. Bagging predictors. *Technical Report 421, University of California at Berkeley*, 1994.
- [3] X. Carreras, L. Màrques, and L. Padró. Named entity extraction using adaboost. In *Proc. of CoNLL-2002*, pages 167–170, 2002.
- [4] X. Carreras and L. Màrquez. Phrase recognition by filtering and ranking with perceptrons. In *RANLP*, pages 205–216, 2003.
- [5] M. Collins and T. Koo. Discriminative reranking for natural language parsing. *Comput. Linguist.*, 31(1):25–70, 2005.
- [6] G. Escudero, L. Màrquez, and G. Rigau. Boosting applied to word sense disambiguation. In *Proc. of 11th ECML*, pages 129–141, 2000.
- [7] Y. Freund and L. Mason. The alternating decision tree learning algorithm. In *Proc. of 16th ICML*, pages 124–133, 1999.
- [8] R. Iida, K. Inui, and Y. Matsumoto. Exploiting syntactic patterns as clues in zero-anaphora resolution. In *Proc. of 44-th ACL*, 2006.
- [9] R. Jin, Y. Liu, L. Si, J. Carbonell, and A. G. Hauptmann. A new boosting algorithm using input-dependent regularizer. In *Proc. of 20th ICML*, 2003.
- [10] T. Kudo and Y. Matsumoto. Use of support vector learning for chunk identification. In *Proc. of CoNLL-2000 and LLL-2000*, pages 142–144, 2000.
- [11] T. Kudo and Y. Matsumoto. Chunking with Support Vector Machines. In *Proc. of NAACL 2001*, pages 192–199, 2001.
- [12] T. Kudo and Y. Matsumoto. A boosting algorithm for classification of semi-structured text. In *Proc. of EMNLP 2004*, pages 301–308, July 2004.
- [13] T. Kudo, J. Suzuki, and H. Isozaki. Boosting-based parse reranking with sub-tree features. In *Proc. of 43rd ACL*, pages 189–196. Association for Computational Linguistics, June 2005.
- [14] S. Morishita. Computing optimal hypotheses efficiently for boosting. *Progress in Discovery Science, Springer*, pages 471–481, 2002.
- [15] B. Pfahringer, G. Holmes, and R. Kirkby. Optimizing the induction of alternating decision trees. *LNCS*, 2035:477+, 2001.
- [16] R. E. Schapire and Y. Singer. Improved boosting using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.
- [17] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.
- [18] F. Sebastiani, A. Sperduti, and N. Valdambrini. An improved boosting algorithm and its application to text categorization. In *Proc. of 9-th CIKM*, pages 78–85, 2000.
- [19] Y. Tsuruoka and J. ichi Tsujii. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *HLT/EMNLP*, 2005.
- [20] T. Zhang, F. Damerou, and D. Johnson. Text chunking using regularized winnow. In *ACL*, pages 539–546, 2001.

# Reusability of a corpus and a treebank to enrich verb subcategorisation in a dictionary

Arantza Díaz de Ilarraza, Koldo Gojenola and Maite Oronoz  
Department of Computer Languages and Systems  
University of the Basque Country, P.O. box 649, E-20080 Donostia  
*jipdisaa, jipgogak, jiporanm@si.ehu.es*

## Abstract

This paper deals with the reusability of a corpus and a treebank to enrich verb subcategorisation in a static resource, a dictionary. Two experiments have been performed to propose: a) new subcategorisation information for verb entries included in the dictionary, and b) new verb entries. For the verb subcategorisation enrichment, inconsistencies between the information obtained from the corpus and the dictionary were found by means of a tool called *Saroi*. The same tool is used to propose new entries. A verb is proposed for its inclusion in the dictionary if it is found in the corpus but not in the dictionary, and it also appears in the treebank.

Our aim is to enrich the dictionary in two ways; a) adding verb subcategorisation information after looking for inconsistencies between the verbs that appear in a corpus and those that appear in the dictionary, and b) enriching EH with verb entries found in the corpus but that are missing in the dictionary after checking its existence in the treebank. The enrichment proposal lists will be presented to linguists. A *feedback* process has been performed as we use the dictionary to enrich itself.

The remainder of this paper is organised as follows: section 2 describes the used resources; in section 3 we will analyse *Saroi*, a dependency-tree inspection tool; section 4 explains the preprocessing work, and sections 5 and 6 show the performed experiments. Finally, some conclusions are outlined in section 7.

## 1 Introduction

This paper deals with the reusability of a corpus and a treebank to enrich verb subcategorisation in a dictionary. Dictionaries are a basic and very rich source of lexical information. However, their creation is very time consuming and sometimes dictionaries do not reflect changes in language usage. Several works have been carried out with the aim of automatically enriching dictionaries. They tackle a great variety of aspects going from the sources from which data was extracted to the output resources to be created. For example, in [8], a dictionary of word combinations was automatically enriched using information extracted by means of a dependency parser. In another work, the Prague Dependency Treebank was used to learn verb subcategorisation frames for Czech by means of machine learning techniques. In [10] frequencies about words were extracted from a corpus and added to the Longman Dictionary.

In our case, the dictionary we want to enrich is a general purpose monolingual dictionary called *Euskal Hiztegia* (EH)[11]. Since its creation the Basque Academy has made new decisions about the standard forms of some words. Moreover, we assume that corpora better reflect the changes in the language.

In order to reduce manual work to the checking of the results, we reuse already developed resources: a) a corpus to extract verbs and their realisation schemas, b) the EH dictionary to obtain verbs and their subcategorisation patterns, and c) the Basque Dependency Treebank. To manage all these resources, we have used a dependency-tree inspection tool called *Saroi*.

## 2 Resources

Basque is an agglutinative language with relative free order among sentence elements. In finite verbs, the verb agrees in tense and mood with the subject, object or indirect object of the sentence. As [9] says, “The simplest forms of intransitive verbs are monovalent and mark agreement with the subject (NOR). Intransitive verbs can also have bivalent forms marking agreement with an absolutive argument (subject) and a dative argument (NOR-NORI). Finite transitive verb forms are minimally bivalent, marking agreement with an ergative argument (subject) and an absolutive (direct object) argument (NOR-NORK). In addition, there are trivalent forms that add agreement with a dative argument (NOR-NORI-NORK)”. The type of auxiliary verb used by each of these four types of verbs has been pointed out between parentheses. Three different resources are used:

- **Corpus.** It consists of verb realisation schemas obtained as a result of the automatic analysis of a corpus composed of 10,032,133 word-forms taken from a Basque newspaper [4]. A group of 2,541 verbs (including 367 multiword verbs) was extracted from this corpus with the aim of identifying their verbal syntactic pattern or realisation schema. In this list each verb is accompanied by the syntactic components found in its context, together with information about the type of auxiliary verb, and the proportion in which each type of auxiliary verb appears. Table 1 shows the data we extracted for the verb *etorri*.

Etorri "to come"(5649 occurrences)		
Aux. type	#	%
NOR-NORK	2	0.03 %
NOR	5331	94.37 %
NOR-NORI-NORK	0	0 %
NOR-NORI	316	5.59 %

Table 1: Auxiliary verb types with the verb etorri.

Corpora offer a vast and complete description of verb structures, nevertheless, as the information is automatically collected, errors can be produced.

- **Dictionary.** All the verb patterns were extracted from the *Euskal Hiztegia* (EH) dictionary.

We have used a TEI-conformant (*Text Encoding Initiative*) XML version of the dictionary as a source of information about 4,016 verbs. Apart from the headword, we extracted a tag that identifies the kind of auxiliary verb. Possible types are DA, DU, DIO, ZAIO, DA-DU . . . . The dictionary specifies the senses of each entry word. For most of the verb senses the type of the auxiliary is marked. For example, the verb *eratu* has two senses with an auxiliary mark, and one without it. The sense similar to *konpondu* ("to adapt") carries out a DA type auxiliary, while the second sense, similar to *moldatu*, *antolatu* ("to repair"), goes with a DU type auxiliary. In this case a combined DA-DU tag is automatically assigned to the verb *eratu* to collect both sense uses. The auxiliary type tags in the dictionary differ from those used in the corpus (see table 2).

Dictionary	Corpus
DA	NOR and NOR-NORI
DU	NOR-NORK and NOR-NORI-NORK
ZAIO	NOR-NORI
DIO	NOR-NORI-NORK

Table 2: Equivalences between auxiliary verb tags.

- **Trebank.** The 3LB project annotated corpora for Catalan, Spanish and Basque. The syntactically annotated Basque corpus contains 25,000 word-forms from a reference corpus [1] and 25,000 from newspapers. The corpus used for the treebank and the one for the realisation schemas are disjoint. The treebank was annotated using a dependency framework similar to [5] and the *Conference on Computational Natural Language Learning 2007* format.

We consider these three resources complementary to each other as the corpus reflects the real use of the nowadays language, the dictionary was compiled after a vast manual linguistic analysis, and the treebank combines both viewpoints.

### 3 A treebank inspection tool

The main goal of the *Saroi* system is to look for linguistic information in dependency-trees by means of rules that express the characteristics of the information we search. This system was also used to look for agreement errors in dependency-trees [6].

The system is composed of three main modules: a) a robust syntactic analyser, b) a rule compiler, and c) a module that coordinates the results of the analyser, applying different combinations of the already compiled rules. The specification language for the description of the rules is abstract, general, declarative, and based on linguistic information. Figure 1 shows the architecture of the system.

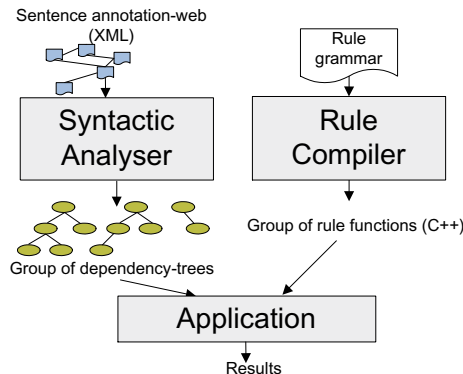


Fig. 1: Architecture of Saroi.

### 3.1 Syntactic analyser

The input of the syntactic analyser module is an annotation-web that follows an XML stand-off markup approach and that represents the linguistic information obtained by the analysis chain. The analysis chain [2] is composed of a morphosyntactic analyser, a tagger/lemmatiser [7], a chunker, and finally, a parser that obtains dependency-trees.

The information gathered in the XML documents that represent the dependency trees is ambiguous. That is, a document can store multiple dependency parses. *Saroi* deals with this ambiguity and creates independent dependency-trees.

In the syntactic analysis module there is an *enrichment module* that carries out two processes: makes explicit the agreement information in auxiliary verbs and enriches main verbs with the information described in section 2. Figure 2 shows part of the morphosyntactic analysis of the verb *etorri* ("to come") after the addition of the information extracted from the corpus (see table 1), and the dictionary.

```

<fs id="V-etorri-1" type="VerbInfo">
  <f name="frequency-features">
    <fs type="verb-frequency">
      <f name="occurrences"><nbr value="5649"/></f>
      <f name="NOR-%"><nbr value="94.37"/></f>
      <f name="NOR-NORK-%"><nbr value="0.03"/></f>
      <f name="NOR-NORI-NORK-%"><nbr value="0"/></f>
      <f name="NOR-NORI-%"><nbr value="5.59"/></f>
    </fs>
  <f name="NOT_in_EH" org="list">
    <sym value="Not_NOR-NORK"/>
    <sym value="Not_NOR-NORI-NORK"/>
  </f>
</fs>
  
```

Fig. 2: Part of the analysis of the verb etorri after the enrichment process.

## 3.2 Rule compiler

The rule grammar that constitutes the input of the rule compiler has been defined by means of a general specification language. The aim of this language is to search for any linguistic structure in a dependency tree. The use of an abstract specification language has several advantages: a) declarativeness, b) maintainability and, c) efficiency, as the abstract rules will be compiled to an object language (C++). The rules allow the traversing of the dependency tree while at the same time checking syntactic constraints.

In the rules we use linguistic information such as tags that define dependency relations between the elements of the sentence (e.g. *nsubj*, *ncobj*,...), as well as tags defining features of the syntactic elements (number, case, ...). Apart from this, some operators have been defined to navigate vertically the dependency-tree and to inspect linguistic features.

The rules, written in an abstract language, cannot be directly applied to a dependency tree because they must first be translated into executable statements. We defined and implemented a syntax-directed translation scheme [3] for that purpose.

## 4 Preprocessing

Therefore, we have three linguistic data resources with very different origins: a) a group of verbs together with information about the types of auxiliary verbs they appear with, extracted from a corpus, b) another group of verbs with the same information but extracted from a dictionary, and lastly, c) a treebank of correct and standard Basque. In addition, we have a system, *Saroi*, that looks for linguistic information in treebanks. So, we can reuse all these elements to enrich the dictionary. As the enrichment module manages verbal information from different origins, we can use this to obtain different lists of verbs. The verbs obtained from the corpus are 2,541 and those extracted from the dictionary, 4,016, with a total of 5,264 different verbs, showing that not all the verbs appear in both sources:

- 1,248 verbs only appear in the corpus (“Corpus Only, CO”): i) Verbs appearing in journalistic style but not in the dictionary, e.g. *klonatu* (“to clone”), ii) Mistyped verbs, and iii) Multiword verbs that do not appear neither as entries nor as subentries in the dictionary.
- 2,723 verbs are exclusively gathered in the dictionary “Dictionary Only, DO”). Examples of these verbs are those marked in the dictionary as: i) Infrequent verbs, e.g. *urgoitu* (“to get tired”), ii) Dialectal variants, e.g. *haurridetu* (“to make sister cities”) used in the French speaking area, and iii) Verb entries marked as highbrow. An example is, *hatsanditu* (“to get out of breath”).
- 1,293 verbs appear in both sources, corpus and dictionary (“Both, B”).

## 5 Finding inconsistencies

The resources used for this experiment are the “Both, B” list of verbs and *Saroi*. The main objective in this first experiment is to look for inconsistencies between the subcategorisation information that appears in the corpus and in the dictionary. For us an inconsistency occurs when the types of auxiliary verb in the corpus and in the dictionary are different. For example, the verb *zauritu* (“to wound”) appears with auxiliaries of type NOR in the corpus and with a DU tag in the dictionary (DU is equivalent to NOR-NORK and NOR-NORI-NORK, see table 2).

### 5.1 The experiment

Let us see step by step the process followed:

1. Analysis of the “B” verb list by means of the analysis chain mentioned in section 3.1.
2. Enrichment of these verbs with the information extracted from the corpus and from the dictionary. After the enrichment process has concluded, each of the verbs will have information similar to the one showed in figure 2.
3. Application of a set of four rules, one for each auxiliary verb type, to the resulting verb list using *Saroi*. Figure 3 shows the rule for detecting inconsistencies in auxiliary verbs of type NOR.

```
RULE INCONSISTENCY_IN_NOR_TYPE
Detect ( @.pos == 'ADI' & @.occurrences >4 &
        @.NOR-% >50 & @.Not.NOR )
```

Fig. 3: Detecting inconsistencies in NOR auxiliaries.

The rule in figure 3 can be paraphrased as: mark that a tree fulfils this rule if the current node (‘@’) has as part of speech (@.pos) ADI (verb), the verb appears in the corpus more than four times and goes with an auxiliary of type NOR with a proportion of more than 50%. But besides this, the entry in the dictionary indicates that the same verb does not usually carry a NOR auxiliary. So, we notice a clear inconsistency between the data extracted from the corpus (the verb appears more than half of the times with the NOR auxiliary) and those extracted from the dictionary (it does not appear with auxiliaries of type NOR).

We have only inspected the verbs with more than 4 occurrences in the corpus to avoid the appearance of mistyped words erroneously marked as verbs. In addition, we think that a clear inconsistency occurs if the proportion of an auxiliary verb in the corpus is more than 50%.

### 5.2 The results

In a list of 1,293 verbs, 53 (4%) present inconsistencies referring the auxiliary verb. In 45 of the cases (84.9%) there is an inconsistency of type NOR. 6 cases (11.3%)

showed a NOR-NORK inconsistency. 2 times (3.77%) a NOR-NORI difference appears, while no NOR-NORI-NORK inconsistencies are marked.

A priori, we expected a high proportion of NOR type inconsistencies before seeing the results. In Basque, when the verbs are used as impersonal, the ergative argument of the sentence (the subject of the clause) is ellided and verbs of type NOR-NORK turn into NOR. This fact is not reflected in the dictionary.

A linguist made manually a deeper analysis of the inconsistencies and found the following casuistry:

- In 36 of the cases (67.9%) there was a lack of some verb alternation (impersonal, inchoative, ...) in the dictionary. In this case, the alternating syntactic structures in the corpus together with their examples can be added to the dictionary.
- In 11 of the cases (20.7%) the verb usage in the corpus and in the dictionary differs. These are interesting for examining the real verb usage and the reasons for changes in language use.
- 5 errors (9.4%) were identified in the dictionary. We manually verified that when the subcategorisation tag in the dictionary indicated an auxiliary type, examples in the dictionary showed others.
- In one of the cases (1.9%), although the word-form was the same, the senses of the verb in the corpus and in the dictionary were different.

We have observed that from a list of 1,293 verbs 53 (4%) are marked by *Saroi* as inconsistencies. A linguist has confirmed that all the proposals present real inconsistencies, so we have obtained reliable results. The inconsistencies have been used to propose the inclusion of new verb alternations and new verb usage in the dictionary, and confirm the usefulness of the corpus as a source of language use information.

## 6 Adding new entries

The objective of this second experiment is to enrich EH with new verb entries found in the corpus and the treebank but that are missing in the dictionary. In this case we have used the “*Corpus Only, CO*” list of verbs, and the treebank. We consider that a verb could be proposed to be part of the dictionary if in addition to being in the corpus, it also appears in the treebank. As the treebank was manually tagged and contains correct linguistic information, we think that it offers enough guarantee for the purpose we follow. Treebanks have the advantage of having less noisy data compared to that obtained by automatic parsers.

### 6.1 The experiment

The process followed to look for verbs that appear in the corpus and in the treebank, but not in the dictionary, is the following one:

1. As we are looking in the treebank for specific verbs lemmas, first, we have automatically created a rule similar to the one in figure 4 for each

of the verbs appearing in the corpus (1,248 rules). In the rule in the figure 4, only the nodes in the dependency-trees with the ADI (verb) POS tag are inspected and if we find one with the lemma “*ados etorri*” (“to agree”) occurring in the corpus more than 10 times, the dependency-tree that fulfils the conditions is marked.

2. The rules are applied to the treebank using *Saroi*.

```
RULE VERB_ADOS_ETORRI
  Detect ( @.pos == 'ADI' &
           @.lemma == 'ados etorri' &
           @.occurrences >10 )
```

Fig. 4: Detection of a verb in the treebank.

## 6.2 The results

Table 3 presents in detail the results of this experiment. The first column shows the candidate verbs. Column 2 indicates the number of occurrences in the corpus while column 3 (Treeb.) shows the number of times in which rules have been activated in the treebank. This column has been divided into two, a) the part of the treebank that is composed of literary texts and, b) the part composed of journalistic texts. Finally, the last column (Propo?) indicates whether an expert proposes or not the verb for its inclusion in the dictionary. The reasons used by the linguist for accepting or rejecting the verbs that appear only in the corpus are diverse:

Verb	Corp. occurs	Treeb.		Propo?	
		EEBS occurs	Journ. occurs	Propo?	Reason
<i>not in dictionary</i>					
baloratu	>50	1	2	Reject	1R
blokeatu	>50	0	5	Accept	1A
diseinatu	>50	0	1	Accept	1A
exijitu	>50	1	2	Doubt	D
inbertitu	>50	0	4	Accept	1A
hitzartu	>50	0	7	Accept	2A
hitzeman	>50	0	2	Accept	2A
kaltetu	>50	0	1	Accept	2A
justifikatu	>50	0	1	Accept	1A
planteatu	>50	2	3	Accept	3A
menperatu	>50	3	0	Reject	2R
afliatu	>10	0	1	Accept	1A
berdintsu izan	>10	0	1	Doubt	D
deskubritu	>10	4	0	Reject	1R
erlazionatu	>10	1	0	Accept	1A
errebindikatu	>10	0	2	Accept	1A
errekurritu	>10	0	3	Doubt	D
finantzatu	>10	0	6	Accept	1A
kargugabetu	>10	0	1	Accept	1A
kartzelaratu	>10	1	0	Accept	1A
kolaboratu	>10	0	1	Accept	1A
komentatu	>10	1	0	Doubt	D
konplikatu	>10	1	0	Accept	1A
lanpetu	>10	1	0	Reject	3R
ingresatu	>10	0	2	Reject	1R
inkomunikatu	>10	0	6	Accept	1A
inkulpatu	>10	0	1	Reject	4R
inspiratu	>10	1	0	Accept	1A
integratu	>10	1	1	Accept	1A
konprometitu	>10	0	1	Accept	1A
kotizatu	>10	0	1	Accept	1A
kriminalizatu	>10	0	1	Doubt	D
merkaturatu	>10	0	5	Accept	1A
praktikatu	>10	1	0	Accept	1A
profitatu	>10	1	0	Accept	1A

Table 3: Candidate verbs.



- A candidate verb is accepted (A) if:
  - 1A. It has been manually looked up in four dictionaries and it is found in at least two.
  - 2A. It does not appear with this word-form in the EH dictionary but appears with a similar form in a subentry. For example, *hitzartu* (“to agree to”) does not appear but *hitz hartu* does with the same sense. The linguist proposes the word-form found in the corpus when it appears with the same spelling in most of the dictionaries.
  - 3A. The candidate verb appears in the dictionary but not as the preferred verb. For example, *planteatu* (“to bring up”) is marked as “*spanish influenced word*” and *ezarri* is proposed. In the rest of the dictionaries, *planteatu* is a standard entry. So, the linguist proposed the form found in the corpus.
- The reasons for rejecting (R) a candidate verb are:
  - 1R. Another form is preferred in all the rest of the dictionaries.
  - 2R. It does not appear as a dictionary entry but as a variant of the verb.
  - 3R. It does not appear in the dictionary as a verb but as an adjective.
  - 4R. It does not appear in any dictionary.
- The doubtful (D) verbs are those that could be found in only one of the four dictionaries.

Two thresholds were defined for this second experiment. One asking each verb to appear more than 50 times in the corpus, and a second one reducing the number of occurrences to 10. Table 3 shows that a high number of occurrences in the corpus does not necessarily mean a guarantee in the proposal. When the verb occurs in the corpus more than 50 times 72.7% of the verbs is accepted and 18% refused. For the second group of verbs (more than 10 times in the corpus) 66% is accepted and 16% refused. In this second group the number of refused verbs is lower, but the number of those marked as doubtful is higher. We have the impression that the verbs marked as doubtful probably would be accepted but the conditions we have established are quite strict. Besides, the contribution of verbs to the dictionary in the second group is higher.

The verb lists proposed in both experiments can be easily extended. When looking for inconsistencies, we could reduce the number of occurrences in the corpus, obtaining more inconsistencies. In the case of verb entries, asking, for example, 5 occurrences in the corpus, the proposed list will probably be larger.

## 7 Conclusions

This work examines the validity of corpora and treebanks in the enrichment of a more static resource, a dictionary. We have explored two different alternatives to enrich verbal information in a dictionary using both an unannotated corpus and a treebank. The experiments have been designed to obtain on the one hand,

a list of verbs that already exist in the dictionary but that present inconsistencies with verbs found in a corpus and, on the other hand, a list of verbs found in the corpus and treebank but that are missing in the dictionary.

By reusing already existing resources, the work carried out to obtain results from the corpus as well as the one to enrich the dictionary, usually a very time consuming task, has been reduced to the minimum.

The experiment for including verb entries in the dictionary shows that, regardless of the threshold used in the corpus, all the verbs appearing more than 4 times in the treebank composed of newspapers are accepted. This part of the treebank combines the actual use of the language with linguistic correctness. The acceptance level of verbs demonstrate the validity of treebanks as information source.

We think that the methodology we use is general for any language although it has a twofold implication: a) the appropriate resources must be available, and b) the linguistic information must be represented following the input specifications for *Saroi* (a general dependency representation in XML).

We are of the opinion that this work is extensible to the rest of words in this dictionary (i.e. *nouns, adjectives, ...*). Information concerning POS, examples, or usage domain could be added to the dictionary. Changing the source corpus, domain specific words could also be added.

**Acknowledgments.** This research is supported by the Univ. of the Basque Country (GIU05/52) and the Basque Government (ANHITZ project, IE06-185). Thanks to I. Aldezabal and A. Atutxa for their invaluable help.

## References

- [1] I. Aduriz, M. Aranzabe, J. M. Arriola, A. Atutxa, A. Díaz de Ilarraza, N. Ezeiza, K. Gojenola, M. Oronoz, A. Soroa, and R. Urizar. Methodology and steps towards the construction of epec, a corpus of written basque tagged at morphological and syntactic levels for the automatic processing. In *Corpus Linguistics Around the World*. Rodopi, 2006.
- [2] I. Aduriz, M. Aranzabe, J. M. Arriola, A. Díaz de Ilarraza, K. Gojenola, M. Oronoz, and L. Uria. A cascaded syntactic analyser for basque. In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: 5th Int. Conf. CICLing2004, Korea*, volume 2945 of *Lecture Notes in Computer Science*, pages 124–134. Springer-Verlag GmbH, 2004.
- [3] A. V. Aho, R. Sethi, and J. D. Ullman. *Compilers: Principles, Techniques, and Tools*. Addison-Wesley, 1985.
- [4] I. Aldezabal, M. Aranzabe, A. Atutxa, K. Gojenola, M. Oronoz, and K. Sarasola. Application of finite-state transducers to the acquisition of verb subcategorization information. *Natural Language Engineering*. Cambridge University Press., 9(1):39–48, 2003.
- [5] J. Carroll, G. Minnen, and T. Briscoe. Corpus annotation for parser evaluation. In *EACL 99 workshop on Linguistically Interpreted Corpora (LINC)*, pages 35–41, Norway, 1999.
- [6] A. Díaz de Ilarraza, K. Gojenola, and M. Oronoz. Design and development of a system for the detection of agreement errors in basque. In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: 6th Int. Conf. CICLing2005, Mexico*, volume 3406 of *Lecture Notes in Computer Science*, pages 793–803. Springer-Verlag GmbH, 2005.
- [7] N. Ezeiza. *Corpusak ustiatzeko tresna linguistikoak. Euskararen etiketatzaile sintaktiko sendo eta malgua*. PhD thesis, University of the Basque Country, Donostia, 2003.
- [8] A. Gelbukh, G. Sidorov, S.-Y. Han, and E. Hernández-Rubio. Automatic enrichment of very large dictionary of word combinations on the basis of dependency formalism. *Lecture Notes in Artificial Intelligence*, (2972):430–437, 2004.
- [9] J. I. Hualde and J. O. de Urbina. *A grammar of Basque*. 2001.
- [10] A. Kilgarrif. Putting frequencies in the dictionary. *International Journal of Lexicography*, 10(2):135–155, 1997.
- [11] I. Sarasola. *Euskal Hiztegia*. Donostia, 1996.

# A Link Grammar for an Agglutinative Language

Ozlem Istek

Department of Computer Engineering  
Bilkent University  
Bilkent 06800, Ankara, Turkey  
oistek@cs.bilkent.edu.tr

Ilyas Cicekli

Department of Computer Engineering  
Bilkent University  
Bilkent 06800, Ankara, Turkey  
ilyas@cs.bilkent.edu.tr

## Abstract

This paper presents a syntactic grammar developed in the link grammar formalism for Turkish which is an agglutinative language. In the link grammar formalism, the words of a sentence are linked with each other depending on their syntactic roles. Turkish has complex derivational and inflectional morphology, and derivational and inflection morphemes play important syntactic roles in the sentences. In order to develop a link grammar for Turkish, the lexical parts in the morphological representations of Turkish words are removed, and the links are created depending on the part of speech tags and inflectional morphemes in words. Furthermore, a derived word is separated at the derivational boundaries in order to treat each derivation morpheme as a special distinct word, and allow it to be linked with the rest of the sentence. The derivational morphemes of a word are also linked with each other with special links to indicate that they are parts of the same word. The adapted unique link grammar formalism for Turkish provides flexibility for the linkage construction, and similar methods can be used for other languages with complex morphology.

**Keywords:** parsing, link grammar.

## 1. Introduction

There are different classes of theories for the natural language syntactic parsing problem and for creating the related grammars. One of these classes of formalisms is categorical grammar motivated by the principle of compositionality. According to this formalism; syntactic constituents combine as functions or in a function-argument relationship. In addition to categorical grammars, there are two other classes of grammars, which are phrase structure grammars, and dependency grammars. Phrase structure grammars construct constituents in a tree-like hierarchy. On the other hand, dependency grammars build simple relations between pairs of words. Since dependency grammars are not defined by a specific word order, they are well suited to languages with free word order, such as Czech and Turkish. Link

grammar [8] is similar to dependency grammar, but link grammar includes directionality in the relations between words, as well as lacking a head-dependent relationship.

There is some research on the computational analysis of Turkish syntax. One of these is a lexical functional grammar of Turkish [4]. There is also an ATN grammar for Turkish [2]. Another grammar for Turkish is based on HPSG formalism [9]. In addition, there are some works on the categorical grammars for Turkish [1,5]. Turkish syntax is also studied from the dependency parsing perspective. Oflazer presents a dependency parsing scheme using an extended finite state approach [6]. This parser is used for building a Turkish Treebank [7]. The Turkish Dependency Treebank is used for training and testing a statistical dependency parser for Turkish [3].

Syntactic analysis underlies most of the natural language applications and hence it is a very important step for any language. Although there are previous works on the computational analysis of Turkish, this paper presents the first link grammar developed for Turkish which is an agglutinative language. In this work, lexicalized structure of link grammar formalism is utilized for expressing the syntactic roles of intermediate derived forms of words in a language with very productive derivational and inflectional morphology. This is achieved by treating each of these intermediate derived forms as separate words. Using the adapted link grammar formalism, a fully functional link parser for Turkish is developed. The adapted link grammar formalism can also be used in the development of link grammars for other languages with very productive morphology.

Section 2 presents a general overview of the link grammar formalism, and Section 3 presents some distinctive features of Turkish syntax. In Section 4, the system architecture of the developed Turkish parser which uses our adapted link grammar formalism is given. Section 5 presents the special method for handling the syntactic roles of the words with derivations is given. Then, the paper continues with the performance evolution in Section 6, and Section 7 presents the concluding remarks.

## 2. Link Grammar

Link grammar is a formal grammatical system developed by Sleator and Temperley in 1993. In their work, they also developed top-down dynamic programming algorithms to process grammars based on this formalism and constructed a wide coverage link grammar for English. In this formalism, the syntax of a language is defined by a grammar that includes the words of the language and their linking requirements. A given sentence is accepted by the system if the linking requirements of all the words in the sentence are satisfied (connectivity), none of the links between the words cross each other (planarity) and there is at most one link between any pair of words (exclusion). A set of links between the words of a sentence that is accepted by the system is called a linkage. The grammar is defined in a dictionary file and each of the linking requirements of words is expressed in terms of connectors in the dictionary file. When a sequence of words is accepted, all the links are drawn above the words.

For example, the linkage requirements of three Turkish words can be defined as follows:

```
yedi (ate): O- & S-;
kadın (the woman): S+ ;
portakalı (the orange): O+;
```

Here, the verb “yedi”(ate) has two left linking requirements, one is “S”(subject) and the other is “O”(object). On the other hand, the noun “kadın” (the woman) needs to attach to a word on its right for its “S+” connector and the noun “portakalı”(the orange) has to attach a word on its right for its “O+” connector. Since the word, “yedi”(ate) and “kadın” (the woman) have the same “S” connector, i.e. same linking requirements, with opposite sign they can be connected by an “S” link. A similar situation occurs between the words “portakalı”(the orange) and “yedi”(ate) for the “O” connector. Therefore, if these words are connected in the following way, all of the linking requirements of these words are satisfied.

- Kadın portakalı yedi.
- (The woman ate the orange)

```

+-----S-----+
|               +---O---+
|               |       |
Kadın          portakalı yedi
The woman     the orange ate

```

In this sentence, “kadın”(the woman) links to word “yedi”(ate) with the S (subject) link and “portakalı”(the orange) links to word “yedi”(ate) with the O (object) link.

## 3. Turkish Syntax

In Turkish, the basic word order is SOV, but order of constituents may change according to the discourse context.

For this reason, all six combinations of subject, object, and verb are possible in Turkish.

Turkish is head-final, meaning that modifiers always precede the modified item. For example, an adjective (modifier) precedes the head noun (modified item) in a noun phrase. In the basic word order of the sentence, the subject and the object (modifiers) precede the verb (modified item). Although the head-final property can be violated at major constituent levels (SOV) of a sentence, it is preserved at sub-clause levels and smaller syntactic structures. For example, the following simple noun phrase demonstrates this property.

- (the girl with the red hat)
- kırmızı şapkalı kız
- red with hat girl

In this phrase, the adjective “kırmızı” modifies the noun “şapka”, and the phrase “kırmızı şapkalı” modifies the noun “kız”.

Like all other Altaic languages, Turkish is agglutinative. Non-functional words can take many derivational suffixes and each of these derivations can take its inflectional suffixes. In addition, in Turkish, inflectional suffixes have important grammatical roles. Inflectional suffixes of intermediate derived forms of a word also contribute to these syntactic roles of the word. Hence, there is a significant amount of interaction between syntax and morphotactics. For example, case, agreement, relativization of nouns and tense, modality, aspect, passivization, negation, causatives, and reflexives of verbs are marked by suffixes. For example, the following single Turkish word contains two derivational morphemes, and it corresponds to a complete English sentence.

- (you had not been able to make him do)
- yaptıramıyormuşsun
- yap+tır<sub>1</sub>+amı<sub>2</sub>+yor<sub>3</sub>+muş<sub>4</sub>+sun<sub>5</sub>
- yap+Verb ^DB+Verb+Caus<sub>1</sub>  
^DB+Verb+AbleNeg<sub>2</sub>+Neg  
+Prog<sub>1</sub> +Narr<sub>4</sub> +A2sg<sub>5</sub>

In this example, “^DB” indicates the derivational morpheme boundary, and the underlined morphemes are derivational morphemes.

## 4. System Architecture

The system architecture of Turkish parser is depicted in Figure 1 as a flowchart by labeling the parsing steps 1 through 5. The parser uses the Turkish morphological analyzer and the link grammar static libraries externally. A given sentence is transformed into certain intermediate forms at each step, and at the end all possible linkages of the sentence are generated by the parser. In the rest of this section, each step is explained separately.

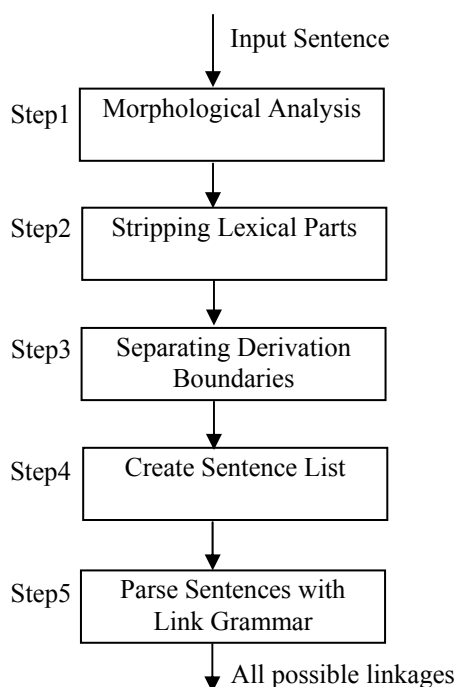


Figure 1. System Architecture of Turkish Parser

### Step 1 - Morphological Analysis:

After taking the input sentence in step 1, the system calls the external morphological analyzer for each word of the sentence to get its morphological structure. A fully functional Turkish morphological analyzer is used in the analysis of the words. The word itself is used in the rest of the system if the morphological analyzer cannot analyze a word.

For example, if the following input sentence is given into step 1, the output from step 1 will be as follows.

Input to Step 1:

- sen kitabı okudun
- (you read the book)

Output from Step 1:

- sen (you)  
i. sen+Pron+A2sg+Pnon+Nom
- kitap (book)  
i. kitap+Noun+A3sg+Pnon+Acc  
ii. kitap+Noun+A3sg+P3sg+Nom
- oku (read)  
i. oku+Verb+Pos+Past+A2sg

### Step 2 - Stripping Lexical Parts:

In step 2, the output of step 1 is preprocessed for the following parsing stages. In this step, lexical parts of the words are removed for all types of words except conjunctions. In fact, Turkish link grammar is designed for the classes of word types and their feature structures, i.e. POS, rather than the words themselves.

When the above output from step 1 is given into step 2, the lexical parts are removed from the morphological structures of the words, and the following output is created in step 2.

Output of Step 2:

- sen (you)  
i. Pron+A2sg+Pnon+Nom
- kitap (book)  
i. Noun+A3sg+Pnon+Acc  
ii. Noun+A3sg+P3sg+Nom
- oku (read)  
ii. Verb+Pos+Past+A2sg

The output of step 2, as shown above, is the list of unlexicalized morphological feature structures of words.

### Step 3 - Separating Derivation Boundaries:

If a word is derived from another word by the help of at least one derivational suffix, then its feature structure must contain at least one derivational boundary. Feature structures of words with derivational boundaries are handled in a special way in our system. In step 3, the words are separated at derivational boundaries and the part of speech tag of each derived form is marked in order to indicate its position in that word. The algorithm for step 3 is given in Figure 2. After step 3, a derived word is represented with a sequence of tokens. Each token starts with a part of speech tag with a position mark, and continues with inflectional feature structures. Below are some examples for step 3.

Input:

Noun+A3sg+Pnon+Acc

Output:

Noun+A3sg+Pnon+Acc

Input:

Noun+A3sg+P1pl+Loc^DB+Adj+Rel  
^DB+Noun+Zero+A3sg+Pnon+Gen

Output:

NounRoot+A3sg+P1pl+Loc  
AdjDB  
NounDBEnd+A3sg+Pnon+Gen

Since the first example does not contain any derivation, no action is taken and the part of speech tag "Noun" at the

```

if ( the feature structure of input word has
      no derivational boundary)
  • Output is equal to input
else {
  • Separate the word from the derivational boundaries
    to create a list of derived forms  $DF_1 \dots DF_n$ 
    where  $n \geq 2$ . In this list,  $DF_1$  is the root word,  $DF_n$ 
    is the last derivation, and others are intermediate
    derivations.
  • Replace POS tag portion of  $DF_1$  with the concatenation
    of POS of  $DF_1$  and the string "Root".
  • Replace POS tag portion of  $DF_n$  with the concatenation
    of POS of  $DF_n$  and the string "DBEnd".
  • Replace POS tag portion of each intermediate derived
    form with the concatenation of POS of that
    intermediate form and the string "DB".
  • Output is the list of derived forms
}

```

**Figure 2. Separating Words from Derivation Boundaries**

beginning of the output indicates that it is a noun without a derivation.

The second example above is divided into three derivational forms. In the example, the POS tag portion of each derived form is underlined, and they are replaced by new strings as described in the algorithm given in Figure 2 in order to indicate their positions in the word. After step 3, each token starts with a part of speech tag (or a part of speech tag followed by one of the strings "Root", "DB", or "DBEnd") and continues with inflectional suffixes. The first token starts with "NounRoot", and it indicates that the root word is a noun and that token is the root word of the derived word. "AdjDB" in the second token indicates that the word is converted into an adjective with a derivational morpheme, and that token is an intermediate derivation of the word. "NounDBEnd" in the last token indicates that the word is reconverted back into a noun again with a derivational morpheme, and that token is the last derivation of the word.

#### **Step 4 - Create Sentence List:**

Since a part-of-speech tagger is not used in our system, the number of feature structures found for the words is very large. For this reason, after step 4, a separate sentence is created for each of the morphological parse combinations of the words in step 3. For the example sentence given in step 2, "sen kitabı okudun" (you read the book), the output of step 4 is shown below.

*Input to Step 4:*

```

i. Pron+A2sg+Pnon+Nom
i. Noun+A3sg+Pnon+Acc
ii. Noun+A3sg+P3sg+Nom
i. Verb+Pos+Past+A2sg

```

*Output from Step 4:*

```

i. Pron+A2sg+Pnon+Nom
   Noun+A3sg+Pnon+Acc
   Verb+Pos+Past+A2sg
ii. Pron+A2sg+Pnon+Nom
   Noun+A3sg+P3sg+Nom
   Verb+Pos+Past+A2sg

```

This means that the sentence has two different representations at the morphological level. Each output is a sequence of tokens, and the first part of each token is a part of speech tag (or a part of speech tag with derivation position information). The rest of each token contains only the inflectional suffixes.

Since each word more than one representation at the morphological level, a sentence can have many representations at the morphological level. Each representation of the sentence will be fed into the parser at Step 5. In the future, the number of possible representations of the sentence at the morphological level will be reduced as a result of the integration of a Turkish morphological disambiguator into the system.

#### **Step 5 - Parsing Sentences:**

At the end, for each of these sentences, the link grammar is called, and each of the sentences is parsed in step 5 with respect to the designed Turkish link grammar. The Turkish link grammar contains a set of link requirements for each part of speech tag (or a part of speech tag followed by one of the strings "Root", "DB", or "DBEnd").

A linking requirement is written for a token, and the link requirements of a token depend on the part of speech tag of the token, and the inflection suffixes in that token. Each link requirement may contain left and right linking requirements.

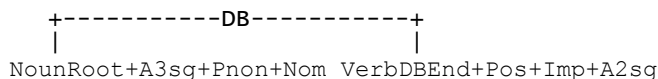
If all the linking requirements of the tokens in a sentence are satisfied, a linkage is created and returned as an output of the parser for the sentence. There is more than one possible linkage connection between tokens; all linkages are returned as the outputs of the parser.

## **5. Linking Requirements Related to Agglutination**

In order to preserve the syntactic roles that the intermediate derived forms of a word play, they are treated as separate words in the grammar. On the other hand, to show that they are the intermediate derivations of the same word, all of them are linked with the special "DB" (derivational bound-

ary) connector. In the following example, the feature structure of each morpheme is marked with the same subscript.

- uzman<sub>1</sub>+laş<sub>2</sub> (specialize)
- uzman+Noun+A3sg+Pnon+Nom<sub>1</sub>  
^DB+Verb+Pos+Imp+A2sg<sub>2</sub>
- NounRoot+A3sg+Pnon+Nom<sub>1</sub>  
VerbDBEnd+Pos+Imp+A2sg<sub>2</sub>



Here, the noun root “uzman”(specialist) is an intermediate derived form and connected to the last derivation morpheme “-laş” (to become) by the “DB” link, to denote that they are parts of the same word. Since the root word (NounRoot) is an intermediate derivation form of this derived word, it can only have left linking requirements by contributing the left linking requirements of the derived word. The last derived form (VerbDBEnd) can have both left and right linking requirements. In general, a derived word consists of a sequence of intermediate derived forms where the first one is the root word, and the last derivation form. However, these intermediate derived forms, IDF, do not contribute to the right linking requirement of the last derived word. In addition, the “DB” linking requirements of the intermediate derived forms are different according to their order. The last derived form can contribute to both left and right linking requirements of the derived word.

In Figure 3, linking requirements of a word, with  $n$  intermediate derived forms (IDF<sub>1</sub>...IDF<sub>n</sub>) are illustrated. In Figure 3, “LL” represents the links to the words on the left hand side of the word, and “RL” represents the links to the words on the right hand side of the word. IDFs of the word are connected by “DB” links. As it can be seen all  $n$  IDFs can connect to the words to the left of, but only the last IDF, IDF<sub>n</sub> can connect to the words on the right hand side of the word. In addition, IDF<sub>1</sub>, which is the root stem, needs only to connect to its right with the “DB” connector,

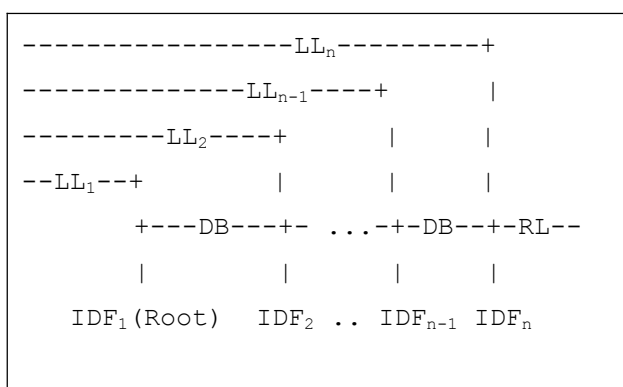


Figure 3. Linking Requirements of Intermediate Forms of a Word

whereas the last IDF (IDF<sub>n</sub>) needs to connect to its left with the same connector. On the other hand, all the IDFs between these two should connect to both to their lefts and to rights with “DB” links to denote that they belong to the same word. Hence, the same IDF, has different linking requirements depending on its place in a word. To handle this situation, different items are placed into the grammar representing each of these three places of the same word.

Tokens with a part of speech tag (without any derivational position marker) can have left and right linking requirements. We call these linking requirements as “non-derivational linking requirements” (NDLR). In addition, NDLLR is used as an abbreviation for “non derivational left linking requirement” and NDRLR is for “non derivational right linking requirement”. Thus, all tokens with a part of speech tag without a derivational position marker will only have NDLR.

Tokens containing a part of speech tag with a derivational position marker may not use all NDLR, and they can have “DB” linking requirements. Their linking requirements depend on their position in the derived word. Figure 4 gives linking requirements of tokens with a part of speech tag with derivational position marker, and they are referred as IDFs (intermediate derivational forms) in Figure 4. In Figure 4, derivational linking requirements are in italics and non-derivational linking requirements are in bold.

As it can be seen in Figure 4, NDLRs of an IDF placed at the beginning and in the middle are the same. In addition, NDLR of the IDF for these two positions is a subset of the whole NDRL of the same IDF placed at the end.

## 6. Performance Evaluation

The performance of our system is tested for coverage with a document consisting of sentences collected from domestic, foreign, sports, astrology, and finance news randomly together with sentences from a storybook for children. Before beginning testing, punctuation symbols are removed from the sentences. In addition, incorrect morphological analyses are removed from the results. Table 1 shows the

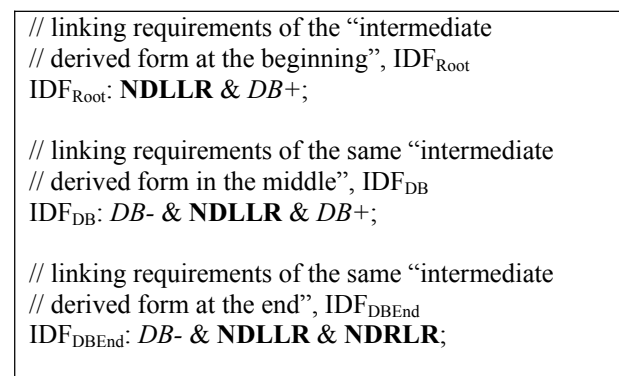


Figure 4. Linking Requirements of an IDF According to Its Place

**Table 1. Statistical Results of the Test Run**

Number of Sentences	250
Average number of words in each sentence	5.19
Percentage of the sentences for which resulting parses contains the correct parse	84.31
Average number of parses	7.49
Average ordering of the correct parse	1.78

results of the test run.

In the experiment, 250 sentences are used. Average number of words in the sentences is 5.19. Average number of parses per sentences is 7.49. However, for two of the sentences, the number of the parses are very high, i.e. 22 and 50. Both of these two sentences contain many consecutive nouns. Since nouns are not subcategorized for time, place, and title, this resulted in many incorrect indefinite and adjectival nominal groups to be generated and this is the problem in these two sentences. Moreover, one of these sentences consists of words with very complex derivational morphotactics, i.e. many derivational intermediate forms, which results in the number of possible links between these intermediate derived forms to increase. In addition, for 84.31% of the sentences, the result set of the parser contains the correct parse. Lastly, average ordering of the correct parse in the result set was 1.78. However, for 62.39% of the sentences, the first parse is the correct parse and for 80.94% of the sentences, one of the first three parses is correct.

## 7. Conclusions and Future Work

In this work, we have developed a grammar of Turkish language in the link grammar formalism. Noun phrases; postpositional phrases; dependent clauses constructed by gerunds, participles, and infinitives; simple, complex, conditional, and ordered/compound sentences; nominal and verbal sentences; regular sentences; positive, negative, imperative, and interrogative sentences; pronoun drop; freely changing order of adverbial phrases, noun phrases acting as objects, and subject are in the scope. In addition, quotations, numbers, abbreviations, hyphenated expressions, and unknown words are handled. However, inverted sentences, idiomatic and multi-word expressions, punctuation symbols, and embedded and some types of substantival sentences are currently out of the scope.

In the grammar, we used a fully described morphological analyzer, which is very important for agglutinative languages like Turkish. The Turkish link grammar that we developed is not a lexical grammar. Although we used the lexemes of some function words, we used the morphological feature structures for the rest of the word classes. In addition, we preserved the syntactic roles of the intermediate derived forms of words in our system by separating the

derived words from their derivational boundaries and treating each intermediate form as a distinct word.

As mentioned above, because of the productive morphology of Turkish, our linking requirements are defined for morphological categories. However, instead of using only the morphological feature structures of words, stems of words can also be added to the current system. Thus, the results of our current Turkish link grammar can be more precise. In addition, statistical information about the relations between the words can be embedded into the system. Moreover, our current system does not use a POS tagger, and its addition will improve the performance of the system in terms of both time and precision. During the tests, we recognized that there are many multi-word expressions in Turkish and a multi-word expression processor is necessary.

Although the adopted unique link grammar approach is used in the development of a Turkish link grammar, it can be used in the development of the link grammars for other languages with complex morphology. The adopted approach can provide flexibility in the development of link grammars for such languages.

## 8. References

- [1] Bozşahin, C. and Göçmen, E. 1995. *A Categorical Framework for Composition in Multiple Linguistic Domains*. In Proceedings of the Fourth International Conference on Cognitive Science of NLP, Dublin, Ireland.
- [2] Demir, Coşkun. 1993. *An ATN Grammar for Turkish*. M.S. Thesis, Bilkent University.
- [3] Eryiğit, G., and Oflazer, K. 2006. *Statistical Dependency Parsing of Turkish*. In Proceedings of EACL 2006 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy.
- [4] Güngördü, Zelal. 1993. *A Lexical Functional Grammar for Turkish*. M.S. Thesis, Bilkent University.
- [5] Hoffman, Beryl. 1995. *The Computational Analysis of the Syntax and Interpretation of 'Free' Word Order in Turkish*. PhD thesis, University of Pennsylvania.
- [6] Oflazer, K. 1999. *Dependency Parsing with an Extended Finite State Approach*. In Proceedings of 37th Annual Meeting of the ACL, Maryland, USA.
- [7] Oflazer, K.; Say, B.; Hakkani-Tür, D.K.; Tür, G. *Building a Turkish Treebank. Invited chapter in Building and Exploiting Syntactically-annotated Corpora*, Anne Abeille Editor, Kluwer Academic Publishers. The treebank is available online at: <http://www.ii.metu.edu.tr/~corpus/treebank.html>
- [8] Sleator, D. D. K. and Temperley, D. 1993. *Parsing English with a Link Grammar*. Third International Workshop on Parsing Technologies.
- [9] Şehitoğlu, O. Tolga. 1996. *A Sign-Based Phrase Structure Grammar for Turkish*. M.S. Thesis, Middle East Technical University.

# Semantic Similarity of Short Texts

Aminul Islam and Diana Inkpen  
University of Ottawa  
School of Information Tech. and Eng.  
Ottawa, Ontario, Canada, K1N 6N5  
{mdislam, diana}@site.uottawa.ca

## Abstract

This paper presents a method for measuring the semantic similarity of texts using a corpus based measure of semantic word similarity and a normalized and modified versions of the Longest Common Subsequence (LCS) string matching algorithm. Existing methods for computing text similarity have focused mainly on either large documents or individual words. In this paper, we focus on computing the similarity between two sentence or between two short paragraphs. The proposed method can be exploited in a variety of applications involving textual knowledge representation and knowledge discovery. Evaluation results on two different data sets show that our method outperforms several competing methods.

## Keywords

Semantic similarity of words, similarity of short texts, corpus-based measures.

## 1. Introduction

Similarity is a complex concept which has been widely discussed in the linguistic, philosophical, and information theory communities. Frawley [9] discusses all semantic typing in terms of two mechanisms: the detection of similarities and differences. For our task, given two input text segments, we want to automatically determine a score that indicates their similarity at *semantic* level, thus going beyond the simple lexical matching methods traditionally used for this task.

An effective method to compute the similarity between short texts or sentences has many applications in natural language processing and related areas such as information retrieval and text filtering. For example, in web page retrieval, text similarity has proven to be one of the best techniques for improving retrieval effectiveness [33] and in image retrieval from the Web, the use of short text surrounding the images can achieve a higher retrieval precision than the use of the whole document in which the image is embedded [3]. The use of text similarity is beneficial for relevance feedback and text categorization [13], [24], text summarization [7], [22], word sense disambiguation [19], methods for automatic evaluation of machine translation [25], [31], evaluation of text coherence [17], and schema matching in databases [26].

One of the major drawbacks of most of the existing methods is the domain dependency: once the similarity method is designed for a specific application domain, it cannot be adapted easily to other domains. To address this drawback, we aim to develop a method that is fully automatic and independent of the domain in applications

requiring small text or sentence similarity measure. The computing of text similarity can be viewed as a generic component for the research community dealing with text-related knowledge representation and discovery.

This paper is organized as follow: Section 2 presents a brief overview of the related work. Our proposed method is described in Section 3. Evaluation and experimental results are discussed in Section 4.

## 2. Related Work

There is extensive literature on measuring the similarity between long texts or documents [15], [27], [28], but there is less work related to the measurement of similarity between sentences or short texts [8]. Related work can roughly be classified into four major categories: word co-occurrence/vector-based document model methods, corpus-based methods, hybrid methods, and descriptive feature-based methods.

The vector-based document model methods are commonly used in Information Retrieval (IR) systems [28], where the document most relevant to an input query is determined by representing a document as a word vector, and then queries are matched to similar documents in the document database via a similarity metric [37].

The Latent Semantic Analysis (LSA) [15], [16] and the Hyperspace Analogues to Language (HAL) model [2] are two well known methods in corpus-based similarity. LSA analyzes a large corpus of natural language text and generates a representation that captures the similarity of words and text passages. The dimension of the word by context matrix is limited to several hundreds because of the computational limit of Singular Value Decomposition (SVD). As a result the vector is fixed and the representation of a short text is very sparse. The HAL method uses lexical co-occurrence to produce a high-dimensional semantic space. The authors' experimental results showed that HAL was not as promising as LSA in the computation of similarity for short texts.

Hybrid methods use both corpus-based measures [38] and knowledge-based measures [18] of word semantic similarity to determine the text similarity. Mihalcea et al. [30] suggest a combined method for measuring the semantic similarity of texts by exploiting the information that can be drawn from the similarity of the component words. Specifically, they use two corpus-based measures, PMI-IR (Pointwise Mutual Information and Information Retrieval) [38] and LSA (Latent Semantic Analysis) [16] and six knowledge-based measures [12], [18], [19], [23],



[34], [39] of word semantic similarity, and combine the results to show how these measures can be used to derive a text-to-text similarity metric. They evaluate their method on a paraphrase recognition task. The main drawback of this method is that it computes the similarity of words from eight different methods, which is not computationally efficient.

Li et al. [20] propose another hybrid method that derives text similarity from semantic and syntactic information contained in the compared texts. Their proposed method dynamically forms a joint word set only using all the distinct words in the pairs of sentences. For each sentence, a raw semantic vector is derived with the assistance of the WordNet lexical database [32]. A word order vector is formed for each sentence, again using information from the lexical database. Since each word in a sentence contributes differently to the meaning of the whole sentence, the significance of a word is weighted by using information content derived from a corpus. By combining the raw semantic vector with information content from the corpus, a semantic vector is obtained for each of the two sentences. Semantic similarity is computed based on the two semantic vectors. An order similarity is calculated using the two order vectors. Finally, the sentence similarity is derived by combining semantic similarity and order similarity.

Feature-based methods try to represent a sentence using a set of predefined features. Similarity between two texts is obtained through a trained classifier. But finding effective features and obtaining values for these features from sentences make this category of methods more impractical.

### 3. Proposed Method

The proposed method derives text similarity of two texts by combining semantic similarity and string similarity, with normalization. We call our proposed method the Semantic Text Similarity (STS) method. We investigate the importance of including string similarity by a simple example. Let us consider a pair of texts,  $T_1$  and  $T_2$  that contain a *proper noun (proper name)* ‘Maradona’ in  $T_1$ . In  $T_2$  the name ‘Maradona’ is misspelled to ‘Maradena’.

$T_1$  : Many consider Maradona as the best player in soccer history.

$T_2$  : Maradena is one of the best soccer players. Dictionary-based similarity measure can not provide any similarity value between these two proper names. And the chance to obtain a similarity value using corpus-based similarity measures is very low. We obtain a good similarity score if we use string similarity measures. The following sections present a detailed description of each of the above mentioned functions.

#### 3.1 String Similarity between Words

We use the *longest common subsequence* (LCS) [1], [14] measure with some normalization and small modifications for our string similarity measure. We use

three different modified versions of LCS and then take a weighted sum of these<sup>1</sup>. Melamed [29] normalized LCS by dividing the length of the longest common subsequence by the length of the longer string and called it *longest common subsequence ratio* (LCSR). But LCSR does not take into account of the length of the shorter string which sometimes has a significant impact on the similarity score.

We normalize the *longest common subsequence* (LCS) so that it takes into account of the length of both the shorter and the longer string and call it *normalized longest common subsequence* (NLCS) which is,

$$v_1 = NLCS(r_i, s_j) = \frac{\{length(LCS(r_i, s_j))\}^2}{length(r_i) \times length(s_j)} \quad (1)$$

While in classical LCS, the common subsequence needs not be consecutive, in text matching, consecutive common subsequence is important for a high degree of matching. We use *maximal consecutive longest common subsequence* starting at character 1,  $MCLCS_1$  (Fig. 1) and *maximal consecutive longest common subsequence* starting at any character  $n$ ,  $MCLCS_n$  (Fig. 2). In Fig. 1, we present an algorithm that takes two strings as input and returns the shorter string or maximal consecutive portions of the shorter string that consecutively match with the longer string, where matching must be from first character (character 1) for both strings. In Fig. 2, we present another algorithm where matching may start from any character (character  $n$ ). We also normalize  $MCLCS_1$  and  $MCLCS_n$ .

We take the weighted sum of the values  $v_1$ ,  $v_2$  (normalized  $MCLCS_1$ ), and  $v_3$  (normalized  $MCLCS_n$ ) to determine string similarity score, where  $w_1$ ,  $w_2$ ,  $w_3$  are weights and  $w_1+w_2+w_3=1$ . Therefore, the similarity of the two strings is:  $\alpha = w_1v_1 + w_2v_2 + w_3v_3$  (2)

We set equal weights for our experiments.<sup>2</sup>

---

#### Algorithm $MCLCS_1$

**Input:**  $r_i, s_j$  //  $r_i$  and  $s_j$  are two input strings where  
//  $|r_i| = \tau, |s_j| = \eta$  and  $\tau \leq \eta$  as mentioned earlier.  
1.  $\tau \leftarrow |r_i|, \eta \leftarrow |s_j|$   
2. **while**  $|r_i| > 0$   
3.     **if**  $r_i \subset s_j$  // i.e.,  $s_j \cap r_i = r_i$   
4.         **return**  $r_i$   
5.     **else**  $r_i \leftarrow r_i \setminus c_\tau$  // i.e., remove the right-  
// most character from  $r_i$   
6.     **end if**  
7. **end while**  
**Output:**  $r_i$  //  $r_i$  is the Maximal Consecutive  
// LCS starting at character 1

---

Fig. 1. Maximal consecutive LCS starting at character 1.

<sup>1</sup> We use modified versions because in our experiments we obtained better results (precision and recall) for text matching on a sample of data than when using the original LCS, or other string similarity measures.

<sup>2</sup> We use equal weights in several places in this paper in order to keep the system unsupervised. If development data would be available, we could adjust the weights.

---

**Algorithm MCLCS<sub>n</sub>**

**Input:**  $r_i, s_j$  //  $r_i$  and  $s_j$  are two input strings  
// where  $|r_i| = \tau$ ,  $|s_j| = \eta$  and  $\tau \leq \eta$ .

1. **while**  $|r_i| > 0$
  2. determine all  $n$ -grams from  $r_i$  where  $n = 1 \dots |r_i|$   
and  $\bar{r}_i$  is the set of  $n$ -grams
  3. **if**  $x \in s_j$  where  $\{x \mid x \in \bar{r}_i, x = \mathbf{Max}(\bar{r}_i)\}$   
//  $i$  is the number of  $n$ -grams and  $\mathbf{Max}(\bar{r}_i)$   
// returns the maximum length  $n$ -gram from  $\bar{r}_i$
  4. **return**  $x$
  5. **else**  $\bar{r}_i \leftarrow \bar{r}_i \setminus x$  // remove  $x$  from set  $\bar{r}_i$
  6. **end if**
  7. **end while**
- Output:**  $x$  //  $x$  is the Maximal Consecutive  
// LCS starting at any character  $n$
- 

**Fig. 2. Maximal consecutive LCS starting at any character  $n$**

---

**Algorithm semanticMatching**

**Input:**  $r_i, s_j$  //  $r_i$  and  $s_j$  are two input words  
// where  $|r_i| = \tau$ ,  $|s_j| = \eta$  and  $\tau \leq \eta$ .

1.  $v \leftarrow \mathbf{SOCPMI}(r_i, s_j)$  // This method determines  
// semantic similarity between two words. Any  
// other similarity method can also be used instead.
  2. **if**  $v > \lambda$  //  $\lambda$  is the maximum possible similarity value
  3.  $v \leftarrow 1$
  4. **else**  $v \leftarrow v / \lambda$
  5. **end if**
- Output:**  $v$  //  $v$  is the semantic similarity value  
// between 0 and 1, inclusively
- 

**Fig. 3. Semantic similarity matching.**

### 3.2 Semantic Similarity between Words

There is a relatively large number of word-to-word similarity metrics in the literature, ranging from distance-oriented measures computed on semantic networks or knowledge base (or dictionary/thesaurus-based measures), to metrics based on models of information theory (or corpus-based measures) learned from large text collections. A detailed review on word similarity can be found in [21], [35]. We focus our attention on corpus-based measures because of their large type coverage.

PMI-IR [38] is a simple method for computing corpus-based similarity of words which uses Pointwise Mutual Information. PMI-IR used AltaVista Advanced Search query syntax to calculate the probabilities. LSA, another corpus-based measure, analyzes a large corpus of natural text and generate a representation that captures the similarity of words (discussed in the Related Work section).

We use the Second Order Co-occurrence PMI (SOC-PMI) word similarity method [10] that uses Pointwise Mutual Information to sort lists of important neighbor words of the two target words from a large corpus. The method considers the words which are common in both

lists and aggregate their PMI values (from the opposite list) to calculate the relative semantic similarity. We define the *pointwise mutual information* function for only those words having  $f^b(t_i, w) > 0$ ,

$$f^{pmi}(t_i, w) = \log_2 \frac{f^b(t_i, w) \times m}{f^t(t_i) f^t(w)},$$

where  $f^t(t_i)$  tells us how many times the word  $t_i$  appeared in the entire corpus,  $f^b(t_i, w)$  tells us how many times word  $t_i$  appeared with word  $w$  in a context window words and  $m$  is total number of tokens in the corpus. Now, for word  $w_1$ , we define a set of words,  $X$ , sorted in descending order by their PMI values with  $w_1$  and taken the top-most  $\beta_1$  words having  $f^{pmi}(t_i, w_1) > 0$ .

$$X = \{X_i\}, \text{ where } i = 1, 2, \dots, \beta_1 \text{ and}$$

$$f^{pmi}(t_1, w_1) \geq f^{pmi}(t_2, w_1) \geq \dots \geq f^{pmi}(t_{\beta_1-1}, w_1) \geq f^{pmi}(t_{\beta_1}, w_1)$$

Similarly, for word  $w_2$ , we define a set of words,  $Y$ , sorted in descending order by their PMI values with  $w_2$  and taken the top-most  $\beta_2$  words having  $f^{pmi}(t_i, w_2) > 0$ . The value of  $\beta$  (either  $\beta_1$  or  $\beta_2$ ) is related to how many times a word  $w$  appears in the corpus, i.e., the frequency of  $w$  as well as the number of types in the corpus. Then we define the  $\beta$ -PMI summation function. For word  $w_1$ , the  $\beta$ -PMI summation function is:

$$f^\beta(w_1) = \sum_{i=1}^{\beta_1} (f^{pmi}(X_i, w_2))^\gamma,$$

where,  $f^{pmi}(X_i, w_2) > 0$  and  $f^{pmi}(X_i, w_1) > 0$

which sums all the positive PMI values of words in the set  $Y$  also common to the words in the set  $X$ . In other words, this function actually aggregates the positive PMI values of all the semantically close words of  $w_2$  which are also common in  $w_1$ 's list. The higher the value of  $\gamma$  is, the greater emphasis on words having very high PMI values with  $w_1$  is given. Similarly, we calculate the  $\beta$ -PMI summation function for word  $w_2$ . Finally, we define the *semantic PMI similarity* function between the two words,  $w_1$  and  $w_2$ ,

$$Sim(w_1, w_2) = \frac{f^\beta(w_1)}{\beta_1} + \frac{f^\beta(w_2)}{\beta_2}$$

We normalize the semantic word similarity (Fig. 3), so that it provides a similarity score between 0 and 1 inclusively. The word similarity method is a separate module in our Text Similarity Method. Therefore any other word similarity method could be substituted instead of SOC-PMI. In that case, we need to set  $\lambda$  to the maximum similarity value specific to that method.

### 3.3 Overall Sentence Similarity

Our task is to derive a score between 0 and 1 inclusively that will indicate the similarity between two texts  $P$  and  $R$  at semantic level. The main idea is to find, for each word in the first sentence, the most similar matching in the second sentence. The method consists in the following six steps:

**Step 1:** We use all special characters, punctuations, and capital letters, if any, as initial word boundary and

eliminate all these special characters, punctuations and stop words. We lemmatize each of the segmented words to generate tokens. After cleaning we assume that the text  $P = \{p_1, p_2 \dots, p_m\}$  has  $m$  tokens and the text  $R = \{r_1, r_2 \dots, r_n\}$  has  $n$  tokens and  $n \geq m$ . Otherwise, we switch  $P$  and  $R$ .

**Step 2:** We count the number of  $p_i$ 's (say,  $\delta$ ) for which  $p_i = r_j$ , for all  $p \in P$  and for all  $r \in R$ . I.e., there are  $\delta$  tokens in  $P$  that exactly match with  $R$ , where  $\delta \leq m$ . We remove all  $\delta$  tokens from both of  $P$  and  $R$ . So,  $P = \{p_1, p_2 \dots, p_{m-\delta}\}$  and  $R = \{r_1, r_2 \dots, r_{n-\delta}\}$ . If all the terms match,  $m-\delta = 0$ , we go to step 6.

**Step 3:** We construct a  $(m-\delta) \times (n-\delta)$  string similarity matrix (say,  $M_1 = (\alpha_{ij})_{(m-\delta) \times (n-\delta)}$ ) using the following process: we assume any token  $p_i \in P$  has  $\tau$  characters, i.e.,  $p_i = \{c_1 c_2 \dots c_\tau\}$  and any token  $r_j \in R$  has  $\eta$  characters, i.e.,  $r_j = \{c_1 c_2 \dots c_\eta\}$  where  $\tau \leq \eta$ . In other words,  $\eta$  is the length of the longer token and  $\tau$  is the length of the shorter token. We calculate the followings:

$$\begin{aligned} v_1 &\leftarrow NLCS(p_i, r_j), \\ v_2 &\leftarrow NMCLCS_1(p_i, r_j) \\ v_3 &\leftarrow NMCLCS_n(p_i, r_j), \\ \alpha_{ij} &\leftarrow w_1 v_1 + w_2 v_2 + w_3 v_3 \end{aligned}$$

i.e.,  $\alpha_{ij}$  is a weighted sum of  $v_1$ ,  $v_2$ , and  $v_3$  where  $w_1$ ,  $w_2$ ,  $w_3$  are weights and  $w_1 + w_2 + w_3 = 1$ . We set equal weights for our experiments.

We put  $\alpha_{ij}$  in row  $i$  and column  $j$  position of the matrix for all  $i = 1 \dots m-\delta$  and  $j = 1 \dots n-\delta$ .

$$M_1 = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{1j} & \alpha_{1(n-\delta)} \\ \alpha_{21} & \alpha_{22} & \alpha_{2j} & \alpha_{2(n-\delta)} \\ \alpha_{i1} & \alpha_{i2} & \alpha_{ij} & \alpha_{i(n-\delta)} \\ \alpha_{(m-\delta)1} & \alpha_{(m-\delta)2} & \alpha_{(m-\delta)j} & \alpha_{(m-\delta)(n-\delta)} \end{bmatrix}$$

**Step 4:** We construct a  $(m-\delta) \times (n-\delta)$  semantic similarity matrix (say,  $M_2 = (\beta_{ij})_{(m-\delta) \times (n-\delta)}$ ) using the following process: We put  $\beta_{ij}$  ( $\beta_{ij} \leftarrow \text{semanticMatching}(p_i, r_j)$ ) (Fig. 3) in row  $i$  and column  $j$  position of the matrix for all  $i = 1 \dots m-\delta$  and  $j = 1 \dots n-\delta$ .

$$M_2 = \begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{1j} & \beta_{1(n-\delta)} \\ \beta_{21} & \beta_{22} & \beta_{2j} & \beta_{2(n-\delta)} \\ \beta_{i1} & \beta_{i2} & \beta_{ij} & \beta_{i(n-\delta)} \\ \beta_{(m-\delta)1} & \beta_{(m-\delta)2} & \beta_{(m-\delta)j} & \beta_{(m-\delta)(n-\delta)} \end{bmatrix}$$

**Step 5:** We construct another  $(m-\delta) \times (n-\delta)$  joint matrix (say,  $M = (\gamma_{ij})_{(m-\delta) \times (n-\delta)}$ ) using

$$M \leftarrow \psi M_1 + \phi M_2 \quad (3)$$

(i.e.,  $\gamma_{ij} = \psi \alpha_{ij} + \phi \beta_{ij}$ ) where  $\psi$  is the string matching matrix weight factor.  $\phi$  is the semantic similarity matrix weight factor, and  $\psi + \phi = 1$ . We set equal weights for our experiments.

$$M = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{1j} & \gamma_{1(n-\delta)} \\ \gamma_{21} & \gamma_{22} & \gamma_{2j} & \gamma_{2(n-\delta)} \\ \gamma_{i1} & \gamma_{i2} & \gamma_{ij} & \gamma_{i(n-\delta)} \\ \gamma_{(m-\delta)1} & \gamma_{(m-\delta)2} & \gamma_{(m-\delta)j} & \gamma_{(m-\delta)(n-\delta)} \end{bmatrix}$$

After constructing the joint matrix,  $M$ , we find out the maximum-valued matrix-element,  $\gamma_{ij}$ . We add this matrix element to a list (say,  $\rho$  and  $\rho \leftarrow \rho \cup \gamma_{ij}$ ) if  $\gamma_{ij} > 0$ . We remove all the matrix elements of  $i$ 'th row and  $j$ 'th column from  $M$ . We repeat the finding of the maximum-valued matrix-element,  $\gamma_{ij}$  adding it to  $\rho$  and removing all the matrix elements of the corresponding row and column until either  $\gamma_{ij} = 0$ , or  $m-\delta-|\rho| = 0$ , or both.

**Step 6:** We sum up all the elements in a value  $\rho$  and add  $\delta$  to it to get a total score. We multiply this total score by the reciprocal harmonic mean of  $m$  and  $n$  to obtain a balanced similarity score between 0 and 1, inclusively.

$$S(P, R) = \frac{(\delta + \sum_{i=1}^{|\rho|} \rho_i) \times (m+n)}{2mn} \quad (4)$$

## 4. Evaluation and Experimental Results

In order to evaluate our text similarity measure, we use two different data sets: 30 sentence pairs [20] and the Microsoft paraphrase corpus [6].

### 4.1 Experiment with Human Similarities of Sentence Pairs

We use the same data set as Li et al. [20] (available at <http://www.docm.mmu.ac.uk/STAFF/D.McLean/SentenceResults.htm>). Li et al. [20] collected human ratings for the similarity of pairs of sentences following existing designs for word similarity measures. The participants consisted of 32 volunteers, all native speakers of English educated to graduate level or above. Li et al. [20] began with the set of 65 noun pairs from Rubenstein and Goodenough [36] and replaced them with their definitions from the Collins Cobuild dictionary [4]. Cobuild dictionary definitions are written in full sentences, using vocabulary and grammatical structures that occur naturally with the word being explained. The participants were asked to complete a questionnaire, rating the similarity of meaning of the sentence pairs on the scale from 0.0 (minimum similarity) to 4.0 (maximum similarity), as in Rubenstein and Goodenough (R&G) [36]. Each sentence pair was presented on a separate sheet. The order of presentation of the sentence pairs was randomized in each questionnaire. The order of the two sentences making up each pair was also randomized. This was to prevent any bias being introduced by order of presentation. Each of the 65 sentence pairs was assigned a semantic similarity score calculated as the mean of the judgments made by the participants. The distribution of the semantic similarity scores was heavily skewed toward the low similarity end of the scale. A subset of 30 sentence pairs was selected to

obtain a more even distribution across the similarity range. This subset contains all of the sentence pairs rated 1.0 to 4.0 and 11 (from a total of 46) sentences rated 0.0 to 0.9 selected at equally spaced intervals from the list. The detailed procedure of this data set preparation is in [20]. Table 1 shows average human similarity scores along with Li et al.'s Similarity Method scores [20] and our proposed Semantic Text Similarity scores. Human similarity scores are provided as the mean score for each pair and have been scaled into the range [0..1].

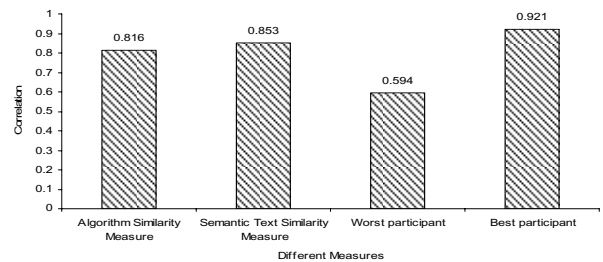
**Table 1. Results on Li et al. sentence data set**

R&G No.	R&G word pair in the sentence	Human Sim. (Mean)	Li et al. Sim. Meth.	STS Meth.	R&G No.	R&G word pair in the sentence	Human Sim. (Mean)	Li et al. Sim. Meth.	STS Meth.
1	Cord Smile	0.01	0.33	0.06	51	Glass Tumbler	0.14	0.65	0.28
5	Autograph Shore	0.01	0.29	0.11	52	Grin Smile	0.49	0.49	0.32
9	Asylum Fruit	0.01	0.21	0.07	53	Serf Slave	0.48	0.39	0.44
13	Boy Rooster	0.11	0.53	0.16	54	Journey Voyage	0.36	0.52	0.41
17	Coast Forest	0.13	0.36	0.26	55	Autograph Signature	0.41	0.55	0.19
21	Boy Sage	0.04	0.51	0.16	56	Coast Shore	0.59	0.76	0.47
25	Forest Graveyard	0.07	0.55	0.33	57	Forest Woodland	0.63	0.7	0.26
29	Bird Woodland	0.01	0.33	0.12	58	Implement Tool	0.59	0.75	0.51
33	Hill Woodland	0.15	0.59	0.29	59	Cock Rooster	0.86	1	0.94
37	Magician Oracle	0.13	0.44	0.20	60	Boy Lad	0.58	0.66	0.60
41	Oracle Sage	0.28	0.43	0.09	61	Cushion Pillow	0.52	0.66	0.29
47	Furnace Stove	0.35	0.72	0.30	62	Cemetery Graveyard	0.77	0.73	0.51
48	Magician Wizard	0.36	0.65	0.34	63	Automobil Car	0.56	0.64	0.52
49	Hill Mound	0.29	0.74	0.15	64	Midday Noon	0.96	1	0.93
50	Cord String	0.47	0.68	0.49	65	Gem Jewel	0.65	0.83	0.65

Fig. 4 shows that our proposed Semantic Text Similarity Measure achieves a high Pearson correlation coefficient of 0.853 with the average human similarity ratings, whereas Li et al.'s Similarity Measure [20] achieves 0.816. The improvement we obtained is statistically significant at the 0.05 level<sup>3</sup>. In the human judging experiment of Li et al. [20] the best human participant obtained a correlation of 0.921 with the mean of the participants and the worst participant obtained 0.594.

## 4.2 Experiment with Microsoft Paraphrase Corpus

We use the semantic text similarity method to automatically identify if two text segments are paraphrases of each other. We use the Microsoft paraphrase corpus [6], consisting of 4,076 training and 1,725 test pairs, and determine the number of correctly identified paraphrase pairs in the corpus using the semantic text similarity measure. The paraphras pairs in



**Fig. 4. Similarity correlations.**

this corpus were labeled by two human annotators who determined if the two sentences in a pair were semantically equivalent paraphrases or not. The agreement between the human judges who labeled the candidate paraphrase pairs in this data set was measured at approximately 83%, which can be considered as an upper bound for an automatic paraphrase recognition task performed on this data set.

We acknowledge, as in [5], that the semantic similarity measure for short texts is a necessary step in the paraphrase recognition task, but not always sufficient. There might be cases when the same meaning is expressed in one sentence and the exact opposite meaning in the second sentence (for example by adding the word *not*). For these situations deeper reasoning methods are needed.

We evaluate the results in terms of accuracy, the number of pairs predicted correctly divided by the total number of pairs. We also measure precision ( $P = TP / (TP + FP)$ ), recall ( $R = TP / (TP + FN)$ ) and F-measure ( $F = 2PR / (P + R)$ ). Here,  $TP$ ,  $FP$  and  $FN$  stand for True Positive, False Positive and False Negative respectively.

We use eleven different similarity thresholds ranging from 0 to 1 with interval 0.1. In Table 2, when we use similarity threshold score of 1 (i.e., matching word by word exactly, therefore no semantic similarity matching is needed), we obtain recall value of 0.0044 for the test data set. We can consider this score as one of the baselines. Mihalcea et al. [30] mentioned two other baselines: Vector-based and Random. See Table 3 for the results of these baselines and the results of several methods from [30] and [5] (on the test set).

For this paraphrase identification task, we can consider our proposed STS method as a supervised method. Using training data set, we obtain the best accuracy of 72.42% when we use 0.6 as the similarity threshold score. Therefore we can recommend this threshold for use on the test set, achieving an accuracy of 72.64% (our method predicts 1369 pairs as correct, out of which 1022 pairs are correct among the 1725 manually annotated pairs). Our results on the test set are shown in Table 3.

For each candidate paraphrase pair in the test set, we first calculate the semantic text similarity score using (4), and then label the candidate pair as a paraphrase if the similarity score exceeds a threshold of 0.6. We obtain the same F-measure (81%) at the combined methods from

<sup>3</sup> We used the test from <http://faculty.vassar.edu/lowry/rdiff.html>?

[30] and [5]. We obtain higher accuracy and precision at the cost of decreasing recall.

**Table 2. Characteristics of the paraphrase evaluation data set and our results**

Number of pairs in (data set)	Number of pairs determined as correct by human annotators ( $TP+FN$ )	Similarity threshold score in our method	Accuracy (%)	Number of correct pairs ( $TP$ )	Number of predicted pairs ( $TP+FP$ )
4076 (Training)	2753	0	67.54	2753	4076
		0.1	67.54	2753	4076
		0.2	67.54	2753	4076
		0.3	67.59	2753	4074
		0.4	67.74	2751	4064
		0.5	69.53	2708	3905
		0.6	<b>72.42</b>	2435	3241
		0.7	68.45	1874	2281
		0.8	56.67	1085	1183
		0.9	37.78	218	219
1725 (Test)	1147	0	66.49	1147	1725
		0.1	66.49	1147	1725
		0.2	66.49	1147	1725
		0.3	66.49	1147	1725
		0.4	66.66	1146	1720
		0.5	68.86	1128	1646
		0.6	<b>72.64</b>	1022	1369
		0.7	68.06	768	940
		0.8	56.29	443	493
		0.9	38.38	86	88
1.0	33.79	5	5		

## 5. Conclusion

Our proposed STS method achieves a very good Pearson correlation coefficient for 30 sentence pairs data set and outperforms the results obtained by Li et al. [20] (the improvement is statistically significant). For the paraphrase recognition task, our proposed STS method performs similar to the combined unsupervised method [30] and the combined supervised method [5]. The main advantage of our system is that it has lower complexity and running time than the other systems [20], [5], [30], because we use only one corpus-based measure, while they combine both corpus-based and WordNet-based measures. For example, Mihalcea et al [30] use six WordNet-based measures and two corpus-based measures. The complexity of the algorithms and their running time is given mainly by the number of searches in the corpus and in WordNet. We don't use WordNet at all, therefore saving a lot of time. We add the string

similarity measure, but this is very fast, because we apply it on short strings (no search needed).

Our method can be used as unsupervised or supervised. For the second task, paraphrase recognition, we used it as supervised, but only to find the best threshold. For the first task, comparing our sentence similarity score to scores assigned by human judges, our system is used as unsupervised (there is no training data available).

**Table 3. Text similarity results for paraphrase identification (test set)**

Metric	Accuracy	Precision	Recall	F-measure
Semantic similarity (corpus-based)				
PMI-IR	69.9	70.2	95.2	81.0
LSA	68.4	69.7	95.2	80.5
<b>STS</b>	<b>72.6</b>	<b>74.7</b>	89.1	<b>81.3</b>
Semantic similarity (knowledge-based)				
J & C	69.3	72.2	87.1	79.0
L & C	69.5	72.4	87.0	79.0
Lesk	69.3	72.4	86.6	78.9
Lin	69.3	71.6	88.7	79.2
W & P	69.0	70.2	92.1	80.0
Resnik	69.0	69.0	96.4	80.4
Combined(S)	71.5	72.3	92.5	81.2
Combined(U)	70.3	69.6	<b>97.7</b>	<b>81.3</b>
Baselines				
Threshold-1	33.8	100.0	0.44	0.87
Vector-based	65.4	71.6	79.5	75.3
Random	51.3	68.3	50.0	57.8

## 6. References

- [1] L. Allison, T.I. Dix, "A Bit-String Longest-Common-Subsequence Algorithm," *Information Processing Letters*, vol. 23, pp. 305-310, 1986.
- [2] C. Burgess, K. Livesay, and K. Lund, "Explorations in Context Space: Words, Sentences, Discourse," *Discourse Processes*, vol. 25, nos. 2-3, pp. 211-257, 1998.
- [3] T.A.S. Coelho, P.P. Calado, L.V. Souza, B. Ribeiro-Neto, and R. Muntz, "Image Retrieval Using Multiple Evidence Ranking," *IEEE Trans. Knowledge and Data Eng.*, vol. 16, no. 4, pp. 408-417, Apr. 2004.
- [4] *Collins Cobuild English Dictionary for Advanced Learners*, J. Sinclair, ed., third ed. Harper Collins Pub., 2001.
- [5] C. Corley and R. Mihalcea, "Measures of Text Semantic Similarity," *Proc. ACL workshop on Empirical Modeling of Semantic Equivalence*, Ann Arbor, MI, June, 2005.
- [6] W. Dolan, C. Quirk, and C. Brockett, "Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources," *Proc. 20th Int'l Conf. Computational Linguistics*, 2004.

- [7] G. Erkan and D.R. Radev, "LexRank: Graph-Based Lexical Centrality As Saliency in Text Summarization," *J. Artificial Intelligence Research*, vol. 22, pp. 457-479, 2004.
- [8] P.W. Foltz, W. Kintsch, and T.K. Landauer, "The Measurement of Textual Coherence with Latent Semantic Analysis," *Discourse Processes*, vol. 25, nos. 2-3, pp. 285-307, 1998.
- [9] W. Frawley, *Linguistic Semantics*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1992.
- [10] A. Islam and D. Inkpen, "Second Order Co-occurrence PMI for Determining the Semantic Similarity of Words," *Proc. Int'l Conf. on Language Resources and Evaluation*, Genoa, Italy, May, 2006.
- [11] M. Jarmasz and S. Szipakowicz, "Roget's Thesaurus and Semantic Similarity," *Proc. Int'l Conf. on Recent Advances in Natural Language Processing*, pp. 212-219, 2003.
- [12] J. Jiang and D. Conrath, "Semantic Similarity based on Corpus Statistics and Lexical Taxonomy," *Proc. Int'l Conf. Research in Computational Linguistics*, 1997.
- [13] Y. Ko, J. Park, and J. Seo, "Improving Text Categorization Using the Importance of Sentences," *Information Processing and Management*, vol. 40, pp. 65-79, 2004.
- [14] G. Kondrak, "N-gram Similarity and Distance," *Proc. Twelfth Int'l Conf. on String Processing and Information Retrieval*, pp. 115-126, 2005.
- [15] T. K. Landauer and S. T. Dumais, "A Solution to Plato's Problem: The Latent Semantic Analysis Theory of the Acquisition, Induction, and Representation of Knowledge," *Psychological Review*, vol. 104, nos. 2, pp. 211-240, 1997.
- [16] T. K. Landauer, P. W. Foltz, and D. Laham, "Introduction to Latent Semantic Analysis," *Discourse Processes*, vol. 25, nos. 2-3, pp. 259-284, 1998.
- [17] M. Lapata and R. Barzilay, "Automatic Evaluation of Text Coherence: Models and Representations," *Proc. 19th Int'l Joint Conf. AI*, 2005.
- [18] C. Leacock and M. Chodorow, "Combining Local Context and WordNet Sense Similarity for Word Sense Identification," *WordNet, An Electronic Lexical Database*, The MIT Press, 1998.
- [19] M. Lesk, "Automatic Sense Disambiguation using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone," *Proc. SIGDOC Conf.*, 1986.
- [20] Y. Li, D. McLean, Z. Bandar, J. O'Shea, and K. Crockett, "Sentence Similarity Based on Semantic Nets and Corpus Statistics," *IEEE Trans. Knowledge and Data Eng.*, vol. 18, no. 8, pp. 1138-1149, Aug. 2006.
- [21] Y.H. Li, Z. Bandar, and D. McLean, "An Approach for Measuring Semantic Similarity Using Multiple Information Sources," *IEEE Trans. Knowledge and Data Eng.*, vol. 15, no. 4, pp. 871-882, July/ Aug. 2003.
- [22] C. Lin and E. Hovy, "Automatic Evaluation of Summaries using n-gram Co-occurrence Statistics," *Proc. Human Language Technology Conf.*, 2003.
- [23] D. Lin, "An Information-theoretic Definition of Similarity," *Proc. Int'l Conf. Machine Learning*, 1998.
- [24] T. Liu and J. Guo, "Text Similarity Computing Based on Standard Deviation," *Proc. Int'l Conf. on Intelligent Computing*, D.-S. Huang, X.-P. Zhang and G.-B. Huang, eds., LNCS 3644, Springer, pp. 456-464, 2005.
- [25] Y. Liu and C.Q. Zong, "Example-Based Chinese-English MT," *Proc. 2004 IEEE Int'l Conf. Systems, Man, and Cybernetics*, vols. 1-7, pp. 6093-6096, 2004.
- [26] J. Madhavan, P. Bernstein, A. Doan, and A. Halevy, "Corpus-based Schema Matching," *Int'l Conf. Data Eng.*, 2005.
- [27] A. Maguitman, F. Menczer, H. Roinestad, and A. Vespignani, "Algorithmic Detection of Semantic Similarity," *Proc. 14th Int'l World Wide Web Conf.*, May 2005.
- [28] C.T. Meadow, B.R. Boyce, and D.H. Kraft, *Text Information Retrieval Systems*, second ed. Academic Press, 2000.
- [29] I.D. Melamed, "Bitext Maps and Alignment via Pattern Recognition," *Computational Linguistics*, vol. 25, no. 1, pp. 107-130, 1999.
- [30] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and Knowledge-based Measures of Text Semantic Similarity," *Proc. American Association for Artificial Intelligence*, Boston, July, 2006.
- [31] G.A. Miller and W.G. Charles, "Contextual Correlates of Semantic Similarity," *Language and Cognitive Processes*, vol. 6, no. 1, pp. 1-28, 1991.
- [32] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller, "Introduction to WordNet: An on-line lexical database," *CSL 43*, Cognitive Science Laboratory, Princeton University, Princeton, NJ, 1993.
- [33] E.K. Park, D.Y. Ra, and M.G. Jang, "Techniques for Improving Web Retrieval Effectiveness," *Information Processing and Management*, vol. 41, no. 5, pp. 1207-1223, 2005.
- [34] P. Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy," *Proc. 14th Int'l Joint Conf. AI*, 1995.
- [35] M.A. Rodriguez and M.J. Egenhofer, "Determining Semantic Similarity among Entity Classes from Different Ontologies," *IEEE Trans. Knowledge and Data Eng.*, vol. 15, no. 2, pp. 442-456, Mar./Apr. 2003.
- [36] H. Rubenstein and J.B. Goodenough, "Contextual Correlates of Synonymy," *Comm. ACM*, vol. 8, no. 10, pp. 627-633, 1965.
- [37] G. Salton and M. Lesk, *Computer evaluation of indexing and text processing*. Prentice Hall, Englewood Cliffs, New Jersey, pp. 143-180., 1971.
- [38] P. Turney, "Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL," *Proc. Twelfth European Conf. Machine Learning*, 2001.
- [39] Z. Wu and M. Palmer, "Verb Semantics and Lexical Selection," *Proc. Ann. Meeting Association for Computational Linguistics*, 1994.

# Exploring the Automatic Selection of Basic Level Concepts\*

Rubén Izquierdo & Armando Suárez  
GPLSI. Departament de LSI. UA.  
Alacant, Spain  
{ruben,armando}@dlsi.ua.es

German Rigau  
IXA NLP Group. EHU.  
Donostia, Spain  
german.rigau@ehu.es

## Abstract

We present a very simple method for selecting Base Level Concepts using basic structural properties of WordNet. We also empirically demonstrate that these automatically derived set of Base Level Concepts group senses into an adequate level of abstraction in order to perform class-based Word Sense Disambiguation. In fact a very naive Most Frequent classifier using the classes selected is able to perform a semantic tagging with accuracy figures over 75%.

## Keywords

WordNet, word-senses, levels of abstraction, Word Sense Disambiguation

## 1 Introduction

Word Sense Disambiguation (WSD) is an intermediate Natural Language Processing (NLP) task which consists in assigning the correct semantic interpretation to ambiguous words in context. One of the most successful approaches in the last years is the *supervised learning from examples*, in which statistical or Machine Learning classification models are induced from semantically annotated corpora [11]. Generally, supervised systems have obtained better results than the unsupervised ones, as shown by experimental work and international evaluation exercises such as Senseval<sup>1</sup>. These annotated corpora are usually manually tagged by lexicographers with word senses taken from a particular lexical semantic resource –most commonly WordNet<sup>2</sup> (WN) [7].

WN has been widely criticised for being a sense repository that often offers too fine-grained sense distinctions for higher level applications like Machine Translation or Question & Answering. In fact, WSD at this level of granularity, has resisted all attempts of inferring robust broad-coverage models. It seems that many word-sense distinctions are too subtle to be captured by automatic systems with the current small volumes of word-sense annotated examples. Possibly, building class-based classifiers would allow to avoid the data sparseness problem of the word-based approach.

Recently, using WN as a sense repository, the organizers of the English all-words task at SensEval-3 reported an inter-annotation agreement of 72.5% [17]. Interestingly, this result is difficult to outperform by state-of-the-art fine-grained WSD systems.

Thus, some research has been focused on deriving different sense groupings to overcome the fine-grained distinctions of WN [8] [14] [12] [1] and on using pre-defined sets of sense-groupings for learning class-based classifiers for WSD [16] [4] [18] [5] [3]. However, most of the later approaches used the original Lexicographical Files of WN (more recently called Supersenses) as very coarse-grained sense distinctions. However, not so much attention has been paid on learning class-based classifiers from other available sense-groupings such as WordNet Domains [10], SUMO labels [13], EuroWordNet Base Concepts [19] or Top Concept Ontology labels [2]. Obviously, these resources relate senses at some level of abstraction using different semantic criteria and properties that could be of interest for WSD. Possibly, their combination could improve the overall results since they offer different semantic perspectives of the data. Furthermore, to our knowledge, to date no comparative evaluation have been performed exploring different sense-groupings.

We present a very simple method for selecting Base Level Concepts [15] using basic structural properties of WN. We also empirically demonstrate that these automatically derived set of Base Level Concepts group senses into an adequate level of abstraction in order to perform class-based WSD.

This paper is organized as follows. Section 2 introduce the different levels of abstraction that are relevant for this study, and the available sets of semi-automatically derived Base Concepts. In section 3, we present the method for deriving fully automatically a number of Base Level Concepts from any WN version. Section 4 reports the resulting figures of a direct comparison of the resources studied. Section 5 provides an empirical evaluation of the performance of the different levels of abstraction. In section 6 we provide further insights of the results obtained and finally, in section 7 some concluding remarks are provided.

## 2 Levels of abstraction

The notion of Base Concepts (hereinafter BC) was introduced in EuroWordNet<sup>3</sup> [19]. The BC are supposed to be the concepts that play the most important role in the various wordnets of different languages. This role

\*This paper has been supported by the European Union under the project QALL-ME (FP6 IST-033860) and the Spanish Government under the project Text-Mess (TIN2006-15265-C06-01) and KNOW (TIN2006-15049-C03-01)

<sup>1</sup> <http://www.senseval.org>

<sup>2</sup> <http://wordnet.princeton.edu>

<sup>3</sup> <http://www.illc.uva.nl/EuroWordNet/>

was measured in terms of two main criteria: a high position in the semantic hierarchy and having many relations to other concepts. Thus, the BC are the fundamental building blocks for establishing the relations in a wordnet. In that sense, the Lexicographic Files (or Supersenses) of WN could be considered the most basic set of BC.

**Basic Level Concepts** [15] (hereinafter BLC) should not be confused with **Base Concepts**. BLC are a compromise between two conflicting principles of characterization: a) to represent as many concepts as possible (abstract concepts), and b) to represent as many distinctive features as possible (concrete concepts).

As a result of this, Basic Level Concepts typically occur in the middle of hierarchies and less than the maximum number of relations. BC mostly involve the first principle of the Basic Level Concepts only. BC are generalizations of features or semantic components and thus apply to a maximum number of concepts. Our work focuses on devising simple methods for selecting automatically an accurate set of Basic Level Concepts from WN.

WordNet synsets are organized in forty five Lexicographer Files, or SuperSenses, based on syntactic categories (nouns, verbs, adjectives and adverbs) and logical groupings, such as person, phenomenon, feeling, location, etc. There are 26 basic categories for nouns, 15 for verbs, 3 for adjectives and 1 for adverbs. Within EuroWordNet, initially, a set of 1,024 Common Base Concepts was selected from WN1.5. The BALKANET project<sup>4</sup> selected his own list of BC extending the original set of BC of EWN to a final set of 4,698 ILI records from WN2.0<sup>5</sup> (3,210 nouns, 1,442 verbs and 37 adjectives). In the the MEANING project<sup>6</sup>, the number of BC selected from WN1.6 was 1,535 (793 for nouns and 742 for verbs).

### 3 Automatic Selection of Base Level Concepts

This section describes a simple method for deriving a set of Base Level Concepts (BLC) from WN. The method has been applied to different WN versions for nouns and verbs. Basically, to select the appropriate BLC of a particular synset, the algorithm only considers the relative number of relations of their hypernyms. We derived two different sets of BLC depending on the type of relations considered: a) all types of relations encoded in WN (All) and b) only the hyponymy relations encoded in WN (Hypo).

The process follows a bottom-up approach using the chain of hypernym relations. For each synset in WN, the process selects as its Base Level Concept the first local maximum according to the relative number of relations. For synsets having multiple hypernyms, the path having the local maximum with higher number of relations is selected. Usually, this process finishes having a number of “fake” Base Level Concepts. That is, synsets having no descendants (or with a very small

#rel.	synset
18	group_1,grouping_1
19	social_group_1
<b>37</b>	organisation_2,organization_1
10	establishment_2,institution_1
<b>12</b>	faith_3,religion_2
5	Christianity_2, <b>church_1</b> ,Christian_church_1
#rel.	synset
14	entity_1,something_1
29	object_1,physical_object_1
39	artifact_1,artefact_1
63	construction_3,structure_1
<b>79</b>	building_1,edifice_1
11	place_of_worship_1, ...
<b>19</b>	<b>church_2</b> ,church_building_1
#rel.	synset
20	act_2,human_action_1,human_activity_1
<b>69</b>	activity_1
5	ceremony_3
<b>11</b>	religious_ceremony_1,religious_ritual_1
7	service_3,religious_service_1,divine_service_1
1	<b>church_3</b> ,church_service_1

**Table 1:** Possible Base Level Concepts for the noun Church in WN1.6

number) but being the first local maximum according to the number of relations considered. Thus, the process finishes checking if the number of concepts subsumed by the preliminary list of BLC is higher than a certain threshold. For those BLC not representing enough concepts according to a certain threshold, the process selects the next local maximum following the hypernym hierarchy. Thus, depending on the type of relations considered to be counted and the threshold established, different sets of BLC can be easily obtained for each WN version.

An example is provided in Table 1. This table shows the possible BLC for the noun “church” using WN1.6. The table presents the hypernym chain for each synset together with the number of relations encoded in WN for the synset. The local maxima along the hypernym chain of each synset appears in bold. Obviously, different criteria will select a different set of Base Level Concepts.

Instead of highly related concepts, we also considered highly frequent concepts as possible indicator of a large set of features. Following the same basic algorithm, we also used the relative frequency of the synsets in the hypernym chain. That is, we derived two other different sets of BLC depending on the source of relative frequencies considered: a) the frequency counts in SemCor (FreqSC) and b) the frequency counts appearing in WN (FreqWN). The frequency of a synset has been obtained summing up the frequencies of its word senses. In fact, WN word-senses were ranked using SemCor and other sense-annotated corpora. Thus, the frequencies of SemCor and WN are similar, but not equal.

### 4 Comparing Base Level Concepts

Different sets of Base Level Concepts (BLC) have been generated using different WN versions, types of relations (All and Hypo), sense frequencies (FreqSC and FreqWN) and thresholds.

Table 2 presents the total number of BLC and its

<sup>4</sup> <http://www.ceid.upatras.gr/Balkanet>

<sup>5</sup> [http://www.globalwordnet.org/gwa/5000\\_bc.zip](http://www.globalwordnet.org/gwa/5000_bc.zip)

<sup>6</sup> <http://www.lsi.upc.es/~nlp/meaning>



average depth for WN1.6<sup>7</sup> varying the threshold and the type of relations considered (All or Hypo) and the type of frequency (WN or SemCor).

Threshold	Relation	BLC		Depth	
		Noun	Verb	Noun	Verb
0	all	3,094	1,256	7.09	3.32
	hypo	2,490	1,041	7.09	3.31
	SemCor	34,865	3,070	7.44	3.41
	WN	34,183	2,615	7.44	3.30
10	all	971	719	6.20	1.39
	hypo	993	718	6.23	1.36
	SemCor	690	731	5.74	1.38
	WN	691	738	5.77	1.40
20	all	558	673	5.81	1.25
	hypo	558	672	5.80	1.21
	SemCor	339	659	5.43	1.22
	WN	340	667	5.47	1.23
50	all	253	633	5.21	1.13
	hypo	248	633	5.21	1.10
	SemCor	94	630	4.35	1.12
	WN	99	631	4.41	1.12

**Table 2:** Automatic Base Level Concepts for WN1.6 using relations or frequencies

As expected, when increasing the threshold, the total number of automatic BLC and its average depth decrease. For instance, using all relations on the nominal part of WN, the total number of BLC ranges from 3,094 (no threshold) to 253 (threshold 50). However, although the number of total BLC for nouns decreases dramatically (around 10 times), the average depth of the synsets selected only ranges from 7.09 to 5.21 using both types of relations (All and Hypo). This fact, possibly indicates the robustness of the approach.

Also as expected, the verbal part of WN behave differently. In this case, since the verbal hierarchies are much shorter, the average depth of the synsets selected ranges from 3.32 to only 1.13 using all relations, and from 3.31 to 1.10 using hypo relations.

In general, when using the frequency criteria, we can observe a similar behaviour than when using the relation criteria. However, now the effect of the threshold is more dramatic, specially for nouns. Again, although the number of total BLC for nouns decreases dramatically, the average depth of the synsets selected only ranges from 7.44 to 4.35 and 4.41. As expected, verbs behave differently than nouns. The number of BLC (for both SemCor and WN frequencies) reaches a plateau of around 600. In fact, this number is very close to the verbal top beginners.

Table 3 summarizes the BALKANET and MEANING Base Concepts including the total number of synsets and their average depth.

Set	PoS	#BC	Depth.
BALKANET	Noun	3,210	5.08
	Verb	1,442	2.45
MEANING	Noun	793	4.93
	Verb	742	1.36

**Table 3:** BALKANET and MEANING Base Concepts

## 5 Sense-groupings as semantic classes

In order to study to what extend the different sense-groupings could be of the interest for class-based

<sup>7</sup> WN1.6 have 66,025 nominal and 12,127 verbal synsets.

	Senses	BLC-A	BLC-S	SS
Nouns	4.93	4.07	4.00	3.06
Verbs	11.00	8.64	8.72	4.08
N + V	7.66	6.13	6.13	3.52

**Table 4:** Polysemy degree over SensEval-3

WSD, we present a comparative evaluation of the different sense-groupings in a controlled framework. We tested the behaviour of the different sets of sense-groupings (WN senses, BALKANET BC, MEANING BC, automatic BLC and SuperSenses) using the English all-words task of SensEval-3. Obviously, different sense-groupings would provide different abstractions of the semantic content of WN, and we expect a different behaviour when disambiguating nouns and verbs. In fact, the most common baseline used to test the performance of a WSD system, is the Most Frequent Sense Classifier. In this study, we will use this simple but robust heuristic to compare the performances of the different sense-groupings. Thus, we will use SemCor<sup>8</sup> [9] to train for Most Frequent Classifiers for each word and sense-grouping. We only used brown1 and brown2 parts of SemCor to train the classifiers. We used standard Precision, Recall and F1 measure (harmonic mean between Precision and Recall) to evaluate the performance of each classifier.

For WN senses, MEANING BC, the automatic BLC, and Lexicographic Files, we used WN1.6. For BALKANET BC we used the synset mappings provided by [6]<sup>9</sup>, translating the BC from WN2.0 to WN1.6. For testing the Most Frequent Classifiers we also used these mappings to translate the sense-groupings from WN1.6 to WN1.7.1.

Table 4 presents the polysemy degree for nouns and verbs of the different words when grouping its senses with respect the different semantic classes on SensEval-3. Senses stand for WN senses, BLC-A for automatic BLC derived using a threshold of 20 and all relations, BLC-S for automatic BLC derived using a threshold of 20 and frequencies from SemCor and SS for the SuperSenses. As expected, while increasing the abstraction level the polysemy degree decreases. Notice that the reduction is dramatic for verbs (from 11.0 to only 4.08). Notice also, that when using the Base Level Concept representations a high degree of polysemy is maintained for nouns and verbs.

Table 5 presents for polysemous words the performance in terms of F1 measure of the different sense-groupings when training the class-frequencies on SemCor and testing on SensEval-3. That is, for each polysemous word in SensEval-3 the Most Frequent Class is obtained from SemCor. Best results are marked using bold.

As expected, SuperSenses obtain very high F1 results for nouns and verbs. Comparing the BC from BALKANET and the best results seems to be achieved by MEANING BC for both nouns and verbs. Notice that the set of BC from BALKANET was larger than the ones selected in MEANING, thus indicating that the BC from MEANING provide a better level of abstraction.

Regarding the relations criteria, all sets of auto-

<sup>8</sup> Annotated using WN1.6.

<sup>9</sup> <http://www.lsi.upc.edu/~nlp/>

Class	All		Hypo		Semcor		WN	
	Nouns	Verbs	Nouns	Verbs	Nouns	Verbs	Nouns	Verbs
Senses	63.69	49.78	63.69	49.78	63.69	49.78	63.69	49.78
Balkanet	65.15	50.84	65.15	50.84	65.15	50.84	65.15	50.84
Meaning	65.28	53.11	65.28	53.11	65.28	53.11	65.28	53.11
BLC-0	66.36	54.30	65.76	54.30	64.45	52.27	64.95	51.75
BLC-10	66.31	54.45	65.86	54.45	64.98	53.21	65.59	53.29
BLC-20	<b>67.64</b>	54.60	<b>67.28</b>	54.60	65.73	53.97	66.30	53.44
BLC-30	67.03	54.60	66.72	54.60	66.46	54.15	66.67	53.61
BLC-40	66.61	55.54	66.77	<b>55.54</b>	68.46	<b>54.63</b>	<b>69.16</b>	54.22
BLC-50	67.19	<b>55.69</b>	67.19	<b>55.54</b>	<b>68.84</b>	<b>54.63</b>	69.11	<b>54.63</b>
SuperSenses	<b>73.05</b>	<b>76.41</b>	<b>73.05</b>	<b>76.41</b>	<b>73.05</b>	<b>76.41</b>	<b>73.05</b>	<b>76.41</b>

**Table 5:** *F1 measure for polysemous words tested on SensEval-3*

matic BLC perform better than those BC provided by BALKANET or MEANING. Also in this case, for nouns, the best results are obtained when using a threshold of only 20. We should highlight this result since this set of BLC obtain better WSD performance than the rest of automatically derived BLC while maintaining more information of the original synsets. That is, BLC-20 using all relatons (558 classes) achieves an F1-score of 67.64, while SuperSenses using a much smaller set (26 classes) achieves 73.05. We can also observe that in general, using hyponymy relations we obtain slightly lower performances than using all relations. Possibly, this fact indicates that a higher number of hyponymy relations is required for a Base Level Concept to compensate minor (but richer) number of relations. These results suggest that intermediate levels of representation such as the automatically derived Base Concept Levels could be appropriate for learning class-based WSD classifiers.

Also in Table 5, we present the results of using frequencies from SemCor and frequencies from WN for selecting the BLC. In this case, not all sets of automatic BLC surpass the BC from BALKANET and MEANING. The best results are obtained when using higher thresholds. However, in this case, verbal BLC obtain slightly lower results than using the relations criteria (both all and hypo). We can also observe that in general, using SemCor frequencies we obtain slightly lower performances than using WN frequencies.

These results for polysemous words reinforce our initial observations. That is, that the method for automatically deriving intermediate levels of representation such the Base Concept Levels seems to be robust enough for learning class-based WSD classifiers. In particular, it seems that BLC could achieve high levels of accuracy while maintaining adequate levels of abstraction (with hundreds of BLC). In particular, the automatic BLC obtained using the relations criteria (All or Hypo) surpass the BC from BALKANET and MEANING. For verbs, it seems that even the unique top beginners require an extra level of abstraction (that is, the SuperSense level) to be affective.

## 6 Discussion

We can put the current results in context, although indirectly, by comparison with the results of the English SensEval-3 all-words task systems. In this case, the best system presented an accuracy of 65.1%, while the “WN first sense” baseline would achieve 62.4%<sup>10</sup>.

<sup>10</sup> This result could be different depending on the treatment of multiwords and hyphenated words.

Class	Relations			Frequencies		
	Noun	Verb	N+V	Noun	Verb	N+V
Senses	71.79	52.89	63.24	71.79	52.89	63.24
Balkanet	73.06	53.82	64.37	73.06	53.82	64.37
Meaning	73.40	56.40	65.71	73.40	56.40	65.71
BLC-0	74.80	58.32	67.35	72.99	55.33	65.01
BLC-10	74.99	58.46	67.52	74.60	57.08	66.69
BLC-20	76.12	58.60	68.20	75.62	57.22	67.31
BLC-30	75.99	58.60	68.14	76.10	57.63	67.76
BLC-40	75.76	59.70	68.51	<b>78.03</b>	58.18	69.07
BLC-50	<b>76.22</b>	<b>59.83</b>	<b>68.82</b>	<b>78.03</b>	<b>58.87</b>	<b>69.38</b>
SuperSns	<b>81.87</b>	<b>79.23</b>	<b>80.68</b>	<b>81.87</b>	<b>79.23</b>	<b>80.68</b>

**Table 6:** *F1 measure for nouns and verbs using all relations and WN frequencies criteria for selecting BLC*

Furthermore, it is also worth mentioning that in this edition there were a few systems above the “WN first sense” baseline (4 out of 26 systems). Usually, this baseline is very competitive in WSD tasks, and it is extremely hard to improve upon even slightly.

Table 6 present for monosemous and polysemous nouns and verbs the F1 measures of the different sense-groupings obtained with all relations and WN frequencies criteria when training the class-frequencies on SemCor and testing on SensEval-3. Best results are marked using bold.

Obviously, higher accuracy figures are obtained when incorporating also monosemous words. Note this naive system achieves for Senses an F1 of 63.24, very similar to those reported in SensEval-3, and SuperSenses obtain a very high F1 of 80.68. Regarding the automatic BLC, the best results are obtained for BLC-50, but all of them outperform the BC from BALKANET and MEANING. However, for nouns and using all relations, BLC-20 (with 558 classes) obtain only slightly lower F1 figures than BLC-50 (with 253 classes). When using WN frequencies instead of all relations, BLC even achieve higher results but not all of them outperform the BC from BALKANET and MEANING.

Surprisingly, these naive Most frequent WSD systems trained on SemCor are able to achieve very high-levels of accuracy. For nouns, using BLC-20 (selected from all relations, 558 semantic labels) the system reaches 76.12, while using BLC-40 (selected from WN frequencies, 132 semantic labels) the system achieves 78.03. Finally, using SuperSenses for verbs (15 semantic labels) this naive system scores 79.23.

To our knowledge, the best results for class-based WSD are those reported by [3]. This system performs a sequence tagging using a perceptron-trained HMM, using SuperSenses, training on SemCor and testing on the SensEval-3. The system achieves an F1-score of 70.74, obtaining a significant improvement from a baseline system which scores only 64.09. In this case,

the first sense baseline is the SuperSense of the most frequent synset for a word, according to the WN sense ranking. Possibly, the origin of the discrepancies between our results and those reported by [3] is twofold. First, because they use a BIO sequence schema for annotation, and second, the use of the brown-v part of SemCor to establish sense-frequencies.

## 7 Conclusions and further work

The WSD task seems to have reached its maximum accuracy figures with the usual framework. Some of its limitations could come from the sense-granularity of WordNet (WN). Moreover, it is not clear how WSD can contribute with the current result to improve other NLP tasks. Changing the set of classes could be a solution to enrich training corpora with many more examples. In fact, our most frequent naive systems are able to perform a semantic tagging with accuracy figures over 75%.

Base Level Concepts (BLC) are concepts that are representative for a set of other concepts. In the present work, a simple method for automatically selecting BLC from WN based on the hypernym hierarchy and the number of stored frequencies or relationships between synsets have been shown. Although, some sets of Base Concepts are available at this moment (e.g. EUROWORDNET, BALKANET, MEANING), a huge manual effort should be invested for its development. Other sets of Base Concepts, like WN Lexicographer Files are clearly insufficient in order to describe and distinguish between the enormous number of concepts that are used in a text. Using a very simple baseline, the Most Frequent Class, our approach empirically shows a clear improvement over such other sets. In addition, our method is capable to get a more or less detailed sets of BLC without losing semantic discrimination power.

Other selection criteria for selecting BLC should be investigated. We are also interested in the direct comparison between automatically and manually selected BLC. Finally, we plan to use BLC for supervised class-based WSD.

## References

- [1] E. Agirre and O. LopezDeLaCalle. Clustering wordnet word senses. In *Proceedings of RANLP'03*, Borovets, Bulgaria, 2003.
- [2] M. J. Atserias J., Climent S. and R. G. A proposal for a shallow ontologization of wordnet. In *Proceedings of the 21th Annual Meeting of the Sociedad Espaola para el Procesamiento del Lenguaje Natural*, Granada, Spain, 2005.
- [3] M. Ciaramita and Y. Altun. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'06)*, pages 594–602, Sydney, Australia, 2006. ACL.
- [4] M. Ciaramita and M. Johnson. Supersense tagging of unknown nouns in wordnet. In *Proceedings of the Conference on Empirical methods in natural language processing (EMNLP'03)*, pages 168–175. ACL, 2003.
- [5] J. Curran. Supersense tagging of unknown nouns using semantic similarity. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL'05)*, pages 26–33. ACL, 2005.
- [6] J. Daudé, L. Padró, and G. Rigau. Validation and tuning of wordnet mapping techniques. In *Proceedings of the International Conference on Recent Advances on Natural Language Processing (RANLP'03)*, Borovets, Bulgaria., 2003.
- [7] C. Fellbaum, editor. *WordNet. An Electronic Lexical Database*. The MIT Press, 1998.
- [8] M. Hearst and H. Schütze. Customizing a lexicon to better suit a computational task. In *Proceedings of the ACL SIGLEX Workshop on Lexical Acquisition*, Stuttgart, Germany, 1993.
- [9] H. Kučera and W. N. Francis. *Computational Analysis of Present-Day American English*. Brown University Press, Providence, RI, USA, 1967.
- [10] B. Magnini and G. Cavaglia. Integrating subject fields codes into wordnet. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, 2000.
- [11] L. Màrquez, G. Escudero, D. Martínez, and G. Rigau. Supervised corpus-based methods for wsd. In *E. Agirre and P. Edmonds (Eds.) Word Sense Disambiguation: Algorithms and applications.*, volume 33 of *Text, Speech and Language Technology*. Springer, 2006.
- [12] R. Mihalcea and D. Moldovan. Automatic generation of coarse grained wordnet. In *Proceeding of the NAACL workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, Pittsburg, USA, 2001.
- [13] I. Niles and A. Pease. Towards a standard upper ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, pages 17–19. Chris Welty and Barry Smith, eds, 2001.
- [14] W. Peters, I. Peters, and P. Vossen. Automatic sense clustering in eurowordnet. In *First International Conference on Language Resources and Evaluation (LREC'98)*, Granada, Spain, 1998.
- [15] E. Rosch. Human categorisation. *Studies in Cross-Cultural Psychology*, I(1):1–49, 1977.
- [16] F. Segond, A. Schiller, G. Greffenstette, and J. Chanod. An experiment in semantic tagging using hidden markov model tagging. In *ACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 78–81. ACL, New Brunswick, New Jersey, 1997.
- [17] B. Snyder and M. Palmer. The english all-words task. In R. Mihalcea and P. Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- [18] L. Villarejo, L. Màrquez, and G. Rigau. Exploring the construction of semantic class classifiers for wsd. In *Proceedings of the 21th Annual Meeting of Sociedad Espaola para el Procesamiento del Lenguaje Natural SEPLN'05*, pages 195–202, Granada, Spain, September 2005. ISSN 1136-5948.
- [19] P. Vossen, L. Bloksma, H. Rodriguez, S. Climent, N. Calzolari, A. Roventini, F. Bertagna, A. Alonge, and W. Peters. The eurowordnet base concepts and top ontology. Technical report, Paris, France, France, 1998.

# Collaborative Entity Extraction and Translation

Heng Ji

Ralph Grishman

Department of Computer Science

New York University

New York, NY, 10003, USA

{hengji, grishman}@cs.nyu.edu

## Abstract

*Entity extraction* is the task of identifying names and nominal phrases ('mentions') in a text and linking coreferring mentions. We propose the use of a new source of data for improving entity extraction: the information gleaned from large bitexts and captured by a statistical, phrase-based machine translation system. We translate the individual mentions and test properties of the translated mentions, as well as comparing the translations of coreferring mentions. The results provide feedback to improve source language entity extraction. Experiments on Chinese and English show that this approach can significantly improve Chinese entity extraction (2.2%-relative improvement in name tagging F-measure, representing a 15.0% error reduction), as well as Chinese to English entity translation (9.1% relative improvement in F-measure), over state-of-the-art entity extraction and machine translation systems.

## Keywords

Named Entities, Machine Translation, Joint Inference

## 1. Introduction

Named entity tagging has become an essential component of many NLP systems, such as question answering and information extraction. Building a high-performance name tagger, however, remains a significant challenge. The challenge is greater for languages such as Chinese and Japanese with neither capitalization nor overt tokenization to aid name detection, or Semitic languages such as Arabic that do not exhibit differences in orthographic case.

This challenge is now generally addressed by constructing, by hand, a large name-annotated corpus. Because of the cost of such annotation, several recent studies have sought to augment this approach through the use of un-annotated data, for example by constructing word classes (Miller et al., 2004) or by annotating additional data automatically and selecting the most confident annotations as further training (Ji and Grishman, 2006).

One further source of information for improving name taggers are bitexts – corpora pairing the text to be tagged with its translation into one or more other languages. Such bitexts are becoming increasingly available for many language pairs, and now play a central role in the creation of machine translation and name translation systems. By aligning the texts at the word level, we are able to infer properties of a sequence  $s$  in language  $S$  from the properties of the sequence of tokens  $t$  with which it is aligned in

language  $T$ . For example, knowing that  $t$  is a name, or merely that it is capitalized (for  $T = \text{English}$ ) makes it more likely that  $s$  is a name. So if we have multiple, closely competing name hypotheses in the source language  $S$ , we can use the bitext to select the correct analysis.

Huang and Vogel (2002) used these observations to improve the name tagging of a bitext, and the NE (named entity) dictionary learned from the bitext. We wish to take this one step further by using information which can be gleaned from bitexts to improve the tagging of data for which we do not have pre-existing parallel text. We will use a phrase-based statistical machine translation system trained from these bitexts; we will translate the source-language entities using the machine translation (MT) and name translation systems; and then we will use this translation to improve the tagging of the original text.

This approach is an example of joint inference across quite disparate knowledge sources: in this case, combining the knowledge from named entity tagging and translation to produce better results for each. Such symbiosis of analysis components will be essential for the creation of high-performance NLP systems.

The translation knowledge source has an additional benefit: because name variants in  $S$  may translate into the same form in  $T$ , translation can also aid in identifying name coreference in  $S$ .

## 2. Task and Terminology

We shall use the terminology of ACE<sup>1</sup> to explain our central ideas.

**entity:** an object or a set of objects in one of the semantic categories of interest, referred to by a set of mentions

**mention:** a reference to an entity (typically, a noun phrase)

**name mention:** a reference by name to an entity

**nominal mention:** a reference by a common noun or noun phrase to an entity

---

<sup>1</sup> The Automatic Content Extraction evaluation program of the U.S. Government. The ACE guidelines are at

<http://www ldc.upenn.edu/Projects/ACE/>

In this paper we consider five types of entities in ACE evaluation: PER (persons), ORG (organizations), GPE ('geo-political entities' – locations which are also political units, such as countries, counties, and cities), LOC (other locations), FAC (facility). *Entity extraction* can then be viewed as a combination of mention detection and classification with coreference analysis, which links coreferring mentions.

### 3. Motivation for Using Bitexts

We present first our motivation for using word-aligned bitexts to improve source language (*S*) entity extraction. Many languages have special features that can be employed for entity extraction. By using the alignment between the entity extraction results in language *S* and their translations in target language *T*, the language-specific information in *T* will enable the system to perform more accurate extraction than a model built from the monolingual corpus in *S* alone. In the following we present some examples for Chinese-English pair.

- **Chinese → English**

Chinese does not have white space for tokenization or capitalization, features which, for English, can help identify name boundaries and distinguish names from nominals. Using Chinese-English bitexts allows us to capture such indicative information to improve Chinese name tagging. For example,

(a) Results from Chinese name tagger

美德联盟立刻委任了一名执行人员出任 <ENAMEX TYPE="ORG">三菱新 </ENAMEX> 总裁。

(b) Bitext

Chinese: 三菱      新  
          |           |

English: *Mitsubishi new*

(c) Name tagging after using bitext

美德联盟立刻委任了一名执行人员出任 <ENAMEX TYPE="ORG">三菱</ENAMEX>新总裁。

Based on the title context word “总裁 (*president*)” the Chinese name tagger mistakenly identifies “*Mitsubishi new*” as an organization name. But the un-capitalized English translation of “*new*” can provide a useful clue to fix this boundary error.

- **English → Chinese**

On the other hand, Chinese has some useful language-specific properties for entity extraction. For example, standard Chinese family names are generally single characters drawn from a fixed set of 437 family names, and almost all first names include one or two characters. The suffix words (if there are any) of ORG and GPE names belong to relatively distinguishable fixed lists. This feature

– particular character or word vocabulary for names – can be exploited as useful ‘feedback’ for fixing name tagging errors.

(a) Results from English name tagger

The flashpoint in a week of bitter <ENAMEX TYPE="ORG">West Bank </ENAMEX> clashes ...

(b) Bitext

English: *West Bank*

          |  
Chinese: 西岸

(c) Name tagging after using translation

The flashpoint in a week of bitter <ENAMEX TYPE="LOC">West Bank</ENAMEX> clashes...

“Bank” in English can be the suffix word of either a ORG or LOC name, while its Chinese translation “岸 (*shore, side*)” indicates that “*West Bank*” is more likely to be a LOC name.

These examples indicate how aligned bitexts can aid entity extraction. However, in most cases the texts from which we wish to extract entities will not be part of such bitexts. We shall instead use a statistical MT system which in effect distills the knowledge in its training bitexts. We will use this MT system to generate entity translations, and then use these translations as we did the bitexts in the examples above.

## 4. General Approach

### 4.1 Combining Entity Extraction and Translation

We propose a new framework to improve source language *S* entity extraction through the indirect use of bitexts as follows.

We first apply a source language ‘baseline’ entity extraction system trained from a monolingual corpus to produce entities (*SEntities*), and then translate these entities into target language *T* (*TEntities*). Coreference decisions are made on the source language level. The *TEntities* carry information from a machine translation system trained from large bitexts, information which may not have been captured in the monolingual entity extraction. The *TEntities* can be used to provide *cross-lingual feedback* to confirm the results or repair the errors in *SEntities*. This feedback is provided by a set of rules which are applied iteratively.

However, in such a framework we face the problem that the translations produced by the MT system will not always be correct. In this paper we address this problem by using confidence estimation based on voting among translations of coreferring mentions, which we shall refer to as a *mention cache*. In section 4.2 and 4.3 we shall verify the

two hypotheses which are required to apply the cache scheme, and in section 4.4 we shall explain the details of these caches.

## 4.2 One Translation per Named Entity

Named entities may have many variants, for example, “IOC” and “International Olympic Committee” refer to the same entity; and “New York City” alternates with “New York”; but all these different variants tend to preserve ‘name heads’ – a brief “key” alternation that represent the *naming function* (Carroll, 1985). Unlike common words for which *fluency* and *vitality* are most required during translation, translating a named entity requires preserving its *functional* property – the real-world object that the name is referring to. Inspired by this linguistic property we propose a hypothesis:

- **Hypothesis (1).** *One Translation per Named Entity:* The translation of different name mentions is highly consistent within an entity.

This hypothesis may seem intuitive, but it is important to verify its accuracy. On 50 English documents (4360 mention pairs) from ACE 2007 Chinese to English Entity Translation training data with human tagged entities, we measure this hypothesis’ *accuracy* by:

| Coreferred mention pairs with *consistent* translations |

---

|Coreferred mention pairs |

We consider two translations *consistent* if one is a name component, acronym or adjective form of the other.

The *accuracy* of this hypothesis for different name types are: 99.6% for PER, 99.5% for GPE, 99.0% for ORG and 100% for LOC. This clearly indicates that Hypothesis (1) holds with high reliability.

## 4.3 One Source Name per Translation

Based on Hypothesis (1), we can select a single ‘best (maximal) name translation’ for each entity with a name; and this best translation can be used as ‘*feedback*’ to determine whether the extracted name mentions in source language are correct or not. If they are incorrect (if their translations are not consistent with the best translation), they can be replaced by a ‘best source language name’. This is justified by:

- **Hypothesis (2).** *One Source Name per Translation:* Names that have the same translation tend to exhibit *consistent* spellings in the source language.

In reviewing 101 Chinese documents (8931 mention pairs) with human translations from ACE07 entity translation training data, the accuracy of this hypothesis for all entity types was close to 100%; the exceptions appeared to be clear translation errors.

Therefore, if we require the name mentions in one entity to achieve consistent translation as well as extraction (name boundary and type), then we can fix within-doc or cross-doc entity-level errors, with small sacrifice of (<1%) exceptional instances.

## 4.4 Cross-lingual Voted Caches

Given an entity in source language  $S$ Entity and its translation  $T$ Entity, let  $SName(i)$  be a name mention of  $S$ Entity and have translation  $TName(i)$ . Then the above two properties indicate that if string  $TName(i)$  appears frequently in  $T$ Entity, then  $SName(i)$  is likely to be correct. On the other hand, if  $TName(i)$  is infrequent in  $T$ Entity and conflicts with the most frequent translation in boundary or word morphology, then  $SName(i)$  is likely to be a wrong extraction.

For a pair of languages  $S$  (source language)  $\rightarrow T$  (target language), we build the following voted cache models in order to get the best *assignment* (extraction or translation candidate) for each entity:

- **Inside-S-T-Cache**

For each name mention of one entity (inside a single document), record its unique translations and frequencies;

- **Cross-S-T-Cache**

Corpus-wide (across documents), for each name and its consistent variants, record its unique translations and their frequencies;

- **Cross-T-S- Cache**

Corpus-wide, for each set of consistent name translations in  $T$ , record the corresponding names in  $S$  and their frequencies.

The caches incorporate simple filters based on properties of language  $T$  to exclude translations which are not likely to be names. For  $T =$  English, we exclude empty translations, translations which are single un-capitalized tokens, and, for person names, translations with any un-capitalized tokens. In addition, in counting translations in the cache, we group together consistent translations. For English, this includes combining person name translations if one is a subsequence of the tokens in the other. The goal of these simple heuristics is to take advantage of the general properties of language  $T$  in order to increase the likelihood that the most frequent entry in the cache is indeed the best translation.

For each entry in these caches, we get the frequency of each unique *assignment*, and then use the following *margin* measurement to compute the confidence of the best assignment:

$$\text{Margin} = \text{Frequency (Best Assignment)} - \text{Frequency (Second Best Assignment)}$$

A large margin indicates greater confidence in the assignment.

## 5. Inference Rules

We can combine the language-specific information in  $S\text{Entity}$ , and its entry in the cross-lingual caches to detect potential extraction errors and take corresponding corrective measures. We construct the following inference rules and an example for some particular rules below.

Based on hypotheses (1) and (2), for a test corpus we aim to achieve a group of entities in both source and target languages, with high consistency on the following levels:

### Rule(1): Adjust Source Language Annotations to Achieve Mention-level Consistency:

#### Rule (1-1): Adjust Mention Identification

If a mention receives translation that has small margin as defined in section 4.4 and violates the linguistic constraints in target language, then do not classify the mention as a name.

#### Rule (1-2): Adjust Isolated Mention Boundary

Adjust the boundary of each mention of  $S\text{Entity}$  to be consistent with the mention receiving the best translation.

#### Rule (1-3): Adjust Adjacent Mention Boundary

If two adjacent mentions receive the same translation with high confidence, merge them into one single mention.

### Rule (2): Adjust Source Language Annotations to Achieve Entity-level Consistency:

If one entity is translated into two groups of different mentions, split it into two entities.

### Rule (3): Adjust Target Language Annotations to Achieve Mention-level Consistency:

Enforce entity-level translation consistency by propagating the high-confidence best translation through corefferred mentions.

For example, for the following Chinese document,

<TEXT>

<sent 1>加拿大第 37 届联邦议会 29 日举行会议, 选举自由党议员 **米利肯** 为众议院新议长。</sent>

The 37th Canadian Federal Parliament held a meeting on the 29th and elected Liberal MP **Miliken** as House of Commons speaker.

<sent2>今年 54 岁的 **彼得. 米利肯** 是来自加拿大安大略省金斯敦地区的议员。</sent>

The 54-year-old **Peter Miliken** is a MP from Kingston, Ontario, Canada.

<sent3>**米利肯** 是在 5 轮投票后当选的。</sent>

**Miliken** was elected after five rounds of voting.

</TEXT>

The baseline system extracts and translates the following entity:

{米利/Mili, 彼得.米利肯/Peter Miliken, 米利肯/Miliken}

By applying rule (1-2), we can fix the boundary of the first name mention “米利” into “米利肯” because “米利肯” has the (maximal) best translation “Miliken”:

{米利肯/Mili, 彼得.米利肯/Peter Miliken, 米利肯/Miliken}

then by applying rule (3) we can change the translation “Mili” into the more frequent translation “Miliken”<sup>2</sup>:

{米利肯/Miliken, 彼得.米利肯/Peter Miliken, 米利肯/Miliken}

These inferences are formalized in Appendix A. They are applied repeatedly until there are no further changes; improved translation in one iteration can lead to improved  $S$  entity extraction in a subsequent iteration.

## 6. System Pipeline

The overall system pipeline for language pair  $(S, T)$  is summarized in Figure 1.

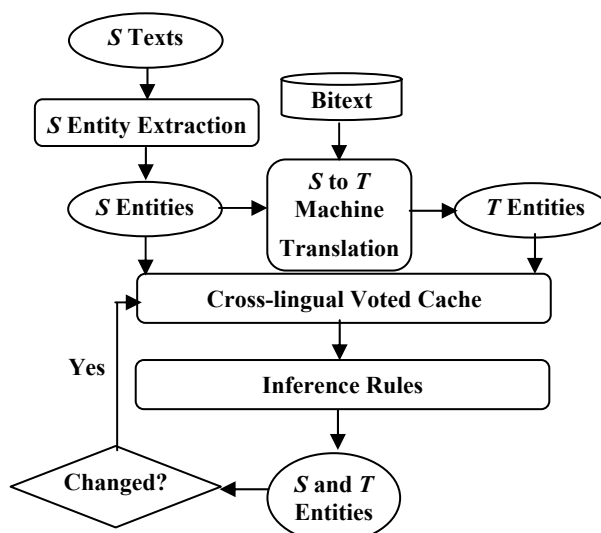


Figure 1. A Symbiotic Framework of Entity Extraction and Translation

## 7. Experiments on Chinese to English

In this section we shall present an example of applying this method using Chinese-to-English translation to improve Chinese entity extraction.

<sup>2</sup> Alternatively we could fix the English in this case by re-translating the corrected mentions. But in other cases rule (3) is needed to correct the translations.

## 7.1 Baseline Systems

We used a Chinese entity extraction system described in (Ji et al., 2005) and a statistical, phrase-based machine translation system (Zens and Ney, 2004) for our experiments. Each source mention is translated independently using the MT system<sup>3</sup>.

## 7.2 Rule Restriction

We tested the rules on a development set, and added a few source-language-specific restrictions on their applicability to improve performance. Also, where the rules allowed for two alternative corrections, we added a language-specific criterion for choosing the correction.<sup>4</sup>

## 7.3 Data

We took the Chinese newswire data from the ACE 2007 Entity Translation training and evaluation corpus as our blind test set, and evaluated our system. The test set includes 67 news texts, with 2077 name mentions and 1907 entities.

## 7.4 Improvement in Entity Extraction

The name tagging performance on different entity types is shown in Table 1 as follows.

Type	Baseline	After Using Inference Rules
PER	89.9%	91.2%
GPE	87.0%	86.9%
ORG	85.7%	88.5%
LOC	89.7%	90.6%
FAC	80.9%	85.3%
ALL	87.3%	89.2%

Table 1. F-Measure (%) of Name Tagging

<sup>3</sup> We tried an alternative approach in which mentions are translated in context and the mention translations are then extracted using word alignment information produced by the MT system, but it did not perform as well. The word alignments are indirectly derived from phrase alignment and can be quite noisy. As a result, noise in the form of words from the target language context is introduced into the mention translations. Manual evaluation on a small development set showed that isolated translation obtains (about 14%) better F-measure in translating names.

<sup>4</sup> Specifically: for Rule (1-2) we added a check that SName (i) and SName (j) are not a name and its acronym. Also for Rule (1-2), if SName (i) includes a conjunction the rule splits the name into two names, otherwise replacing it by SName (j). For Rule (1-1), since in Chinese most ambiguities between name and nominal arise in GPE or ORG names, GPE or ORG names are corrected into nominals, while PER names are deleted. Rule (1-3) was limited to merging mentions of selected entity type pairs, such as “PER-GPE” and “ORG-LOC” because they are unlikely to appear adjacent in Chinese.

Except for the small loss for GPE names, our method achieved positive corrections on most entity types. Significant improvements were achieved on ORG and FAC names for all three language sources, mainly because organization and facility names in English texts have less boundary ambiguity than in Chinese texts. So they are better aligned in bitexts and easier to translate. The small loss in GPE names for the Chinese source is due to the poor quality of the translation of country name abbreviations.

The rules can also improve nominal tagging by disambiguating mention types (name vs. nominal), and improve coreference by merging or splitting incorrect entity structures. All of these improvements benefit entity extraction.

## 7.5 Improvement in Entity Translation

A further benefit of our system is a boost in the translation quality of Chinese entities. We used the official ACE 2007-ET scorer<sup>5</sup> to measure the F-scores. The performance for translating different entity types is presented in Table 2.

Type	Baseline	After Using Inference Rules
PER	34.8%	36.7%
GPE	44.7%	49.8%
ORG	37.0%	39.9%
LOC	18.3%	18.1%
FAC	23.1%	23.3%
ALL	35.1%	38.3%

Table 2. F-Measure (%) of Entity Translation

The inference based on voting over mentions of an entity particularly improved GPE name abbreviation translation and fixed translated person foreign name boundaries. Thus we have succeeded in using the interaction of entity extraction and translation to improve the performance of both.

## 7.6 Error Analysis

The errors reveal both the shortcomings of the MT system and consistent difficulties across languages.

For a name not seen in training bitexts the MT system tends to mistakenly align part of the name with an uncapitalized token. Also, there are words where the ambiguity between name and nominal exists in both Chinese and English, such as “国会-parliament”. Rule (2) fails in these cases by mistakenly changing correct names

<sup>5</sup> The description of the ACE entity translation metric can be found at <http://www.nist.gov/speech/tests/ace/ace07/doc/ET07-evalplan-v1.6.pdf>



into nominal mentions. In these and other cases, we could apply a separate name transliteration system developed from larger name-specific bitexts to re-translate these difficult names. Or we could incorporate the confidence values such as (Ueffing and Ney, 2005) generated from the MT system into our cross-lingual cache model. Nevertheless, as Table 1 and 2 indicate, the rewards of using the bitext/translation information outweigh the risks.

## 8. Related Work

The work described here complements the research described by (Huang and Vogel, 2002). They presented an effective integrated approach that can improve the extracted named entity translation dictionary and the entity annotation in a bilingual training corpus. We expand their idea of alignment consistency to the task of entity extraction in a *monolingual test* corpus. Unlike their approach requiring reference translations in order to achieve highest alignment probability, we only need the source language unlabeled document. So our approach is more broadly applicable and also can be extended to additional information extraction tasks (nominal tagging and coreference).

Aligned bitexts have also been used to project name tags from French to English by Riloff et al. (2002) and from Japanese to English by Sudo et al. (2004), but their approaches only use the entity information from the source language.

In addition, our approach represents a form of cross-lingual joint inference, which complements the joint inference in the monolingual analysis pipeline as described in (Ji and Grishman, 2005) and (Roth and Yi, 2004).

## 9. Conclusion and Future Work

Bitexts can provide a valuable additional source of information for improving named entity tagging. We have demonstrated how the information from bitexts, as captured by a phrase-based statistical machine translation system, and then used to generate translations, can be used to correct errors made by a source-language named-entity tagger. While our approach has only been tested on Chinese and English so far, we can expect that it is applicable to other language pairs. The approach is independent of the baseline tagging/extraction system, and so can be used to improve systems with varied learning schemes or rules.

There are a number of natural extensions and generalizations of the current approach. In place of correction rules, we could adopt a joint inference approach based on generating alternative source language name tags (with probabilities), estimating the probabilities of the corresponding target language features, and seeking an optimal tag assignment. Although the current approach

only relies on limited target language features, we could use a full target-language entity extractor (as Huang and Vogel (2002) did), providing more information as feedback (for example, name type information). Furthermore, we intend to pass the name tagging hypotheses to a name transliteration system and use the transliteration results as additional feedback in assessing name hypotheses.

## 10. Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency under Contract No. HR0011-06-C-0023, and the National Science Foundation under Grant IIS-00325657. Any opinions, findings and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the U. S. Government.

## 11. References

- [1] John M. Carroll. 1985. *What's in a Name?: An Essay in the Psychology of Reference*. New York, US.
- [2] Fei Huang and Stephan Vogel. 2002. Improved Named Entity Translation and Bilingual Named Entity Extraction. *In ICMI 2002:253-258*. Pittsburgh, PA, US.
- [3] Heng Ji and Ralph Grishman. 2005. Improving Name Tagging by Reference Resolution and Relation Detection. *Proc. ACL2005*. pp. 411-418. Ann Arbor, USA.
- [4] Heng Ji and Ralph Grishman. 2006. Data Selection in Semi-supervised Learning for Name Tagging. *In ACL 2006 Workshop on Information Extraction Beyond the Document:48-55*. Sydney, Australia.
- [5] Heng Ji, Adam Meyers and Ralph Grishman. 2005. NYU's Chinese ACE 2005 EDR System Description. ACE 2005 PI Workshop. Washington, US.
- [6] Scott Miller, Jethran Guinness and Alex Zamanian. 2004. Name Tagging with Word Clusters and Discriminative Training. *In HLT/NAACL 2004: 337-342*. Boston, Massachusetts, US.
- [7] Dan Roth and Wen-tau Yih. 2004. A Linear Programming Formulation for Global Inference in Natural Language Tasks. *Proc. CONLL 2004*. pp. 1-8
- [8] Ellen Riloff, Charles Schafer, and David Yarowsky. 2002. Inducing Information Extraction Systems for New Languages via Cross-Language Projection. *In COLING 2002:828-834*. Taipei, Taiwan.
- [9] Kiyoshi Sudo, Satoshi Sekine and Ralph Grishman. 2004. Cross-lingual Information Extraction System Evaluation. *In COLING 2004:882-888*. Geneva, Switzerland.
- [10] Nicola Ueffing and Hermann Ney. 2005. Word-Level Confidence Estimation for Machine Translation using Phrase-Based Translation Models. *In HLT/EMNLP 2005:763-770*. Vancouver, Canada.
- [11] Richard Zens and Hermann Ney. 2004. Improvements in Phrase-Based Statistical Machine Translation. *In HLT/NAACL 2004*. New York City, NY, US

## Appendix A: Inference Rules of Using Translation to Improve SEntity Extraction

Terms	
<i>TConstraint</i>	Some constraint that name entities must satisfy in language <i>T</i> . For example, in the setting of <i>S</i> =Chinese and <i>T</i> =English, it includes the capitalization constraint.
<i>CorefMentionNum(i)</i>	the number of name mentions coreferring to <i>SName(i)</i> in <i>SEntity</i>
<i>BestTName(Cache)</i>	the best (most frequent) translation in <i>Cache</i>
<i>FreBestTName(Cache)</i>	the frequency of the best (most frequent) translation in <i>Cache</i>
<i>FreSeBestTName(Cache)</i>	the frequency of the second best (most frequent) translation in <i>Cache</i>
<i>Margin(i, Cache)</i>	the <i>margin</i> (defined in section 4.4) of name <i>SName(i)</i> in <i>Cache</i>
Predicates	
<i>ViolateTConstraint(i)</i>	<i>TName(i)</i> does not satisfy <i>TConstraint</i>
<i>HasBestTran(j, Cache)</i>	<i>SName(j)</i> has translation <i>BestTName(Cache)</i> in <i>Cache</i>
<i>ConflictBoundary(i, j)</i>	<i>SName(i)</i> is consistent with <i>SName(j)</i> at one boundary but not the other
<i>HasFewCorefMentions(i)</i>	$CorefMentionNum(i) < \delta_1$
<i>HasLowConf(i, Cache)</i>	$Margin(i, Cache) < \delta_2$
<i>ShareTranslation(i, j)</i>	$TName(i) = TName(j)$
<i>Adjacent(i, j)</i>	<i>SName(i)</i> and <i>SName(j)</i> are adjacent to each other
<i>EqualConf(SEntity)</i>	$FreBestTName(Inside-S-T-Cache) > \delta_3 \wedge FreSeBestTName(Inside-S-T-Cache) > \delta_4$
<i>Overlap(i, j)</i>	<i>SName(i)</i> and <i>SName(j)</i> overlap in spelling
Rule (1-1): Adjust Mention Identification	
if $(ViolateTConstraint(i) \wedge HasFewCorefMentions(i) \vee HasLowConf(i, Cross-T-S-Cache))$ then Change <i>SName(i)</i> into nominal or delete it	
Rule (1-2): Adjust Isolated Mention Boundary	
for all $j \neq i$ do if $(ViolateTConstraint(i) \wedge HasBestTran(j, Inside-S-T-Cache) \wedge ConflictBoundary(i, j)) \vee$ $(HasBestTran(j, Cross-T-S-Cache) \wedge ConflictBoundary(i, j))$ then Replace <i>SName(i)</i> with <i>SName(j)</i> or split it into <i>SName(j)</i> and another mention	
Rule (1-3): Adjust Adjacent Mention Boundary	
for all $j \neq i$ do if $ShareTranslation(i, j) \wedge Adjacent(i, j)$ then Merge <i>SName(i)</i> and <i>SName(j)</i> into a single mention	
Rule (2): Adjust Entity-level Consistent Source Language Annotation (Coreference Resolution)	
if $EqualConf(SEntity) \wedge \neg Overlap(i, j)$ then Split <i>SEntity</i> into two entities	
Rule (3): Adjust Mention-level Consistent Target Language Annotation (Mention Translation)	
if $\neg HasLowConf(i, Inside-S-T-Cache)$ then Replace <i>TName(i)</i> with <i>BestTName(Inside-S-T-Cache)</i> if $\neg HasLowConf(i, Cross-S-T-Cache)$ then Replace <i>TName(i)</i> with <i>BestTName(Cross-S-T-Cache)</i>	

# Biomedical Term Recognition Using Discriminative Training

Sittichai Jiampojamarn, Grzegorz Kondrak and Colin Cherry  
Department of Computing Science,  
University of Alberta, Edmonton  
Alberta, Canada T6G 2E8  
{sj, kondrak, colinc}@cs.ualberta.ca

## Abstract

We investigate the Perceptron HMM algorithm, an instance of the *averaged perceptron* approach, which incorporates discriminative training into the traditional Hidden Markov Model (HMM) approach. We demonstrate the efficiency of the algorithm by applying it to the biomedical term recognition problem. We show that the Perceptron HMM overcomes the limited expressiveness of the traditional, generative HMMs by incorporating additional, potentially overlapping features. This simple and elegant learning method produces performance that is comparable to the current state-of-the-art, while using only straightforward features derived from the provided training data. Our experiments illustrate the relative value of competing techniques that employ more complex learning algorithms and semantic features constructed from external resources.

## Keywords

discriminative training, averaged perceptron, HMMs, biomedical term recognition, gene tagging, named entity extraction

## 1 Introduction

In recent years, discriminative training has become increasingly popular in natural language processing. Discriminative approaches allow us to incorporate a large number of features without concern for their independence. This gives these learners a significant advantage over more traditional generative techniques. However, some discriminative techniques, such as Conditional Random Fields (CRFs), are complex, difficult to implement, and expensive to train. Is it possible to combine the flexibility of feature independence with the elegance and conceptual simplicity of generative techniques?

In this paper, we investigate the Perceptron HMM algorithm, an instance of the *averaged perceptron* approach proposed by Collins [1]. The perceptron makes it possible to incorporate discriminative training into the traditional Hidden Markov Model (HMM)

approach, and to augment it with potentially overlapping features. The Perceptron HMM uses the Viterbi algorithm with a simple perceptron update to train its feature weights. The Viterbi algorithm finds the best answer based on the current parameters while the perceptron algorithm updates the parameters when errors are made. The updating and decoding processes are iterated over the training data until the system converges.

We demonstrate the efficiency of the Perceptron HMM algorithm by applying it, along with a traditional HMM approach, to a specific problem — biomedical term recognition. We show that Perceptron HMM overcomes the limited expressiveness of the traditional HMM by incorporating additional interdependent features, such as part-of-speech, orthographic patterns, and affixes. Using a relatively small number of features that can be derived directly from the training data, we achieve results that are comparable to the current state-of-the-art systems that utilize external features derived from the Web or semantic knowledge-bases.

In the next section, we define the biomedical term identification task. The related work is discussed in Section 3. In Section 4, we describe a basic HMM approach. In Section 5, we introduce our proposed system based on the Perceptron HMM algorithm. In Section 6, we discuss our feature set. Experimental results and conclusions are given in Sections 7 and 8, respectively.

## 2 Biomedical term recognition

Every day, new scientific articles in the biomedical field are published and made available on-line. The articles contain many new terms and names involving proteins, DNA, RNA, and a wide variety of other substances. Given the large volume of new research articles, it is important to develop systems capable of extracting meaningful relationships between substances from these articles. Such systems need to recognize and identify biomedical terms in unstructured texts. Biomedical term recognition is thus a step toward information extraction from biomedical texts.

```
High-dose growth hormone does not
affect <protein>proinflammatory
cytokine</protein> (<protein>tumor
necrosis factor-alpha</protein>,
<protein>interleukin-6</protein>, and
<protein>interferon-gamma</protein>)
release from activated
<cell_type>peripheral blood mononuclear
cells</cell_type> or after minimal to
moderate surgical stress.
```

**Fig. 1:** An annotated example of a biomedical research article

The term recognition task attempts to locate biomedical terminology in unstructured texts. The texts are unannotated biomedical research publications written in English. Meaningful terms, including proteins, DNA, RNA, cell types and cell line names, are identified in order to facilitate further text mining tasks. The ability to identify important terms that represent biomedical concepts in the text is crucial to understanding research publications.

The biomedical term recognition task can only be adequately addressed with machine-learning methods. A straightforward dictionary look-up method is bound to fail because of the term variations in the text, especially when the task focuses on locating exact term boundaries [8]. Rule-based systems can achieve good performance on small data sets, but the rules must be defined manually by domain experts, and are difficult to adapt to other data sets [4, 3]. On the other hand, systems based on machine-learning employ statistical techniques, and can be easily re-trained on different data.

Biomedical term recognition involves the identification of biomedical terms in documents. The input documents are assumed to be written in English without any additional annotation. The identified terms may comprise several words. We also classify the identified terms into biomedical concepts: proteins, DNA, RNA, cell types, and cell lines. An example of an annotated biomedical research publication from the Genia corpus<sup>1</sup> is shown in Fig. 1, where each identified term is annotated by a pair of XML tags.

Another annotation method, referred to as IOB, is more appropriate for learning. It utilizes three types of tags: <B> for the beginning word of a term, <I> for the remaining words of a term, and <O> for non-term words. For the purpose of term classification, the IOB tags are augmented with the names of the biomedical classes; for example, <B-protein> indicates the first word of a protein term. The total number of IOB tags is thus  $2n + 1$ , where  $n$  is the number of classes.

Our biomedical term recognition task is defined as

<sup>1</sup> The Genia corpus 3.02 is available at: <http://www-tsujii.is.s.u-tokyo.ac.jp/~genia>

follows: for every document in a set, find and mark each occurrence of a biomedical term. A term is considered to be annotated correctly only if all its composite words are annotated correctly. Precision, recall and F-measure are determined by comparing the identified terms against the terms annotated in the gold standard.

### 3 Related work

Apart from early rule-based systems [4, 3], most biomedical term recognition systems employ machine-learning techniques, which have the advantages of scalability and generalization. We can divide machine-learning techniques used for this task into two main approaches: word-based methods, and sequence-based methods.

The word-based methods annotate each word without taking previously assigned tags into account. The ABTA system [5] approaches term annotation as a classification problem on a sliding window of words across sentences. Park *et al.* [11] and Lee *et al.* [9] proposed systems based on Support Vector Machines (SVMs), which classify each word in text as an IOB tag. These systems performed poorly in the Bio-Entity recognition task JNLPBA [7]. However, the SVM approach appears to lead to substantial improvements if used in combination with HMMs [18] or if incorporated in a sequence-based method [10].

The sequence-based methods take other annotation decisions into account in order to decide on the tag for the current word. Zhou and Su [18] employed a combination of the HMM and SVM approaches with rich features, obtaining the best performance at the JNLPBA. The features were word formation patterns, morphological patterns, part-of-speech tag information and dictionaries constructed from Swiss-Prot, LocusLink and annotated terms in the training data. Finkel *et al.* [2] used a large list of words, containing over a million names, to train a model based on the Maximum Entropy Markov Model (MEMM) technique. Words in gazetteers along with biomedical concept class indicators were submitted to the Google API in order to determine biomedical concept classes with the highest number of hits. Conditional Random Fields (CRFs) were used by Settles [13] with orthographic features playing the main role, and biomedical concept classes representing semantic features.

By combining the results submitted by the eight participants in the Bio-Entity recognition task at JNLPBA, Si *et al.* [14] were able to achieve a 0.92 F-measure. Since the submitted results involve only the test data, a portion of them were used to train a CRF model that learned relative weights to be assigned to each system.

In the BioCreAtIvE Task 1A [16], MEMM, CRF

and SVM systems achieved best results. In general, these systems incorporate both internal and external features. The internal features are the ones that can be extracted directly from the training data, and include sets of words, part-of-speech information, orthographic patterns, and sub-string affixes. The external features utilize larger resources such as the world-wide-web, gazetteers, and biomedical dictionaries.

## 4 The basic HMM system

We begin by presenting a traditional first-order HMM, which finds the best sequence of IOB tags  $t_1 t_2 \dots t_n$  for a sequence of words  $w_1 w_2 \dots w_n$ . The HMM involves a number of trained parameters. The initial probability  $\pi_{t_i}$  is the probability of the tag  $t_i$  being the starting tag in the tag sequence. The transition probability  $a_{t_i, t_j} = P(t_j | t_i)$  is the probability of the current tag  $t_j$  given the previous tag  $t_i$ . The emission probability  $b_{t_j, w_j} = P(w_j | t_j)$  is the probability of the word  $w_j$  given the tag  $t_j$ . The add-one smoothing technique is applied to prevent the occurrence of zero probability values.

The initial, transition and emission probabilities are calculated using maximum likelihood statistics from the training data. These probabilities are then used to find the most likely tag sequences in the test data. The probability value of a candidate tag sequence  $t_{1..n}$  given a sequence of words  $w_{1..n}$  is the product of the partial probabilities as shown in Equation 1.

$$P(t_{1..n} | w_{1..n}) = \pi_{t_1} b_{t_1 w_1} \prod_{i=1}^{n-1} a_{t_i, t_{i+1}} b_{t_{i+1}, w_{i+1}} \quad (1)$$

Given a sequence words  $w_1 w_2 \dots w_n$  and the model probabilities, the mostly likely tag sequence can be found by using the Viterbi algorithm [6].

## 5 The Perceptron HMM algorithm

The Perceptron HMM algorithm combines the Viterbi and perceptron algorithms to replace a traditional HMM's conditional probabilities with discriminatively trained parameters. Adapting an HMM for perceptron learning and arbitrary features requires a substantial shift in notation. First of all, given a complete tag sequence  $t$  for a word sequence  $w$ , we define  $\Psi(w, t)$  to be a vector of features describing  $t$  and its interactions with  $w$ . Our learned parameters are also represented by a vector  $\alpha$ , which assigns a weight to each component feature of  $\Psi(w, t)$ . The weight of each feature can be either positive, to indicate evidence that  $t$  is the correct tag sequence for  $w$ , negative to indicate evidence against  $t$ , or zero to indicate no evidence.

Given a useful weight vector  $\alpha$ , we also need a way to find the tag sequence  $t$  with the most evidence. That is, we need to search for:

$$\hat{t} = \arg \max_{t \in T} [\alpha \cdot \Psi(w, t)] \quad (2)$$

where  $T$  is the set of all possible tag sequences. If we formulate our features carefully, the Viterbi algorithm will provide the necessary  $\arg \max$  operator. We will define our  $\Psi(w, t)$  so that it never needs more information than what is available during a first-order Viterbi search:

$$\Psi(w, t) = \sum_{i=1}^{n-1} \psi(w, t_i, t_{i+1}) \quad (3)$$

where  $\psi$  is a feature vector that describes the subset of  $\Psi$ 's features that are relevant to the interactions between an adjacent tag pair and a word sequence.

Now that we have a feature representation for a tag sequence, and a method to find the tag sequence with the most evidence according to  $\alpha$ , our goal in learning  $\alpha$  is clear. We want to find an  $\alpha$  that separates the correct tag sequence from all other possible tag sequences. For every sentence-tag sequence pair  $(w, t)$  in our training set, we require:

$$\forall \bar{t} \in T \setminus t : \alpha \cdot \Psi(w, t) > \alpha \cdot \Psi(w, \bar{t}) \quad (4)$$

It has been shown in [1] that a perceptron algorithm will find a separating  $\alpha$  if it exists. In the case of unseparable data, an averaged perceptron will provide a useful approximation to this separator.

The training algorithm for the Perceptron HMM is sketched in Algorithm 1. In each iteration, for each training example, the perceptron adjusts its weight parameters  $\alpha$  according to the features of its current best guess. The Viterbi algorithm finds the best sequence of tags  $\hat{t}$  for  $w$ , given the current  $\alpha$ . If this  $\hat{t}$  is not the correct tag sentence, then  $\alpha$  is altered slightly to prefer  $\Psi(w, t)$  over  $\Psi(w, \hat{t})$ .

---

### Algorithm 1 The perceptron training algorithm

---

```

1:  $\alpha = \vec{0}$ 
2: for  $K$  iterations over training set do
3:   for all sentence-tag sequence pairs  $(w, t)$  in the training set do
4:      $\hat{t} = \arg \max_{\bar{t} \in T} [\alpha \cdot \Psi(w, \bar{t})]$ 
5:      $\alpha = \alpha + \Psi(w, t) - \Psi(w, \hat{t})$ 
6:   end for
7: end for
8: return  $\alpha$ 

```

---

For example, suppose that in our training data we have the following sentence  $w$  with its correct annotation  $t$ . The current best guess found by our Viterbi algorithm is  $\hat{t}$ :

$w$	IL-2	gene	expression	and
$t$	B-DNA	I-DNA	O	O
$\hat{t}$	B-DNA	I-protein	O	O

If our features consist only of indicators for word-tag pairs and tag bigrams, the weight vector  $\alpha$  is altered as follows:

- Weights corresponding to the features  $(gene, I-DNA)$  and  $(B-DNA, I-DNA)$  are incremented by 1
- Weights corresponding to  $(gene, I-protein)$  and  $(B-DNA, I-protein)$  are decremented by 1.

Term annotation is a complex problem; we are unlikely to find an  $\alpha$  that perfectly separates our training data, no matter how good our features are. In order to compensate for this, instead of returning the final  $\alpha$  as shown in Algorithm 1, we return the average  $\alpha$  over all updates. This averaged perceptron tends to be more effective on unseen data [1].

## 6 The extended feature set

Our feature set is composed entirely of standard, internal features that have been incorporated in many systems [7]. These features can be divided into three broad classes according to how they generalize the training data: by words, characters or part-of-speech. Word features allow the system to remember common annotations for words that occur frequently in the training data. More general character-based features, such as orthography, prefix and suffix features, help the system recognize unseen words by memorizing linguistic patterns. Part-of-speech features provide syntactic information at the sentence level, which allows the system to take advantage of the fact that most terms are noun phrases. An example sequence of words and tags in the training set is shown below. Its corresponding features are shown in Table 1.

word	...	of	E1A-immortalized	cells	...
tag	...	O	B-cell_line	I-cell_line	...
POS	...	IN	CD	NNS	...

The part-of-speech tag features are obtained by using the `Lingua::EN::Tagger`<sup>2</sup>. The orthography features encode the spelling characteristics of a word, such as uppercase letters (U), lowercase letters (L), digits (D), and symbols (S). For example, the orthography feature for the word “E1A-immortalized” has the following value: “U D U S L”. The prefix and suffix features are the  $k$  first and last characters of words. For  $k = 3$ , the prefix and suffix features for the word “E1A-immortalized” have the values “E1A” and “zed”, respectively.

<sup>2</sup> `Lingua-EN-Tagger-0.13` by Aaron Coburn is available at <http://search.cpan.org/~acoburn>

Feature template	Example
<b>Word features &amp; Current tag</b>	
Current word	E1A-immortalized & B-cell_line
Previous word	of & B-cell_line
Next word	cells & B-cell_line
Bigram word	of E1A-immortalized & B-cell_line E1A-immortalized cells & B-cell_line
<b>Part-of-Speech tag features &amp; Current tag</b>	
Current POS	CD & B-cell_line
Previous POS	IN & B-cell_line
Next POS	NNS & B-cell_line
Bigram POS	IN CD & B-cell_line CD NNS & B-cell_line
<b>Orthography features &amp; Current tag</b>	
Current ORTH	U D U S L & B-cell_line
Previous ORTH	L & B-cell_line
Next ORTH	L & B-cell_line
Bigram ORTH	L U D U S L & B-cell_line U D U S L L & B-cell_line
<b>Prefix features &amp; Current tag</b>	
Current PRE	E1A & B-cell_line
Previous PRE	of & B-cell_line
Next PRE	cel & B-cell_line
Bigram PRE	of E1A & B-cell_line E1A cel & B-cell_line
<b>Suffix features &amp; Current tag</b>	
Current SUF	zed & B-cell_line
Previous SUF	of & B-cell_line
Next SUF	lls & B-cell_line
Bigram SUF	of zed & B-cell_line zed lls & B-cell_line

**Table 1:** The feature template and example used in the experiments

## 7 Results and discussions

We evaluated our system on the JNLPBA Bio-Entity recognition task. The training set contains 2,000 Medline abstracts labeled with biomedical classes in the IOB style. Our development set was constructed by randomly selecting 10% of the sentences from the available training set. The number of iterations for training was determined by observing the point where the performance on the held-out set starts to level off. The test set is composed of new 404 Medline abstracts.

The performance of the basic HMM system on the test data is shown in Table 2. Overall, the F-measure performance on the testing data was about 10% lower than on the training data. The highest F-measure was obtained on the protein class. The basic HMM completely fails to identify cell line terms.

Table 3 shows the results of our Perceptron HMM system on all five classes. Notice the impressive improvement over the basic HMM system, which is particularly evident for the terms of type RNA, cell type, and cell line.

Table 4 presents a comparison of our results with the results of eight participants at the JNLPBA shared tasks, which are taken from the task report [7]. The table also includes the basic HMM described in Sec-

Class (# of terms)	Recall	Precision	F-measure
Protein (5,067)	59.33%	58.84%	59.08%
DNA (1,056)	50.76%	53.17%	51.94%
RNA (118)	21.19%	55.56%	30.67%
cell_type (1,921)	49.97%	48.41%	49.18%
cell_line (500)	0.00%	0.00%	0.00%
ALL(8,662)	52.26%	55.57%	53.86%

**Table 2:** The performance of the basic HMM system on the testing set

Class	Recall	Precision	F-measure
protein	76.73 %	66.04 %	70.99 %
DNA	63.54 %	65.53 %	64.52 %
RNA	66.10 %	64.46 %	65.27 %
cell_type	64.65 %	78.56 %	70.93 %
cell_line	53.20 %	51.65 %	52.41 %
ALL	70.94 %	67.32 %	69.08 %

**Table 3:** The performance of the proposed system on the test set with respect to each biomedical concept class

tion 4, and the baseline system provided for the competition, which is based on longest string matching against a list of terms from the training data. The “Ext.” column in Table 4 indicates whether a system includes a use of external resources. The external resources include gazetteers from dictionaries and Gene Ontology, various Word Wide Web (WWW) resources, British National Corpus, MEDLINE corpus, Penn Treebank II corpus, and tags from other gene/protein name taggers.

In terms of F-measure, our system ranks fourth. The performance gap between our system and the best systems in Table 4 can be attributed to the use of external features. When compared against other systems that use only internal features, our system achieves the highest F-measure.

The listed systems stratify into several categories, which should help elucidate the importance of external data. The three systems at the bottom of the list (our basic HMM, [9], [11]) use either sequence-based or discriminative learning, but not both; only the discriminative methods use external data. This shows that the use of an expressive sequence-based method is important in achieving competitive results. Among the next four systems, we have three methods that combine discriminative and sequence learning ([12], [15], and our P-HMM), along with the only generative sequence method to use external data [17]. Finally, the sequence-based discriminative systems that incorporate external data dominate the top of the list. With our approach, we have shown nearly a 3-point improvement in achievable performance when no external information sources are employed, greatly narrowing the gap between data-poor and data-rich features.

The full system uses all features described in Sec-

System	Method	Ext.	F-measure
Zhou and Su [18]	SVM-HMM	Y	72.6 %
Finkel <i>et al.</i> [2]	MEMM	Y	70.1 %
Settles [13]	CRF	Y	69.8 %
<b>Our system</b>	<b>P-HMM</b>	<b>N</b>	<b>69.1 %</b>
Song <i>et al.</i> [15]	SVM-CRF	N	66.3 %
Zhao [17]	HMM	Y	64.8 %
Rössler [12]	SVM-HMM	N	64.0 %
Park <i>et al.</i> [11]	SVM	Y	63.0 %
<b>Basic HMM</b>	<b>HMM</b>	<b>N</b>	<b>53.9 %</b>
Lee <i>et al.</i> [9]	SVM	Y	49.1 %
Baseline	Matching	N	47.7 %

**Table 4:** The performance comparison

Features	Precision	Recall	F-measure
word	64.27	61.85	63.04
word+POS	66.71	60.53	63.47
word+ORTH	65.59	61.97	63.73
word+PRE	61.53	65.31	63.37
word+SUF	64.48	64.75	64.61

**Table 5:** The complete match performance of each included feature on the test set

tion 5: word, part-of-speech tag (POS), orthography (ORTH), prefix (PRE), and suffix (SUF) features. In order to measure the impact of these feature types, we trained several systems using a single feature class along with the basic word features. As one can see, each type of feature contributes very little on its own, increasing F-measure by at most 1.5 points. But together, these features are literally worth more than the sum of their parts, increasing F-measure by 6 points from 63 to 69. These additional features are internal features which can be directly obtained from the training set.

In order to compare the performance between traditional HMM and Perceptron HMM learning objectives, we limited the feature set in the Perceptron HMM to only the current word feature (the first line in Table 1). Thus, both the HMM and the Perceptron HMM have the same feature set, but the Perceptron HMM trains those features discriminatively. While the traditional HMM system achieves a 53.9% F-measure, the Perceptron HMM system achieves an F-measure of 56.9%. This 3-point increase shows the value of discriminative training when all other variables are held constant; performance increases before we even begin to take advantage of the perceptron’s smooth handling of overlapping features.

## 8 Conclusion and future work

We have proposed a new approach to the biomedical term recognition task using the Perceptron HMM algorithm. Our system achieves a 69.1% F-measure with a simple and elegant machine-learning method,

and a relatively small number of features that can be derived directly from the training data. The performance we achieve with this approach is comparable to the current state-of-the-art.

CRFs, SVM-HMMs and Perceptron HMMs are all discriminative training methods that have similar feature representations and learning objectives. Among them, the Perceptron HMM is by far the most straightforward in its implementation. It is our hope that our experiments help illustrate the relative value of the slower CRF and SVM approaches. Along the same lines, we have demonstrated just how far one can advance without having to resort to features mined from the web or semantic knowledge-bases.

Finally, we have provided a detailed comparison of the Perceptron HMM with a traditional HMM with maximum-likelihood parameters. We have illustrated the value of discriminative training, and we have shown that overlapping features allow a giant leap forward in performance while using the same Viterbi algorithm.

## Acknowledgments

We would like to thank Susan Bartlett and other members of the Natural Language Processing research group at University of Alberta for their helpful comments and suggestions. This research was supported by the Natural Sciences and Engineering Research Council of Canada.

## References

- [1] M. Collins. Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.
- [2] J. Finkel, S. Dingare, H. Nguyen, M. Nissim, G. Sinclair, and C. Manning. Exploiting context for biomedical entity recognition: From syntax to the web. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.
- [3] K. Franzen, G. Eriksson, F. Olsson, L. Asker, P. Liden, and J. Coster. Protein names and how to find them. In *International Journal of Medical Informatics special issue on Natural Language Processing in Biomedical Applications*, pages 49–61, 2002.
- [4] K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Toward information extraction: Identifying protein names from biological papers. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 707–718, 1998.
- [5] S. Jiampojamarn, N. Cercone, and V. Keselj. Biological named entity recognition using N-grams and classification methods. In *Proceedings of the Conference Pacific Association for Computational Linguistics (PACLING'05)*, 2005.
- [6] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing*. Prentice Hall, 2000.
- [7] J. Kim, T. Ohta, Y. Tsuruoka, Y. Tateisi, and N. Collier. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, 2004.
- [8] M. Krauthammer and G. Nenadic. Term identification in the biomedical literature. In *Journal of Biomedical Informatics (Special Issue on Named Entity Recognition in Biomedicine)*, volume 37(6), pages 512–526, 2004.
- [9] C. Lee, W. Hou, and H. Chen. Annotating multiple types of biomedical entities: A single word classification approach. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.
- [10] T. Mitsumori, S. Fation, M. Murata, K. Doi, and H. Doi. Gene/protein name recognition based on support vector machine using dictionary as features. In *BMC Bioinformatics 2005, 6(Suppl 1):S8*, 2005.
- [11] K. Park, S. Kim, D. Lee, and H. Rim. Boosting lexical knowledge for biomedical named entity recognition. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.
- [12] M. Rössler. Adapting an NER-system for german to the biomedical domain. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.
- [13] B. Settles. Biomedical named entity recognition using conditional random fields and novel feature sets. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.
- [14] L. Si, T. Kanungo, and X. Huang. Boosting performance of bio-entity recognition by combining results from multiple systems. In *BIOKDD '05: Proceedings of the 5th international workshop on Bioinformatics*, pages 76–83, New York, NY, USA, 2005. ACM Press.
- [15] Y. Song, E. Kim, G. G. Lee, and B. Yi. POSBIOTM-NER in the shared task of BioNLP/NLPBA 2004. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.
- [16] A. Yeh, A. Morgan, M. Colosimo, and L. Hirschman. BioCreAtIvE Task 1A: gene mention finding evaluation. In *BMC Bioinformatics 2005, 6(Suppl 1):S2*, 2005.



- [17] S. Zhao. Name entity recognition in biomedical text using a HMM model. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.
- [18] G. Zhou and J. Su. Exploring deep knowledge resources in biomedical name recognition. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.

# Giving Semantic Structure to Verbs in the Context of VN Collocations

Kate H. Kao                      James M. Lee                      Richard Y. Chang                      Jason S. Chang  
National Tsing Hua University   National Tsing Hua University   National Tsing Hua University   National Tsing Hua University  
101, Section 2, Kuang-Fu Rd.   101, Section 2, Kuang-Fu Rd.   101, Section 2, Kuang-Fu Rd.   101, Section 2, Kuang-Fu Rd.  
Hsinchu, Taiwan 30013        Hsinchu, Taiwan 30013        Hsinchu, Taiwan 30013        Hsinchu, Taiwan 30013  
msgkate@gmail.com            jamesmlee@gmail.com           richtrf@gmail.com            jason.jschang@gmail.com

## Abstract

New computational tools for extracting collocations are a great boon to both language learners and lexicographers alike. In this paper, we use two general similarity measures to organize the extremely numerous collocates that these tools can return into semantically relevant clusters. As it is most relevant to language learners, we focus on V-N pairs and cluster over the verbs. We find that, somewhat unexpectedly, general similarity measures used in conjunction with unsupervised learning can be used to automatically discover verb classes with reasonable performance in the specific context of collocations.

## Keywords

Verb-noun collocations, similarity measures, semantic similarity, semantic organization

## 1. Introduction

The teaching of foreign languages has long favored grammar and memorization of lexical items over learning larger linguistic units. However, several studies have shown the importance in acquisition of collocations, and moreover, they have found specifically that most important is learning the right verbs in verb-noun collocations (Nesselhauf, 2003, Liu, 1999). For example, Liu (2002) found that in a study of English learner's essays from Taiwan, 87% of miscollocations were attributed to the misuse of V-N collocations and of those, 96% were due to the selection of the wrong verb. A simple example will suffice to illustrate: In English, one *writes a check* (also: *write a letter*) while the equivalent Mandarin Chinese is *kai zhi piao* "lit: open a check" (also: *kai men* "open a/the door"). This type of language-specific idiosyncrasy is not encoded in either pedagogical grammars or the lexicon but is of utmost importance to fluent production of a language.

The recognition of the importance of collocations to language learning and the development of tools for their automatic discovery has been an important step towards ameliorating the bias towards teaching atomic lexical units. Applications of automatic extraction of significant collocations such as Word Sketch (Kilgarriff and Tugwell,

2001) or TANGO<sup>1</sup> (Jian, Chang and Chang, 2004) have been created to answer queries of collocation usage.

With corpus sizes rapidly growing and collocation extraction being an inexpensive, high accuracy computation, there is no reason these tools cannot aim for comprehensiveness. Approaching comprehensiveness not only serves the needs of the lexicographer hunting for uncatalogued usages, but also provides a learner with better confidence in the result of not finding a collocation in a set—an unobserved collocation is likely to be incorrect collocation. This objective, however, comes into competition with the need for presenting digestible amounts of information at a time—a system like TANGO can sometimes return over one hundred collocations for a search keyword—quite an intimidating amount. Fortunately, a compromise can be reached between the twin objectives of comprehensiveness and digestible presentation through the exploitation of the structured semantic relationships among verbs.



Figure 1. 126 Instances of verb collocations for the noun "relationship." From TANGO

<sup>1</sup> TANGO 1.0 is funded by a CANDLE project grant from the National Science Council of Taiwan (NSC92-2524-S007-002). For more information on CANDLE, please refer to the website: <http://candle.cs.nthu.edu.tw/>.

Consider the query *balance*. Instead of generating a long list of verbs, a better response would be composed of clusters of verbs inserted into distinct semantic categories such as: (*find, achieve, strike*), (*alter, shift, tilt, tip*), and (*hold, keep, maintain*).

We developed an automatic classification scheme for verbs from a set of verb-noun collocations centered around a common noun. We present an unsupervised learning algorithm and test it using both the similarity measures of Resnik (1995) and Lin (1998). We use a version of Hierarchical Agglomerative Clustering Algorithm (Jain, 1999). The varying number of clusters given the set of verbs and the difficulty in having predefined seeds for clustering made our task a good candidate for using hierarchical clustering. Further, since we observe words and not word senses, we use Resnik and Lin similarity measures in order to simultaneously disambiguate and calculate similarity. We also take additional precautions by considering multiple links where possible. The algorithm has been tested on nine sets of verb-noun collocations extracted from the large monolingual corpus, *British National Corpus* (BNC).

We will first take some time to discuss our view of collocations and idioms, and how metaphors relate to our attempt to cluster verbs, then we will look at work related to semantic classification, and finally we will discuss our methodology and results.

## 2. Collocations, Idioms, and Metaphor

The traditional analysis of collocations and idioms focuses on consumption of language and distinguishes between the two on the basis of compositionality. Idioms, such as “kick the bucket” are thought to be completely non-compositional, since for a reader, there is no way to infer the meaning of the phrase from the meaning of the individual words, while collocations are thought to be mainly compositional, as in “write a check.”

Our view is that the distinguishing feature should be the rigidness of the collocation, that is whether or not other words can take the place of one of the words in the collocation. To a language learner that is looking to produce idiomatically correct language, there may be nothing obvious about even the benign looking example of “write a check.” In the traditional view, because writing a check does not literally point to an act of writing, it may be considered idiomatic. In our view, irrespective of compositionality, once we find that we can also say “write a money order,” we consider it a collocation and group the two together, not as simply a unique lexical item that needs to be remembered separately.

Essentially, we assume that except in the case of true idioms that are rigid, most expressions that admit substitute words will be substitutable in a way such that there is some

semantic relation between them. This is our key organizational principle and motivation in clustering. It is not true that we can confidently predict a collocation by substituting, say, a synonym—which is why we need to extract collocations in the first place—but we can look for productive collocations that do allow some substitution, presumably based on a semantic relation, once we observe a set of collocations.

Additional support for the view that there is a high level of semantic structure in among verbs of a collocation comes from the work of Lakoff and Johnson (1980). In their seminal work, *Metaphors We Live By*, Lakoff and Johnson showed that contrary to common belief, metaphors are pervasive, inconspicuous, yet partially shape and order the logic of our thought and language. Most relevantly, they showed that these metaphors, such as that TIME IS MONEY, are systematic in nature and will produce a set of semantically related usages: wasting time, save you hours, spend your time. This provides us with fertile ground for semantic clustering.

## 3. Collocations and ESL/EFL Learners

The past decade has seen an increasing interest in the studies on collocations. This has been evident not only from a collection of papers introducing different definitions of the term “collocation” (Firth, 1957; Benson, 1985; Nattinger & DeCarrico, 1992; Nation, 2001), but also from the inclusive review of research on collocation teaching and the relation between collocation acquisition and language learning (Lewis, 1997; Hall, 1994).

### 3.1 Collocation Tools

Several print dictionaries, the BBI Dictionary of English Word Combinations, the Oxford Collocation Dictionary for Students of English, and the Explanatory Combinatorial Dictionaries of Contemporary French—to name a few—provide listings and partial classifications of collocates. However, the manual compilation and classification of collocation dictionaries from large corpora is a time consuming and prohibitively cost-intensive procedure, and classification of collocates is often very coarse or incomplete.

### 3.2 Meaning Access Indexing

Some attention has been given to the investigation of the dictionary needs and reference skills of language learners (Scholfield, 1982; Béjoint 1994), and one important cited feature is structure to support user's neurological processes in meaning access.

Tono (1984) suggests that the dictionary layout should be more user-friendly. According to Tono (1992), menus that summarize or subdivide definitions into groups at the beginning of entries in dictionaries would help users with limited reference skills to access the information in the dictionary entries more easily. The Longman Dictionary of

Contemporary English, 3rd edition, has just such a system called "Signposts," as does the Cambridge International Dictionary of English, which created an index called "Guide Word" with provides similar functionality, as well as the Macmillan English Dictionary, which has menus for words with many senses.

<p><b>balance</b> (unclustered) <i>n.</i>  maintain, achieve, shift, tip, hold, lose, upset, alter, strike, find, keep, have, recover, tilt</p> <p><b>balance</b> (clustered) <i>n.</i></p> <ol style="list-style-type: none"> <li>1. tilt, tip</li> <li>2. maintain, hold, keep, have, alter</li> <li>3. find, recover</li> <li>4. achieve, strike, upset</li> <li>5. lose</li> <li>6. shift</li> </ol>
--

Figure 2. A possible, clustered Word Sketch entry

## 4. Method

### 4.1 Unsupervised Learning Algorithm

Again, if a verb collocate for a given noun is a true collocate and not part of an idiomatic expression, then it is often just one lexicalization of a concept and similar verb collocates can easily be found. We can use this to group verb collocates together by concept, giving learners meaning-based access to the collocates. We now formally state the problem we are addressing:

*Problem Statement:* We are given a set of verb collocates  $V_1, V_1, \dots, V_k$  with unknown word senses, for a given noun  $N$  extracted from a corpus of English texts (e.g., British National Corpus). Our goal is to present the list of collocated verbs in conceptual groups so learners can have meaning-based access to the data. For that, we group the verb collocates into a set of clusters,  $C_1 = \{U_{11}, U_{12}, \dots, U_{1k(1)}\}$ ,  $C_2 = \{U_{21}, U_{22}, \dots, U_{2C(2)}\}$ ,  $C_m = \{U_{m1}, U_{m2}, \dots, U_{mC(m)}\}$ .

For the rest of the section we describe our solution to this problem. This agglomerative approach builds a hierarchy of clusters from bottom up. Initially, each verb collocate is put into a cluster by itself. Then we calculate pair-wise cluster similarity based on word similarity. After that, we merge the pairs of nodes sharing the same verb collocates exclusively. Finally, we check the stopping condition. If the condition is satisfied, we stop and output the clusters. Otherwise, iterate the merging process.

1. Each of the collocates  $V_1, V_1, \dots, V_k$ , is put into a cluster by itself  $C_1 = \{V_1\}, C_2 = \{V_2\}, \dots, C_k = \{V_k\}$ .
2. In general, we assume we have the clusters,  $C_1 = \{U_{11}, U_{12}, \dots, U_{1k(1)}\}$ ,  $C_2 = \{U_{21}, U_{22}, \dots, U_{2C(2)}\}$ ,  $C_m =$

$\{U_{m1}, U_{m2}, \dots, U_{mC(m)}\}$ . For all pairs  $(C_a, C_b)$ , we define the similarity as:

$$\text{Sim}(C_a, C_b) = \mathbf{avg}_{i,j} \text{Sim}(U_{ai}, U_{bj})$$

3. Find the pair of closest clusters  $C_a$  and  $C_b$

$$(a,b) = \mathbf{argmax}_{i,j} \text{Sim}(C_i, C_j)$$

4. If  $\text{Sim}(C_i, C_j) < \lambda$ , then stop and output the clusters  $C_1, C_2, \dots, C_m$  Step (5)
5.  $C_a = C_a \cup C_b$  and  $C_b = C_m$
6. Remove  $C_m$  from the list of clusters
7. Go to Step 2

The number of verbs is always relatively small from a computational perspective, so we can afford to use hierarchical clustering, and we can also afford to compute the average of all-links in deciding whether or not to merge a cluster. The assumption is that merging the two closest clusters first will help avoid error in selecting the wrong word sense and that looking at all-links will lessen the risk that two clusters are erroneously merged because an incorrect word sense is selected.

### 4.2 Distance Measure

Although several distance measures could have been used for this task, we choose two that seemed most promising—the similarity measures proposed by Resnik (1995) and Lin (1998). Resnik's similarity measure is based on WordNet's ontology, that is measuring the link distance between two words through their lowest common ancestor. However, Resnik takes into account word frequencies, so that areas of the ontology that are subdivided very finely, such as biological species, do not result in dissimilar words. Lin's similarity measure is a general measure trained on a general corpus using dependency triples that does not rely on WordNet. Since the similarity of two words in WordNet is through a common ancestor, it could be used to provide labels for classes based on the ancestor; however, WordNet is a static resource that has an imperfect, incomplete ontology.

Ciaramita and Johnson (2003) presented a framework for word sense classification, based on the WordNet lexicographer classes, but this is too coarse of a classification. Simple link distance in WordNet is unsuitable for the same reason that Resnik's is—each link can often represent vastly different amounts of granularity.

### 4.3 Evaluation

To evaluate the agreement of the generated clusters with our hand clustered set, we use a recognized standard for evaluating clusters, the Adjusted Rand Index (Hubert and Arabie, 1985). The Rand Index (Rand, 1971) is essentially a 0 to 1 measure that looks at all possible pairs of nodes:

$$\frac{ss + dd}{ss + dd + sd + ds}$$

where *ss* is the number of pairs of nodes in the same cluster in both the standard set and the experimental set, *dd* is the number of pairs of nodes in different clusters in both the standard set and the experimental set, and *sd* and *ds* being the pairs where the two sets differ. The Adjusted Rand Index takes this index and adjusts it such that the expected value of the index given a random clustering is 0 and the maximum remains 1.

$$\frac{rand\_index - E[random]}{1 - E[random]}$$

## 5. Experimental Setup

We selected nine nouns—balance, conclusion, damage, disease, impact, issue, plan, relationship, wing—for their varying level of abstractness and extracted a subset of their respective verb collocates from the BNC. In total, 228 verbs were extracted.

Two raters were instructed to independently cluster each set of verb collocates based on similar usage and meaning. We did not filter the collocations for mistakes or idioms as we assume a collocation extractor would not have the benefit of such a filter either; we allowed judges to create clusters of size 1 with the expectation that idioms could be placed in such clusters.

For our baseline, we used a most frequent heuristic based on WordNet 2.0 lexicographer classes. WordNet organizes word senses into sets of synonyms called *synsets*; e.g., one word sense of each of the verbs {grow, develop, produce, get, acquire} form a synset (Fellbaum, 1998). Each synset is classified in one of several lexicographer classes (15 for verbs). Although each lexicographer class has fairly naturally defined features, classifying VN collocations based on the file class is a multiclass problem with too much ambiguity—word senses may fall under several classes. WordNet does contain a frequency estimate for each word sense, and we choose that word sense and its associated lexicographer class as our label. Our most frequent heuristic model, then, is the clustering of the collocated candidates' based on the most frequent lexicographer labels.

## 6. Results and Discussion

**Table 1. Experimental Results—Adj. Rand Index**

Compared to	Standard	<b>Rater 1</b>	<b>Rater 2</b>

<b>Rater 1</b>	1.00	0.56
<b>Rater 2</b>	0.56	1.00
<b>Most freq</b>	0.00	0.00
<b>LIN</b>	0.35	0.28
<b>RES</b>	0.27	0.21

We find that the most frequent heuristic model performs no better than random. The Adjusted Rand Index shows our result is significant, though without visual inspection, it is difficult to tell just how good the performance is in absolute terms. Because the index is based on pair-wise comparisons of all nodes, the non-inclusion or mistaken inclusion of a node in a cluster can often unexpectedly alter the index value greatly. We find that regardless of which rater we take as standard, Lin's similarity function outperforms Resnik's. For a visual inspection of a sample of the clustered results please see the Appendix. Each machine cluster is aligned with the human cluster with which it shares the greatest number of nodes (verbs). Human clusters are allowed to appear more than once.

## 7. Conclusion

In this paper, we have used an unsupervised method to give semantic organization to collocations. The method described here is a good starting point for a much improved and useful presentation of collocations. Somewhat unintuitively, we have also shown that it is possible to use general WordNet-based similarity measures without knowing the word sense in the specific context of a verb-noun collocations. We speculate that this is related to metaphor and new usages created by analogy—if one can combat the enemy, fight the enemy, and combat inflation, then, by natural extension we can also fight inflation. Thus, if the more common, more primitive word senses of two verbs are selected, their similarity will often hold in more abstract or metaphorical uses as well, so the “wrong” word-sense being selected works out for the best in the end.

We plan to further improve the accuracy of our method and explore non-WordNet based methods for determining semantic similarity. WordNet restricts our method to languages where WordNet or WordNet-like tools are available. WordNet lacks antonymy or polar relations for verbs (“accept a conclusion” and “resist a conclusion”), which ideally would be noted as well. We are currently working on compiling more hand-clustered data and exploring supervised methods in addition to unsupervised methods for classifying verbs.

## 8. Appendix

<b>Noun</b>	<b>Clustering of Verb Collocates by</b>
-------------	---

Seeds (RAND)	Similarity Measure (LIN)	Rater
balance 0.241	<b>tilt, tip</b>	<b>tilt, tip</b> , upset
	<b>maintain, hold, keep, have, alter</b>	<b>maintain, hold, keep</b>
	<b>recover, find</b>	<b>recover</b>
	<b>achieve, strike, upset</b>	<b>achieve, strike, find, have</b>
	<b>shift</b>	<b>shift, alter</b>
	<b>lose</b>	<b>lose</b>
damage 0.480	<b>avoid, escape</b>	<b>avoid, escape</b> , exclude, prevent
	<b>limit, reduce, minimize, prevent</b>	<b>limit, reduce, minimize</b>
	<b>repair, mend</b>	<b>repair, mend</b> , undo, recover
	<b>apportion, award</b>	<b>apportion, award</b>
	<b>do, cause, make, induce, wreak, give, have, sustain, pay, provide, admit, diffuse, prove, assess, inflate</b>	<b>do, cause, make, induce, wreak, inflict, diffuse, give, have, inflate, provide</b>
	<b>inspect</b>	<b>inspect</b> , assess
	exclude, <b>recover, undo</b>	repair, mend, <b>undo, recover</b>
	<b>seek, risk</b>	<b>seek, risk</b>
	<b>inflict</b>	do, cause, make, induce, wreak, <b>inflict</b> , diffuse, give, have, inflate, provide
impact 0.598	<b>limit, minimize, reduce</b>	<b>limit, minimize, reduce</b> , soften
	<b>have, make, soften</b>	<b>have, make, feel</b>
	<b>assess, evaluate, measure</b>	<b>assess, evaluate, measure</b>
	<b>see, examine, investigate, feel</b>	<b>see, examine, investigate</b>
wing 0.781	<b>flap, beat</b>	<b>flap, beat</b>
	<b>raise, lift</b>	<b>raise, lift</b>
	<b>build, make</b>	<b>build, make</b>
	<b>stretch</b>	open, spread, <b>stretch</b>
	<b>open, spread</b>	<b>open, spread, stretch</b>
	<b>fold</b>	<b>fold</b>

## 9. References

- [1] D. S. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, New Jersey, 2000.
- [2] Benson, M. 1985. Collocations and Idioms. In R. Ilson (Ed.), *Dictionaries, Lexicography and Language Learning* (ELT Documents 120; Oxford: Pergamon), pp.61-8.
- [3] Benson, M., E. Benson, and R. Ilson., 1986a. *The BBI Dictionary of English Word Combination*. John Benjamins Publishing Company.
- [4] Béjoint, H. 1994. *Tradition and Innovation in Modern English Dictionaries*. Oxford: Clarendon Press.
- [5] Béjoint, H. 1989. 'The Teaching of Dictionary Use: Present State and Future Tasks,' in F. J. Hausmann et al. (Eds.) *Wörterbücher / Dictionaries / Dictionnaires*. International Encyclopedia of Lexicography. Berlin: W. de Gruyter, pp.208-15.
- [6] Ciaramita, M., T. Hofmann, and M. Johnson. 2003. Hierarchical Semantic Classification: Word Sense Disambiguation with World Knowledge. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, Acapulco, Mexico.
- [7] Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.
- [8] Firth, J.R. 1957. *The Semantics of Linguistics Science*. *Papers in linguistics 1934-1951*. London: Oxford University Press.
- [9] Hall, G. 1994. Review of the *Lexical Approach: The State of ELT and a Way Forward*, by Michael Lewis. *ELT Journal* 44, 48.
- [10] Hubert, L. and P. Arabie. 1985. Comparing Partitions. *Journal of Classification*, 193-218.
- [11] Jian, J. Y., Y. C. Chang, and J. S. Chang. 2004. TANGO: Bilingual Collocational Concordancer, Post & demo in *ACL 2004*, Barcelona.
- [12] Jain, A. K., M. N. Murty, and P. J. Flynn. 1999. Data Clustering: A Review. *ACM Computing Surveys*, 31(3):276-280.
- [13] Kilgarriff, A. 2003a. Thesauruses for natural language processing. In *Proceedings of the 2003 International Conference on Natural Language Processing and Knowledge Engineering*. Beijing: Beijing Media Center.
- [14] Kilgarriff, A. and D. Tugwell. 2001b. "WORD SKETCH: Extraction and Display of Significant Collocations for Lexicography" In *Proceedings of the COLLOCATION: Computational Extraction, Analysis and Exploitation workshop*, 39th ACL and 10th EACL, pp.32-38.
- [15] Koren, S. 1997. 'Quality Versus Convenience: Comparison of Modern Dictionaries from the Researcher's, Teacher's and Learner's Points of View.' *TESL-EJ* 2.3. <http://www.kyoto-su.ac.jp/information/tesl-ej/ej07/a2.html> (13/04/00)
- [16] Lakoff, G., and M. Johnson. 1980. *The Metaphors We Live By*. Chicago: The University of Chicago Press.
- [17] Lewis, M. 1997. *Implementing the Lexical Approach*. Hove, England: Language Teaching Publications.
- [18] Lin, D. 1998. Automatic Retrieval and Clustering of Similar Words. *COLING-ACL*, Montreal, pp.768-774.

- [19] Liu, L. E. 2002. A Corpus-based Lexical Semantic Investigation of Verb-Noun Miscollocations in Taiwan Learners' English. Tamkang University, Taipei.
- [20] Mason, Z. J. 2004. CorMet: A Computational, Corpus-Based Conventional Metaphor Extraction System. *Computational Linguistics*, 30(1).
- [21] Nation, I. S. P. 2001. *Learning Vocabulary in Another Language*. Cambridge: Cambridge Press.
- [22] Nattinger, J.R. and J.S. DeCarrico. 1992. *Lexical Phrases and Language Learning*. Oxford: Oxford University Press.
- [23] Nesselhauf, N. 2003. 'The Use of Collocations by Advanced Learners of English and Some Implications for Teaching,' *Applied Linguistics* 24/2: 223–42.
- [24] Nirenburg, S. and V. Raskin. 1987. The Subworld Concept Lexicon and the Lexicon Management System, *Computational Linguistics*, v. 13.
- [25] Rand, W. M. 1971. Objective Criteria for the evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66, 846-850.
- [26] Resnik, P. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pp. 448-453.
- [27] Scholfield, P. 1982. Using the English dictionary for comprehension. *TESOL Quarterly* 16: 185-194
- [28] Tono, Y. 1984. *On the Dictionary User's Reference Skills*. Unpublished B.Ed. Thesis. Tokyo: Tokyo Gakugei University.
- [29] Tono, Y. 1992. Effect of Menus on EFL Learners' Look-up Processes. *LEXICOS 2 (AFRILEX Series)* Stellenbosch: Buro Van de Watt.
- [30] Wanner, L., B. Bohnet, and M. Giereth. 2006. What is beyond Collocations? Insights from Machine Learning Experiments. *EURALEX*.
- [31] Yarowsky, D. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd Annual Meeting of the ACL*, pages 189-196, Cambridge, MA.

# Enforcing Consistency on Coreference Sets

Manfred Klenner  
Institute of Computational Linguistics  
University of Zurich  
*klenner@cl.uzh.ch*

## Abstract

We show that intra-sentential binding constraints can act as global constraints that - via transitivity - enforce consistency on coreference sets. This yields about 5 % increase in performance. In our model, the probabilities of a baseline classifier for coreference resolution are used as weights in an optimization model. The underlying integer linear programming (ILP) specification is straightforward - binding constraints give rise to (local) exclusiveness of markables, transitivity propagates these restrictions and enforces the re-computation of coreference sets.

## Keywords

Coreference Resolution, Global Constraints, Optimization, Intra-sentential Binding Constraints

## 1 Introduction

In many NLP fields including coreference resolution, approaches are still striving to improve empirical results by a rather traditional (after about twenty years one might use this term) machine learning system design: given some annotated data for the problem to be solved, find appropriate features and train a classifier, e.g. maximum entropy, decision trees or k-nearest neighbor. Actually, these attempts are successful, since there is still room for improvements, for example with respect to coreference resolution through the integration of semantic knowledge from new resources such as Wikipedia [16]. One problem with these approaches, however, is that they can't adhere to - globally operative - prescriptive knowledge. Such strong linguistic principles exist and some of them are never violated even in real scenarios. Take intra-sentential binding constraints as given in the following example:

A [man] stole/sold [him] [his] car.  
[Peter] was angry/happy.

It is known from binding theory that "man" as the subject and "him" as the indirect object of the same (non-predicative) verb have exclusive referents. On the other hand, "his" can be coreferent with either, depending on the verb and world knowledge: "his" refers to "him" in the case of "steal", "his" refers to "man" in the case of "sell"<sup>1</sup>.

Assume a binary classifier that incorporates binding constraints in form of a hard filter. This would

<sup>1</sup> There are also verbs such as "to give" where both resolution alternatives are allowed, given an appropriate context.

prevent exclusive pairs such as (man him) from being generated. Unfortunately, such local decisions do not impose any restrictions on the resolution of subsequent pairs: (man his), (him his), (man peter), (him peter), (his peter). However, only some combinations form a consistent solution. For example, {(man his), (him his)} does not, since, via transitivity of the anaphoric relation, (man him) deductively follows. An inconsistent coreference set, thus, evolves. Transitivity of the anaphoric relation is a constraint that cannot be integrated in a binary classifier since it classifies candidate pairs independently of each other.

The crucial point is that a local constraint (intra-sentential binding constraint) becomes via transitivity of the anaphoric relation globally operative. Most of the time no simple repair mechanism operating on the inconsistent classifier output could do a good job, since these dependencies can get rather complex and often there is no single but multiple consistent solutions. The question is, which is the optimal solution.

We introduce a model of coreference resolution that bases its decisions on the output (probabilities) of a traditional classifier. Our model is formalized within the framework of Integer Linear Programming, a constraint-based numerical optimization algorithm. As long as no inconsistencies arise, the decisions of the baseline classifier are left unaltered. Violations of exclusiveness restrictions as indicated by binding constraints cause a reordering of coreference sets. Our architecture, thus, combines theory-based, apriori linguistic knowledge of the problem at hand with empirically derived preferences.

## 2 Integer Linear Programming

Integer Linear Programming (ILP)[13] is the name of a class of constraint satisfaction algorithms which are restricted to a numerical representation of the problem to be solved. The goal is to optimize the numerical solution, where optimization means maximization or minimization of linear equations. An ILP specification has two parts, an objective function and constraints. The general form of the objective function is:

$$\max : f(X_1, \dots, X_n) := y_1 X_1 + \dots + y_n X_n$$

The general form of the constraints is:

$$a_{i1} X_1 + a_{i2} X_2 + \dots + a_{in} X_n \begin{pmatrix} \leq \\ = \\ \geq \end{pmatrix} b_i,$$

with  $i = 1, \dots, m$



$X_i$  are variables,  $y_i$ ,  $b_i$  and  $a_{ij}$  are constants. The goal is to maximize (or minimize) a  $n$ -ary function  $f$ , which is defined as the sum of  $y_i X_i$ .

ILP as a scheme for global inference in NLP has been introduced by [18] and applied to various NLP tasks, including generation of coherent discourse [1], shallow dependency labeling [7] and semantic role labeling [17].

### 3 Binary ILP Models

Coreference resolution can be modeled as a binary classification task: two markables are or are not coreferent. Given  $n$  markables, numbered according to their occurrence in a text,  $1..n$  (the markable indices), a binary relation  $C_{ij}$  with  $i < j$  represents a classification decision: If  $C_{ij} = 1$ , then the markables with position index  $i$  is the antecedent of markable  $j$ , which is the anaphor. If  $C_{ij} = 0$ , the two markables are not coreferent.  $C_{ij}$  represents a candidate pair. Whether it is (set to) one or zero, depends on a number of constraints (e.g. do they agree) and the strength or weight that such a coupling receives according to an underlying statistical model. We rely on a machine learning (baseline) classifier to fix these weights of a candidate pair. Constraints can be formulated with (in)equalities, e.g.  $C_{ij} \leq C_{ik}$ . This is the ILP equivalent to implication from statement logic. It might be instructive to relate binary ILP modeling to statement logic and find mappings from logical connectives to their counterparts within the ILP framework. Such a bridge could ease the understanding of our formalization and help to evaluate the potential of ILP for NLP. The binary relation  $C_{ij}$  can be reinterpreted as a propositional variable: it can be true or false. Constraints correspond to formulas, i.e. expressions combining propositional variables and logical connectives such as implication or disjunction. The most obvious difference between these two reasoning schemes is that in the case of ILP the inferences are driven by optimization. ILP is - in a sense - model building under the supervision of optimization.

In Fig. 1, we give a mapping from logic formulas to ILP equations ( $X_i$  are binary variables)

1.	$X_1 \vee .. \vee X_n$	$X_1 + .. + X_n = 1$
2.	$X_1 \vee .. \vee X_n$	$X_1 + .. + X_n \geq 1$
3.	$X_1 \wedge .. \wedge X_n$	$X_1 + .. + X_n = n$
4.	$X_1 \rightarrow X_2$	$X_1 \leq X_2$
5.	$X_1 \leftrightarrow X_2$	$X_1 = X_2$

Fig. 1: Statement Logic and ILP

Exclusive OR (cf. line 1 in Fig. 1) requires that exactly one propositional variable is set to one, i.e. the sum of all variables must be one. Logical OR excludes the possibility that all variables are set zero, thus, the sum of all variables must be at least one. AND of  $n$  variables must sum up to  $n$  - setting each variable to one. Implication is false, if the antecedent is true, but the consequent is false. This means that the value of the antecedent is less or equal to the consequent. If the antecedent is one, then the consequent must also be one, otherwise the formula is not fulfilled. Equivalence of two variables means that they must be equal.

## 4 Transitivity

The anaphoric relation is transitive. For three markables  $i, j, k$  to be coreferent it must hold that  $C_{ij} \wedge C_{jk} \rightarrow C_{ik}$ , or, in a simpler notation:  $X_1 \wedge X_2 \rightarrow X_3$ . According to Fig. 1 line 4,  $X_2 \rightarrow X_3$  corresponds to  $X_2 \leq X_3$ . Adding a further antecedent  $X_1$  to the lefthand side of the (in)equality increases the amount at most by 1 (if  $X_1 = 1$ ). Accordingly, we must add the amount of 1 on the righthand side - to keep the balance. So our transitivity statement becomes:  $X_1 + X_2 \leq X_3 + 1$  (cf. also [4]).

The point here is, if  $X_1$  and  $X_2$  are set to one, then  $X_3$  is forced also to be one by ILP. If only one antecedent is set to one, nothing can be deduced with respect to the consequent. And since the sum of the antecedents are restricted to be less (or equal) to the consequent, the value of the consequent has no influence on the values of the antecedents.

Note that  $X_1 \wedge X_2 \rightarrow X_3$  is only one incarnation of transitivity constraints on these three binary relations  $X_1, X_2, X_3$ . Also  $X_3 \wedge X_1 \rightarrow X_2$  and  $X_3 \wedge X_2 \rightarrow X_1$  are valid and must be generated to take full advantage of ILP's reasoning capabilities.

Coming back to our former notational conventions, the full definition of transitivity is:

$$C_{ij} + C_{jk} \leq C_{ik} + 1, \quad \forall i, j, k \ (i < j < k) \quad (1)$$

$$C_{ik} + C_{jk} \leq C_{ij} + 1, \quad \forall i, j, k \ (i < j < k) \quad (2)$$

$$C_{ij} + C_{ik} \leq C_{jk} + 1, \quad \forall i, j, k \ (i < j < k) \quad (3)$$

In our model, transitivity is mainly used to propagate exclusiveness. It is the primary mechanism to enforce consistency on coreference sets. Exclusiveness stems from binding and agreement constraints. We first discuss our ILP model and then come back to transitivity as a mechanism that "globalizes" local exclusiveness constraints.

## 5 ILP's Objective Function

Our ILP model is straightforward<sup>2</sup>. The objective function introduces for each positive (and only for positive) classification decision of the baseline classifier a indicator variable,  $C_{ij}$ , that is weighted by the corresponding probability,  $P_{ij}$ .

The objective function is:

$$\max : \sum_{(i,j) \in \oplus} P_{ij} * C_{ij} \quad (4)$$

$\oplus$  is the set of positively classified pairs,  $P_{ij}$  is the probability of such a pair used as a weight and  $C_{ij}$  is the indicator variable that eventually is set to zero or one. If it is set to one, then ILP has adopted the classification decision of the baseline classifier, otherwise, if set to zero, ILP has revised it. It is important to note that our model relies exclusively on positively classified pairs. This is due to the fact that binary classifiers (including our baseline classifier) are unaware of transitivity: as a consequence, a pair that is transitively implied by two positively classified pairs might - at the

<sup>2</sup> We use `lp_solve`, cf. <http://lpsolve.sourceforge.net/>.

same time and inconsistently- get a negative classification.

To illustrate this: given two positively classified pairs,  $(i, j)$  and  $(j, k)$ . Although  $(i, k)$  transitively follows, the binary classifier often assigns a negative classification to such pairings  $(i, k)$ . This is due to its “global blindness”, but it is - from a local perspective - quite reasonable. Assume a proper name,  $i$ , at the beginning and a pronoun,  $k$ , at the end of a text. Both might be in the same coreference set (via a long chain of intermediary mentions), but there is no a priori or empirical reason that their direct linkage must form a good candidate pair as well. On the contrary, a personal pronoun hardly refers to the same referent throughout a (long) text, it shifts forth and back acting as a local variable .

Thus, positive classifications (of the baseline classifier) are better indicators of coreference than negative ones are indicative of exclusiveness: some of the negative classifications are - from a transitive (global) perspective - contradictory. A model that takes transitivity into account as our model does, must not get confused by flaws inherent to lower level models that ignore transitivity. Therefore, we don’t consider negative classifications.

## 6 Linguistic Constraints

Transitivity is a structural constraint, as it defines how the (truth) values of indicator variables depend on each other. We have already introduced transitivity in section 4. Here we deal with linguistic constraints namely intra-sentential binding constraints. In our ILP model, binding constraints are used for exclusiveness restrictions. In the literature, various variants of the so-called binding theory are being discussed. Often the coindexing of arguments (mostly pronominal or non-pronominal NPs) is restricted by a structural relation over phrase structure trees - the c-command. We give here a simple version of such a binding theory following [5]. Note that there are also definitions of binding constraints in other syntactic frameworks such as dependency grammar (cf. the d-bind command in [20]).

- C1 A reflexive pronoun must be coindexed with a c-commanding argument within the minimal NP or S that contains it.
- C2 A nonreflexive pronoun must not be coindexed with a c-commanding NP within the minimal NP or S that contains it.
- C3 A nonpronominal NP must not be coindexed with a c-commanding NP.

Only [C2] and [C3] define exclusiveness, we therefore discard [C1] from consideration. Moreover, since we can’t rely on perfect parse trees (a statistical parser is being used in our experiments), we do not work with the c-command. Instead, we define a simple predicate, *clause\_bound*, that most of the time correctly captures the restrictive function of the c-command. Two mentions,  $i$  and  $j$ , are *clause bound*, if they occur in the same (sub)clause, none of them being a reflexive or a

possessive pronoun and they don’t form an apposition. There are only 16 cases in our data set where this predicate produces false negatives. Some of these cases are country names reoccurring in the same clause as part of an adjectival phrase (“Russia<sub>i</sub> and Russian<sub>i</sub> people ...”). False negatives might also stem from clauses with predicative verbs (“He<sub>i</sub> is still prime minister<sub>i</sub>”).

- ILP version of [C2] and [C3]

$$C_{ij} = 0, \quad \forall i, j \text{ (clause\_bound}(i, j)) \quad (5)$$

Two markables  $i, j$  that are clause bound (in the sense defined above) are exclusive.

A possessive pronoun is exclusive to all markables in the (base) noun phrase it is contained in (e.g. “[her<sub>i</sub> manager<sub>j</sub>]” with  $i \neq j$ ), but might get coindexed with markables outside of such a local context (“Anne<sub>i</sub> talks to her<sub>i</sub> manager”). We define a predicate *np\_bound* that is true of two markables  $i, j$ , if they occur in the same (base) noun phrase. In general, two markables that *np bind* each other are exclusive. This is captured by the following constraint.

- Exclusiveness in local contexts

$$C_{ij} = 0, \quad \forall i, j \text{ (np\_bound}(i, j)) \quad (6)$$

A structural or technical constraint completes our ILP model: the definition of variables as being binary variables:

- Variables are binary

$$C_{ij} \in \{0, 1\}, \quad \forall i, j \text{ (} i < j \text{)} \quad (7)$$

## 7 Resolving Inconsistencies

Our ILP model accepts the decisions of the binary classifier as long as no inconsistencies are detected. A coreference set is inconsistent, if at least two of its members are exclusive as indicated by a violation of binding constraints.

For every two mentions  $i, j$  occurring in the same clause, the exclusiveness constraints are checked. If a pair is found exclusive, the corresponding ILP indicator variable is set to zero, i.e.  $C_{ij} = 0$ . If these two mentions are not in the same coreference set, nothing happens - although their exclusiveness might serve as an additional restriction, if a reorganization is triggered from other pairs.

On the other hand, an exclusive pair that is part of a (thus inconsistent) coreference set is a starting point for reorganization. All those mentions  $(k, l)$  from such an inconsistent coreference set that do not directly violate an exclusiveness restriction are introduced into the ILP model simply as indicator variables without a predetermined value. Their values get fixed as part of the reorganization process.

Two options are available: remove one of the exclusive mentions from the coreference set (and include it in another one, or even declare it non-anaphoric) or split the coreference set. To illustrate this assume that the output of the binary classifiers gives rise to

reference	corefset1	corefset2	MUC			ECM		
	[m1,m2,m3]	[m4,m5]	P	R	F	P	R	F
classifier	[m1,m2,m3,m4,m5]		3/4	3/3	0.857	3/5	3/5	0.6
ILP	[m2,m3]	[m1,m4,m5]	2/3	2/3	0.667	4/5	4/5	0.8
ILP effect			drop			boost		

Fig. 2: Illustration: MUC score compared to ECM score

two coreference sets,  $\{g, h\}$  and  $\{i, j, k, l\}$ . Each pair (e.g.  $C_{gh}$ ) has an corresponding weight ( $\mathcal{P}_{gh}$ ). Let  $(i, j)$  be an exclusive pair, i.e.  $C_{ij} = 0$ . Removing  $i$  (or  $j$ ) from the inconsistent coreference set produces costs, e.g.  $\mathcal{P}_{ik} + \mathcal{P}_{il}$  no longer contribute to the value of the objective function. If no constraints are violated, the integration of  $i$  into  $\{g, h\}$  gives some profit, namely  $\mathcal{P}_{gi} + \mathcal{P}_{hi}$ . If neither  $i$  nor  $j$  are allowed to be combined with  $\{g, h\}$ , splitting the inconsistent set might be appropriate. Assume  $\{i, k\}$  and  $\{j, l\}$ . The costs here are  $\mathcal{P}_{il}$ ,  $\mathcal{P}_{jk}$  and  $\mathcal{P}_{hk}$ , no profit can be made. Which decision is to be taken depends on the concrete weights and the given constraints. ILP searches for an optimal and at the same time consistent solution.

In order to run an ILP solver, ILP models must be extensionalized. Fortunately, programs (e.g. Zimpl, [8]) exist that compile ILP models. Input is a ILP model written in a certain specification language and some data, output is an instantiation of the model in an executable format. Depending on the amount of data, such an instantiation might result in a vast number of formula statements. Most of the time, ILP comes up very quickly with a solution even if given thousands of particular constraints<sup>3</sup>. However, it is impossible to extensionalize transitivity for longer texts (e.g. whole books). Fortunately, there is a natural division, i.e. splitting texts in paragraphs or chapters. Coreference resolution could then be done incrementally, preserving found solutions by defining segmentation overlaps (i.e. a window moving over the text).

## 8 Empirical Evaluation

As a baseline system we use a reimplementation of the Soon coreference classifier (cf. [19]). The baseline system features and its performance with respect to the MUC-6, MUC-7 and ACE data are described in [16]<sup>4</sup>. The ACE data [12] is used as a corpus.

We report in the following tables the MUC score [21], but we also have measured the system's performance with a metric called ECM-F, introduced by [9] (the Bell tree approach). ECM-F is an acronym for entity-constrained mention F-measure. It first aligns the system entities (i.e. the found coreference sets) with the reference entities (i.e. the gold standard coreference sets). This is done in a way such that the number of common mentions is maximized. However, each system entity is constrained to align with at most one entity from the reference set (and vice versa). ECM is

a very tough metric that has its shortcomings, but it is sensitive to the primary (splitting and reordering) effect of our ILP model. As we will argue, it is better suited than the MUC score.

Consider as an illustration the schematic example from Fig. 2. Here  $[[m1,m2,m3],[m4,m5]]$  is the gold standard (two coreference sets, 5 mentions), the baseline classifier produces, say,  $[[m1,m2,m3,m4,m5]]$  (one coreference set with all 5 mentions in it). Assume that m3 and m4 are exclusive and that ILP generates  $[[m2,m3],[m1,m4,m5]]$ , for example. This would be a reasonable splitting. However, the MUC score prefers the baseline classifier's partition (0.857 F-measure) over the ILP results (0.667). Quite contrary, the ECM metric prefers ILP's solution (0.8 versus 0.6)<sup>5</sup>. As discussed in [2], every metric has its strengths and shortcomings - it sheds light on certain aspects and hide others. It seems reasonable to choose at least one suitable measure for the problem at hand.

Given the MUC score, in our schematic example ILP even reduces the performance of the baseline classifier (cf. ILP effect "drop" from Fig. 2), while with ECM ILP boosts performance ("boost"). Please note that in our real experiments, even the MUC score admits ILP an increase over the baseline classifier, although a considerably smaller one than the one of the more suitable ECM (cf. Fig. 3).

Before we come to discuss our experimental results, we would like to stress another point. Recent work on coreference resolution (not only with the ACE texts but more generally) often works with true mentions (i.e. only markables that are in the gold standard). This is a considerable simplification, since the classification of a markable as being a true mention itself is an error prone task. Performance thus considerably drops, if one returns to a realistic scenario where all markables are to be related. We don't want to criticize the "perfect settings" - the reason why we run our experiments in a realistic setting is not purism<sup>6</sup>. Remember that our model becomes active only in those situations where coreference sets are inconsistent. With perfect data (BNEWS and NWIRE) only a few exclusiveness violations take place (namely 34 violations as compared to 180 given the realistic data). Even with the perfect data some improvements were achieved (2% ECM-F-measure, no improvement according to MUC score). However, our model seems to contribute more to the realistic setting than to the perfect.

Fig. 3 shows the empirical results. We give the MUC scores for each text collection as well as the ECM

<sup>3</sup> Note that our ILP model belongs to the class of binary (or zero/one) problems for which special purpose algorithms exist (cf. [3]).

<sup>4</sup> I would like to thank Simone P. Ponzetto and Michael Strube for supplying me with the results of their Soon reimplementation applied to the ACE texts.

<sup>5</sup> ECM evaluation: 5 true mentions, ILP postulates 5 mentions, corefset1 and corefset2 of ILP and the gold standard align, respectively. 4 mentions are in common, thus recall and precision are 4/5.

<sup>6</sup> Note that with the perfect setting, a very simple strategy that sets all linkages positive, receives (with ACE texts) a very high MUC score (F-measure around 80 %), cf. also [2].

	NWIRE			BNEWS		
	P	R	F	P	R	F
BL <sub>MUC</sub>	46.6	62.6	53.4	55.6	60.9	58.1
ILP <sub>MUC</sub>	56.1	56.5	<b>56.3</b>	63.4	56.1	<b>59.5</b>
BL <sub>ECM</sub>	45.6	50.5	47.0	56.0	44.1	46.8
ILP <sub>ECM</sub>	53.2	54.4	<b>53.3</b>	60.1	44.9	<b>49.3</b>

**Fig. 3:** *MUC score compared to ECM score*

results. 'BL' is the Soon baseline classifier, 'ILP' our ILP model. According to the MUC score, ILP pushes NWIRE results by 2.9% and BNEWS by 1.4%. Note that precision rises significantly but at the same time recall drops. This is due to the bias the MUC score obeys to (as discussed above). With ECM evaluation the situation is better, NWIRE is pushed by 6.3% and BNEWS by 2.5%, so we have an improvement of 4.4% for the whole collection.

With NWIRE, there are 114 direct exclusiveness violations, while in BNEWS there are 66 violations. 'direct' means 'locally observable', the actual number of violations (considering also transitivity) is higher. Instead of counting these cases, we measured the impact of transitivity more directly. Fig. 4 gives the results of this experiment, where we removed transitivity from the ILP model. Thus, ILP selects (and optimizes) new coreference sets only according to the given weights (adhering still to local exclusiveness).

	NWIRE			BNEWS		
	P	R	F	P	R	F
BL <sub>ECM</sub>	45.6	50.5	47.0	56.0	44.1	46.8
ILP <sub>trans</sub>	53.2	54.4	53.3	60.1	44.9	49.3
ILP <sub>no-trans</sub>	50.5	53.0	<b>51.0</b>	59.8	45.5	<b>49.5</b>

**Fig. 4:** *Does transitivity matter (ECM score)?*

With the NWIRE texts (112 violations), transitivity contributes 35% to the improvement (47.0% was the baseline, 53.3% the effect of full ILP and 51.0% has been achieved without transitivity). There is no effect with the BNEWS texts (66 violations). Whether transitivity helps might correlate with the number of violations or even the cardinality of the coreference sets (larger sets might profit from transitivity propagation). Further experiments are necessary to fix this.

## 9 Related Work

ILP as a tool to utilize the output of an underlying classifier to come to a consistent solution has been used so far only in few approaches (e.g. [1], [7], [10]). The architecture of all these systems is very similar (including ours), it more or less follows the design principles introduced in [18].

There is one recent approach to coreference resolution with ILP (cf. [6]). The most striking differences to our approach are: their ILP model does not integrate transitivity, it does not integrate binding constraints and uses perfect ACE data (only true mentions). Moreover, *all* mention pairs are combined and integrated into the objective function, whereas in our model only the positively classified pairs are being

used. The authors discuss two models, the difference between them is the integration of indicator variables for anaphoricity. In our model, all positively classified instances of the baseline classifier are interpreted as anaphoric, so we don't need a separate indicator variable. Given these numerous differences, it is interesting to see that the impact of ILP in both approaches is similar, that is about 5%. However, the reasons for the improvement are quite different: it (mainly) stems from an increase in recall in their approach, while in our approach an increase in precision was obtained. It would be interesting to combine the two models in order to reap the benefits of both settings.

In [22] an np-cluster based approach to coreference resolution (in the biomed domain) is introduced. To integrate a (definite) np into one of the evolving coreference sets, the np is compared to every element of the set. A similarity measure is used to find the best set. Intra-sentential binding constraints could easily be integrated in this approach: the similarity of an np,  $np_1$ , to a coreference set that contains an np,  $np_2$ , where  $np_1$  and  $np_2$  are exclusive according to binding constraints, could be set to zero. However, the selection of a best cluster for each of the two exclusive nps is a local decision. Due the incremental nature of this clustering process, everything depends on the quality of the initial coreference seeds.

This problem is solved in [9]) where candidate coreference sets are being pursued in a (n-) best-first search by constructing a Bell tree (a tree with branches according to the Bell number which quantifies the number of coreference sets, given  $n$  markables). [9] do not integrate binding constraints, nor do they formalize transitivity within their model. The integration of a markable into a coreference set depends on its probability with respect to the so-called active (i.e. last) element of the set. One difference to our model is the representation of coreference sets. In our formalization, coreference sets are given intensionally (transitivity together with constraints), while for the Bell tree they are explicitly maintained, forcing the systems to prune (on performance grounds). Pruning of coreference sets, however, - if based on numerical measures only - is a local decision. In our approach, coreference sets are constructed simultaneously under the regiment of global optimization.

The statistical model described in [11] explicitly deals with the problem of consistency of coreference sets. On deciding whether a candidate pair  $C_{ij}$  actually is coreferent, transitivity is checked for all  $k$  mentions that could be coreferent with either,  $i$  or  $j$  (i.e.  $\langle C_{ij}, C_{ik}, C_{jk} \rangle$  must be consistent  $\forall k$ ). The corresponding graph partition problem is NP-hard (ILP is NP-complete), [11] rely on approximations to cope with exponential complexity. An evaluation was done for proper noun coreference only (MUC, ACE).

Although not concerned with consistency of coreference sets, the global ranking approach (cf. [14]) might serve as a reference point of the performance of our model. In Fig. 5, we give the MUC scores including the baseline (BL) of [15] (Ng & Cardie, 2002) as reported by [14]. While for BNEWS the global ranking approach (GRA) is clearly superior, for NWIRE, ILP is scored best. Integrated as one of the clustering algorithms ranked by global ranker, our ILP model

presumably would boost performance further.

	NWIRE			BNEWS		
	P	R	F	P	R	F
BL	59.9	43.1	50.1	58.6	56.5	57.5
ILP	56.5	56.1	<b>56.3</b>	56.1	63.4	59.5
GRA	60.3	50.1	54.7	67.9	62.2	<b>64.9</b>

Fig. 5: MUC scores: Baseline, ILP, Ranking

## 10 Conclusions

We have introduced a constraint-based model for coreference resolution within the framework of ILP. We have demonstrated that local linguistic constraints (intra-sentential binding constraints that give rise to exclusiveness of mentions) become globally operative via transitivity. Violated restrictions trigger the re-computation of coreference sets, transitivity guarantees consistency. If no violations occur, our model leaves the coreference sets of the baseline classifiers as they are.

We have empirically demonstrated that our system - especially if applied in a realistic setting - can boost the results of a baseline classifier. We believe that a tighter coupling of empirical and theoretical knowledge in such (still numerical, but also normative) models is a step towards better NLP models. Future work will focus on the definition of a full ILP model of coreference. Such a model no longer only corrects inconsistent coreference sets, but autonomously generates coreference sets in the first place. In order for such an approach to work, more and tighter linguistic constraints are necessary.

Acknowledgment. I would like to thank Simone P. Ponzetto and Michael Strube for their support, be it with data or advice.

## References

- [1] E. Althaus, N. Karamanis, and A. Koller. Computing locally coherent discourses. In *Proc. of the ACL*. 2004.
- [2] A. Bagga and B. Baldwin. Algorithms for scoring coreference chains. In *Proc. of the Linguistic Coreference Workshop at the 1st LREC*, pages 563–566. 1998.
- [3] E. Balas. An additive algorithm for solving linear programs with zero-one variables. *Operations Research* 13, 1965.
- [4] R. Barzilay and M. Lapata. Aggregation via set partitioning for natural language generation. In *Proc. of the HLT-NAACL*, pages 359–366. 2006.
- [5] G. Chierchia and S. McConnell-Ginet. *Meaning and Grammar*. Cambridge, MA: MIT Press, 2001.
- [6] P. Denis and J. Baldridge. Global, joint determination of anaphoricity and coreference resolution using integer programming. In *Proc. of the NAACL*. 2007.
- [7] M. Klenner. Shallow dependency labeling. In *Proc. of the ACL, Demo and Poster Proceedings*. 2007.
- [8] T. Koch. Zimpl user guide. <http://citeseer.comp.nus.edu.sg/480956.html>.
- [9] X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proc. of the ACL*, pages 135–142. 2004.
- [10] T. Marciniak and M. Strube. Beyond the pipeline: Discrete optimization in NLP. In *Proc. of the CoNLL*. 2005.
- [11] A. McCallum and B. Wellner. Conditional models of identity uncertainty with application to noun coreference. In *Neural Information Processing Systems (NIPS)*. 2004.
- [12] A. Mitchell, S. Strassel, M. Przybocki, J. Davis, G. Doddington, R. Grishman, A. Meyers, A. Brunstain, L. Ferro, and B. Sundheim. TIDES extraction (ACE 2003 multilingual training data). In *LDC2004T09, Philadelphia, Penn.: Linguistic Data Consortium*. 2003.
- [13] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. New York: Wiley, 1999.
- [14] V. Ng. Machine learning for coreference resolution: From local classification to global ranking. In *Proc. of the ACL*. 2005.
- [15] V. Ng and C. Cardie. Improving machine learning approaches to coreference resolution. In *Proc. of the ACL*. 2002.
- [16] S. P. Ponzetto and M. Strube. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proc. of the HLT-NAACL*, pages 192–199. 2006.
- [17] V. Punyakanok, D. Roth, W. Yih, and D. Zimak. Semantic role labeling via integer linear programming inference. In *Proc. of the COLING*. 2004.
- [18] D. Roth and W. Yih. A linear programming formulation for global inference in natural language tasks. In *Proc. of the CoNLL*. 2004.
- [19] W. Soon, H. Ng, and D. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.
- [20] M. Strube and U. Hahn. ParseTalk about sentence- and text-level anaphora. In *Proc. of EACL*, pages 237–244. 1995.
- [21] M. Vilain, J. Burger, J. Aberdeen, D. Conolly, and L. Hirschman. A model-theoretic coreference scoring scheme. In *Proc. of the 6th MUC*, pages 45–52. 1995.
- [22] X. Yang, J. Su, G. Zhou, and C. L. Tan. An np-cluster approach to coreference resolution. In *Proc. of COLING*. 2004.

# Discovering the Underlying Meanings and Categories of a Name through Semantic and Domain Information

Zornitsa Kozareva, Sonia Vazquez and Andres Montoyo  
DLSI, University of Alicante  
Carretera de San Vicente,S/N  
Alicante, 03080  
*zkozareva,svazquez,montoyo@dlsi.ua.es*

## Abstract

We present a person name disambiguation and fine-grained categorization approach which uses semantic similarity and domain information. In order to discover the underlying meanings of a name, we generate descriptive and discriminative labels which are related to topic signatures. The developed approach is evaluated on 16 ambiguous person names and 10 fine-grained categories. The obtained results show a significant improvement over a baseline.

## Keywords

fine-grained named entity classification, name disambiguation, semantic information, relevant domains.

## 1 Introduction

### 1.1 Background

Proper names play an important role in current NLP applications, therefore the need for specialized NE categories arises. In addition, the fact that the web is growing dynamically and is increasing in coverage influences the growth of web name ambiguity and makes the searching for people, places or organizations potentially very confusing. We focus on the resolution of name disambiguation and fine-grained categorization, which are challenging tasks due to the fact that one and the same name can refer to hundred or thousand of individuals, or the fine-grained category of a name changes over time.

According to [3], although one and the same name belongs to various people, each of these referents appears in distinct contextual characteristics. In order to determine the different mentions of John Smith, they developed a first order contextual representation approach which reaches 84% f-score. Similar disambiguation approach which considered bi-gram co-occurrences was developed by [11].

In order to identify the different fine-grained categories of a name, [6] used information extraction patterns from a set of seed facts. To conduct the fine-grained classification, syntactic features, topic signature information and synonym expansion with WordNet were encoded. According to their study, to improve the 65% performance, more sophisticated features need to be incorporated, a better person name

fine-grained corpus has to be generated and finally the name ambiguity problem has to be tackled a-priori.

Therefore, [8] continued this work by developing an unsupervised approach to name discrimination. According to their approach, words that are seen more often in a pattern obtain more weight. This information is combined with contextual characteristics regarding the age, the date of birth, the name of the wife, son, daughter, as well as associations such as country, company or organization if present. The approach is evaluated on pseudonym repository where people with similar backgrounds such as Tom Cruise and Tom Hanks are conflated.

### 1.2 Motivation and Contribution

Compared to the previously developed NE fine-grained approaches which suffer from low coverage because of the name ambiguity issue, this paper focuses on the a-priori disambiguation of person names and their posterior fine-grained classification. In concrete, the main goals of the research reported in this paper are to:

1. Discover the underlying meaning of multiple NEs that are denoted by the same proper name (disambiguation).
2. Assign categories to the disambiguated person names on the basis of domain information (fine-grained classification).
3. Determine the descriptive and discriminative features of the person names.

According to the distributional hypothesis of [9], words with similar meaning are used in similar contexts. We adapt this hypothesis to discover the underlying meaning of different person names and to cluster the sentences referring to the same individual. To address the lack of global contextual representation in the previous approaches, we developed an approach that captures the contextual and semantic meaning of a text using the relevant domain (RD) resource [12]. In order to establish the global context of the text, we assign to each word its corresponding domains and then we rank the majority domain for the text. Our person name semantic classification is based on the hypothesis that names occurring in the same domain belong to the same fine-grained category. In addition, for each name, we generate descriptive and discriminative labels. With this information, we can establish that a person is a president related to **SPORT** or **POLITICS**. According to our knowledge, none of the previously devel-

oped fine-grained named entity approaches focused on such distinction that a person is a sport president and not a president of a country or political party. In this paper we focus only on the resolution of person names, because they are more challenging and need deeper semantic knowledge derived from the surrounding text [6].

## 2 Approach

### 2.1 NE Disambiguation with Semantic Information

Our NE disambiguation approach is based on Latent Semantic Analysis (LSA)<sup>1</sup> which extracts and infers relations of words in discourse by representing explicitly terms and documents in a rich, high dimensional space, allowing the underlying “latent”, semantic relationships between terms and documents to be exploited. LSA relies on the constituent terms of a document to suggest the document’s semantic content. However, the LSA model views the terms in a document as somewhat unreliable indicators of the concepts contained in the document. It assumes that the variability of word choice partially obscures the semantic structure of the document. By reducing the original dimensionality of the term-by-document space, the underlying, semantic relationships between documents are revealed, and much of the “noise” (differences in word usage, terms that do not help distinguish documents, etc.) is eliminated. LSA statistically analyzes the patterns of word usage across the entire document collection, placing documents with similar word usage patterns near to each other in the term-document space, and allowing semantically-related documents to be closer even though they may not share terms.

Taking into consideration these properties of LSA, we thought that instead of constructing the traditional term-by-document matrix, we can construct a term-by-sentence matrix with which we can find a set of sentences that are semantically related and talk about the same person. The rows of the term-by-sentence matrix correspond to the words of the sentences with ambiguous names, while the columns correspond to the whole sentences. The cells show the number of times a given word occurs in a sentence. For each sentence with ambiguous name  $s_i$ , LSA returns a list of the semantic similarity scores which indicate how good the match between the returned sentence and the target sentence is judged to be.

It is known that LSA treats the words as tokens without making distinction among the different syntactic categories. However, in our approach, we encode the grammatical categories in the following way: “president#n Bush#np welcomes#v the#det guests#n”. This modification improves the performance of LSA for name disambiguation.

### 2.2 Graph-Based Sentence Clustering

Once the semantic similarity scores among all sentences are obtained with LSA, we use this information

to build a new similarity matrix  $S$ . This matrix is given as input to a Pole-Based Overlapping Clustering Algorithm (PoBOC) [5] which searches for poles, constructs a membership matrix of objects<sup>2</sup> to poles, assigns the objects to poles and finally gives a hierarchical organization of the obtained groups. These groups correspond to the underlying meanings<sup>3</sup> behind a name.

[5] defined poles as: given a set of objects  $X = \{s_1, \dots, s_n\}$ , where  $s_i$  is a sentence and a similarity matrix  $S = X \times X$  where the cells correspond to the semantic similarity values obtained by LSA, the pole represents a subset of homogeneous area which appears in a region with uniform density. The poles are constructed on the basis of a similarity graph denoted by  $G_s(X, V)$ . The graph  $G_s$  is defined by the set of vertices  $X$  and the set of edges  $Y$  in a way that  $(s_i, s_j) \in V$  when  $s(s_i, s_j) \geq \max\{\frac{1}{n} \sum_{x_k \in X} s(s_i, s_k), \frac{1}{n} \sum_{x_k \in X} s(s_j, s_k)\}$ .

According to the definition of PoBOC, there is an edge between  $s_i$  and  $s_j$  when the similarity is greater than the average similarity between  $s_i$  and the whole set of objects, and between the average similarity between  $s_j$  and the whole set of objects.

There are many available clustering algorithms, however we have selected PoBOC for our research study because it is shown that the algorithm reaches good results in word clustering and web page categorization [4], besides it does not require complex parameter settings and finds the number of clusters automatically.

### 2.3 NE Categorization with Domain Information

In order to decide whether a name belongs to one fine-grained category or another, we determine the global meaning of the text by estimating the domain of the sentence. In this sense, a pair of sentences sharing the same domain are highly probable to belong to the same fine-grained category.

We use the WordNet Domains (WND)<sup>4</sup> of [7], where each WordNet synset is annotated with at least one domain label selected from about 200 labeled hierarchy. The number of domain labels is related to the polysemy of the word, therefore words with many senses are linked to several domain labels. A very important property of WNDs is that they can relate semantically words of different syntactic categories, for instance MEDICINE groups sense from nouns such as **doctor#1** and **hospital#1**, and verbs such as **operate#7**. Taking advantage of this property, for each word in the text with ambiguous name, we extract the possible domain labels and we rank them by their relevance score. The most probable domain for a text is obtained from the domain label which is seen with the highest frequency in the text.

[7] annotated in a semi-automatic way the WNDs to the synsets in WordNet<sup>5</sup> so that triplets of the synset

<sup>2</sup> in our case the objects are the sentences  $s_i$  with ambiguous names

<sup>3</sup> number of senses per person name

<sup>4</sup> <http://wndomains.itc.it/wordnetdomains.html>

<sup>5</sup> <http://wordnet.princeton.edu/>

<sup>1</sup> [www.informap.com](http://www.informap.com)



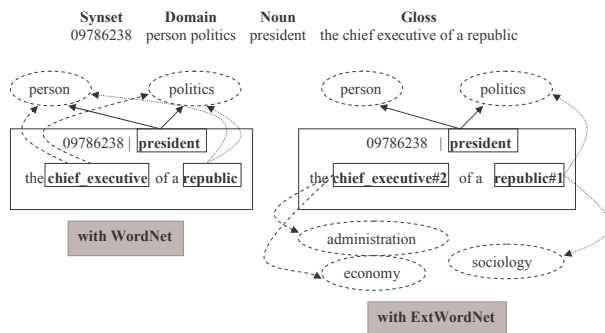


Fig. 1: Comparison of WordNet and ExtWordNet relevant domain annotation

$syn_i$ , word sense with its gloss  $wg_{i,1}, \dots, wg_{i,n}$  and domain labels  $D_{i,k}$ , where  $i$  corresponds to the number of the synset and  $k$  is the number of domains for  $syn_i$  are formed. We use this information to create the repository of relevant domains (RD) by taking  $\forall wg_{i,n} \in syn_i$  and assign their corresponding domain labels  $D_{i,k}$  for  $syn_i$ . At the end of this process, we obtain a list with all  $wg$  of WordNet, associated to various domains  $D$ . This is done, because we consider that  $wg_{i,n} \in syn_i$  are semantically related to  $D_{i,k}$  of  $syn_i$ .

However, we noticed that during the usage of the standard WordNet, we do not have information about the senses of the words in the gloss and this can lead to inaccurate assignment of domain labels as shown in Figure 1. In this example, the synset  $syn_{09786238}$  of the noun **president#1** is defined as the chief executive of a republic. The domain labels for this synset are  $D_{09786238,1} = politics$  and  $D_{09786238,2} = person$ . If we use WordNet, the two words in the gloss  $wg_{09786238,1} = chief\_executive$  and  $wg_{09786238,2} = republic$  relate to  $D_{09786238,1}$  and  $D_{09786238,2}$ . In the RD resource,  $wg_1$  and  $wg_2$  will appear with the domains politics and person. However, through the usage of ExtendedWordNet<sup>6</sup>, we can establish that for the disambiguated word  $\#wg_{09786238,1} = chief\_executive\#2$ , their corresponding domains are  $D_{wg_1,3} = administration$  and  $D_{wg_1,4} = economy$ , while for the second disambiguated word  $\#wg_{09786238,2} = republic\#1$  the domains are  $D_{09786238,1} = politics$  and  $D_{wg_2,6} = sociology$ . The domains for **president#1** are still  $D_{09786238,1}$  and  $D_{09786238,2}$ , the politic domain for **republic#1** remained the same, however the other domains were not assigned during the usage of WordNet. This example shows how the word disambiguation increases the robustness of the RD resource and leads to a more accurate word-domain generation.

The purpose of the creation of the RD resource is not only to generate a word-domain ( $wg - D$ ) list, but also to rank it according to some relevancy score. To indicate the representativeness and the importance of the word-domain pairs, we apply Mutual Information (MI)  $MI(w; D) = \log_2 \frac{Pr(w, D)}{Pr(w)Pr(D)}$  and Association Ratio (AR)  $AR(w; D) = Pr(w, D) \log_2 \frac{Pr(w, D)}{Pr(w)Pr(D)}$  formulae, where  $w$  is the word and  $D$  is the domain.

<sup>6</sup> <http://xwn.hlt.utdallas.edu/>

MI arranges the word-domain pairs according to the most representative domain that corresponds to a word. Representativeness measures how often a word tends to appear in the context of a given domain. However, MI cannot establish the importance of the  $wg - D$  relation, therefore AR is applied. This measure provides a significance score information of the most relevant and common domain of a word. AR is able to capture the words that appear many times in several domains and associate them as non common words. Finally, the  $wg - D$  pairs are arranged and ranked by their AR values with which the creation of the RD resource terminates. The RD resource is applied to the person name categorization by disambiguating all words in the sentence of the ambiguous name and then associating to the disambiguated words their most relevant domains. At the end, we determine the domain probability for the whole text and associate the determined domains as probable name categories.

## 2.4 Descriptive and Discriminative NE labels

With the help of the RD, we establish the global context in which the ambiguous NE appears. With the domain information, we establish that a person is related to MUSIC, however we do not know whether this person is a music composer, singer or dancer. In order to deepen into the fine-grained classification of a name, we create descriptive and discriminative labels.

The descriptive label of a disambiguated name includes the top ten words of the cluster of a name. These words are nouns or verbs and can be shared by other clusters. For the examples in Table 1, Jordi Pujol is the president of Cataluña and his descriptive label is {regeneration, alliance, statement, declaration, coalition, union, chance, nation, government, debate}, González Pujol is a president of a government whose descriptive label is {social security, negotiation, debates, coalition, loyalty, statement, government, minister, politics, statesmen}. These words describe them as politicians and presidents, and meanwhile distinguish them from the Agustin and Carlos Pujols who refer to the SPORT and LITERATURE domains. However, to understand whether we talk about Jordi or González, the discriminative labels are formed. They are built up from the top ten words which are unique for the cluster and are not shared by the other names or clusters.

Once we construct the descriptive and discriminative labels, we match their set of words to the topic signatures [1]. For instance, for Jordi Pujol {regeneration, alliance, statement, declaration, coalition, union, chance, nation, government, debate}, we found that this definition corresponds to "a person with the highest political position, usually the leader of the government". We related the newly obtained information to the already established domains and we classify Jordi Pujolo as a **president** of POLITICS, ADMINISTRATION. The simultaneous assignment of the domains and the topics allows for the distinction between a president of a sport club and a president of a country. To our knowledge, the previously developed fine-grained approaches do not make such fine-grained



Surname	Category	Count	%	Surname	Category	Count	%
Pujol	Agustín: Sport President	30	19%	Martínez	Conchita: Tennis Player	763	54%
	Carlos: Writer	11	7%		Jorge: Rider	306	21%
	González: President	16	10%		Pedro: Driver	218	15%
	Jordi: President of Cataluña	100	64%		Miguel: President of Sp. Council	139	10%
	total	157	100%		total	1426	100%
García	Alan: President of Peru	75	26%	Franco	Carlos: Golf Player	47	13%
	Carlos: Football Player	81	28%		Darío: Football Player	110	30%
	Luis: President of Bolivia	50	17%		Francisco: Spanish State President	172	47%
	Manuel: Parliamentary President	85	29%		Zenon: Chess Player	38	10%
	total	291	100%		total	367	100%

Table 1: NE

distinction and simply assign to a person the president category but do not subcategorize it into sport or politics.

### 3 Experimental Setup

#### 3.1 Data Set

One of the main problems in name disambiguation and fine-grained categorization is related to the evaluation process, because there are no freely available hand-annotated corpora. This is because the creation of annotated corpus is time-consuming and labor intensive process which requires the supervision of specialist.

In order to surmount this obstacle, the researchers in this area [8], [11] build pseudo-name pairs and thus create ambiguity in the data by conflating names that are largely unambiguous. For instance, they took all occurrences of Bill Clinton and all occurrences of Tony Blair and made them ambiguous by replacing them with the single label BC-TB. This pseudo-name creation eases the evaluation process, because the data is ambiguous to the method, but the underlying or pre-conflated name identity is already known. According to the hypothesis of [10], pseudo-word pairs created from words that are individually unambiguous are yet still related in some way.

For this reason, we decided to compile our own disambiguation and fine-grained corpus using pseudo-word name pairs. Several surnames are selected and for each one of the surnames we have found from four to five different individuals who refer to the categories singer, actor, president, politician or football player. Table 1 shows the name distribution and the fine-gained categories we worked with. The percentages shown the number of examples per category.

Further in our experimental work, we considered as a baseline a system which returns the majority sense per surname category. For instance, for the conflated names of Pujol, the majority sense is determined by Jordi Pujol, because it has the highest number of examples. The majority baseline for the surname Pujol is obtained from the normalization of the number of examples corresponding to Jordi divided by the total number of examples for Pujol. The 64% baseline shows the disambiguation performance of a system whose answer is always Jordi Pujol.

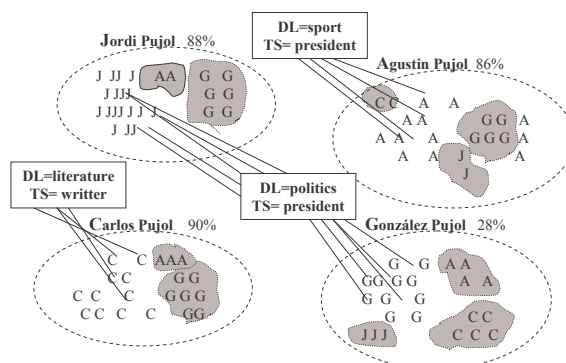


Fig. 2: Fine-grained NE evaluation

The examples for this experimental study are extracted from the EFE 1994–1995 Spanish news corpus. The *xml* tags of the corpora are stripped away, the texts are tokenized, and the boundaries of the document are maintained. On the basis of a rule-based NE recognizer, we extract the names of interest so that each name is surrounded by a context of 100 words.

Once we obtained the necessary data for all surnames, we created the pseudo-word pairs by conflating the examples of Agustín, Carlos, Jordi and Gonzalo Pujol and mingled them together. The names are obfuscated with the label Pujol. During the name disambiguation, the four underlying meanings of Pujol: Agustín, Carlos, Jordi and Gonzalo have to be discovered and in the second stage their fine-grained categories: sport president, writer, president of a country have to be determined. The same process is repeated for the rest of the surname pairs.

#### 3.2 Evaluation Measures

The performance of the NE disambiguation approach is measured in terms of

$$Precision = \frac{|\{relevant\ NEs \in c_i\} \cap \{retrieved\ NEs \in c_i\}|}{|\{retrieved\ NEs \in c_i\}|},$$

$$Recall = \frac{|\{relevant\ NEs \in c_i\} \cap \{retrieved\ NEs \in c_i\}|}{|\{relevant\ NEs \in c_i\}|}$$

and f-score<sup>7</sup> which reflects the harmonic mean of precision and recall.

<sup>7</sup>  $c_i$  refers to cluster  $i$ , where  $i$  stands for the number of different individuals behind a name

The fine-grained evaluation is performed over the results of the name disambiguation. For each cluster, we determine the majority sense as shown in Figure 2 and then we evaluate the precision, recall and f-score per named entity category. For instance, although Jordi, Gonzalez and Agustin Pujol are grouped into three distinct clusters, they all belong to the president fine-grained category. Therefore, we measure the performance for this category considering all sentences that refer to the president. The categorization for Carlos Pujol is evaluated only over the 90% correctly disambiguated examples. The same evaluation is conducted for the rest of the surnames and their corresponding clusters and name categories.

## 4 Experimental Evaluation

### 4.1 NE disambiguation

The results of the NE disambiguation for four different person surnames are shown in Table 2. According to  $z'$  statistics<sup>8</sup>, the f-scores for the complete name disambiguation, as well as for the majority sense disambiguation, outperform significantly the baseline system. The performances of the approach vary from 81% to 90% f-score depending on the type of the surname and its underlying categories. In general, the results for precision are higher than those of recall, which indicates that the majority of the retrieved names in the clusters point out to one individual.

It is interesting to observe that although both surnames Pujol and García include names which refer to president and sport, García obtains better results. This is due to the performance of the clusters of Pujol which is low due to the misclassification of the name González that shares many semantically related words with the names Jordi and Agustín. The easiest name for disambiguation among the four Pujols is Carlos. As a writer his discriminative label contained words related to literature, linguistics, poetry which are quite different from those of the presidents. The sport president Agustín is also well separated from the other two president clusters, as it was seen with words related to sport, tennis, football. The two best discriminable names for the clusters of Martínez are the tennis player Conchita and the President Miguel. However many of the sentences of the rider Jogle were found in the cluster of the driver Pedro and vice versa.

The best discriminable name among all Francos is Francisco. The discriminative label of this Spanish State president contains words which separated him very well from the rest of the names. We decided to conduct an experiment, where we took the nouns and the verbs from the discriminative label of Francisco Franco and searched for news in the BBC repository. It is interesting to see that among the first documents related to such a query stayed documents related to Francisco Franco the Spanish State president<sup>9</sup>. This shows that our semantic similarity disambiguation approach is able to encounter not only the different un-

Surname	Individual	P	R	F
Pujol	Agustín	86.67	86.60	86.67
	Carlos	90.00	91.00	90.95
	González	23.81	31.25	27.03
	Jordi	92.39	85.00	<b>88.54</b>
	Total	80.25	81.82	<b>81.03</b>
Martínez	Conchita	97.84	77.46	<b>86.47</b>
	Jorge	93.44	79.09	85.66
	Pedro	51.64	86.70	64.73
	Miguel	93.08	85.21	88.97
	Total	80.15	84.10	<b>82.08</b>
García	Alan	96.92	84.00	90.00
	Carlos	82.76	88.88	85.71
	Luis	77.96	92.00	84.40
	Manuel	97.47	90.58	<b>93.90</b>
	Total	88.97	88.67	<b>88.81</b>
Franco	Carlos	91.18	65.96	76.54
	Dario	95.372	93.64	94.50
	Francisco	97.14	98.84	<b>97.98</b>
	Zenon	96.96	84.21	90.14
	Total	96.00	91.55	<b>93.72</b>

Table 2: Results for NE disambiguation

derlying meanings of a name, but also to generate discriminative labels characterizing the name which later on can be used to approximate people searches to an individual or pre-clustered names.

### 4.2 NE classification

For each one of the disambiguated NEs, we assign their RDs in order to study the context in which the NE resides. This information is used during the specification of the fine-grained categorization. Table 3 shows the a-priori disambiguated names with their most representative relevant domains, their descriptive labels and the performance of the fine-grained categorization. The descriptive labels are related to the topic signatures which determine the fine-grained category of the name and the evaluation of the categorization is done by considering how many of the derived topic signatures correspond to the NE categories of the names in Table 1.

According to the obtained results, around 80% of the names are correctly categorized and 95% of the times the domains of the names are determined correctly. During the analysis of the obtained results, we found that most of the erroneous classification is related to sentences whose context contains only proper names. It is obvious that it is impossible to associate the correct relevant domain or name category given such examples, because we cannot determine the domain of the proper names. Other errors are produced for texts whose ambiguous names appear in domain different from the one we have expected. For instance, some of the text related to the president of a government included sentences talking about how he attended a football game or opened a new sport hall. The domain for these sentences is not related to ECONOMICS but to SPORT. Therefore, the examples of the president were placed in the sport cluster. Meanwhile, the football player Carlos García had an injury and the context was related to hospital, recuperation, rehabilitation hence the domain of these sentences was MEDICINE and not SPORT. The writer Carlos Pujol talked about his research study in geography and it was difficult to relate it to the domain WRITING.

<sup>8</sup> confidence level of 0.975

<sup>9</sup> the query was performed on 16th of January 2007 and the retrieved page is <http://news.bbc.co.uk/2/hi/europe/5151504.stm>

Name	Domains	Descriptive Labels	%
P_A	tennis, sport, athletics, radio, money, banking, publishing,	federación, oficina, tenis, presidente, equipo, asamblea, jugador, dimisión	83
P_C	literatura, astronomy, publishing, grammar, linguistics	escritor, miembro, edición, época, actualidad, obra, bellas_artes	91
P_G	diplomacy, banking, comerse, politics, economy	cotización, seguridad, negociación, presidente, ponencia, organización	56
P_J	politics, enterprise, engineering, economy, law, industry	alianza, proyecto, declaración, presidente, pluralidad, convergencia, cúpula, comité	88
G_A	diplomacy, politics, sociology, law, banking, industry, commerce, economy	cooperación, construcción, presidente, miembro, comisión, Lima, justicia, campaña, congresista, juez, funcionario	92
G_C	soccer, football, sport, play, athletics, time_period, body_care	centro, fondo, balón, red, area, jugador, temporada, descenso, carrera, marcador, equipo	68
G_L	diplomacy, politics, telecommunication, school, administration, law, money	detención, extradición, seguridad, país, asesinato, prisión, indulto, ejército, delito	87
G_M	doctrines, law, insurance, enterprise, military, banking	labor, juez, investigación, audiencia, tribunal, magistrado, gobernador, ministro	91
M_C	sport, tennis, play, fashion, Money, free.time, table.tennis,	torneo, temporada, clasificación, figura, tennis, jornada, raqueta, semifin, exhibición	82
M_J	money, publishing, athletics, sport, racing, engineering	logro, entrenamiento, posición, equipo, carrera, pieza, podio, vuelta, marca, motociclismo	86
M_P	athletics, sport, racing, insurance, aeronautic, Money	pista, carrera, vuelta, suspension, entrenamientos, velocidad, automovilismo	67
M_M	diplomacy, politics, literature, law, geography, administration	consejo, organismo, presidente, cámara, diputado, conflicto, congreso, palacio, candidatura	88
F_C	golf, play, sport, athletics, time_period, cricket, free_time	desempate, golf, golpe, término, tarjeta, clasificación, hoyo, jornada, golfista, posición, recorrido, torneo	69
F_D	football, sport, athletics, tv, fashion	equipo, selección, país, jugador, peroné, afición, jornada, defensa, estadio, balón, área, meta, disparo, escuadra	87
F_F	banking, diplomacy, economy, law, military, religion	idiosincrasia, comunismo, generalísimo, líder, transición, referéndum, paz, república, levantamiento, extradición, revolución	90
F_Z	chess, number, play, sport, card, free.time	movimiento, ajedrez, pieza, defensa, tabla, jornada, maestro, ataque, rey, torneo, jugador, partida, caballo, peón	89

Table 3: NE fine-grained categorization

## 5 Conclusions and Future Work References

In this paper we have presented a novel approach for the discovery of the underlying meanings and fine-grained categories of ambiguous person names. In particular, we have shown how to separate into meaningful clusters semantically similar sentences that refer to the same fine-grained category or individual. The obtained results are very promising. Our method yielded 86% f-score for the disambiguation of 16 person names and 80% for their fine-grained categorization into 10 categories. The statistical tests show that the obtained results outperform the baseline with 30%.

In order to improve the performance of our approach, in the future we want to incorporate the domain information not only during the NE categorization, but also during the NE disambiguation process. Presently, we have evaluated our name disambiguation and fine-grained categorization approach with name entity examples gathered from static corpora, but we want to expand our approach to automatic web page clustering [2]. We want to conduct queries from the produced discriminative name labels and to gather information for people, organizations, products or locations. Finally, we want to evaluate the contribution of our approach in Question Answering and Information Extraction applications.

## Acknowledgments

We would like to thank Mihai Surdeanu for his useful comments on the paper. This research has been partially funded by the European Union under the project QALLME number FP6 IST-033860 and by the Spanish Ministry of Science and Technology under the project TEXT-MESS number TIN2006-15265-C06-01.

- [1] E. Agirre, E. Alfonseca, and O. Lopez. Approximating hierarchy-based similarity for wordnet nominal synsets using topic signatures. In *Proceedings of the 2nd Global WordNet Conference*, 2004.
- [2] J. Artilles, J. Gonzalo, and F. Verdejo. A testbed for people searching strategies in the www. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 569–570, 2005.
- [3] A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of ACL*, pages 79–85, 1998.
- [4] R. Campos, G. Dias, and C. Nunes. Wise: Hierarchical soft clustering of web page search results based on web content mining techniques. *wi*, 0:301–304, 2006.
- [5] G. Cleuziou, L. Martin, and C. Vrain. Poboc: An overlapping clustering algorithm, application to rule-based classification and textual data. In *ECAI*, pages 440–444, 2004.
- [6] M. Fleischman and E. Hovy. Fine grained classification of named entities. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, 2002.
- [7] B. Magnini and G. Cavaglia. Integrating subject field codes into wordnet. In *Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation*, pages 1413–1418, 2000.
- [8] G. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 33–40, 2003.
- [9] G. Miller and W. Charles. Contextual correlates of semantic similarity. In *Language and Cognitive Processes*, pages 1–28, 1991.
- [10] P. Nakov and M. A. Hearst. Category-based pseudowords. In *HLT-NAACL*, 2003.
- [11] T. Pedersen and A. Kulkarni. Discovering identities in web contexts with unsupervised clustering. In *Proceedings of the IJCAI-2007 Workshop on Analytics for Noisy Unstructured Text Data*, 2007.
- [12] S. Vázquez, A. Montoyo, and G. Rigau. Using relevant domains resource for word sense disambiguation. In *IC-AI*, pages 784–789, 2004.

# Parsing the Medline Corpus

Cornelis H.A. Koster, Marc Seutter and Olaf Seibert  
Computing Science Institute  
Radboud University  
Nijmegen, The Netherlands  
{kees,marcs,olafs}@cs.ru.nl

## Abstract

For the development of PHASAR, an experimental system for literature mining in the BioSciences which uses dependency triples as search terms, we have parsed a snapshot of the Medline collection of biomedical abstracts (18 million short documents, 17 Gbytes of text) using the EP4IR dependency parser of English. The resulting dependency trees were un-nested into triples and indices from words and triples to documents were constructed.

We describe the linguistic resources, the parsing technique used (best-only top-down chart parsing) and the un-nesting and indexation processes. We describe the parsing and indexation process and show the results of some performance measurements.

## Keywords

Natural language parsing, Best-Only parsing, weighted attributed grammars, text mining.

## 1 Introduction

There is a growing demand, coming from natural language based applications in Information Retrieval and Text Mining, for fast and accurate parsers for natural languages. In their overview article on Text Mining, [Shatkey and Feldman, 2003] stated that

Efficient and accurate parsing of unrestricted text is not within the reach of current techniques. Standard algorithms are too expensive to use on very large corpora and are not robust enough.

The present paper shows that this statement is no longer appropriate - deep parsers are at least fast enough for serious applications.

Text Mining (TM) has been defined in [Hearst, 1999] as

the combined, automated process of analyzing unstructured natural language text in order to discover information and knowledge that are typically difficult to retrieve.

An example to clarify the difference with traditional retrieval, including most Question Answering approaches: if you search Google for Aspirin, with the intention to look for side effects, the first hit will satisfy you. Text mining should be capable of giving you

insight in the other 16.7 million hits, including probably the papers in which these side effects and others were first described.

The state-of-the-art in biomedical Text Mining is characterized by the use of thesauri and co-occurrence at the document level (e.g. the [IKNOW search engine] and the CoPub Mapper [Alako et al, 2005]). Research in TM is mostly aimed at Information Extraction using NLP techniques and the use of Classification techniques (e.g. in the presentation of results and for “fingerprinting”).

Co-occurrence techniques work well enough for Medline abstracts, in which the mention of two words or concepts in one abstract usually points at some relation between the two, but it will not work so well on longer full-text articles. In such a wider context, the precise analysis of phrases becomes more important. Shallow parsing will no longer be enough to identify the important noun phrases in a document and the relations between them – in the longer term, even discourse structure will have to be taken into account. Therefore, in TM the demand for accurate (and fast) deep parsing is on the rise.

In the mean time, NLP-based search engines over a parsed version of the Medline abstracts are starting to appear, such as MEDIE [Matsuzaki et al, 2007] and the PHASAR literature mining system [Koster et al., 2006]. Their performance depends critically on the accuracy and speed of the parser used.

In this paper we give some highlights of the EP4IR parser (section 2), briefly describe the Best-Only parsing strategy (section 3), describe the parsing and indexation process of Medline and show some results (section 4).

## 2 The EP4IR parser of English

EP4IR (English Phrases for IR) is a dependency grammar of English, developed specifically for NLP-based Information Retrieval applications in the course of the EC/IST projects DORO and PEKING. The EP4IR parser is generated automatically from the EP4IR (English Phrases for IR) grammar and lexicon, using the AGFL parser generator system <sup>1</sup>.

The grammar is rule-based, written in the AGFL formalism. The main body of the grammar consists of 588 productions (including 58 for lexical robustness)

<sup>1</sup> <http://www.cs.ru/agfl/>

and 15 affix rules. The associated lexicon consists of two components:

1. a *general lexicon* of 238004 single-word entries and 79949 multi-word collocations (found in Wordnet or extracted from Patent texts)
2. a *domain lexicon* consisting of another 87852 single word entries and 210554 collocations, found in the UMLS thesaurus.

In the next sections, we describe the AGFL formalism in which the grammar is written and the particular form of dependency grammar used. Then we describe the transduction of dependency trees from running text and the syntactic normalizations incorporated in this transduction.

## 2.1 The AGFL formalism

Affix Grammars over a Finite Lattice (AGFL) are a form of CF grammars extended with features. Just like in PROLOG with DCGs, these are passed as parameters to the rules of the grammar:

```
noun group (NUMBER):
  adjective, noun group (NUMBER);
  subst (NUMBER).
```

The domain of every affix is described by a *meta rule* as a (finite) set of terminal affixes. Two typical meta rules:

```
NUMBER :: sing | plur.
PERSON :: first | second | third.
```

Affixes obey the *consistent substitution rule*, i.e. in rewriting all occurrences of a certain affix in a rule obtain the same value. In AGFL, affixes are *set-valued*, i.e. an affix variable can take as value any subset of its terminal productions except the empty set. The possible values of an affix variable form a finite lattice (hence the name of the formalism).

The *top*-element  $\top$  of the lattice can be seen as the union of all possibilities (the value may be *first* or *second* or *third* or any combination). As we obtain more information, the number of possibilities may be narrowed down to a particular value, or even further to the *bottom*-element  $\perp$ , which indicates inconsistency. We denote the top-element by the affix PERSON itself.

An affix may obtain a value by means of an (implicit or explicit) *guard*. A guard is an operation which restricts the set of values the affix may take. For example:  $\{X:Y\}$  restricts both  $X$  and  $Y$  to the intersection value  $X \cap Y$  (*set unification*). Guards may be implicit at parameter positions: a call  $p(Y)$  with  $Y \subset X$  can be seen as a shorthand for  $p(X), \{X:Y\}$ .

The following is a complete AGFL for a tiny fragment of English.

```
NUMBER :: sing | plur.
PERSON :: first | second | third.
sentence:
  pers pron(NUMBER,PERSON),
  to be(NUMBER,PERSON), adjective.
```

```
pers pron(sing,first): "I".
pers pron(NUMBER,second): "you".
pers pron(sing,third): "he"; "she"; "it".
```

```
pers pron(plur,first): "we".
pers pron(plur,third): "they".
```

```
to be(sing,first): "am".
to be(NUMBER,second): "are".
to be(sing,third): "is".
to be(plur,first | third): "are".
```

```
adjective: "great"; "small".
```

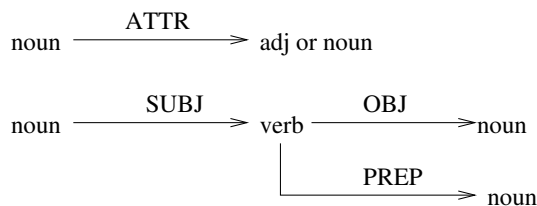
Pre-terminal rules like the above are usually stored in the lexicon.

The parser generated from this grammar will parse the sentence *you are great* with  $NUMBER=\{sing, plur\}$  and  $PERSON=\{second\}$ . The input *I are great* will not be recognized due to the fact that it is impossible to give PERSON a value satisfying the consistent substitution rule.

## 2.2 Dependency Trees and Dependency Triples

By a *dependency tree* [Melčuk, 1988] we mean a graph (a tree with possibly additional confluent arcs) whose nodes are marked with words and whose arcs are marked with directed syntactic relations.

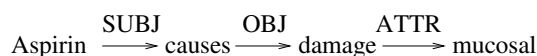
The following dependency tree shows the typical structure of the attributed noun and of the SVOC-sentence in English (the arrows go from Head to Modifier).



A dependency tree represents the main structure of a sentence in an abstract way, much more abstract than a constituent tree (parse tree), in terms of syntactic word relations from which semantic relations can be derived.

By a *dependency triple* we mean a triple [word,relation,word], which forms part of a dependency tree, from which it can be obtained by *unnesting* the tree to the triples contained in it. Triples are a compact notation for the syntactic structure of phrases. They are closely related to the *Head/Modifier pairs* which have been used as terms in Information Retrieval by many researchers [Fagan, 1988, Lewis, 1992, Strzalkowski, 1995] and to the Index Expressions of [Grootjen and van der Weide, 2004]. Table 1 shows the most important dependency relations.

As an example, the sentence *Aspirin causes mucosal damage* corresponds (after lemmatization) to the dependency tree

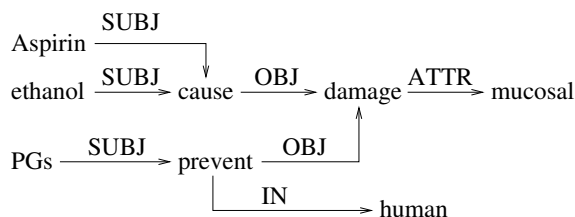


The more complicated example *In humans, PGs prevent the mucosal damage caused by aspirin and ethanol*, consisting of three sentences intertwined, is transduced to

<i>relation</i>	concrete notation	<i>example</i>
subject relation	[noun,SUBJ,verb]	[picture,SUBJ,show]
object relation	[verb,OBJ,noun]	[show,OBJ,view]
predicate relation	[noun,PRED,noun]	[Bush,PRED,president]
attribute relation	[noun,ATTR,noun]	[theatre,ATTR,movie]
attribute relation	[noun,ATTR,adj]	[monument,ATTR,large]
prepos relation	[noun,PREP,noun]	[president,of,United States]
prepos relation	[verb,PREP,noun]	[sit,on,chair]
prepos relation	[adj,PREP,noun]	[dark,from,age]
modifier relation	[verb,MOD,adverb]	[destroy,MOD,not]

**Table 1:** *The major dependency relations*

the following dependency "tree" (after transforming one of the sentences from passive to active, see 2.4):



The parsing process takes into account the subcategorization information of verbs, nouns and adjectives supplied by the EP4IR lexicon.

## 2.3 Transduction

Every alternative of an AGFL grammar can be accompanied by a list of transduction elements: nonterminals or affixes from that alternative or texts between quotes. Together, these recursively specify a compositional transduction for every construct described in the grammar. As an example, the rule

```

noun group(NUMB,PERS,CASE) :
  adjective, noun group(NUMB,PERS,CASE)/
  "[" , noun group, " , ATTR, " , adjective, " ]";
perspron(NUMB, PERS, CASE);
noun part(NUMB,CASE), {PERS :: third}.

```

indicates that a noun group (with affixes NUMBER, PERSON and CASE) may be realized by a personal pronoun with the same number, person and case, or by a noun part (in which its PERSON is third), preceded (right-recursively) by zero or more adjectives.

## 2.4 Normalization

The probability of finding a phrase consisting of many words repeated elsewhere is small, because language allows us to express the same meaning in many different ways. In order to gain recall, we therefore do our best to map different formulations of the same phrase onto one same representative form. This *syntactical normalization* is expressed in the grammar itself, using again the transduction facility.

- Words which are unimportant for the aboutness of the text are elided during transduction: articles, quantifiers, auxiliary verbs and connectors – which is much like applying a stop list.

- SVOC sentences contain embedded constructions like relative clauses and participial constructions from which additional SVOC sentences can be derived – albeit sometimes lacking an explicit subject or object. An example: The doctor came in, stinking of gin, and proceeded to lie on the table (unnested and lemmatized)

```

[doctor,SUBJ,came in]
[doctor,SUBJ,stink]
[stink,of,gin]
[doctor,SUBJ,proceed]
[doctor,SUBJ,lie]
[lie,on,table]

```

- one of the most important normalizing transformations is *de-passivation*: transforming a passive sentence into an active sentence with the same aboutness. For this transformation, the sentence *the damage was caused by Aspirin* is considered equivalent to *Aspirin caused damage*.

After the parsing/transduction and unnesting, lemmatization is applied to the words and collocations occurring in the resulting triples. As an example, the triple [model,SUBJ,stand] may be obtained both from both a model was standing at the window and the model stands at the window.

## 3 Best-Only Parsing

Parsing for NLP applications poses different requirements than parsing for linguistic research. Research may require an enumeration of all possible parse trees of a sentence, but applications in IR require only a single best analysis. What is "best" depends ideally on the semantics of the sentence, but there is no (useful) way to do this automatically. A second best is to take the most probable one, given some probability distribution of sentences. This what motivates most forms of probabilistic parsing.

Linguistic parsing mostly deals with complete sentences, whereas in IR all recognizable parts must be found in a stream of text. Unknown words, irregular punctuation and strange constructions will disturb the parsing. For syntactic robustness, we adopt the *segment parsing* approach: From left to right the parser attempts to recognize the longest parsable sequence of words (segment), and then begins a new segment. Unrecognizable words are skipped.

As the best analysis, the parser takes *the longest segment with the lowest penalty level*.



### 3.1 Penalties

In AGFL there are two mechanisms to attribute a price to a parse tree:

- *penalties* written into the grammar by the grammar writer, to distinguish between preferred and non-preferred constructions. Penalties act as *tie breakers* (in case more analyses are possible, the one with the lowest penalty is preferred) but they also have an interpretation as a crude approximation to the negative logarithm of a syntactic probability. Penalties are used to regulate verb subcategorization and to achieve syntactic and lexical robustness.
- *lexical probabilities* derived from counts in tagged corpora. As an example the lexicon entries

```
"time"    NOUN(sing)      3509
"time"    VERBI(none, trav) 59
```

indicate that the word “time” is about 60 times more likely to occur as a noun than as a verb form. The lexical probability  $P(POS|word)$  is converted into a penalty (as a negative 10-log), so that the parser has to deal only with integer valued penalties.

Because of the penalty mechanism, AGFL is a form of *weighted attribute grammar*.

### 3.2 The Best-Only heuristic

*Best-Only Parsing* (BOP) can be characterized as Top-Down chart parsing with Best-Only memoization for weighted attribute grammars [Jones, 2000]. We describe BOP very briefly, a more complete description exceeds the scope of this paper.

The chart of the parser is realized as a *memo function*. The first call of a certain nonterminal at a certain position of the input will succeed zero or more times, with as its result a set of tuples:

1. the final position reached
2. the resulting parameter values
3. the parse tree and
4. the price (in penalties).

The following calls of this nonterminal at this position will be satisfied from the memo (once for every result).

```
memo : pos × nonterminal ↦ Bool ×
      {< pos, parameters, tree, price >}
```

The Boolean indicates whether this memo position is still empty. Each parameter obtains the largest set of values satisfying the restrictions caused by the guards involved in the call (a Most General Unifier). Notice that the resulting tree is not a string but a functional object, since the values of the parameters occurring in it may still be narrowed by other calls.

Only the best parse (the one with the smallest price) is memoized for each *distinguishable* call in the memo, where two calls are indistinguishable if they concern the same non-terminal with the same parameter set and the same initial and final position.

This heuristic greatly reduces the number of non-terminal calls to be tried. The same memo is used to implement *left-recursion*.

Generating a Best Only parser takes a matter of seconds for an attributed grammar with 400 productions and a lexicon of 200 000 entries. BO parsers make efficient use of memory ( $O(m.n^2)$  with a low constant factor,  $n$  = input size,  $m$  = number of nonterminals). These properties make BO parsing also ideal for quick prototyping, for the development of large grammars, and for use on small machines.

## 4 Some results

In this section we describe the lexical coverage, speed and accuracy of the EP4IR parser.

### 4.1 Lexical coverage

The coverage of the various lexica was tested on one file (*medline06n0233.xml*) chosen arbitrarily among the 839 files of Medline 2006. Its size is 94457624 bytes including XML and 12661897 bytes after XML removal.

First we counted the words covered by the UMLS thesaurus, then among the remaining words those present in our general lexicon and finally the words found only by lexical robustness rules. Those not found at all (after skipping of numbers and special characters) are counted as unrecognizable. In a table:

485582	words covered by UMLS	25.9 %
1337695	added by general lexicon	71.5%
17441	robustly recognized	0.9%
30428	unrecognized words	1.6%
1871146	total words	

The total lexical coverage was 98.4 percent. The unrecognized words contain, besides typos, a large number of formulae and non-standard names not occurring in the UMLS thesaurus.

### 4.2 Parsing speed

The following table shows the parsing speed on a 2.2 Ghz machine for the same Medline file as above, and for a 141486-word sample of the WSJ-corpus.

corpus	#words	time	words/sec
Medline file	1828141	12 m 18 s	2477
WSJ sample	141486	59 sec	2399

Unquestionably, a parsing speed of 2400 words per second is enough for serious applications.

### 4.3 Overall performance

A snapshot of Medline containing all abstracts which were on-line in February 2007 (more than 18 million documents, 17 Gbytes of text after XML removal) has been parsed using the EP4IR parser. The occurrences of all terms (2 G words, 794 M triples) were indexed using the indexing system MRS [Hekkelman and Vriend, 2005].

The analysis process consists of the following steps:

- removal of XML, retaining only the bodies of the ArticleTitle and AbstractText fields
- splitting the text into sentences

- robust lexicalization, including tokenization and lexicon-based Named Entity Recognition
- robust syntax analysis
- syntactic normalization
- transduction to dependency trees
- unnesting to dependency triples
- lemmatization of all words in the triples.

During the analysis, indices are constructed from 7.2 M different lemmatized words, 121 M triples and 196000 UMLS concepts to the set of documents in which they occur. The total size of the combined indices is 77 GB, of which 2.2 MB for the words and concepts index and 30 GB for the various triple indices; 45 GB is used for storing the (compressed) documents.

The whole analysis and indexation process took 1294 CPU-hours on the LISA parallel computer system of SARA in Amsterdam<sup>2</sup>, spread over up to 100 processors. Most of this time was taken up by the indexation processes.

#### 4.4 Accuracy

The accuracy of the current EP4IR parser has not yet been seriously measured, but previous measurements on sentences from Medline showed around 65-70 percent accuracy (computed as precision and recall in terms of the dependency triples generated, see [Lin, 1995]). This is well below the 80% reached by the best parsers at this time (see [Clegg and Shepherd, 1996]).

Medline is not easy to parse correctly using a rule-based parser: sentences are long and tortuous, with often totally unclear attachment of Preposition Phrases and relative phrases. Long combinations of nouns are preferred, requiring a reliable lexicon of collocations for disambiguation.

Our main source of domain collocations, the UMLS thesaurus, is very unhelpful because it does not provide Part-Of-Speech information and is polluted with many linguistically impossible entries; a more suitable thesaurus is still under construction.

Although the parser is adequate for the present experimental stage of PHASAR, we hope to improve its accuracy drastically by introducing hybrid parsing based on the triples extracted from Medline.

## 5 Conclusion

We have described the EP4IR grammar of English and its lexicon. Based on syntactic penalties and lexical frequencies, it is a weighted attribute grammar, defining a compositional transduction from text segments to normalized dependency trees. Due to the use of Top-Down chart parsing with Best-Only memoization, The EP4IR parser generated by the AGFL system is very fast (about 2400 words/second), fast enough to parse the whole medline corpus. Its accuracy is good enough for our experimental Text Mining system, but we are working on its improvement using hybrid parsing.

<sup>2</sup> [www.sara.nl](http://www.sara.nl)

The EP4IR parser, which is available under the GPL license [Koster and Verbruggen, 2002], forms a free resource for full-text mining and other applications of deep parsing.

## References

- [Alako et al, 2005] Blaise T. Alako, A. Veldhoven, S. van Baal, R. Jelier, St. Verhoeven, T. Rullmann, J. Polman and G. Jenster (2005), CoPub Mapper - A Text Mining Tool Based on Co-Publication. [www.biomedcentral.com/1471-2105/6/51/abstract](http://www.biomedcentral.com/1471-2105/6/51/abstract)
- [Clegg and Shepherd, 1996] A.B. Clegg and A.J. Shepherd (2007), Benchmarking natural-language parsers for biological applications using dependency graphs, *BMC Bioinformatics* 2007,8:24.
- [Fagan, 1988] J.L. Fagan (1988), *Experiments in automatic phrase indexing for document retrieval: a comparison of syntactic and non-syntactic methods*, PhD Thesis, Cornell University.
- [Grootjen and van der Weide, 2004] F. A. Grootjen and T. P. van der Weide (2004), Effectiveness of Index Expressions. *NLDB 2004*, Springer LNCS 3136 pp. 171-181.
- [Hearst, 1999] M.A. Hearst (1999), Untangling text data mining. *Proc. 37th Annual Meeting of the Association for Computational Linguistics*, 3-10.
- [Hekkelman and Vriend, 2005] M.L. Hekkelman and G. Vriend (2005), MRS: A fast and compact retrieval system for biological data. *Nucleic Acids Res.* 2005 July 1; 33(Web Server issue): W766W769. Also <http://mrs.cmbi.ru.nl/>.
- [IKNOW search engine] <http://www.iknow.be/>
- [Jones, 2000] P. A. Jones (2000), *Best First Search and Document Processing Applications*, Dissertation University of Nijmegen, The Netherlands.
- [Koster and Verbruggen, 2002] C.H.A. Koster and E. Verbruggen (2002), The AGFL Grammar Work Lab, *Proceedings FREENIX/Usenix*, pp 13-18.
- [Koster et al., 2006] C.H.A. Koster, O. Seibert and M. Seutter (2006), The PHASAR Search Engine. *Proceedings NLDB 2006*, Springer LNCS 3999, pp 141-152.
- [Lewis, 1992] Lewis, D.D. (1992), An evaluation of phrasal and clustered representations on a text categorization task. *Proceedings ACM SIGIR'92*.
- [Lin, 1995] D. Lin (1995), A dependency-based method for evaluating broad-coverage parsers. *Proceedings IJCAI-95*, pp. 1420-1425.
- [Matsuzaki et al, 2007] Matsuzaki, Takuya, Yusuke Miyao and Jun'ichi Tsujii. Efficient HPSG Parsing with Supertagging and CFG-filtering. *Proceedings IJCAI '07*, 1671-1676.
- [Melčuk, 1988] I. A. Melčuk. *Dependency Syntax: Theory and Practice*. State University of New York Press, Albany, NY, 1988.
- [Shatkay and Feldman, 2003] H. Shatkay and R. Feldman (2003), Mining the Biomedical Literature in the Genomic Era: An Overview. *Journal of computational biology*, 10,6, 821-855.
- [Strzalkowski, 1995] T. Strzalkowski (1995), Natural Language Information Retrieval, *Information Processing and Management*, 31 (3), pp. 397-417.



# Processing of Beliefs extracted from Reported Speech in Newspaper Articles

Ralf Krestel and René Witte

Institut für Programmstrukturen und Datenorganisation (IPD)

Universität Karlsruhe (TH), Germany

*krestel|witte@ipd.uka.de*

Sabine Bergler

Department of Computer Science and Software Engineering

Concordia University, Montréal, Canada

*bergler@cs.concordia.ca*

## Abstract

The growing number of publicly available information sources makes it impossible for individuals to keep track of all the various opinions on one topic. The goal of our *artificial believer* system presented in this paper is to extract and analyze statements of opinion from newspaper articles.

Beliefs are modeled using a fuzzy-theoretic approach applied after NLP-based information extraction. A fuzzy believer models a human agent, deciding what statements to believe or reject based on different, configurable strategies.

also varies between different humans, not only depending on different background knowledge but on different attitudes towards a coherent worldview or importance and ability of logical thinking.

A computational system, whose task should be to simulate a human newspaper reader by imitating his belief processing, must take into account not only the beliefs (of others) stated in an article, but also the existing beliefs held by the system. Such an *artificial believer*<sup>2</sup> should also be able to distinguish between different belief strategies, modeling the different human approaches.

Our *Fuzzy Believer* system models a human newspaper reader who develops his own point of view for current events described in newspaper articles. More specifically, we only rely on information stated within the grammatical construct of *reported speech*. This allows a clear assignment of statements to sources and enables the system to judge according to different degrees of reliability in a source.

Our approach differs from existing work by addressing two different problems usually dealt with in isolation: opinion extraction/mining and recognizing textual entailment. Solving these two tasks is necessary to implement an artificial believer. The area of opinion mining [3, 5, 7] is dominated by systems limited to extraction, where the processing of the extracted opinions is rather rudimentary. On the other side are systems that deal with the relation of two sentences to each other [6, 10, 12, 14]. The Pascal RTE Challenge [2, 4] has led to the development of a number of new systems dealing with inference or entailment.

Our fuzzy believer combines these approaches and thereby presents an application capable of “reading” and evaluating newspaper articles. To internally represent the extracted statements and process the different beliefs, we employ fuzzy set theory techniques [18]. Fuzzy set theory explicitly expresses the intrinsic fuzziness in natural language, and the handling of ambiguities and similarities in natural languages is done in a more robust way than crisp approaches. Another reason we chose a fuzzy approach are the existing fuzzy operations for representation and belief revision [15].

To summarize, the system we present in this paper

## Keywords

Belief Processing, Textual Entailment, Artificial Believer

## 1 Introduction

With the possibility to gain access to huge amounts of information, for example via the Internet, the Natural Language Processing (NLP) research community has developed whole branches that deal explicitly with vast amounts of information encoded in written natural language<sup>1</sup>. One goal is to gain knowledge about irrefutable facts like “The number of inhabitants of city X” or the “Name of the president of country X.” But a lot of information, especially within newspaper articles, are not hard facts that could be easily proven right or wrong. Often newspaper articles contain different views of the same event, or state controversial opinions about a certain topic. In this case the notion of *belief* becomes relevant.

For humans this is a daily task, sometimes a conscious act, sometimes unconsciously adopted. Depending on context information and background knowledge, together with other belief structures, humans tend to believe certain statements while other statements get rejected. Although everybody uses the term *belief*, the definition is rather vague, and the processes taking place inside the brain while “someone is believing something” are not understood. The process of believing

<sup>1</sup> for example, *Information Extraction, Summarization, or Information Retrieval*

<sup>2</sup> this term was coined by [1]

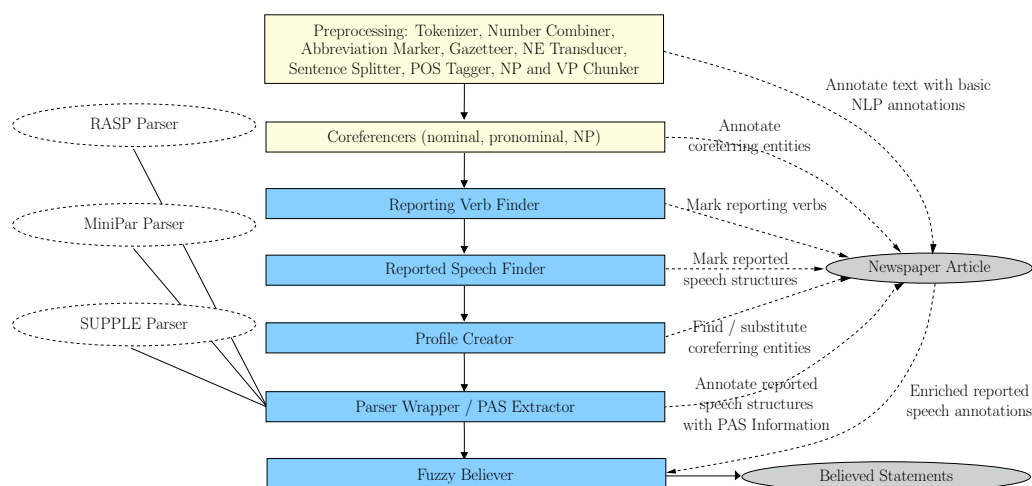


Fig. 1: Overview of the Fuzzy Believer system components

addresses various problems within the NLP domain. Our main contributions making our research significant are: 1. Developing rules to identify and extract reported speech from newspaper articles; 2. processing the gained information by applying fuzzy set theory to natural language processing; 3. creating a working implementation of these ideas, together with an evaluation environment.

The remainder of this paper is structured as follows: In the next section, we give an overview of our fuzzy believer system, followed by a more detailed description of the main component in Section 3. An evaluation of our approach, using different corpora and evaluation methods, is presented in Section 4. Section 5 discusses related work, followed by conclusions in Section 6.

## 2 System Overview

The starting point of our system is a selection of newspaper articles. Different components are used to realize specific tasks within the system to process the input documents, as can be seen in Fig. 1.

After preprocessing an input document, a first important step is to identify noun phrases. These structures are important to identify acting entities, like persons within a text. We do full noun phrase coreference resolution making use of an existing coreferencer [17].

The next step is to identify and *extract reported speech* within the document (see Fig. 1: *Reporting Verb Finder*, *Reported Speech Finder*). For this part, patterns had to be developed representing the different ways to express reported speech.

Afterwards, we have to combine the results found in the last two steps. The coreference component can identify the same source of two different reported speech utterances enabling us to *build profiles* (see Fig. 1: *Profile Creator*).

The core of our system is the processing of the information encoded in the profiles. We use external parsers to extract predicate-argument structures (see Fig. 1: *PAS Extractor*) as basis for further processing. Our focus lies thereby on the analysis of the extracted PASs of the reported speech utterance and the gener-

ation of held beliefs from it in the last step.

Finding the *entailment* relation between two sentences is the most complex part, and an active research field [2, 4]. Do they express the same, similar things, contradicting things, or are they totally independent? Our approach uses fuzzy set theory and WordNet<sup>3</sup> to tackle this question.

The final step is, after trying to “understand” what has been said and by whom, to define what the system should actually believe. The Fuzzy Believer thus has to do processing on the created belief structure. To model different human “believers,” the Fuzzy Believer component (see Fig. 1: *Fuzzy Believer*) uses different *believe strategies*.

The result of the system is a set of propositions the system “believes,” and a set of propositions the system rejected.

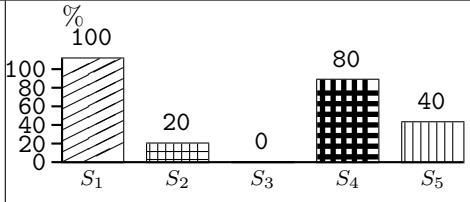
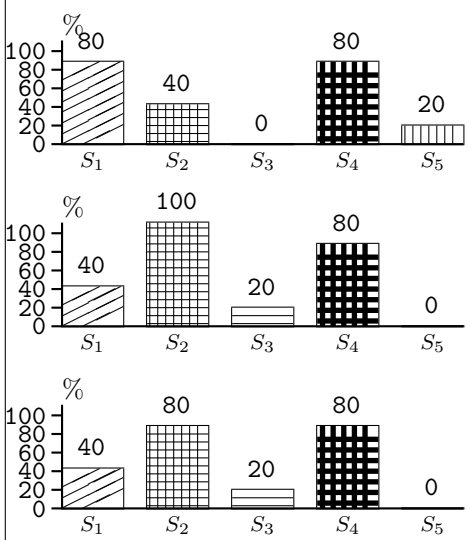
In the next section, we will describe the main component of our system. For more details about the other components concerned with steps 1 to 3 covered in the example shown in Fig. 2, we direct the reader to [9].

## 3 Fuzzy Believer

The Fuzzy Believer component uses predicate-argument structures extracted from the output of a parser to group the statements of the newspaper articles according to common topics. To process these predicate-argument structures using fuzzy set theory later on, we need to consider a few constraints: The basic set for fuzzy operations to work on has to be limited to statements dealing with the same topic or fact in the world. This is due to the character of fuzzy processing always considering the whole set to perform its operations on. And with beliefs having nothing to do with each other stored in only one single set, we could not use a similarity measure between statements to perform our computations, because this would, for example, lead to the deletion of dissimilar statements dealing with independent topics.

The fuzzy processing task therefore has to consist of four steps:

<sup>3</sup> WORDNET, <http://www.wordnet.princeton.edu>

Step	Example	Description
1.	“Preisig worked as a consultant”, one of the employees said.	Sentence in a newspaper article.
2.	[Preisig worked as a consultant](content) [one of the employees](source) [said](reported verb)	Identified reported speech structure.
3.	[Preisig](subject) [work](verb) [consultant](object)	Extracted predicate-argument structure from the output of a parser.
4.	Preisig – work – consultant ( $S_1, domain_{38}$ )	predicate-argument structure assigned to a domain according to the topic.
5.	 <p style="text-align: right;">Atoms <math>A_{S_1,j}</math></p>	Atoms $A_{S_1,j}$ for Predicate-argument structure $S_1$ with correlation grades for all statements in $domain_{38}$ ( $S_1, \dots, S_5$ ) as computed by heuristic $H_1$ . $(S_1, S_1) = 100$ means a 100% possibility that the two statements express the same meaning. $(S_1, S_2) = 20$ indicates on the other hand that the two statements have probably different meanings.
6.	 <p style="text-align: right;">Formula <math>F_{S_1}</math></p> <p style="text-align: right;">Formula <math>F_{S_2}</math></p> <p style="text-align: right;">Formula <math>F_{S_1} \oplus_{\gamma} F_{S_2}</math></p>	Fuzzy belief revision: Result of $\gamma$ -revision with $\gamma = 0.8$ of the two formulas on top. The first formula represents the existing statements within a domain by combining the different atoms to form literals, then clauses, and then formulas of each statement. The second formula represents the new statement added to the domain. The resulting formula contains only these clauses that are not contradicting the new one, or in other words having a similarity degree of at least 0.8. The interpretation of the result is that the system believes the new statement and all older statements about the same topic that are not contradicting the new one.

- (1) Grouping statements into domains (topics),
- (2) Finding a fuzzy representation for the statements,
- (3) Identifying the polarity of statements,
- (4) Computing beliefs according to a strategy.

The different strategies make it necessary to identify the *topics* statements deal with. And the Fuzzy Believer has to identify the *polarity* of the statements to detect opposite opinions.

The first task is handled by two heuristics (semantic, based on WORDNET, and syntactic, based on string similarity) that compare the extracted predicate-argument structures (PAS) of two statements. If the heuristics recognize a similarity degree higher than a given threshold between two statements, they are grouped into one domain (topic).

The second and third task is solved by using fuzzy set theory and representing all statements as degrees of similarity between the verbs of the statements in one domain. This similarity is again computed using WORDNET together with the detection of negations and antonyms.

For the fourth task we use three fuzzy set operations (*Union*, *Expansion*, and *Revision*, see [16]) to model various belief strategies.

**Domain Finding.** The task of this component is to group similar statements together according to their topics to form a *domain*. We use a WORDNET distance measure to find similar, related words and assign a score to each word pair. The threshold of this score can be adjusted as a run-time parameter, allowing a more lenient or a more strict domain classification. As another run-time parameter, the maximum WORDNET distance<sup>4</sup> can be defined.

A second heuristic currently in use compares the string representation of two words. This is particularly useful for proper nouns that do not occur within the WORDNET database. The score of this heuristic depends on the character overlap of the two words, thus a perfect match is not necessary to gain a score.

To ensure that we compare the appropriate words, an analysis of the main verb is mandatory. We have to differentiate between active and passive constructs, exchanging the syntactic subject and the syntactic object.

The requirements for two predicate-argument structures to match are that at least two element pairs have a matching score of at least the defined threshold. This threshold can be set as a run-time parameter, and al-

<sup>4</sup> we use the same WORDNET distance as [17]

lows for more strict or more lenient domain classification.

An advantage of dividing the domain classification and the actual matching finding process is that we can use different thresholds for the fuzzy process of assigning statements to different domains and discover supporting and conflicting statements. One statement can belong to more than one domain, exploiting the possibilities of a fuzzy set representation again. The result of this component is shown in Fig. 2 in step 4.

**Fuzzy Representation.** Every predicate-argument structure is presented as its degrees of similarity with other PASs in the same domain. Fig. 2 step 5 gives an example of the representation of one PAS that is an element of a domain containing five PASs.

**Polarity Identification.** To identify opposing statements, the fuzzy representation of the PASs is evaluated. If the heuristics yielded small values for the degree of similarity, the meaning of the two statements are considered opposing. A threshold makes it possible to decide whether two statements are similar enough to be considered as expressing the same sense or are likely to contain opposing views.

**Computing Beliefs.** The phenomena of opposing opinions compelling the human reader to take either one side or to believe neither has to be reflected in our fuzzy believer system as well. To model different human believe behavior, the fuzzy believer makes the decision which statements to believe based on different strategies. The result is a set of held beliefs and rejected beliefs after processing newspaper articles. The strategies used to model different human behavior are: (1) Believe everything, (2) believe old news, (3) believe new news, (4) believe majority, (5) believe certain source/reporter/newspaper, and (6) believe weighted majority – a combination of (4) and (5). Let’s take a closer look at one of the strategies. The “believe new news” strategy uses a fuzzy operation called *revision*. The result of a revision of formula 1 with another formula 2 depends on the order of the formulas, as well as on an ordering of the clauses of the formulas. The ordering can be chronological depending on the timestamp of the insertion of the clause into the formula, or any other ordering like ordering according to degrees of certainty or an order relying on the reporter or newspaper. We chose as an order the first way enabling us to model a belief strategy concerned with the chronological order of news.

The revision process compares statement sets, formally represented by fuzzy formulas in conjunctive normal form [16], with each other. If the two statements sets are compatible, the revision process results in a new set containing the fuzzy union of both sets. However, in case some of the statements are contradicting to a degree that is larger than the prescribed minimal consistency  $\gamma$ , the revision operator will remove individual, inconsistent statements from the first set, according to a preference ordering [16]. In the example in Fig. 2 at step 6, we can see the formula generated in previous steps containing two clauses, and below it, the new formula, with which we start the revision. The

result shown at the bottom in Fig. 2 is a new formula containing two clauses. The ordering of the clauses, which determines the sequence of processing, is again defined by the date of the statements.

## 4 Evaluation

So far, we performed a detailed evaluation of the individual components of our system. For the evaluation of the reported speech component, as well as for a more detailed evaluation, see [9].

**Domain Finding.** The evaluation of the domain finding component includes the comparison of the results obtained with RASP, MiniPar, and manually annotated predicate-argument structures. The test data we use is taken from the MSR corpus<sup>5</sup> and comprises 116 paraphrase pairs. We treat all sentences as content of a reported speech construct. The best result for recall is 81% and best precision value obtained 52% with a different configuration. Detailed results also including manual PAS annotated test data can be found in [9].

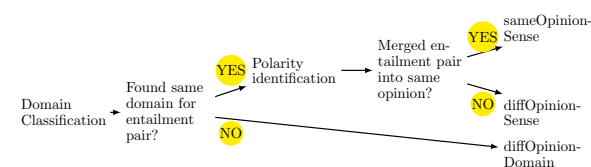


Fig. 2: Polarity identification evaluation strategy

**Polarity Finding.** The data that comes closest to the conditions we need are the entailment pairs of the PASCAL challenge corpus [2]. We tested different configurations and computed accuracy for two settings. For one experiment, we included all results in the evaluation counting the entailment pairs that were not grouped into the same domain by the domain classification as non-entailing. In Fig. 2, these are the pairs in the “diffOpinion-Domain” category. Here, the best results were around 55% accuracy. The other test setting only considered the sentence pairs that were actually grouped into the same domain by the domain classification component (in Fig. 2 the “same/diffOpinion-Sense” category) yielding an accuracy of 58% using MiniPar-extracted PASs. Table 1 gives an overview of the obtained results with the configuration settings in the table meaning, from left to right: Maximum WordNet Distance between (1) subjects, (2) verbs, (3) objects of two statements. (4) indicates whether a new statement has to match with one (lenient) or all (strict) statements within one domain and (5) is the threshold for assigning the same polarity to a statement.

## 5 Related Work and Discussion

The extraction of opinions from newspaper articles [3] or customers reviews [5, 7] has become an active re-

<sup>5</sup> MSR-corpus, [http://research.microsoft.com/research/nlp/msr\\_paraphrase.htm](http://research.microsoft.com/research/nlp/msr_paraphrase.htm)



Configuration	Accuracy			
	Sense & Domain		Sense Only	
	Rasp	MiniPar	Rasp	MiniPar
3-3-3-strict-0.7	0.52	0.55	0.53	0.58
5-5-5-lenient-0.7	0.51	0.53	0.51	0.53
5-5-5-strict-0.3	0.52	0.53	0.55	0.51
5-5-5-strict-0.7	0.51	0.54	0.50	0.56
7-7-7-strict-0.7	0.51	0.52	0.51	0.52

**Table 1:** *Polarity Identification: Accuracy values for different parse methods*

search field. Those approaches are usually only concerned with the identification and extraction of information without processing it further, except for binary classification within a clearly specified domain.

In the wake of the PASCAL challenge [2,4], systems have been developed to deal with the relation of sentences to each other. The different approaches include the recognition of false entailment [14], or learning entailment [10]. Others are concerned with relatedness between words and how to measure it [8]. We were not interested in concentrating on one of these areas but rather to develop an all-embracing system incorporating different aspects.

For the domain classification, our best results for 300 paraphrase pairs from the MSR-corpus are, for recall, 81% (with a precision of 38%), and for precision 52% (with a recall of 58%). These values can probably be improved by using more sophisticated heuristics, although there will be a ceiling set by the parser and by the use of language in general. The same meaning can be expressed by various different sentences whose words are not in close relations to each other and therefore hard to detect by current NLP tools. Keeping these facts in mind, the obtained numbers are rather satisfactory and promising for future development.

The rather shallow semantic approach sets a practical limit to the achievable results. This can be inferred by comparing the numbers obtained using manually parsed predicate-argument structures with the numbers obtained by the parsers. It shows that there is space for improvement on the side of the parsers, as well as on the side of the PAS extractor. Combining the results of different parsers could also lead to better results, but a precision of 55% and a recall of 85%, as obtained for the best configuration of the system using manually parsed PASs, shows that it needs more and/or better heuristics to get a really significant improvement.

The polarity identification task was expectedly the hardest one. This is illustrated by the rather poor results we obtained by trying to find different opinions within one domain. Best accuracy values were obtained using MiniPar and were around 58%. This task is very hard for computational systems. But with more elaborated heuristics it is possible to increase these numbers, comparable to the Pascal challenge [2, 4], where systems also started with around 50% accuracy and improved over time.

Testing of the different strategies revealed that the fuzzy processing operators perform in accordance to their assigned tasks. Further evaluation of the results would need some kind of measure to get quantitative,

comparable results. This is beyond the scope of this paper and deferred to future work.

## 6 Conclusions and Future Work

We developed an artificial believer system that can be applied in different scenarios: (1) companies evaluating product reviews on web sites or blogs, (2) governmental organizations interested in dispositions of people, or (3), as we demonstrated here, assist individuals in news analysis.

Apart from the evaluation described above, tests of the system on actual newspaper articles showed accepted and rejected beliefs that reflect the desired results. Embedding the system within an Internet agent and measuring its effectiveness for a real user will be the next major step.

## References

- [1] A. Ballim and Y. Wilks. *Artificial Believers: The Ascription of Belief*. Lawrence Erlbaum Associates, Inc., 1991.
- [2] R. Bar-Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. The Second PASCAL Recognising Textual Entailment Challenge. In *Proc. of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.
- [3] S. Bethard, H. Yu, A. Thornton, V. Hatzivassiloglou, and D. Jurafsky. Automatic extraction of opinion propositions and their holders. In Qu et al. [13], pages 20–27.
- [4] I. Dagan, O. Glickman, and B. Magnini. The PASCAL Recognising Textual Entailment Challenge. In *Proc. of the PASCAL Challenges Workshop on Recognising Textual Entailment*, 2005.
- [5] M. Gamon, A. Aue, S. Corston-Oliver, and E. K. Ringger. Pulse: Mining customer opinions from free text. In A. F. Famili, J. N. Kok, J. M. Peña, A. Siebes, and A. J. Feelders, editors, *Advances in Intelligent Data Analysis VI, 6th International Symposium on Intelligent Data Analysis, IDA 2005, Madrid, Spain, September 8-10, 2005, Proceedings*, volume 3646 of *Lecture Notes in Computer Science*, pages 121–132. Springer, 2005.
- [6] S. Hahn, R. Ladner, and M. Ostendorf. Agreement/disagreement classification: Exploiting unlabeled data using contrast classifiers. In Moore et al. [11], pages 53–56.
- [7] S.-M. Kim and E. Hovy. Identifying and analyzing judgment opinions. In Moore et al. [11], pages 200–207.
- [8] B. B. Klebanov. Measuring Semantic Relatedness Using People and WordNet. In Moore et al. [11], pages 13–16.
- [9] R. Krestel, R. Witte, and S. Bergler. Creating a Fuzzy Believer to Model Human Newspaper Readers. In Z. Kobti and D. Wu, editors, *Proc. of the 20th Canadian Conference on Artificial Intelligence (Canadian A.I. 2007)*, LNAI 4509, pages 489–501, Montréal, Québec, Canada, May 28–30 2007. Springer.
- [10] B. MacCartney, T. Grenager, M.-C. de Marneffe, D. Cer, and C. D. Manning. Learning to recognize features of valid textual entailments. In Moore et al. [11], pages 41–48.
- [11] R. C. Moore, J. Bilmes, J. Chu-Carroll, and M. Sanderson, editors. *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. Association for Computational Linguistics, New York City, USA, June 2006.
- [12] R. Nairn, C. Condoravdi, and L. Karttunen. Computing relative polarity for textual inference. In *Proceedings of the 5th Workshop on Inference in Computational Semantics (ICoS-5)*, Buxton, UK, April 2006.
- [13] Y. Qu, J. Shanahan, and J. Wiebe, editors. *Exploring Attitude and Affect in Text: Theories and Applications*. Technical Report SS-04-07. AAAI Press, Stanford, CA, USA, March 22–25 2004.
- [14] R. Snow, L. Vanderwende, and A. Menezes. Effectively using syntax for recognizing false entailment. In Moore et al. [11], pages 33–40.
- [15] R. Witte. *Architektur von Fuzzy-Informationssystemen*. BoD, 2002. ISBN 3-8311-4149-5, <http://rene-witte.net>.
- [16] R. Witte. Fuzzy Belief Revision. In *9th Intl. Workshop on Non-Monotonic Reasoning (NMR'02)*, pages 311–320, Toulouse, France, April 19–21 2002.
- [17] R. Witte and S. Bergler. Fuzzy Coreference Resolution for Summarization. In *Proc. of 2003 Intl. Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS)*, pages 43–50, Venice, Italy, June 23–24 2003.
- [18] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

# A Syntactic Candidate Ranking Method for Answering Questions with a Main Content Verb

Abolfazl Keighobadi Lamjiri, Leila Kosseim, Thiruvengadam Radhakrishnan  
Department of Computer Science and Software Engineering  
Concordia University, Montréal, Canada  
{a\_keigho,kosseim,krishnan}@cse.concordia.ca

## Abstract

We present a technique for ranking the candidate answers of questions that have a main content verb. This novel ranking method uses the question head (the most important noun phrase) as an anchor for selecting the target subtree in the parse tree of the candidate sentence. The semantic similarity of the action in the selected subtree to the action asked by the question is verified using WordNet::Similarity. For verifying the syntactic similarity of the target subtree to the question's parse tree, syntactic restrictions as well as word-based measures compute the unifiability of critical syntactic participants in the trees. Finally, in order to apply web redundancy statistics into our linguistic method, we fed Aranea answers into our linguistic QA system.

Results show a precision of 48% on the TREC 2003 to 2006 non-copula questions. This confirms our hypothesis of the applicability of a basic syntactic mapping for answering non-copula factoid questions.

## Keywords

Question Answering, Syntactic Mapping, Non-copula Verbs

## 1 Introduction

In this paper, we present a technique for ranking the candidate answers of questions that have a main content verb. Researchers in QA have classified questions based on various features, such as, their *semantic type*, arranged hierarchically in taxonomies (ex. comparison, definition, spatial or temporal, procedural, etc.) [2], or their *structure*, into factoid, that ask for names, dates, locations, quantities, etc. versus complex questions, that require syntactic, semantic or contextual processing, relation detection, etc. or ask for a list of answers, or any important information about a topic [14].

We categorize questions based on their main verb type into copulative (that have a 'to be' main verb) such as *Q66.2- "Who was the on-board commander of the submarine?"*, versus non-copula (with main content verb) questions, such as *Q149.2- "The Daily Show parodies what other type of TV program"*<sup>1</sup>. The initial idea behind this categorization comes from our

<sup>1</sup> Although there are other copula verbs in English, such as 'look', 'feel', 'taste', 'smell', 'sound', etc., that can be used to connect the subject to an adjective, we only make a distinction between 'to be' versus non-'to be' verbs.

Question Type	#Questions	ratio
2003 copula	171	59.0%
2003 non-copula	119	41.0%
2004 copula	149	64.8%
2004 non-copula	81	35.2%
2005 copula	230	62.7%
2005 non-copula	137	37.3%
2006 copula	264	65.5%
2006 non-copula	139	34.5%
Total copulas	814	63.1%
Total non-copulas	476	36.9%

**Table 1:** The number of copula versus non-copula questions in each TREC QA question set.

previous work in closed-domain [11]. As opposed to factoid questions, questions posed in a closed domain are longer and usually tend to be more open-ended and ask for properties, procedures or conditions [4]. As a result, they usually contain a main content verb, with critical syntactic relations (subject and object). In our closed domain corpus [11], this type accounts for 70% of the questions. We showed that syntactic analysis is quite successful for measuring the relatedness of candidate answers to these non-copula questions. In open domain, the distribution of questions is significantly different. For example, the TREC QA [20] data set contains only about 1/3 of non-copula questions (Table 1). In this paper, we show that our syntactic ranking technique is also applicable to open domain. Analysis of previous TREC results shows that all participating QA systems perform slightly better on copulative questions, practically showing that non-copulas questions are more difficult to answer. To our knowledge, work on categorizing questions based on their main verb type has not been investigated before.

## 2 Related Work

To rank the candidate sentences returned by the IR, we compute their syntactic similarity to the question. Pure linguistic criteria for measuring the similarity of parse trees impose very strict syntactic constraints that result in low recall [17]. On the other hand, statistical systems that learn and score syntactic links such as [10] and [18] are very lenient in considering the importance of primary roles (such as subject and object) over less important roles (such as determiner modifier). An interesting effort towards improving this syntactic measure is weighting the matching links according to their Inverse Document Frequency (IDF)<sup>2</sup>;

<sup>2</sup> Two links match if they have similar head, relation and tail.

rare link types have more information content than frequent relation types and hence, will contribute more when matching two subtrees. However, this has not solved the recall problem.

Most current TREC type question answering systems choose the answer from the candidate sentence that statistically has some lexical similarity to the question; For example, with one of the best performing QA systems in the TREC 2004 track, the university of Singapore QA [17] uses the Jaccard coefficient to test pairwise similarity of frames marked by the AS-SERT predicate argument recognizer. This coefficient ignores stop-words and uses the bag-of-words feature for scoring.

Katz and Lin [9] have a ternary  $\langle \text{subject} - \text{verb} - \text{object} \rangle$  scheme and use predicate logic; the constraint satisfaction to find an answer that satisfies the syntactic/semantic constraints is binary while we use a fuzzy scoring schema. Applicability of their comprehensive state-of-the-art method is shown successfully on five questions. Breaking the text into small grains in predicate-logic form is less feasible to apply in large scale and open-domain.

Salvo et al., in [3] introduce a hierarchical knowledge representation for *meaning entailment*: a sentence is entailed by a paragraph if its context graph can be unified with that of the paragraph. A cost function determines the goodness of a unification. Unified nodes must be at the same level in the hierarchy, and the cost of unifying nodes at higher levels dominates those of the lower levels. Nodes in both hierarchies are checked for subsumption in a top-down manner: The hierarchy level  $H_0$  consists of verbs that unify if they are synonyms based on WordNet and their constituent phrases at  $H_1$  level unify. Hierarchy set  $H_2$  corresponds to word-level nodes. As it can be seen, syntax is used only at the topmost level  $H_0$ . As we will see later, subject and object relations are considered to be critical in our matching algorithm.

Raina et al. [6] learn weights for matching subtree at the source and destination nodes for this task: matching of the modifier of two verb nodes may contribute less to the unification score than matching of their subjects. Nyberg [5] also introduces a light-weight fuzzy unification as an extension to their earlier work, JAVELIN; here, counterpart syntactic links and their head and tail tokens contribute to the final match score. Unlike PiQASso [1], they weigh syntactic links so that a matching ‘subject’ link has higher contribution than a ‘determiner’ link. For this linguistic work however, no evaluation result is provided.

In this paper, we present a syntactic solution for question answering by parse tree matching for the questions that have a content main verb. As we will see, our approach uses syntax while not being dependent on having a perfect parse tree matching.

### 3 Candidate Answer Extraction

In this section, we review the processes involved in the information retrieval phase for question answering.

### 3.1 Question Analyzer

The question analyzer module extracts the *expected answer type* and a ranked list of *question keywords* to be fed to the Lucene IR engine<sup>3</sup>. *Question keywords* will be used to retrieve the documents and passages relevant to the question, based on the assumption that relevant passages contain words in common with the question. We use the existing work done in the Aranea QA system [15] to extract the expected answer type. Aranea was one of the top 5 QA systems at TREC 2002.

*Important* words are then marked as question keywords. Two factors contribute in deciding if a word is important: its part-of-speech and its number of modifiers. The question analyzer first filters out non-content words and keeps nouns, verbs, adjectives and adverbs. It then processes *important* parse links provided by the Minipar parser [12] for identifying the question keywords. To do so, based on the type of the parse link seen, both the head and the tail or only the tail is kept:

- For a nominal complement of a preposition such as “for convenience” and determiner relation such as “the network” (shown as ‘pcomp-n’, ‘det’ respectively, in the Minipar notations), only the tail (‘convenience’ and ‘network’) is marked as a question keyword.
- For an adjunct modifier link, a lexical modifier such as “electric guitar”, a conjunction, a subject or object links, a noun complement and a passive verb modifier of a noun such as “the service provided...” (shown as ‘mod’, ‘lex-mod’, ‘conj’, ‘subj’, ‘obj’, ‘nn’, and ‘vrel’ in Minipar), both head and tail words are considered.

Finally, the number of modifiers of a word is computed and stored as a feature of that keyword. This feature will be used later in scoring these keywords. This strategy is based on the hypothesis that if a word has more modifiers, it acts as a central idea in the question and is therefore more important<sup>4</sup>. For example, in the question Q76.4- “What is the title of his all-time best-selling record?”, the noun ‘record’ has three modifiers (‘his’, ‘all-time’ and ‘best-selling’). Question keywords are ranked based on the following heuristic function:

$$Score_{kw} = (\#modifiers + 1) \times Score_{POS}(kw)$$

where,  $Score_{POS}$  is assigned as the following: proper nouns are favored (given a weight value of 3), then common nouns (1), verbs (0.75), adjectives (0.5), and finally auxiliary verbs, adverbs and determiners (0.25). The rationale behind these values is to boost proper nouns in the list, since they convey a unique meaning. Verbs on the other hand are more ambiguous [13] and can have more synonyms (alternatives for conveying the same meaning), so they are slightly pushed down the list. Adverbs usually relate to verbs, and not nouns, so they receive the lowest rank. These values were determined experimentally with our development set (the first 50 non-copula questions from the TREC

<sup>3</sup> Available at <http://lucene.apache.org/>

<sup>4</sup> It is interesting to note that the PiQASso QA system [1] ranks question keywords based on their depth in the parse tree.

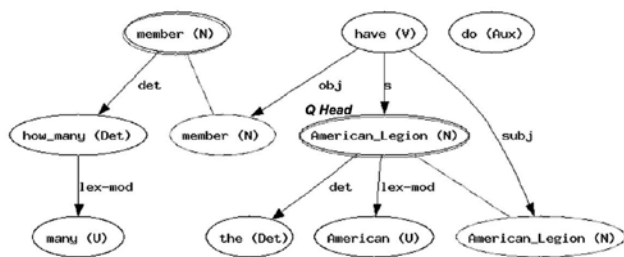


Fig. 1: Parse structure for the question “How many members does the American Legion have?”

2005 set). The accuracy of our scoring method is not sensitive to the values chosen as long as the order of importance of each category is preserved. As an example, the following ranking is computed for *Q98.4*- “What organization has helped to revitalize Legion membership?”:

Legion membership\* revit\* help\* organiz\*

### 3.2 Candidate Sentence Selection

The candidate answer extraction module processes the top  $n$  documents that are returned by Lucene and are found in the PRISE top document list<sup>5</sup>. Sentences that contain  $\alpha$  percent of the keywords are recorded as candidate sentences to be ranked by our linguistic unifier. Since our rather strict unifier filters out bad candidates later on, we chose a low threshold of  $\alpha = 65\%$  to have a high recall at this stage.

Experiments with different values of  $\alpha$  for this boolean bag-of-words sentence selection show that varying this threshold affects the candidate ranking and the answer extraction running time, but does not have much effect on the quality of the results. On average, the number of candidates retrieved decreases from 55 (with a standard deviation of 23) to 25 (std.dev. 16) when increasing the threshold from 35% to 75% for our development question set.

## 4 Syntactic Unification for Candidate Ranking

Statistical approaches in QA inspired us to relax a strict syntactic mapping. We force critical syntactic roles to eliminate the candidates with no syntactic consistency with the question, and score the remaining links for the candidates that pass the first criterion.

### 4.1 Choosing the Target Subtree

Essentially, we believe that the best subtree in a candidate sentence is the one that has a similar verb to the question’s main verb, and equivalent arguments. A strong verb similarity should co-occur with an essential entity similarity (question head) match in the candidate’s parse tree. This suggests that a strong seed point is the root of the subtree in the candidate that contains the question’s head noun phrase.

<sup>5</sup> This list is compiled by the TREC organization, running their IR engine on question keywords and the answer phrase.

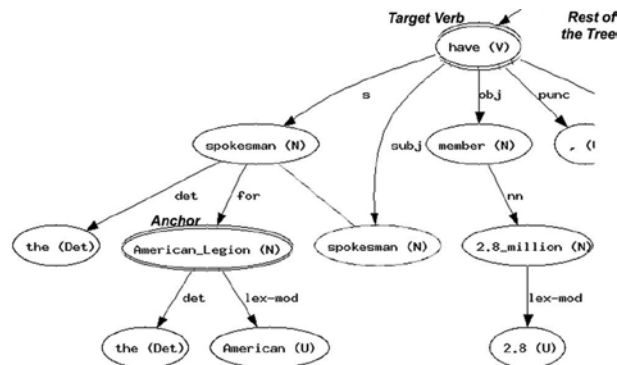


Fig. 2: The parse structure for the sentence “...said Phil Budahn, spokesman for the American Legion, which has 2.8 million members.”

#### 4.1.1 Finding the Question Head

To choose the question head, we rank all the noun phrases in the question and pick the one that contains the most valuable question keywords (with higher  $Score_{kw}$  value, see Section 3.1). If this head phrase is found in the candidate sentence, it becomes an anchor to find the relevant verb. We then move up from this noun phrase in the candidate parse tree to reach the first parent verb. For example, in the question *Q98.5*- “How many members does the American Legion have?”, two noun phrases exist (the double lined nodes in Figure 1). The noun phrase ‘American Legion’ is chosen as the question head, since it has the highest  $Score_{kw}$  value. In the candidate sentence, “...said Phil Budahn, spokesman for the American Legion, which has 2.8 million members.”, this anchor is found in the left subtree of the verb ‘have’ (Figure 2). Moving up from this anchor skips the noun node ‘spokesman’ and marks the verb node ‘have’ as the root of the target subtree. This root will then be used as the seed point for starting the unification. In such long candidate sentences, using an anchor reduces the candidate verbs to the ones that include the question head (or a reference to it).

#### 4.1.2 Semantic Similarity of Verbs

Since the main action specified in a non-copula question is typically realized by a verb, our first step is to verify the semantic relatedness of the question’s main verb to the candidate’s target verb.

To do this, we use WordNet::Similarity [16]. This package provides six similarity measures which use information found in the *is-a*, *has-part*, *is-made-of*, and *is-an-attribute-of* relations in a hierarchy of concepts (or synsets) and quantify how much concept A is similar to concept B.

Among these six measures, Leacock and Chodorow (lch) worked best for verbs in our development set. This measure basically finds the shortest path between two concepts, and scales that value by the maximum path length found in the *is-a* hierarchy in which they occur. We proceed with unification for the candidates that have a target verb with a similarity value more than 1.8 to the question’s main verb.



## 4.2 Unifying Two Subtrees

Finally, we check whether the target verb relates the same entities as the question’s main verb. The fuzzy statistical method we describe in this section, evaluates the similarity of two *subject* subtrees, and likewise for *object* or *modifier* subtrees, if any.

To do so, we apply a heuristic that uses two measures: the number of overlapping words based on a bag-of-words approach and the number of overlapping links.

These similarity scores are summed to produce the final score of a candidate sentence:

$$\text{Similarity}(Verb_q, Verb_T) + \sum_i \text{Counterpart Subtree Score}(Q_i, T_i)$$

where,  $Q$  is the question,  $T$ , the target subtree, and

$$\text{Score}(Q_i, T_i) = \beta \times \text{WordOverlap} + (1 - \beta) \times \text{LinkOverlap}$$

is the unification score of two counterpart subtrees. The parameter  $\beta$  shows the relative importance of each feature:  $\beta = \frac{1}{3}$  (our configuration) considers the link-overlap to be twice as important as the bag-of-words feature. Note that the absolute value of the final score is not important since the scores are used only to rank the candidates.

Each subtree can be seen as a paraphrase, since its focus is an entity (noun) that possibly has some modifiers. For example, the noun phrase ‘*the American Legion*’ in the question “*How many members does the American Legion have?*” (Figure 1) appears as “*spokesman for the American Legion*” in the answer sentence, depicted in Figure 2. Here, a score of 6.25 is returned by matching the words (‘the’, ‘American’ and ‘Legion’) and 2.0 for the matching links in the subject subtrees.

The reason we relax our linguistic constraints at this stage is that we are focusing on a sentence that conveys a similar event or state to the question; only a clue about similarity of its verb arguments is sufficient to conclude that its verb modifies the same entities as the question. Syntactic differences of verb arguments (subtrees) should not critically affect our judgment.

By analyzing a few unification cases, we realized that matching different types of links should have a variable contribution to the final unification score. Compare a modifier (‘*mod*’) link matching in the candidate “*wireless network*” as opposed to a determiner (‘*det*’) link in the candidate “*a network*” matching with the phrase “... *a wireless network* ...” in the question. The first case shows a stronger similarity since it narrows down the meaning of the noun (‘*network*’). To account for this, we weight links differently: a lexical modifier link has the highest weight because it connects two proper nouns, while a determiner has the lowest score. Table 2 shows the classes of equivalent links we selected and the values we obtained experimentally for each class. These values can also be learned given a tagged set of equivalent, but syntactically different phrases, such as an appropriately selected subset of the Microsoft Research Paraphrase corpus<sup>6</sup>.

For the previous example (Figure 2), the value of the *LinkOverlap* feature will therefore be  $1.0 + 0.25 = 1.25$  (for the lexical modifier and the determiner link).

Category	Minipar Relation	weight
Lexical modifier	lex-mod	1.0
Adjective/Nominal mod	mod,pnmod,pcomp-n,nn	0.5
(pre)Determiner	(pre)det	0.25
Possessives	gen	0.25

**Table 2:** Weights of different syntactic links used in scoring the similarity of two phrases.

### 4.2.1 Using Statistics from the Web

One simple way of embedding statistics in our linguistic QA system is to feed it with the top answers given by a redundancy-based QA system.

Aranea [15] is a QA system that extracts answers from the Web using two different techniques: *knowledge annotation* and *knowledge mining*. Knowledge annotation is an approach to answering large classes of frequently occurring questions by utilizing semi-structured and structured Web sources. Knowledge mining is a statistical approach that leverages massive amounts of Web data to overcome many natural language processing challenges.

Aranea’s answers are initially used to expand the information retrieval query and later on, are used in the unifier to boost candidates that include Aranea’s answers based on their position in the syntactic tree in these candidates. Ideally, an Aranea’s answer should fill in the role that is asked for in the question by the question word. However, since such a perfect mapping rarely occurs, we implemented a heuristic to compute high relevance of the suggested answer with the question words using the parse tree as opposed to using the linear form of the sentence.

Although this heuristic does not guarantee correctness of that answer, with the level of detail we understand the semantics of sentences, such an assumption is reasonable. Candidates whose parse tree contain are boosted by  $\frac{1}{D} \times \text{Score}(Cand_{Aranea})$ , where  $D$  is the path distance of the Aranea’s answer from the root of the target subtree. In effect, this considers closer positions as more relevant.

Finally, the noun phrase that is of the expected answer type in the target subtree is extracted and returned from the best candidate sentence.

### 4.2.2 Inter-type Syntactic Mapping

When the question or the answer sentence is copula while the other one has a non-copula main verb, they cannot be mapped to each other without syntactic modification or semantic reasoning. The answer to the non-copula question *Q109.2* “*How many countries does it operate in?*” about “*Telefonica of Spain*” is answered by the propositional attachment in the copula sentence “*Telefonica is the largest supplier of telecommunications services in the Spanish and Portuguese speaking world with operations in 17 countries and over 62 million customers.*”.

Multiple mapping cases can happen in this situation; for example, the answer to a copula question might appear as a noun modifier or a propositional attachment in a non-copula sentence. Manual modeling of all possible mapping cases is difficult and will not cover many cases. This should be done automatically and with a larger data set in order to significantly improve the results. For this reason, we leave this task as

<sup>6</sup> Available at <http://research.microsoft.com/research/>

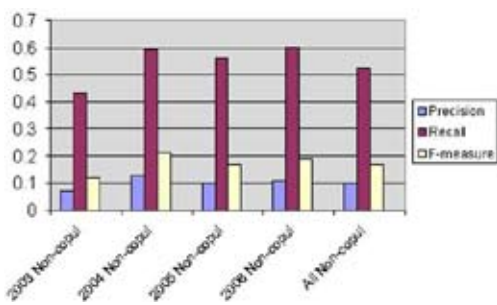


Fig. 3: Precision and recall at the document level for non-copula test questions.

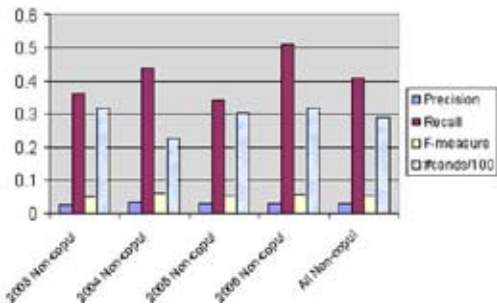


Fig. 4: Precision and recall of candidate sentence selection for non-copula test questions.

future work. Alternatively, one might use the edit distance measure to find the degree of similarity of such sentences (similar to Vilares, et al.[19]).

## 5 Evaluation

### 5.1 Candidate Extraction Results

Figure 3 shows the precision and recall of document retrieval for the 426 non-copula questions in the TREC 2003 to 2006 data sets (50 non-copula questions from TREC 2005 were kept for development). Precision and recall at the document level (at level 35) are around 10% and 53% respectively<sup>7</sup> (except for the 2003 where most participants performed significantly more poorly). Figure 4 shows the input accuracy at the sentence level. As shown in this diagram, sentence level recall drops to around 42% from 53% at the document level. The precision obtained for this recall level in candidate sentence extraction is 2.9%. The ‘#cand/100’ bar shows the number of candidates that are selected for each question (divided by 100). The more candidates extracted, the harder the ranking task. On average, around 30 candidates are passed to the unifier for ranking. The recall column in this figure shows the percentage of questions that have at least one correct answer in their candidate set. Sentence level recall and the number of candidate sentences are two factors that create an upper bound on the expected accuracy of our unifier.

### 5.2 Candidate Ranking Accuracy

The accuracy of a question answering system is reported as the Mean Reciprocal Ranking (MRR) score which is equal to the inverse of the position of the first correct answer in the list [20].

<sup>7</sup> Compared to the state of the art in IR for open domain QA systems (86.1% [7]), our IR method has space for improvement.

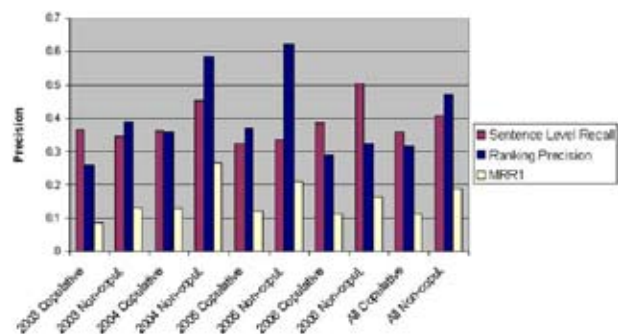


Fig. 5: The accuracy of our syntactic ranking method on the TREC test question sets.

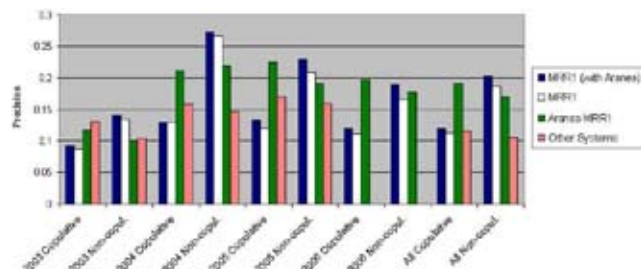


Fig. 6: Comparison with the modified Aranea and other QA participants on the TREC factoid questions.

The theoretical baseline for this task is the precision of randomly selecting a candidate as the answer: on average, we have 1.7 correct answers in a set of 30 candidates. This results in a theoretical precision baseline  $\frac{1.7}{30}$  of 5.7%. However, the experimental baseline ranking accuracy is 5.0%, because of high deviation in the size of candidate sets.

Figure 5 shows the accuracy of our QA system. The column labeled ‘Ranking Precision’ shows the unifier’s accuracy when the error in the IR’s output is excluded from the final result. The results show a high performance for the candidate ranking algorithm especially for non-copula questions (twice as high compared to copulas). High accuracy for the questions with a main content verb shows the important role of the main verb and the syntactic structure mapping for answering these questions.

Figure 6 shows the final accuracy that our QA system achieves. MRR1 shows the final QA precision, without receiving answer hints from Aranea. The ‘Other Systems’ column shows how other TREC participants performed on copula versus non-copula questions. Note that based on the performance of previous TREC submissions, the questions in TREC 2003 were harder to answer, with an average accuracy of MRR=12.2% for the year 2003, compared to the precision MRR1=15.5% in the year 2004 and MRR1=16.7% in 2005. Aranea and other participating TREC QA systems tend to work slightly better on copula questions, practically indicating that non-copula questions are generally harder to answer.

Most current open domain QA systems use redundancy from the Web and the corpus to rank their candidate answers. By combining such a list with the syntactically ranked candidate answer list returned by our QA system, we have a chance to apply one’s information to the other. To test this, we used the statistical Aranea QA system again. This time, the output of Aranea was used to im-

prove our IR result ( $m$  documents from the result of *AranAnswers AND QuestionKeywords* were added to the regular keyword document list) and provided the expected answer type (MRR1 (with Aranea)).

### 5.3 Analysis

To better understand where the system goes wrong, we manually analyzed the errors in the 139 non-copula questions from TREC 2006. As we mentioned earlier, lack of query expansion prevents our system to extract candidates that have different wordings from the question (low IR recall of around 60%). The most frequent error source is when the answer to a non-copula question is given in a copula sentence (6.5%). Finally, the *lcs* semantic similarity measure does not return a correct value for main verbs in 6% of the correct answer sentences. We do not specify the sense of verbs, while WordNet::Similarity has the capability of accepting the sense numbers in order to compute a more precise semantic relation between verbs.

## 6 Conclusion and Future Work

In this paper we presented a method that imposes simple linguistic constraints to select only the candidates that refer to the same event that the question asks for. At the same time, candidate sentences are syntactically chunked. A heuristic measure then computes the similarity of each chunk in a candidate to its counterpart in the question. The similarity of the verb and its entities show high resemblance of that candidate to the question. The final answer is extracted and returned from the top ranked candidate. We evaluated this algorithm on the TREC 2003 to 2006 QA question sets and showed that our unification based scoring method achieves an accuracy of 47% for non-copula factoid questions (comprising 1/3 of these questions).

Although we have a relatively low accuracy at the sentence extraction level, optimizing this phase will make the ranking task more difficult by extracting more candidates. Based on our closed domain experiments, however, we believe that the accuracy of our linguistic method is robust towards having a larger candidate set [11].

Improving the inter-type syntactic mapping strategy and doing word sense disambiguation for improving the semantic similarity measure should improve our overall accuracy considerably. Backing off to phrase equivalence recognition (such as Jacquemin's technique [8]) for cases where the question head is not connected to any verb in the candidate sentence, or when none of the candidate sentences have any syntactic similarity to the question.

Defining or learning other linguistic features than the main verb type in order to categorize and feed questions based on the specialty of different QA systems might give an ultimate solution to tackle the question answering problem.

### Acknowledgement

This research was financially supported by a grant from NSERC and Bell University Laboratories.

## References

- [1] G. Attardi, A. Cisternino, F. Formica, M. Simi, and A. Tommasi. PiQASso: Pisa Question Answering System. In *Proc. of the TREC-12 Conference*, pages 599–607, Gaithersburg, MD, 2001.
- [2] J. Burger and et al. Issues, Tasks and Program Structures to Roadmap Research in Question Answering. 2001.
- [3] R. de Salvo Braz, R. Girju, V. Punyakanok, D. Roth, and M. Sammons. An Inference Model for Semantic Entailment in Natural Language. In *AAAI05*, pages 261–286, Illinois, USA, 2005.
- [4] H. Doan-Nguyen and L. Kosseim. Using Terminology and a Concept Hierarchy for Restricted Domain Question Answering. In *Research on Computing Science, Special issue on Advances in Natural Language Processing*, pages 183–194, 2006.
- [5] B. V. Durme, Y. Huang, A. Kupsc, and E. Nyberg. Towards light semantic processing for Question Answering. In *Proc. of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 54–61, NJ, USA, 2003.
- [6] R. R. et al. Robust Textual Inference using Diverse Knowledge Sources. In *Proc. of the First PASCAL Challenge*, pages 57–60, UK, 2005.
- [7] S. Harabagiu, A. Hickl, J. Williams, J. Bensley, K. Roberts, Y. Shi, and B. Rink. Question Answering with LCC's CHAUCER at TREC 2006. In *Proc. of the TREC 2006 Conference*, pages 283–292, Gaithersburg, MD, 2006.
- [8] C. Jacquemin. Syntagmatic and paradigmatic representations of term variation. In *37th Annual Meeting of the Association for Computational Linguistics (ACL'99), Proceedings*, pages 341–348, USA, 1999.
- [9] B. Katz and J. Lin. Selectively Using Relations to Improve Precision in Question Answering. In *Proc. of the EACL 2003 Workshop on NLP for Question Answering*, pages 50–60, Hungary, 2003.
- [10] M. Kouylekov and H. Tanev. Document filtering and ranking using syntax and statistics for open domain question answering. In *Proc. of ESSLLI 2004 Workshop on Combining Shallow and Deep Processing for NLP*, pages 21–30, Nancy, France, 2004.
- [11] A. K. Lamjiri, L. Kosseim, and T. Radhakrishnan. A Hybrid Unification Method for Question Answering in Closed Domains. In *Proc. of the 3rd International KRAQ'07 Workshop*, pages 36–42, Hyderabad, India, 2007.
- [12] D. Lin. Principle-based Parsing without Overgeneration. In *Proc. of ACL-93*, pages 112–120, Ohio, USA, 1993.
- [13] D. Lin. *Review of WordNet: An Electronic Lexical Database*. The MIT Press, 1998.
- [14] J. Lin. The role of information retrieval in answering complex questions. In *COLING/ACL 2006 Poster Sessions*, pages 523–530, Sydney, Australia, 2006.
- [15] J. Lin and B. Katz. Question Answering from the Web Using Knowledge Annotation and Knowledge Mining Techniques. In *Proc. of CIKM'03*, pages 116–123, USA, 2003.
- [16] T. Pedersen, S. Patwardhan, and J. Michelizzi. WordNet::Similarity - Measuring the Relatedness of Concepts. In *Proc. of the Nineteenth National Conference on Artificial Intelligence*, pages 1024–1025, San Jose, USA, 2004.
- [17] R. Sun, J. Jiang, Y. F. Tan, H. Cui, T.-S. Chua, and M.-Y. Kan. Using Syntactic and Semantic Relation Analysis in Question Answering. In *Proc. of the TREC-13 Conference*, Gaithersburg, MD, 2004.
- [18] H. Tanev, M. Kouylekov, B. Magnini, M. Negri, and K. Simov. Exploiting Linguistic Indices and Syntactic Structures for Multilingual Question Answering: ITC-first at CLEF 2005. In *CLEF-2005 Working Notes*, pages 21–23, Vienna, Austria, 2005.
- [19] M. Vilares, F. J. Ribadas, and J. Vilares. Phrase similarity through the edit distance. In F. Galindo, M. Takizawa, and R. Traunmüller, editors, *Database and Expert Systems Applications*, volume 3180 of *Lecture Notes in Computer Science*, pages 306–317. Springer-Verlag, USA, 2004.
- [20] E. Voorhees and D. Tice. The TREC-8 Question Answering Track Evaluation. In *Proc. of the TREC-8 Conference*, pages 83–106, Gaithersburg, MD, 1999.

# Disambiguating Levin Verbs Using Untagged Data

Jianguo Li

Department of Linguistics  
The Ohio State University  
Columbus, Ohio, USA  
jianguo@ling.ohio-state.edu

Chris Brew

Department of Linguistics  
The Ohio State University  
Columbus, Ohio, USA  
cbrew@ling.ohio-state.edu

## Abstract

Lapata and Brew [8] (hereafter LB04) obtain from untagged texts a statistical prior model that is able to generate class preferences for ambiguous Levin [9] verbs. They also show that their informative priors, incorporated into a Naive Bayesian classifier deduced from hand-tagged data, can aid in verb class disambiguation. We re-examine the parameter estimation of LB04's prior model and identify the only parameter in LB04's prior model that determines the predominant class for a particular verb in a particular frame. In addition, we propose a method for training our classifier without using hand-tagged data. Our experiments suggest that although our verb class disambiguator does not match the performance of the ones that make use of hand-tagged data, it consistently outperforms the random baseline model. Our experiments also demonstrate that the informative priors derived from untagged texts help improve the performance of the classifier trained on untagged data.

## Keywords

Keywords: lexical semantics, verb class disambiguation, Levin verb class, informative priors, untagged corpus, Naive Bayesian classifier

## 1 Introduction

Much research in lexical acquisition has concentrated on verb classification [18, 11, 14, 7]. Many scholars hypothesize that the meaning of a verb determines to a large extent its syntactic behavior, particularly the realization and interpretation of its arguments, and therefore base their verb classification on the relation between verbs and their arguments [5, 4, 9, 16]. Such classifications can capture generalizations over a range of linguistic properties, and therefore can be used as a means of reducing redundancy in the lexicon and for filling gaps in lexical knowledge.

Much of the work on verb classification in NLP has adopted the classification proposed by Levin [9]. Levin [9] argues that verbs that exhibit the same diathesis alternation can be assumed to share certain semantic components and to form a semantically coherent class. Applying this observation inductively, one can use surface syntactic cues to infer a verb's semantic class. In this paper, we focus on the task of verb classification for ambiguous Levin verbs (verbs belonging

to two or more Levin classes). To be exact, given a verb in a particular frame, we want to assign it to one of its possible Levin classes. As noted by Lapata and Brew [8], this is a wide-spread and important disambiguation problem. Levin's verb inventory covers 3,024 verbs. Although only 784 verbs are polysemous verbs, the total frequency of polysemous verbs in the British National Corpus (BNC) is comparable to the total frequency of monosemous verbs (48.4%:51.6%).

Consider the verb *call* in the following two sentences:

1. He *called* me a fool.
2. He *called* me a taxi.

The verb *call* is ambiguous between the class of DUB and GET when occurring in the double object frame. We want to automatically identify *call* as a DUB verb in sentence (1) and a GET verb in sentence (2). The verb class of a particular verb token provides a significant amount of information about the verb. At semantic level, for example, knowing a token's verb class helps determine the thematic role of its arguments [14, 19] and at the syntactic level, it indicates what subcategorization frames and alternations are allowed [9, 6].

Word sense disambiguation (WSD) is usually cast as a problem in supervised learning, where a word class disambiguator is induced from hand-tagged data. The context within which the ambiguous word occurs is typically represented by a set of more or less linguistically-motivated features from which a learning algorithm induces a representative model that performs the disambiguation task. One classifier that has been extensively used is the Naive Bayesian classifier. A Naive Bayesian classifier usually consists of two parts: the prior and the posterior. Lapata and Brew [8] estimate an informative prior over Levin verb classes for a given verb in a given frame, training on untagged texts. Their prior model is able to generate a class preference for an ambiguous verb. Consider the verb *call* again. It is ambiguous between the class of DUB and GET when occurring in double-object frame. Their prior model predicts DUB to be the predominant class. The model's outcome is considered correct since hand-tagged corpus tokens also reveal a preference for the class DUB. To compute the posterior probability, LB04 uses contextual features (e.g. word collocation) extracted from a small hand-tagged corpus. Their experiments demonstrate that the informative priors obtained from untagged texts helps achieve improved disambiguation performance.

The major contribution of LB04 is that it highlights the importance for WSD of a suitable prior derived from untagged text:

- A prior model derived from untagged texts can help find with reasonable accuracy the predominant sense of a given target ambiguous word. Knowing the predominant sense of a target ambiguous word is valuable as the first sense heuristics which usually serve as a baseline for supervised WSD systems outperform many of these systems which take the surrounding contexts into account. McCarthy et al. [12] have recently also demonstrated the usefulness of a prior in WSD. They use parsed data to find distributionally similar words to the target ambiguous word and then use the associated similarity scores to discover the predominant sense for that target word. One benefit of both LB04 and McCarthy’s method is that the predominant senses can be derived without relying on hand-tagged data, which may not be available for every domain and text type. This is important because the frequency of the senses of words depends on the genre and domain of the text under consideration.
- A prior model derived from untagged texts can also help improve the performance of a classifier over a uniform prior. This is exactly what is shown in LB04. However, although the informative priors in LB04 are derived from untagged texts, the posteriors are deduced from hand-tagged data. Using hand-tagged data to derive the posterior probability assumes the existence of such a corpus. However, if there is a hand-tagged corpus, then an empirical prior can be derived from such a corpus. We would expect a prior obtained from a hand-tagged corpus to be more accurate, therefore when combined with some contextual features should yield better performance. In this paper, we want to evaluate the usefulness of priors derived from a large unlabelled corpus or a small hand-labelled corpus.

Two experiments are conducted in this paper. First, we examined the estimation of LB04’s prior model because we suspected that some of its parameters are irrelevant to the ultimate outcome of the decision process. This examination confirmed our suspicion. We identified the only parameter that determines the predominant class and reformulated LB04’s prior model accordingly. Our reformulation shows that LB04’s prior model ignores the identity of individual verbs in determining the predominant class for a particular verb. We implemented LB04’s prior model using data parsed by two different full parsers [2, 1]. Second, we proposed a new way to train the verb disambiguator without relying on a hand-tagged corpus. More precisely, we used examples containing unambiguous verbs in a particular verb class as the training data for the ambiguous ones in that class. In doing so, both our informative priors and posteriors are obtained without using hand-tagged data. This method is available even if we are dealing with an unusual text type. We also tested the usefulness of our informative priors in aiding verb class disambiguation.

## 2 Experiment 1: The Prior Model

### 2.1 LB04’s Prior Model

LB04’s prior model views the choice of a class  $c$  for a polysemous verb  $v$  in a given frame  $f$  as a maximization of the joint probability  $P(c, f, v)$ , where  $v$  is a verb subcategorizing for the frame  $f$  with Levin class  $c$ :

$$P(c, f, v) = P(v)P(f|v)P(c|v, f) \tag{1}$$

The estimation of  $P(c|v, f)$  relies on the frequency of  $F(c, v, f)$ , which could be obtained if a parsed corpus annotated with semantic class information were available. Without such a corpus, LB04 assumes that the semantic class determines the subcategorization patterns of its members independently of their identity:

$$P(c|v, f) \approx P(c|f) \tag{2}$$

By applying Bayes’ rule,  $P(c|f, v)$  is rewritten as

$$P(c|f) = \frac{P(f|c)P(c)}{P(f)} \tag{3}$$

Substituting (3) into (1), LB04 expresses  $P(c, f, v)$  as

$$P(c, v, f) = \frac{P(v)P(f|v)P(f|c)P(c)}{P(f)} \tag{4}$$

### 2.2 Examination of LB04’s Parameter Estimation

To estimate  $P(c, f, v)$ , LB04 has to estimate five parameters:  $P(v)$ ,  $P(f|v)$ ,  $P(f)$ ,  $P(f|c)$  and  $P(c)$ , as shown in (4). However, for a given verb  $v$  in a given frame  $f$ , the value of  $P(v)$ ,  $P(f|v)$  and  $P(f)$  do not vary with the choice of the class  $c$ . If we are only interested in knowing which class  $c$  is the predominant class for a given verb in a given frame, we could simply ignore them. Therefore, it is the value of  $P(f|c)$  and  $P(c)$  that determines the predominant class for the verb. According to LB04,  $P(f|c)$  and  $P(c)$  are estimated as

$$P(f|c) = \frac{F(f, c)}{F(c)} \tag{5}$$

$$P(c) = \frac{F(c)}{\sum_i F(c_i)} \tag{6}$$

With (5) and (6), the value that determines the predominant class for a given verb in a given frame is calculated as

$$\frac{F(f, c)}{F(c)} \times \frac{F(c)}{\sum_i F(c_i)} = \frac{F(f, c)}{\sum_i F(c_i)} \tag{7}$$

The value of the denominator  $\sum_i F(c_i)$  is only a normalizing constant to ensure that we have a probability function. Again, if we are simply interested in which class is the predominant class for a given verb in a given frame, we can ignore it. It turns out that

Rank	Class	F(Class,V-NP)
1	CONT. LOCATION	70,471
2	ADMIRE	66,352
3	HURT	12,730
4	WIPE MANNER	10,294
5	ASSESS	9,872
6	PUSH-PULL	9,828

**Table 1:** Frequency of six classes with V-NP

$F(f, c)$  is the only value that determines the predominant class. According to LB04,  $F(f, c)$  is obtained by summing over all occurrences of verbs that are members of class  $c$  and attested in the corpus with frame  $f$ :

$$F(f, c) = \sum_i F(c, f, v_i) \quad (8)$$

For monosemous verbs,  $F(c, f, v)$  reduces to the number of times these verbs have been attested in the corpus with a given frame. For polysemous verbs,  $F(c, f, v)$  is obtained by dividing the frequency of a verb with the given frame by the number of classes that the verb belongs to when occurring in the given frame.

Note that our reformulation of LB04 does not result in a different model. All we did is getting rid of the parameters of LB04's model that are irrelevant to the decision regarding the predominant class of a verb  $v$  in a frame  $f$ .

Note two facts about the model from LB04:

First, due to the independence assumption, the only parameter that matters for the prior model is  $F(c, f)$ . The identity of a given verb is totally irrelevant. In other words, for a verb  $v$  that is ambiguous between class  $c_1$  and  $c_2$  in a given frame  $f$ , the predominant class for the verb  $v$  is  $c_1$  if  $F(c_1, f)$  is greater than  $F(c_2, f)$  and  $c_2$  otherwise. Table 1 ranks six verb classes according to their frequency of occurring with the transitive frame. For a verb that is ambiguous between any two of the classes listed in Table 1 when occurring in the transitive frame, the preferred class is determined by the rank of the class in the table. For example, both *miss* and *support* are ambiguous between the class ADMIRE and CONT. LOCATION when occurring in the transitive frame. Since  $F(\text{CONT. LOCATION}, \text{V-NP})$  is greater than  $F(\text{ADMIRE}, \text{V-NP})$ , the model selects CONT. LOCATION as the predominant class for both verbs. However, the prevalence in the manually annotated corpus data (BNC) suggests that CONT. LOCATION is the preferred class for *miss* while ADMIRE is the preferred class for *support*. The independence assumption makes it impossible for the model to select the right preferred class for both *miss* and *support*.

The second fact about LB04's prior model is that without our reformulation, LB04 has to estimate  $F(c)$ :

$$F(c) = \sum_i F(c, v_i) \quad (9)$$

$$F(v, c) = F(c)P(c|v) \quad (10)$$

LB04 proposes two ways to estimate the value of  $P(c|v)$ :

1. Equal Distribution: dividing the overall frequency of a verb by the number of classes it belongs to:

$$P(c|v) = \frac{1}{|\text{classes}(v)|} \quad (11)$$

2. Unequal Distribution: distributing a verb's frequency unequally according to class size:

$$P(c|\text{amb\_class}) = \frac{|c|}{\sum_{c \in \text{amb\_class}} |c|} \quad (12)$$

LB04 shows that in selecting the predominant class for a verb in a given frame, for the 34 ambiguous verbs with the transitive frame, its prior model is about 6% better using the equal distribution for the estimation of  $F(c)$  than using the unequal distribution. According to our reformulation of its prior model, the value of  $F(c)$  is totally irrelevant in choosing the predominant class for a verb. There should be no difference in the performance of the prior model between using equal and unequal distribution to estimate  $F(c)$ .

## 2.3 Experiments on the Prior Model

### 2.3.1 Methodology

LB04 used a parsed version of the whole BNC made with GSearch [3], a tool that facilitates the search of arbitrary part-of-speech-tagged corpora for shallow syntactic patterns. It used a chunk grammar for recognizing the verbal complex, NPs and PPs, and applied GSearch to extract tokens matching frames specified in Levin. A set of linguistic heuristics were applied to the parser's output in order to filter out unreliable cues.

**Our implementation** used two sets of frames acquired from the whole BNC using two different statistical parsers. (1) We parsed the whole BNC with Charniak's parser. (2) In addition, we obtained the frame set from Schulte im Walde [18]. This frame set was acquired from the whole BNC using a head-entity parser described in Carroll and Rooth (1998) (hereafter CR). We implemented LB04's prior model (based on our reformulation) using these two separate sets of frames.

We obtained test data from LB04. This test data, summarized in Table 2, consists of 5,078 ambiguous verb tokens involving 64 verb types and 3 frame types<sup>1</sup>. It includes verbs with double object frame (V-NP-NP) (3.27 average class ambiguity), verbs with dative frame (V-NP-PP(to)) (average 2.94 class ambiguity) and verbs with transitive frame (V-NP) (2.77 average class ambiguity).

### 2.3.2 Results of the Prior Model's Performance

We report the results of our implementation of LB04 using accuracy by verb type. This accuracy is the percentage of verb types for which the prior model correctly selects the predominant class. The outcome

<sup>1</sup> The test data we used here is not identical to that used in LB04. It has undergone both additional corrections and systematic adjustments before being released to us.

Frame	Number of Verb Types
V-NP-NP	12
V-NP-PP(to)	16
V-NP	34

**Table 2:** *Test data*

of the prior model is considered correct if the class selected by the prior model agrees with the most frequent class found in the hand-tagged corpus. Table 3 provides a summary of the results for our implementation of LB04’s prior model. We also computed a baseline by randomly selecting a class out of all the possible classes for a given verb in a particular frame<sup>2</sup>.

Parser	Charniak	CR
LB04	53.2%	56.4%
Baseline	39.7%±0.01	

**Table 3:** *Type accuracy for the prior model*

Table 3 shows that our reformulation of LB04’s prior model achieves a better performance (using either set of frame frequency) than the baseline. However, our results are lower than that reported in LB04. LB04’s prior model achieves an accuracy of 74.6%. This may be due to the different test data we used and different parsers we used to obtain frame frequency.

## 3 Experiment 2: Verb Class Disambiguation Using Untagged Texts

### 3.1 Motivation

Recall that LB04 derives the informative priors from untagged texts, but the posteriors from a hand-tagged corpus. In this experiment, we attempt to address this weakness of LB04’s method:

- LB04 does not compare the performance of the Naive Bayesian classifier between using the informative priors derived from untagged texts (IPrior) and the empirical priors derived from a hand-tagged corpus (EPrior). It would be helpful to know if an IPrior outperforms an EPrior estimated from a very small hand-tagged corpus.
- LB04 derives IPrior from untagged texts. If the posteriors can also be deduced without using hand-tagged data, it will free us from our dependence on hand-tagged data for disambiguating Levin verbs. As noted above, Levin [9] has classified verbs according to their syntactic behavior. Verbs that show similar diathesis alternation are assumed to share certain semantic components and to form a coherent semantic class. Neighboring words are not taken into consideration in her verb classification. On the other hand, many scholars have shown that words with similar contextual features, typically neighboring words, are also semantically similar [17, 10]. Faced with these two dif-

<sup>2</sup> We replicated this random selection 100 times and the result reported in Table 3 was obtained by averaging the results on the 100 selections.

class	ambiguous verbs	unambiguous verbs
DUB	<i>call, make, vote</i>	<i>anoint, baptize, brand, christen, consecrate, crown, decree, dub, name, nickname, pronounce, rule, stamp, style, term</i>
GET	<i>call, find, leave, vote</i>	<i>book, buy, cash, catch, charter, choose, earn, fetch, gain, gather, hire, keep, order, phone, pick, pluck, procure, pull, reach, rent, reserve, save, secure, shoot, slaughter, steal, win</i>

**Table 4:** *DUB and GET class*

ferent approaches to identifying semantically similar words, we may ask the following two questions:

- Are the semantic components shared by verbs in a Levin class correlated with their contexts words?
- Can we use the context words of the unambiguous verbs in a particular Levin class to disambiguate the ambiguous verbs in that class?

To perform verb class disambiguation without relying on a hand-tagged corpus, we decided to train our verb class disambiguator using only data containing unambiguous verbs. Consider the verb *call* again, it is ambiguous between the class of DUB and GET when occurring in the double-object frame. However, most verbs in these two classes are not ambiguous, as shown in Table 4. For an unambiguous verb, we know for sure the class it belongs to without even examining the sentences in which it occurs. To disambiguate *call* in a double-object frame, we therefore used all sentences that are identified as double object frame and contain an unambiguous verb in the class DUB as the training data for the class DUB and did the same for the class GET.

### 3.2 Constructing Training Data

We picked all the example sentences containing the relevant unambiguous verbs from Charniak-parsed BNC that are identified as double-object frame, transitive frame or dative frame. We understand that the training data constructed this way is noisy in that some false instances of the target frames are included in the training data. For example, a sentence like *I fed the boy myself* is incorrectly recognized as a double-object frame. Thus the training data we used may potentially have a negative effect on the verb class disambiguator.

### 3.3 Classifier and Feature Space

#### 3.3.1 A Naive Bayesian Classifier

We employed a Naive Bayesian classifier for our disambiguation task. Although the Naive Bayesian classifier is simple, it is quite efficient and has shown good performance on WSD. Another reason for using a Naive Bayesian classifier is that it is easy to incorporate the prior information. Within a Naive Bayesian approach, the choice of the predominant class for an ambiguous verb *v* when occurring in a frame *f* given its context can be expressed as



$$C(v, f) = \underset{c_i}{\operatorname{argmax}} [P(c_i, f, v) \prod_{i=1}^n P(a_1, \dots, a_n | c_i, f, v)] \quad (13)$$

Where  $C(v, f)$  represents the predominant class for an ambiguous verb  $v$  when occurring in a frame  $f$ .  $P(c_i, f, v)$  is the prior probability of the ambiguous verb  $v$  belonging to class  $c_i$  when occurring in frame  $f$  and  $\prod_{i=1}^n P(a_1, \dots, a_n | c_i, f, v)$  is the posterior probability.

### 3.3.2 Feature Space

As common in WSD, we used as features the neighboring words of a target ambiguous verb. We considered 8 different window sizes: L1R1, L1R2, L1R3, L1R4, L2R1, L2R2, L2R3 and L2R4. A window size such as L1R2 represents one word to the left and two words to the right of an ambiguous verb. Neighboring words are lemmatized using the English lemmatizer described in [15].

## 3.4 Results and Discussion

We used the same test data from the first experiment. We compare the performance of six different models. They differ from each other in whether the priors are derived from hand-tagged data and whether the classifier is trained on hand-tagged data.

- **Prior**

- IPrior: The informative priors derived from untagged texts as described in experiment 1.
- EPrior: The empirical priors derived from hand-tagged data. In our experiment, the empirical priors are derived from the test examples.
- UPrior: The uniform priors.

- **Classifier**

- NHTD: The classifier is trained without using hand-tagged data. In our experiment, the training data consists of all the examples containing only unambiguous verbs. The classifier is tested on all test examples.
- HTD: The classifier is trained on hand-tagged data. In our experiment, the classifier is trained and tested using 10-fold cross-validation on the test examples.

The six models we experimented with are as follows: **UPrior+NHTD**, **IPrior+NHTD**, **EPrior+NHTD**, **UPrior+HTD**, **IPrior+HTD** and **EPrior+HTD**<sup>3</sup>.

For the purpose of comparison, we also report the performance of three different baseline models:

- **Random Baseline (RB)**: We randomly selected a class from all those that are compatible with the given verb and frame. Selection was based on a uniform distribution.

<sup>3</sup> We also estimated a prior from the unambiguous examples only, but its performance is about the same as the IPrior.

model	average accuracy	highest accuracy
UPrior+NHTD	58.1%	64.8%(L1R3)
IPrior+NHTD	62.3%	68.8%(L1R4)
EPrior+NHTD	64.1%	71.0%(L2R4)
UPrior+HTD	64.2%	72.3%(L1R4)
IPrior+HTD	64.9%	73.9%(L2R4)
EPrior+HTD	68.9%	77.4%(L2R4)
RB		37.9%
IPB		57.9%
EPB		74.2%

**Table 5:** Results for verb class disambiguation

- **IPrior Baseline (IPB)**: We selected the class whose IPrior was the largest of the available possibilities.
- **EPrior Baseline (EPB)**: We selected the class whose EPrior was the largest of the available possibilities.

The results are summarized in Table 5. The average accuracy was obtained by averaging the accuracy over all 8 window sizes. We also report the highest accuracy and the window size where the highest accuracy were achieved. For example, using the window size L1R3 (see Table 5) the model UPrior+NHTD achieves its best performance of 64.8%. Several things are worth noting in the result Table 5:

- When the classifier is trained on hand-tagged data (HTD), using IPrior (IPrior+HTD) outperforms using UPrior (UPrior+HTD). This agrees with what is shown in LB04. However, using IPrior (IPrior+HTD) does not match the performance of using EPrior (EPrior+HTD). Both the average accuracy and the highest accuracy for model EPrior+HTD are higher than IPrior+HTD. A pair-wise  $t$ -test indicates that the difference is statistically significant ( $p$ -value = 0.021). This suggests that it is not always best to incorporate a prior derived from untagged texts into a classifier trained on hand-tagged data. It is better to derive a prior from a hand-tagged corpus if such a corpus is available.
- When the classifier is trained without using hand-tagged data (NHTD), neither using UPrior (UPrior+NHTD) nor using IPrior (IPrior+NHTD) performs better than any of the supervised models (HTD). However, they both (UPrior+NHTD and IPrior+NHTD) consistently outperform the random baseline, suggesting that verbs in the same Levin class do tend to share their context words. Our verb class disambiguator using untagged data can be used in the absence of a hand-tagged corpus. In addition, using the EPrior (EPrior+NHTD) achieves a performance comparable to that achieved by the supervised model with a UPrior (UPrior+HTD), suggesting a tagged corpus, if available, helps derive more accurate priors.
- When the classifier is trained without using hand-tagged data (NHTD), using IPrior (IPrior+NHTD) outperforms using UPrior (UPrior+NHTD). A pair-wise  $t$ -test indicates that the improvement achieved by using IPrior is statistically significant ( $p$ -value = 0.026), suggesting that the IPrior derived from untagged data, though not as accurate as the EPrior, can still help improve the performance of the classifier.
- All five models we experimented with outperform the IPB, but fail to achieve the performance of the EPB



with the exception of the model EPrior+HTD, the highest accuracy of which is about 3% better than the EPB. Again, annotation, if available, helps.

## 4 Conclusions and Future Work

The main conclusions of this paper are the following: Our experiments confirm the importance of syntactic frame information in verb class disambiguation. In addition, we have also re-confirmed the importance of a good prior derived from untagged texts in WSD. However, instead of deriving the classifier from hand-tagged data like what LB04 did, we trained our classifier using examples containing unambiguous verbs. This offers us a way to disambiguate Levin verbs without relying on hand-tagged data.

### 4.1 About the Prior Model

A contribution of our paper is a clearer reformulation of LB04's prior model. This reveals that LB04's prior model cannot distinguish between different verbs of the same class. This is a direct result of the independence assumption built into the model. To improve the performance of the prior model, we believe it is worthwhile finding new ways to bring the identity of each individual verb into the prior model [13].

### 4.2 Disambiguation without a Hand-tagged Corpus

We proposed a method for disambiguating Levin verbs that completely avoids the need for a hand-tagged corpus and analysed how it compares to various alternatives. Our experiments show that our verb class disambiguator is not as accurate as the supervised ones that make use of a hand-tagged corpus. One reason is that we relied on a statistical parser for identifying the target frames (double-object, transitive and dative) in constructing the training data. The training data obtained this way is noisy in that some false instances of the target frames are included. On the other hand, the training data (in this case it is the 5,078 test examples) used to train the supervised models has been examined by human annotators and is free of any false instances of the target frames. However, our method of disambiguating Levin verbs without using hand-tagged data consistently outperforms the random baseline, suggesting that it is feasible to use examples containing unambiguous verbs to disambiguate ambiguous ones.

Levin's verb classification covers about 79 frames and many of them involve some ambiguity. In this paper, we only tested our verb class disambiguator on three of Levin's frames. It remains to be shown that it works equally well for other frames. We also plan to test our disambiguation method, namely using unambiguous words to disambiguate ambiguous ones, on different WSD data sets.

## 5 Acknowledgments

This study was supported by NSF grant 0347799. We are grateful to Mirella Lapata for providing the test data. Our thanks also go to Sabine Schulte im Walde for making available to us the frame set she acquired from BNC.

## References

- [1] G. Carroll and M. Rooth. Valence induction with a head-lexicalized PCFG. In *Proceedings of 3rd Conference of Empirical Methods of Natural Language Processing*, pages 58–63, 1998.
- [2] E. Charniak. A maximum-entropy-inspired parser. In *Proceedings of the 2000 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 132–139, 2000.
- [3] S. Corley, M. Corley, F. Keller, M. Cocker, and S. Trewin. Finding syntactic structure in unparsed corpora. *Computers and the Humanities*, 35(2):81–94, 2000.
- [4] A. Goldberg. *Constructions*. University of Chicago Press, Chicago, 1st edition, 1995.
- [5] R. Jackendoff. *Semantics and Cognition*. MIT Press, Cambridge, MA, 1983.
- [6] A. Korhonen. *Subcategorization Acquisition*. PhD thesis, Cambridge University, 2002.
- [7] A. Korhonen, Y. Krymolowski, and Z. Marx. Clustering polysemic subcategorization frame distributions semantically. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 48–55, 2003.
- [8] M. Lapata and C. Brew. Verb class disambiguation using informative priors. *Computational Linguistics*, 30(1):45–73, 2004.
- [9] B. Levin. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago, 1st edition, 1993.
- [10] D. Lin. Automatic retrieval and clustering of similar words. In *COLING-ACL 98*, 1998.
- [11] D. McCarthy. Using semantic preference for identifying verbal participation in role switching alternations. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, pages 58–63, 2000.
- [12] D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. Finding predominant senses in untagged text. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 280–287, 2005.
- [13] P. Merlo, E. Joanis, and J. Henderson. Unsupervised verb class disambiguation based on diathesis alternations. manuscripts, 2005.
- [14] P. Merlo and S. Stevenson. Automatic verb classification based on statistical distribution of argument structure. *Computational Linguistics*, 27(3):373–408, 2001.
- [15] G. Minnen, J. Carroll, and D. Pearce. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223, 2000.
- [16] S. Pinker. *Learnability and Cognition: The Acquisition of Argument Structure*. MIT Press, Cambridge, MA, 1989.
- [17] D. Rohde, L. Gonnerman, and D. Plaut. An improved method for deriving word meaning from lexical co-occurrence. *Cognitive Science*, 2004. submitted.
- [18] S. Schulte im Walde. Clustering verbs semantically according to alternation behavior. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 747–753, 2000.
- [19] R. Swier and S. Stevenson. Unsupervised semantic role labelling. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 95–102, 2004.

# Combining Information Extraction and Knowledge Acquisition for Spoken Dialog Systems

Berenike Loos & Hans-Peter Zorn  
European Media Laboratory GmbH  
69118 Heidelberg  
Germany

*firstname.lastname@eml-d.villa-bosch.de*

## Abstract

The manual acquisition and modeling of tourist information (as e.g. addresses of points of interest) for natural language understanding systems are time-consuming and, therefore, expensive. Furthermore, the already encoded knowledge is static and has to be refined for newly emerging sightseeing objects, restaurants or cinemas. The automatic acquisition can support and enhance the manual process and needs to be implemented as a run-time approach in order to deal with information not included in the data or knowledge base of the system.

This paper, therefore, proposes an incremental process for extracting relevant information from the Internet for extending the system's ontology.

context of a specific location. This concept inherits a **has-address** property from a superclass **building**. With the help of this hint and a contextual specification with respect to the location of the user, the corresponding address of the place can be retrieved on the Internet with information extraction methods. In combination with a Web Service navigation assistance the address can be applied for finding the way to the place the user asked for.

The advancement of the described framework in connection with a spoken dialog system (e.g. SmartWeb, as described in Section 3) is that new words can be processed and can be used for information extraction and knowledge acquisition during the time of the user's enquiry.

## Keywords

Ontology Population, Spoken Dialog Systems, Information Extraction, Knowledge Acquisition

## 1 Introduction

In an open-domain spoken dialog system the automatic learning of ontological concepts and corresponding relations between them is essential, as a complete manual modeling of them is neither realistic nor feasible. The real world and its objects, models and processes are continuously changing and so are their denotations.

A viable approach to this challenging problem is to learn ontological instances and property values relevant for a certain user - and only those - incrementally, i.e. at the time of the user's inquiry as described in [18]. Hypernyms of named entities (NEs) that are not part of the speech recognizer lexicon and are hence lacking any mapping to the employed knowledge representation of the language understanding component are to be found in texts from the Internet. With the found hypernym the framework can assign the concept in the system's ontology to add the new NE to the system as an instance.

Once the right concept for adding the unknown NE as an instance is found, the corresponding ontological properties are investigated. E.g. the user asks the question "How do I get to the Auerstein" and the unknown word is **Auerstein**, therefore, the appropriate concept in the ontology might be **restaurant** in the

## 2 Related Work

There are two related but distinct research areas associated with this work: On the one hand, the field of information extraction (IE), which deals with the identification of certain types of information within unstructured text or Web sites; and on the other hand, ontology learning and population, which aims at extending ontologies by finding the proper place for new instances in the ontological hierarchy.

During the last two decades significant progress has been made in IE. Starting in the late 80's and during the 90's the Message Understanding Conferences (MUC) have played a major role in research standardization by putting together a set of corpora and tasks to be completed, employing these standardized corpora and thereby enabling the research community to compare their results. The focus of the MUC community was mainly to extract specific entities like locations or persons from unstructured text. Since the late 90's one can notice a strong diversification of research areas associated with natural language processing (NLP).

One of the early approaches towards IE was the usage of handcrafted grammatical patterns to extract knowledge, like hyponym-hypernym relations, from natural language texts [12]. The main problem with pattern-based attempts at that time was the sparseness of data, which means that these patterns appeared very rarely in common corpora.

Modern applications of such patterns circumvent the sparse data problem by using the Internet as a information source, as demonstrated by [13], [7] and [4].

Soon machine learning techniques were put to work to either learn patterns or to directly extract named entities from textual sources with an a priori set of patterns. To further enhance the precision of the analysis, a variety of methods have been applied including LSA (Latent Semantic Analysis, see [3]), integrating data repositories like WordNet [9] and gazetteers, while further error reduction could be accomplished by combining multiple extraction approaches as shown by [10].

Frameworks like Web->KB [5] aimed at combining information extraction with ontology population. With the emergence of sophisticated machine learning algorithms it was shown that information extraction by itself could be solved by adding more data and thus diminishing the differences between extraction algorithms. Since then the focus has been on integrating and automatizing the extraction process.

Besides statistical approaches and rules referring more to the context there are more linguistic approaches to information extraction. The SProUT information extraction [1] uses linguistic annotation to extract data and later integrates the result by a semantic transformation component. In a similar way construction grammars [24] can be used to extract information from short text snippets. Here as well the transformation to ontology instances takes places in a separate step.

In both cases the mapping between extracted information and the ontology is static rule based, meaning that new types of information will not be integrated into the knowledge base because the algorithms do not know the correct ontology class the information belongs to.

### 3 Our Framework

The framework we propose for incremental ontology learning was integrated as a module into the Smartweb system.

Smartweb [25] is a project which aims at establishing an open-domain spoken dialog system. Here open-domain means, that it needs to cover information which is not encoded in the system so far. Therefore, some kind of ontology learning is needed, which can deal with lacking information in both the speech recognizer lexicon as well as in the internal knowledge base of the system. To find out which information is needed it is necessary to look at the questions a potential user utters. One frequent class of lacking words in the system are those naming recently established locations as e.g. restaurants, shops or cinemas.

Here, it is not only necessary to find the correct classification as well as the appropriate concept in the system's ontology but also to find more information (such as the address of those objects). Generally, one can not depend only on a mercantile directory as people today often rely on search engines for finding information about new locations and do not look up such directories. Therefore, new locations often advertise on their homepage or on more popular lists for their domains.

In Subsection 3.1 the architecture of SmartWeb is described. Subsection 3.2 and 3.3 will explain the speech recognition and natural language understand-

ing components of Smartweb, which are also relevant for our module.

Subsection 3.4 will deal with the integration of the ontology learning module into the overall system.

Subsection 3.5 and 3.6 describe already implemented parts of the module and finally Subsection 3.7 will explain the representation of newly extracted information in the system's ontology.

### 3.1 SmartWeb's Architecture

SmartWeb is built as a 3-tier architecture as shown in Figure 1. The client is running on either a handheld or is integrated into a car or motorbike. The dialog manager is controlling the client and is responsible for speech recognition, language interpretation and turn management. It connects to the Semantic Mediator which acts as an interface to various interactive semantic access components.

The components are a knowledge database server which is used as a fact base, semantic wrapper agents for online crawling of Web pages, a semantic Web Services composer for accessing commercial Web Services and a free text open-domain question answering component. The dialog system creates a semantic representation of the multi-modal user input including gestures. This semantic query is then sent to a Semantic Mediator, which sends the query to all access components that are likely able to answer the question.

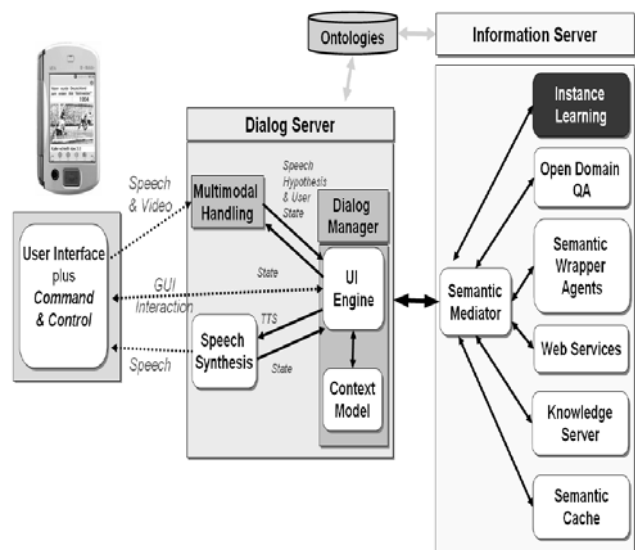


Fig. 1: The SmartWeb Architecture

The semantic access components are designed as Web Services. For communication between components SmartWeb has adopted the W3C EMMA <sup>1</sup> by using RDF Schema instead of basic XML Schema. Semantic access components receive, modify and return SWEMMA documents containing meta-information as well as the semantic representation of the query.

The Semantic Mediator also combines the answers of the different components and returns the combined

<sup>1</sup> see <http://www.w3.org/TR/emma> (last access: 19th July 2007)

semantic representation to the dialog manager. Here the results are preprocessed for presentation on the handheld or motorbike GUI. Additionally, a Text-to-Speech (TTS) component creates speech output.

In the following the speech recognition module of the system is described, which presents the first step of our framework as it delivers out-of-vocabulary information about words from user's utterances.

### 3.2 Speech Recognition

The speech recognizer in SmartWeb classifies all words of the user's utterance not found in the lexicon as out-of-vocabulary (OOV). That means the automatic speech recognition (ASR) system has to process words, which are not in the lexicon of the speech recognizer [14]. A solution for a phoneme-based recognition is the establishment of corresponding best-rated grapheme-chain hypotheses [11]. These grapheme-chains are constructed with the help of statistical methods to predict the most likely grapheme order of a word not found in the lexicon. Those chains are then used for a search on the Internet combined with an automatic spelling correction in the final version of the framework.

### 3.3 Language Understanding

The speech recognizer builds a SWEMMA document, which contains the word-lattices of the speech recognition hypothesis. In the case of OOV tokens, the word lattice contains an n-best list of grapheme-chain hypotheses as well as a confidence score for each grapheme-chain.

The dialog manager is responsible for interacting with the user. After recognition, the speech interpretation dialog manager component (SPIN [6]) will employ transformation rules to a working memory based production system to produce a set of ontology instances representing the utterance.

Within the SmartWeb ontology, names are modeled as separate `denomination` instances which are connected to the corresponding instance by a `has-denomination` slot. Distinct instances of different classes can be assigned the same name and thus are named by the same denomination instance. In the case of OOV words, the speech interpretation creates an instance of a more generic instance based on coarse-grained OOV categories provided by the recognizer. The instance will then be named by a special denomination instance (`OOVDenomination`). This way the unknown word is integrated into the semantic representation (given that the other parts of the utterance are specific enough to allow this). The `OOVDenomination` instance carries all grapheme-chain hypotheses and their scores for further processing. Figure 2 shows the ontological modeling of the description.

Furthermore, the knowledge about the concepts of the other words of the utterance can help to evaluate the results: When there is more than one concept proposal for an instance (i.e. on the linguistic side a proper noun like **Auerstein**) found in the system's ontology, the semantic distance between each proposed concept and the other concepts of the user's question can be calculated as described in [21].

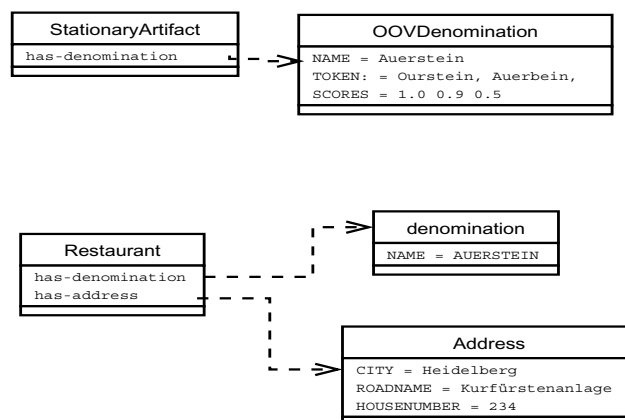


Fig. 2: Representation of out-of-vocabulary words in the semantic representation

### 3.4 Integration of the Ontology Learning Framework into SmartWeb

The goal of integrating the ontology learning framework was, on the one hand, to improve the system's coverage and, on the other hand, not to interfere with other components or to degrade the systems overall performance. This has been achieved by integrating the ontology learning as an additional knowledge source to the SmartWeb system (also referred to as Instance Learning (IL) module in the technical description). As described before, the Semantic Mediator component queries all appropriate knowledge sources for answers to a certain query, which are implemented as Web Services.

After the dialog manager has built a semantic representation from the user query, it invokes the Semantic Mediator Web Service and passes the query as a SWEMMA document. If the Semantic Mediator detects an `OOVDenomination` instance in the query, the IL Web Service will be called in addition to the other knowledge sources as depicted in Figure 3. The instance learning algorithm now tries to resolve the NE encoded in the `OOVDenomination`. If this process was successful, the named entity instance in the query is replaced by the result of the IL process: a more specific and enriched set of ontology instances.

The IL module now restarts the whole answer search process by building a cascade. The enriched SWEMMA document is passed again to the Semantic Mediator Web Service, which then queries the other information sources with this enhanced query. This is done in parallel while the old information search is probably still running.

In case one of the knowledge sources can answer the semantically enhanced query, this result is passed back to the Semantic Mediator cascade, which in turn will deliver this result to the IL. The IL then takes this result and passes it back to the first mediator.

If in the meanwhile one of the knowledge sources was able to provide answers to the original, underspecified query, the results from the IL module will be simply discarded by the dialog manager. If on the other hand, only the IL module could deliver the answer, this result will be presented to the user.

Of course, the second stage information search running in parallel to the first information search will consume computing resources. However, the IL Web Service could be running on a separate server. In this case, the whole second chain of information search will not degrade the normal overall system performance.

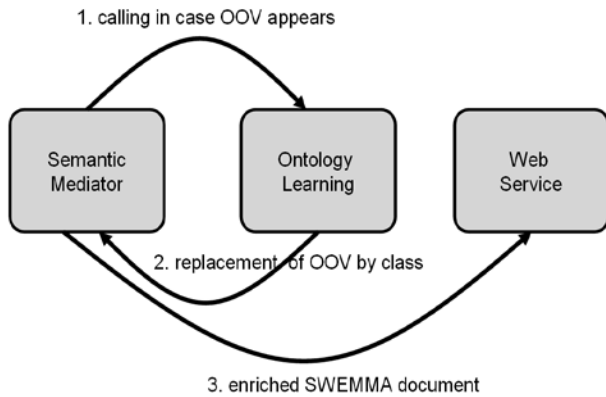


Fig. 3: Calling the Ontology Learning Module

### 3.5 Classification of Named Entities

The task of classifying NEs we published in [8] was solved with a combination of text mining approaches using structural and non-structural features. We used POS-Tags of all tokens in a sequence surrounding the NE and Chunk-Tags of all tokens in a sequence including the NE's Chunk-Tag. As boolean features we used the information if the NE appears in the document's title, if a hypernym candidate appears in the document's title, if the NE is part of a lexico-syntactic pattern (see [12]) and if a word hints towards coreference.

In an initial test we tested more than a dozen algorithms and variants and selected the most promising ones for further analyzes. The more suitable ones were the following taken from the WEKA [26] ML library, namely, the Averaged One-Dependence Estimators (AODE), which averages over various Bayesian learners, the Alternating Decision Tree (ADTree), J48 and the Naïve Bayes tree (NBTree).

The employed ML framework allows for altering the behavior of some of the different classifiers. In those cases we included some of the different variations into our evaluation.

With the help of this set of features we obtained a precision of nearly 60 % for the Bayesian learner Averaged One-Dependence Estimator (AODE).

In an approach to cluster the resulting web pages with the help of the non-hierarchical Single-Link [16] and the Clique algorithm [15] for semantic similarity as described in [20] we obtained more fine grained results. It appeared that for a threshold value of 0.5 the results of Clique outperformed Single-Link considerably as well as for the recall.

As soon as an appropriate classification for the OOV word can be found in the system's ontology, relevant slots are distinguished and used to find more information about the new instance to fill corresponding on-

tological slots. The following subsection will describe, how to find this kind of information.

### 3.6 Information Search

As soon as the corresponding ontological concept (in the named example **restaurant**) can be found in the system's ontology with the help of term widening techniques it can be integrated into the knowledge base as an instance of the found concept. For term widening we applied both a machine-readable thesaurus and the linguistic information contained in a meta-ontology as described in [2]. The direct and inherited properties of this concept are then analyzed to receive hints for further information extraction and knowledge acquisition.

In case the named entity is classified as a **restaurant** in the ontology, the corresponding properties indicate that a **restaurant** is a **building** and that all **buildings** have **addresses**. Therefore, the address of such a new instance can be retrieved and integrated into the knowledge base.

In [19] we demonstrated that unsupervised tagging substantially increases performance of a supervised learning for address extraction. As the unsupervised learning method is in need of large amounts of data, we used a list with about 20.000 Google queries each returning about 10 pages to obtain an appropriate amount of plain text. After filtering the resulting 700 MB raw data for German language and applying cleaning procedures we ended up with about 160 MB totaling 22.7 million tokens. This corpus was used for training the unsupervised tagger.

For performing the supervised task, we trained the MALLET tagger [22], which is based on Conditional Random Fields (CRFs, see [17]). Features per instance for the CRF were the word itself, the relative position to the target name and the unsupervised tag.

### 3.7 Representing the Extracted Information in the Ontology

According to the SmartWeb Ontology SWintO [23], it is possible that in some cases there exists the corresponding concept to a hypernym. This can be discovered with the help of term widening and a word-to-concept lexicon. The concept labels in the SmartWeb Ontology are generally English words. Therefore, the found German hypernym has to be translated into English. An English thesaurus is used to increase the chance of finding the right label in the ontology. The OOV word can then be added as an instance of the corresponding concept.

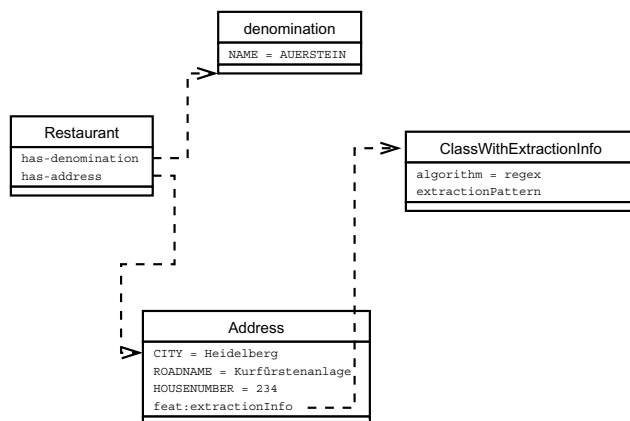
With the help of the information extraction component the properties of the newly learned instance can then be filled with the appropriate information. E.g. the property **has-address** of the instance "Auerstein" can be filled with the address found on the Internet.

## 4 Conclusion and Future Work

The evaluation of the different components showed, that the task of creating an incremental ontology

learning system is generally viable. The actual integration into the overall system showed, that the module significantly helps to find better answers for particular types of questions.

The next step in the establishment of the ontology learning framework is to set up a decision making process about which information extraction patterns are needed for the classified instances. Another important step will be the overall evaluation of the Smartweb system and the gain in dialog efficiency due to the integrated ontology learning framework.



**Fig. 4:** Proposed Representation of Extraction Information in the Ontology

To promote integration of information extraction with the ontology infrastructure would be beneficial for ontology engineers. Each class within the ontology would be annotated with the necessary information about how to find and extract instances of this class from text. This could be done analogous to LingInfo [2] by defining a metaclass and encoding the extraction metadata in instances thereof. Figure 4 shows the idea of embedding meta information in the knowledge base.

## References

- [1] P. Buitelaar, P. Cimiano, S. Racioppa, and M. Siegel. Ontology-based information extraction with soba. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 2321–2324. ELRA, MAY 2006.
- [2] P. Buitelaar, T. Declerck, A. Frank, S. Racioppa, M. Kiesel, M. Sintek, R. Engel, M. Romanelli, D. Sonntag, B. Loos, V. Micelli, R. Porzel, and P. Cimiano. Linginfo: Design and applications of a model for the integration of linguistic information in ontologies. In *Proceedings of the OntoLex06 Workshop at LREC*. Genoa, Italy, 2006.
- [3] S. Cederberg and D. Widdows. Using isa and noun coordination information to improve the precision and recall of automatic hypernymy extraction. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL*, 2003.
- [4] P. Cimiano and S. Staab. Learning by googling. In *SIGKDD Explorations*, 2004.
- [5] M. Craven, D. DiPasquo, D. Freitag, A. K. McCallum, T. M. Mitchell, K. Nigam, and S. Slattery. Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118(1/2):69–113, 2000.
- [6] R. Engel. SPIN: Language understanding for spoken dialogue systems using a production system approach. In *Proceedings of the International Conference on Speech and Language Processing 2002*, Denver, USA, 2002.
- [7] R. Evans. A framework for named entity recognition in the open domain. In *Proceedings of Recent Advances in Natural Language Processing*, Borovetz, Bulgaria, 2003.

- [8] A. Faulhaber, B. Loos, R. Porzel, and R. Malaka. Towards understanding the unknown: Open-class named entity classification in multiple domains. In *Proceedings of the OntoLex Workshop at LREC*. Genoa, Italy, 2006.
- [9] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Mass., 1998.
- [10] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named entity recognition through classifier combination. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL*, 2003.
- [11] F. Gallwitz. *Integrated Stochastic Models for Spontaneous Speech Recognition*. Logos, Berlin, 2002.
- [12] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING*, Nantes, France, 1992.
- [13] A. Kilgarriff. Web as corpus. In *Proceedings of Corpus Linguistics*, 2001.
- [14] D. Klakow, G. Rose, and X. Aubert. Oov-detection in a large vocabulary system using automatically defined word-fragments as filler. In *Proceedings of EUROSPEECH'99*, Budapest, Hungary, 1999.
- [15] I. Koch. Enumerating all connected maximal common subgraphs in two graphs. *Theoretical Computer Science*, 250(1-2):1–30, 2001.
- [16] G. Kowalski. *Information Retrieval Systems: Theory and Implementation*. Kluwer Academic Publishers, USA, 1997.
- [17] J. Lafferty, A. K. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of ICML-01*, 2001.
- [18] B. Loos. On2l - a framework for incremental ontology learning in spoken dialog systems. In *Proceedings of the Student Research Workshop at ACL*, 2006.
- [19] B. Loos and C. Biemann. Supporting web-based address extraction with unsupervised tagging. In *Studies in Classification, Data Analysis, and Knowledge Organization*. Springer, 2007.
- [20] B. Loos and M. DiMarzo. A two-stage approach for context-dependent hypernym extraction. In *Studies in Classification, Data Analysis, and Knowledge Organization*. Springer, 2007.
- [21] B. Loos and R. Porzel. Towards ontology-based pragmatic analysis. In *Proceedings of SIGDial*, Cambridge, Massachusetts, USA, 2004.
- [22] A. K. McCallum. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>, 2002.
- [23] D. Oberle, A. Ankolekar, P. Hitzler, P. Cimiano, M. Sintek, M. Kiesel, B. Mougouie, S. Vembu, S. Baumann, M. Romanelli, P. Buitelaar, R. Engel, D. Sonntag, N. Reithinger, B. Loos, R. Porzel, H.-P. Zorn, V. Micelli, C. Schmidt, M. Weiten, F. Burkhardt, and J. Zhou. Dolce ergo sumo: On foundational and domain models in smartweb integrated ontology (swinto). In *Journal of Web Semantics: Sci. Services Agents World Wide Web*, 2007.
- [24] R. Porzel, V. Micelli, H. Aras, and H.-P. Zorn. Tying the knot: Ground entities, descriptions and information objects for construction-based information extraction. In *Proceedings of the OntoLex Workshop at LREC*, pages 35 – 40, Genoa, Italy, May 2006.
- [25] W. Wahlster. SmartWeb: Mobile applications of the semantic web. In *Proceedings of Informatik*, Ulm, Germany, 2004.
- [26] I. H. Witten and E. Frank, editors. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, Secaucus, NJ, USA, 2005.

# Corpus-driven Enhancement of a BCI Spelling Component

Tanja Mathis<sup>1</sup> and Dennis Spohr<sup>2</sup>

<sup>1</sup>Dept. of Medical Psychology and Behavioural Neurobiology  
Eberhard Karls University  
Tübingen, Germany  
tanja.mathis@med.uni-tuebingen.de

<sup>2</sup>Inst. for Natural Language Processing  
University of Stuttgart  
Stuttgart, Germany  
spohrds@ims.uni-stuttgart.de

## Abstract

We present a general methodology to enhance the spelling component of brain-computer interfaces by deriving a computational *trie lexicon* from word forms extracted from a corpus. The theoretical framework of our approach is provided by the cohort theory [10]. Although the data structure and theory themselves are not subject of current research in NLP as such, it has – to our knowledge – not been attempted to implement a scalable combination of the two so far. We have evaluated our method on a simulation of the *P300 matrix speller* GUI with German texts from four different genres.

## Keywords

Brain-computer interfaces, data structures, psycholinguistics, cohort theory, mental lexicon

## 1 Introduction

The work described in this paper is to contribute to current research that aims at providing communication facilities for people suffering of severe or total motor paralysis – as triggered by injuries to the spinal cord or by degenerative neuromuscular diseases such as amyotrophic lateral sclerosis (ALS) – and which are thus unable to communicate. In order to compensate this loss of communication, brain-computer interfaces (BCI; see e.g. [6]) have been developed which evade the need of voluntary muscle control by directly interlinking the brain with a computer. For instance, these interfaces utilise electric brain activity in order to control a cursor or select characters on a screen. This task is rather time-consuming ([11] report possible selection times between *15s* and *10min* for the *P300 matrix speller* [4]), and therefore current research is investigating possibilities in order to reduce these times.

While these primarily focus on technical improvements of the BCIs themselves, we propose a linguistically motivated enhancement which is based on lexical information derived from corpora. In particular, we build a computational lexicon in a *trie* data structure (cf. [1]) from word lists that are extracted from a large collection of German newspaper texts (more than 200 million tokens). During the character selection process, certain subtrees in the trie are activated – similar to the so-called *cohorts* in Marslen-Wilson and Welsh's *cohort theory* [10]. This way, we achieve automatic addition of substrings of a word as soon as the activation algorithm encounters a single-branching

node, which may incrementally lead to completion of the entire word. However, we emphasise a fundamental caution that any method which aims at being an enhancement in this field has to take into account, namely that making errors has far-reaching effects. For a patient, having to correct errors is even more time-consuming than the actual selection of a character, and thus, even slight mistakes by a “supporting” device might lead to its being completely rejected by the patient. So the primary goal of this approach is not to optimise the number of characters that can be added in a word (although this is certainly a desired achievement), but to provide support that is *error-free* – although absolute correctness is without a doubt too unrealistic a goal. We thus exclude any statistical information (such as word frequency) from the lexicon creation, since statistical models are rather prone to making errors, and put up with an expected lower performance wrt. completion.

In Section 3, we provide more detailed background on BCIs and the trie data structure. Section 4 focuses on technical aspects as to the methodology and implementation, and Section 5 evaluates our approach on German texts from different genres. We conclude in Section 6.

## 2 Related work

We believe that it is necessary to distinguish our work from remotely related approaches found in the literature, such as spelling correction systems or the commercial system T9<sup>®</sup> (“Text on 9 keys”<sup>1</sup>) known from mobile phone technology. While both are obviously targeted towards very different groups of people, T9<sup>®</sup> differs from our approach in a number of very fundamental respects: (i) it has been designed for the purpose of accommodating more than 30 different characters onto a keyboard of only nine keys, which means that a single key corresponds to several different characters; (ii) the number of keys that have to be pressed in order to produce a word always corresponds to (at least) the number of characters that the word contains; (iii) T9<sup>®</sup> does not attempt to complete entire words or character subsequences; and (iv) our approach has its theoretical justification in psycholinguistics. In addition to this, the target group is more forgiving with errors made by the system, and thus the aim of the approach is rather different since the overall benefit is able to “outscore” the occasional need for correction.

As far as spelling correction is concerned (cf. e.g. [3]), we believe that these systems have very different goals in mind, since we aim at minimising the effort needed in order

<sup>1</sup> <http://www.tegic.com/>

for a patient to produce a word, as opposed to correcting typologically malformed input from users.

Our approach is more closely related to that of Li and Hirst, who propose an approach to interactive word-completion enriched with semantic knowledge (see [7]). However, this *interaction* is also what distinguishes their approach from ours. Proposing alternatives – which is what they do to users with linguistic disabilities, and for which they report a very high rate of keystroke saving – does not work quite the same way with patients who have almost completely lost their ability to communicate. Therefore, proposing alternatives and prompting the patient to pick one of these is not applicable in our context.

Probably most closely related to our approach is command line completion known e.g. from UNIX shells, differs however fundamentally wrt. theoretical background, implementation – character addition is not triggered by an additional keystroke (cf. TAB key) but performed automatically, and no alternatives are displayed – and finally scale of the application.

## 3 Background

### 3.1 Brain-computer interfaces

BCIs support people who have lost their ability to speak due to severe or total motor paralysis. In order to provide a communication facility without the contribution of muscles, a BCI is based on electrical brain activity originating from the patient. These measured signals are scanned according to a specific brain activity pattern which is elicited time-locked to presented stimuli (or behavioural responses), and which then is utilised as a trigger to select characters on a computer screen. The P300 matrix speller BCI (cf. [11]; [4]) works on the basis of involuntarily elicited (cognitive) event-related potentials, which – in the case of a P300 component – are characterised as positive peaks occurring 300ms after each stimulus onset. Appropriate rare task-relevant events, which are needed to measure a P300 response, are presented visually on a screen within a 6x6 symbol matrix consisting of cells labelled with alphanumeric characters and a blank space. Randomly, the rows and columns of the matrix flash with every matrix element being highlighted 30 times. Since each flashing of the focussed and simultaneously counted symbol activates a P300 response, while other illuminated cells do not influence the brain activity in that way, the interface is capable of identifying the selected character. By passing further runs, the patient is able to communicate character by character.

### 3.2 Trie data structure

Our approach is theoretically grounded on the psycholinguistic *cohort theory* (see [10]; [8]), which describes the different stages and functions involved in the human word recognition process. Mainly, a speech signal is to be related to wordforms in the mental lexicon, where all words sharing the uttered word-initial acoustical pattern are activated and form a preliminary word-initial cohort. By continuous comparison of activated candidates with auditory input during word recognition, the cohort is incrementally reduced up until the *recognition point*, at which either the word is

safely identified among the members of the cohort or only a single candidate remains [9].

The trie data structure not only in a way simulates the described processes but also represents ideal technical properties wrt. the dense amount of lexical data and its high demands on the efficiency of storage and retrieval (see section 4.2). The special tree structure is characterised by the fact that the key values represent the initial part of the data themselves. Therefore, it is particularly suited for efficient retrieval concerning character sequences, since the first characters of the respective string represent the query term. Concerning the psycholinguistic background, accessing wordforms is provided by traversing the trie according to the character sequence given by the user (cf. key value), which specifies the path to a subtree in the trie that represents the current cohort, i.e. the activated word candidates. As long as several possible branches starting from the root of this subtree exist, further disambiguating input by the user is needed. In case of a single edge, or even unique paths, the respective characters are incrementally added to the previous user input, thus extending the notion of secure identification of word candidates in terms of the cohort theory to secure substring identification and completion.

## 4 Methodology

### 4.1 Lexicon acquisition

**Corpus extraction.** The basis of our lexicon consists of lists of different types of word forms extracted from a collection of German newspaper corpora containing more than 200 million tokens. In order to further increase its coverage, we also extracted lemma information from the corpora – since most of the hapax legomena in the list were inflected – and unified them with those words already in the list. The thus obtained list counts 2,465,172 types, excluding numbers and any kind of punctuation, as these items can only have negative impact on the spelling component.

**Noise reduction.** Naturally, the extracted word lists contain a large amount of noisy data, such as typographical errors (e.g. “*Aafang*” vs. “*Anfang*” (*beginning*)). These certainly affect the performance of the spelling component, since on the one hand they may block the completion of a word, while on the other they could lead to wrong completions. In order to reduce the amount of noise in the data, we had the word list checked with a finite-state morphological analyser (see [5] for details); if the analyser returned at least one possible analysis for a word, the respective word was kept in the list, whereas items that did not receive analyses were deleted. This process, in combination with deletion of upper-case duplicates, reduced the size of the list by more than 800,000 items. Although we are aware of the fact that the morphological analyser does not provide complete coverage and that hence not all of the deleted items are actual typographical errors, we thus still ensure that the resulting lexicon (1,651,471 different upper-case items) is of much higher quality than the raw list that had been extracted from the corpora.

### 4.2 Data structure and activation

**Trie implementation.** The trie data structure for storing the lexicon has been implemented with two major goals in



mind, namely to optimise access and selection time while requiring as little main memory as possible. We therefore implemented the trie as a series of integer records in a binary file format, with records corresponding to *nodes* in the trie, and integers in a record representing *edges* starting from the respective node. More precisely, each record contains an array of 30 unsigned integers: 26 upper-case characters, 3 German umlauts ('Ä', 'Ö', 'Ü') and a special end-of-word marker ('^'). The position of an integer  $x$  in the array corresponds to one of these characters, while the value of  $x$  indicates the byte address of the continuing record in the binary file. For the standard letters of the alphabet, the index in the array is calculated on the basis of the character's ASCII decimal code, so that e.g. the position corresponding to the letter 'A' is at index 0, while the position corresponding to the letter 'E' is at index 4. Similarly, the byte position of the continuing record/node is calculated by multiplying the integer value with the size of the record, which is 120B. Thus, if e.g. the letter 'B' is encountered (cf. first row in Figure 1), the continuing record is located at  $435,324 \times 120\text{B} = 52,238,880\text{B}$  offset in the binary file, while zeros in the array indicate that there is no continuing edge for the respective character. Therefore, as is indicated in the last row of the figure, continuation from this particular record is only possible via the letter 'D'.

A	B	C	D	E	
1	435324	872629	914168	1113356	...
3	395	68365	73383	79653	...
			⋮		
0	0	0	23	0	...

Figure 1: Sample records of the trie implementation

Although the final size of the trie amounts to 6,188,665 nodes and 7,840,134 edges, character look-up and adding require only imperceptible amounts of time, while main memory usage is kept at a minimum since basically only one record at a time needs to be stored.

**Character look-up and completion.** We briefly discuss the functionality of the node activation process by means of the word "Änderung" (change; see Figure 2).

Basically, each time a character is entered and passed on to the lexicon component, the record representing the respective node is activated and retrieved from the trie. In the first case, this is an array containing all possible continuations of 'Ä', such as 'M', 'N' or 'O'. As soon as 'ÄND' is encountered, the algorithm identifies a *preliminary recognition point*, as there is a unique edge starting from the 'D' in the trie (similar to the last row in Figure 1) and incrementally adds Characters as long as there are only single possible continuations (cf. bold edges and nodes in Figure 2). After this, the user is required to provide further input, since there is more than one way of continuing the current string, such as 'ÄNDERN', 'ÄNDERST' or 'ÄNDERT' (to change, (you) change, (he/she/it) changes). After the letter 'U' is entered by the user, the algorithm correctly identifies that the current string has to be 'ÄNDERUNG' and adds the respective characters.

If an unknown word is entered, i.e. a word that is not represented in the trie structure, it is not necessarily the case that a wrong word is retrieved from the lexicon. For

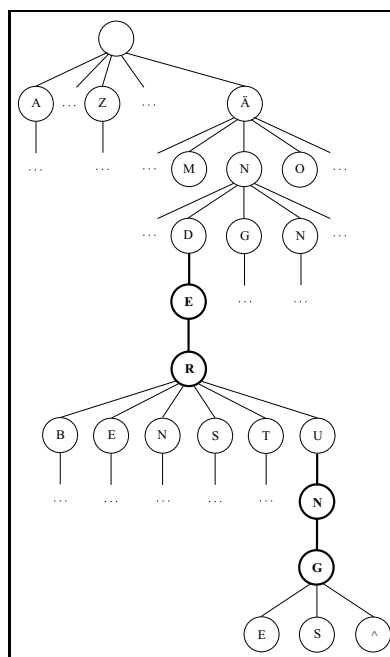


Figure 2: Trie fragment for "Änderung" (change)

example, if 'ÄNDERUNG' and its continuations were not in the lexicon, the algorithm would still add the character sequence 'ER' as in the example above. Since the desired word continues with 'U', no suitable continuing edge would be available in the trie (only 'B', 'E', 'N', 'S' and 'T' would be possible; cf. Figure 2), and thus the user would have to type in the remaining characters on his or her own.

## 5 Evaluation

### 5.1 Text selection and methodology

**Text selection.** In order to cover a rather diverse set of language styles, we have evaluated our method on texts from four different genres, namely *authentic*, *open letter*, *literary* and *newspaper*. The authentic text (80 tokens) has been taken from a letter written with a BCI by a patient suffering from ALS. The second set of texts (15,702 tokens) consists of open letters that have been retrieved from the web irrespective of content. The set of literary texts (106,263 tokens) comprises a collection of 30 randomly selected short stories from which we expect more creative language use. Finally, the fourth set consists of texts (3,989,424 tokens) from a German magazine covering various fields of interest, such as politics, culture and sports.<sup>2</sup> Since newspaper text is easily available in large amounts, the sizes of the evaluation sets differ significantly, and thus direct comparison of the results e.g. between the authentic text and newspaper text is not possible. However, since the authentic text we used represents the only publicly available sample of *real* data from a person suffering from ALS, we still decided to include it in this evaluation – despite its insignificance to the actual evaluation figures.

<sup>2</sup> These newspaper texts were, of course, not included in the set of corpus texts from which the lexicon was built.

		Authentic		Open letters		Literary		Newspaper	
		abs	rel	abs	rel	abs	rel	abs	rel
tokens	Words	80	100.00	15,702	100.00	106,263	100.00	3,898,424	100.00
	Wrong completions	0	<b>0.00</b>	91	<b>0.58</b>	358	<b>0.34</b>	33,235	<b>0.85</b>
	Characters	433	100.00	97,593	100.00	548,198	100.00	23,081,763	100.00
	Characters (user)	394	90.99	83,478	85.54	497,144	90.69	20,119,708	87.17
	Characters (system)	39	<b>9.01</b>	14,115	<b>14.46</b>	51,054	<b>9.31</b>	2,962,055	<b>12.83</b>
types	Words	63	100.00	4,769	100.00	16,157	100.00	221,353	100.00
	Wrong completions	0	<b>0.00</b>	74	<b>1.55</b>	284	<b>1.76</b>	16,231	<b>7.33</b>
	Characters	357	100.00	44,843	100.00	140,578	100.00	2,458,122	100.00
	Characters (user)	324	90.76	35,617	79.43	115,180	81.93	1,889,166	76.85
	Characters (system)	33	<b>9.24</b>	9,226	<b>20.57</b>	25,398	<b>18.07</b>	568,956	<b>23.15</b>

**Table 1:** Evaluation results for authentic, open letter, literary and newspaper text

**Evaluation methodology.** While the authentic text has been reproduced on a simple reimplementation of the GUI that is visible to users of the P300 matrix speller, namely the 6x6 matrix explained in Section 3.1 above, it is not feasible to evaluate large amounts of text in this manner. For the three other text sets, we therefore opted for a fully automated evaluation procedure that directly accesses the trie lexicon. So for every word that is looked up in the lexicon, the number of characters that (would) have been entered by the patient as well as the number of characters added by the system are counted. In case the lexicon added a character that does not match the next character in the current word, this word is marked as wrongly completed and the process resumes with the next word. Therefore, if e.g. the word in the text is 'ALMSICK' (surname of a German swimmer) but the word 'ALMSIEDLUNG' (alp settlement) is retrieved from the lexicon, then only the wrong completion is counted – not the number of characters that have been added. Moreover, punctuation is not taken into account.

## 5.2 Results and discussion

**Results.** Table 1 above displays the results for each individual type of text. The rows that are marked as *tokens* contain the results obtained by simply passing the text on to the lexicon look-up component, while the rows marked as *types* display the results after omission of duplicate words, e.g. 'DER MANN, DER GESTERN AUF DER STRASSE ...' vs. 'DER MANN GESTERN AUF STRASSE'. This was done in order to minimise the effects of highly frequent stop words, such as "und" or "der/die/das" (and, the), and thus to get a better picture of how word completion actually performs. However, the figures for the tokens are of course those that are more important, since these correspond to what the patient experiences in interaction with the system.

Setting a baseline to compare the results against is very difficult in our case, since existing approaches such as the one described in [7] have different aims, and these will definitely outperform our approach in terms of keystroke saving. What we want our system to achieve is an error rate of 0% when processing a text (i.e. *tokens*) as provided by the patient, and this is probably what our approach needs to be evaluated against. Table 2 below shows the overall results irrespective of genre.

**Discussion of results.** The results for the tokens in Tables 1 and 2 show very low rates of wrong word comple-

		Overall	
		abs	rel
tokens	Words	4,020,469	100.00
	Wrong completions	33,684	<b>0.84</b>
	Characters	23,727,987	100.00
	Characters (user)	20,700,724	87.24
	Characters (system)	3,027,263	<b>12.76</b>
types	Words	242,342	100.00
	Wrong completions	16,589	<b>6.85</b>
	Characters	2,643,900	100.00
	Characters (user)	2,040,287	77.17
	Characters (system)	603,613	<b>22.83</b>

**Table 2:** Overall evaluation results for the four text sets

tions, performing best on the authentic and literary texts (0.00% and 0.34% respectively), and with an overall rate of only 0.84%. Conversely, the number of automatically added characters is highest for the open letters and newspaper texts, resulting in an overall rate of system-given characters of 12.76%. When minimising the effects of frequent stop words, the rates of both wrong word completions and automatically added characters rises, showing the most significant changes for the largest text sample, of course.

An analysis of the mistakes reveals that many of these have been caused by proper names (cf. Table 3 below). Of the 50 most frequent word completion errors, at least 41 can be classified as proper names, in addition to eight foreign language terms and one abbreviation. Table 4 shows that of a sample of 61 wrong word completions taken from a preliminary study (cf. [2]), 16 have been caused by morphologically related wordforms absent from the lexicon. For example, the philosophical term "Weltenwanderer" (traveller between worlds) was evoked by the character sequence "Weltenwander", although "Weltenwanderung" (travelling between worlds) was intended in the text. The analyses are shown in Tables 3 and 4 below.

## 6 Conclusion

We have presented an efficient implementation of a cohort-based approach to substring completion and have shown how the application of this method to a BCI spelling component is able to significantly improve its performance. We have evaluated our approach partly on a simulation of the

Freq.	Word	Freq.	Word
150	MFS	46	WIKTOR
149	MURDOCH	46	LUBBERS
137	BENETTON	46	KEEGAN
134	PIECH	45	JAGGER
115	BISKY	45	COMPUSERVE
91	HÖPPNER	44	KIESLOWSKI
85	AMERICAN	44	HEMINGWAY
78	HYDAC	44	CUTLER
74	KUWEIT	43	WHITEWATER
68	ALMSICK	42	GORDIMER
65	BRITISH	42	DORMANN
62	NICHOLS	42	COMPACT
59	SILICON	41	KURINS
57	HARVARD	40	PROCEDO
55	WEINRICH	40	LIFE
54	WIECZOREK	39	KAPOR
54	JONES	38	UPDIKE
53	MILLER	38	TSCHERNOMYRDIN
53	FOSTER	38	GOMBRICH
49	SCIENCE	37	GAMBINO
49	HAVEMANN	36	VÖLKE
49	GORAZDE	36	VALLEY
48	WHITE	36	SOMBART
48	PANDOSCH	36	NÖLDNER
48	LOVE	36	GIRLS

**Table 3:** 50 most frequent errors in word completion

Type of error	Number
Proper names	19
Compounds	17
Specialised terminology	13
Creative language use	5
Foreign language material	4
Absent entry	45
Absent inflected form	16

**Table 4:** Distribution according to error type

P300 matrix speller GUI and partly by automated evaluation procedures. The results have shown that on average every eighth character was added automatically, while retaining a very low rate of 0.84% wrong word completions.

The methodology can be easily applied to languages other than German, and preliminary experiments with an English lexicon – which was, however, only a quarter of the size of the German lexicon and did not undergo the noise reduction steps discussed in Section 4.1 – produced a rate of automatically added characters of 4.09% at a very low error rate of only 0.16% for a sample of roughly one million tokens from the Wall Street Journal. In the future, we will further investigate possibilities to try and completely eliminate errors produced by the system, e.g. by increasing the coverage of named entities. A further possibility would be to reduce the errors caused by missing inflectional or derivational forms by extending the coverage of the lexicon with a morphological generation component, which would generate derivations and missing forms in the inflectional paradigm from the stem of the form that had been present in the corpus. These steps are further improvements we will consider before evaluating our method in real-life situations.

## Acknowledgements

We would like to thank the four anonymous reviewers for their valuable comments and suggestions.

## References

- [1] A. V. Aho, J. E. Hopcroft, J. Ullman, J. D. Ullman, and J. E. Hopcroft. *Data Structures and Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1983.
- [2] T. Becker. *Applying the Cohort Theory to a BCI Spelling Component*. Diploma thesis. Institute for Natural Language Processing, University of Stuttgart, Germany, 2005.
- [3] S. Cucerzan and E. Brill. Spelling Correction as an Iterative Process that Exploits the Collective Knowledge of Web Users. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Barcelona, Spain, 2004.
- [4] L. A. Farwell and E. Donchin. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, 70(6):510–523, December 1988.
- [5] A. Fitschen. Ein computerlinguistisches Lexikon als komplexes System, volume 10(3) of *Arbeitspapiere des Instituts für maschinelle Sprachverarbeitung*. IMS, University of Stuttgart, Germany, 2004.
- [6] A. Kübler, B. Kotchoubey, J. Kaiser, J. Wolpow, and N. Birbaumer. Brain-Computer Communication: Unlocking the Locked In. *Psychological Bulletin*, 127:358–375, 2001.
- [7] J. Li and G. Hirst. Semantic knowledge in word completion. In *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*, Baltimore, MD, USA, 2005.
- [8] W. D. Marslen-Wilson. Functional parallelism in spoken word-recognition. *Cognition*, pages 71–102, 1987.
- [9] W. D. Marslen-Wilson and L. K. Tyler. The temporal structure of spoken language understanding. *Cognition*, 8:1–71, 1980.
- [10] W. D. Marslen-Wilson and A. Welsh. Processing interactions during word-recognition in continuous speech. *Cognitive Psychology*, 10:29–63, 1978.
- [11] J. Mellinger, F. Nijboer, H. Pawelzik, G. Schalk, D. J. McFarland, T. M. Vaughan, J. R. Wolpaw, N. Birbaumer, and A. Kübler. P300 for Communication: Evidence from patients with amyotrophic lateral sclerosis (ALS). In *Proceedings of the 2nd International BCI Workshop*, Graz, Austria, 2004.

# Asymmetric Association Measures

Lukas Michelbacher  
Institute for NLP  
Universität Stuttgart  
*michells@ifnlp.org*

Stefan Evert  
Cognitive Science  
Universität Osnabrück  
*stefan.evert@uos.de*

Hinrich Schütze  
Institute for NLP  
Universität Stuttgart  
*hs999@ifnlp.org*

## Abstract

Human word associations are asymmetric or directed. When hearing a word like *mango*, *fruit* is one of the first associations that come to mind. But when hearing *fruit*, we are more likely to come up with common fruits like *apple* or *orange* than the less frequent *mango*. Similar asymmetry effects have been observed for collocations, recurrent syntagmatic word combinations that are often lexically determined. Despite these intuitions, virtually all corpus-based measures of the statistical association between words are symmetric. In this paper, we propose two asymmetric, directed association measures, viz. conditional probability and a rank measure derived from the  $\chi^2$  test. The goal of this paper is to determine to what extent these two measures of directed “corpus association” can be used as a model for directed “psychological association” in the human mind. Both measures were implemented and applied to a large data set, the *British National Corpus* (BNC). The results were evaluated against directed human association data obtained from the *University of South Florida (USF) Free Association Norms* database. We find that the new measures are able to distinguish between highly symmetric and highly asymmetric pairs to some extent, but the overall accuracy in predicting the degree of asymmetry is low.

## Keywords

Association measures, collocations

## 1 Introduction

Statistical association measures (see e.g. [2]) are commonly used to quantify collocational strength [3], i.e. the tendency of words such as *day* and *night* to “keep each other company”. Although Firth speaks of a *mutual expectancy* between collocates [3] and virtually all association measures are symmetric (i.e. the calculated scores do not depend on the order in which the two words are given), native speakers have strong intuitions that in many cases one term in a collocation is more “important” for the other than vice versa.

Several authors discuss this phenomenon. Sinclair distinguishes between *upward* and *downward collocation* based on the occurrence frequencies of the two collocates [17, p. xxiii]. In a similar vein, Kjellmer [7] distinguishes three kinds of collocations: (i) *right and left predictive* collocations like *aurora borealis*, where

the first word suggests the second as much as the second suggests the first; (ii) *right predictive* collocations, in which the first word suggest the second but not vice versa (e.g. *wellington boots*); and (iii) *left predictive* collocations like *arms akimbo*, where the second word suggests the first but not the other way around. Hausmann’s definition of collocations [5], which focuses on learner dictionaries, distinguishes between *base* and *collocate*. The base of a collocation, typically a noun, retains its regular meaning whereas the collocate is lexically determined and its meaning is modified or weakened. A classic example is *heavy smoker* with base *smoker*. The relation between the two words in such a collocation is clearly directed from base to collocate. Nouns like *smoker* have a small number of typical collocates, whereas *heavy* does not select a particular noun.

Similar asymmetry effects are observed in human intuitions about associated words, which can be measured, e.g., with free association tasks (see Sec. 5.1). For instance, consider the pair (*mango*, *fruit*). When hearing the word *mango*, *fruit* is one of the first associations that come to mind. But when hearing *fruit*, more common fruits like *apple* are more likely to be the first associations rather than a less frequent fruit like *mango*. We call *fruit*  $\rightarrow$  *mango* a *forward association* of the pair (*fruit*, *mango*) and a *backward association* of (*mango*, *fruit*). In this case, the forward association of (*fruit*, *mango*) is weak whereas its backward association is stronger.

We chose the terms *forward association* and *backward association* because we look at a broader class of associations between words than Sinclair, Kjellmer and Hausmann. While most collocations are lexically determined combinations of syntagmatically related words, human associations also include many paradigmatically related words (e.g. *boy* and *girl*). Note that statistical association measures have also been applied to the identification of such paradigmatic relations, in particular synonymy [19] and antonymy [6].

There are several reasons why human associations can be asymmetric. According to prototype theory [15], some members of a category are more prototypical than others. In our example, *apple* is a more prototypical example of fruit than *mango* (at least in North America), so that the directional association *fruit*  $\rightarrow$  *apple* is stronger than *fruit*  $\rightarrow$  *mango*.

Another possible reason for asymmetry is the degree of generality of terms. There is a tendency for a strong forward association from a specific term like *adenocarcinoma* to the more general term *cancer*, whereas the association from *cancer* to *adenocarcinoma* is weak.

We hypothesize that in many other cases the asymmetry is simply caused by frequency effects, corresponding to Sinclair’s concepts of upward and downward collocation. For example, one of the most asymmetric pairs in our evaluation data is (*moo*, *cow*), with a very strong forward and a weak backward association. The word *moo* only occurs in the context of cows, but this is not true vice versa. Words like *milk* and *bull* are more frequent than *moo* in contexts where *cow* is used. This may not be an effect of prototypicality or specificity since the two terms are not related by a relationship such as hyponymy or meronymy.

Despite the strong intuitive support for the widespread existence of directed association provided by such examples, collocation studies still rely on symmetric association measures such as the well-known *pointwise MI*, *t-score* or *log-likelihood*. Our goal in this paper is to propose new measures that take asymmetric association into account and calculate separate scores for forward and backward association. We make use of psychological association norms (in particular, the *USF Free Association Database*, cf. Sec. 5.1) to evaluate how well these measures correspond to human intuitions about directed association.

It is important to distinguish “psychological association”, the association of words in the human mind, which can be measured with reaction times and cue-target experiments, from “corpus association”, the statistical association between terms in corpora. We will test in this paper to what extent directed corpus association can be used as a model for directed psychological association. We expect the results to carry over when the statistical measures are applied to collocation extraction tasks, for which no suitable (directed) reference data are currently available.

The paper is organized as follows. In Section 2, we discuss related work. Then, two asymmetric association measures are introduced in Section 3. Section 4 describes our methodology and corpus data. Section 5 presents results and evaluation, followed by a conclusion in Section 6.

## 2 Related work

In most cases, association measures are not used to examine the asymmetrical aspect of collocations. According to Evert [2, p. 75]:

[t]he scores computed by an association measure can be interpreted in different ways: (i) They can be used directly to estimate the magnitude of the association between the components of a pair type.<sup>1</sup> (ii) They can be used to obtain a ranking of the pair types in the data set. In this case, the absolute magnitude of the score is irrelevant. (iii) They can also be used to rank pair types with a particular first or second component. [...]. I do not go further into (iii), which is closely tied to a “directional” view of cooccurrences and casts an entirely different light on the properties of association measures.

The first two approaches, which are symmetric, are predominant in computational linguistics [2] and sta-

<sup>1</sup> The term *pair type* refers to a representation of a collocation that is independent of surface form.

tistical natural language processing [10]. In order to model the asymmetric relationship between two given words, this work focuses on the *directional* view which Evert [2, p. 27] describes as follows:

An alternative is the “directional” view, which starts from a given *keyword* and aims to identify its *collocates*. [...] the evaluation of directional methods is more complicated and not as clear-cut. So far, published experiments have been limited to impressionistic case studies for a small number of keywords [1, 16, 18].

We are not aware of systematic research on asymmetrical association measures. In their comprehensive survey [13], Pecina and Schlesinger mention the two measures “conditional probability” and “reversed conditional probability”, but do not discuss and evaluate them. Asymmetry has played a more important role in models of distributional similarity (which in turn have sometimes been used to model human associations), and several asymmetric similarity measures have been developed [4, 9, 11, 14]. Since these approaches focus on a different statistical aspect than association measures and cannot be compared directly, we do not go into further detail here.

## 3 The asymmetric association measures

### 3.1 Conditional probability

As our first measure, we use simple conditional probabilities, defined as the ratio between the joint probability of the pair and the probability of either the first word  $w_1$  or the second word  $w_2$ :

$$P(w_2|w_1) = \frac{P(w_1, w_2)}{P(w_1)} \quad P(w_1|w_2) = \frac{P(w_1, w_2)}{P(w_2)} \quad (1)$$

All probabilities are maximum-likelihood estimates without any smoothing.  $P(w_2|w_1)$  is interpreted as a quantitative measure for the forward corpus association  $w_1 \rightarrow w_2$  of the pair  $(w_1, w_2)$ , and  $P(w_1|w_2)$  as a measure for the backward association  $w_2 \rightarrow w_1$ .

Example: In the BNC, the conditional probabilities for the pair (*tomato*, *soup*) are:  $P(\textit{tomato}|\textit{soup}) = 0.03194$  and  $P(\textit{soup}|\textit{tomato}) = 0.05652$  (see Sec. 4 for details of our experimental setup). This conforms with the intuition that the forward association  $\textit{tomato} \rightarrow \textit{soup}$  is stronger than the backward association  $\textit{soup} \rightarrow \textit{tomato}$ .

### 3.2 Rank measure

We chose to base the rank measure on the  $\chi^2$  test because it is a well-established statistical test for association and is easy to implement. Using a different association measure would result in a different rank measure. To compute the rank measure, we first compute the  $X^2$  statistic for each pair  $(w_1, w_2)$  in the corpus data as follows:

$$X^2(w_1, w_2) = \frac{O_{..} \cdot (O_{11}O_{22} - O_{12}O_{21})^2}{O_{1.}O_{.1}O_{2.}O_{.2}} \quad (2)$$

Using standard notation for contingency tables,  $O_{22}$  is the number of cooccurrence pair tokens that do not contain either of the two words,  $O_{12}$  the number containing only  $w_2$ ,  $O_{21}$  the number with only  $w_1$ , and  $O_{11}$  the number with both words;  $O_{1.}$  is the number of tokens containing  $w_1$  regardless of whether they also contain  $w_2$ ,  $O_{.2}$  the number of tokens *not* containing  $w_1$  regardless of  $w_2$ , etc.; and  $O_{..}$  is the total number of cooccurrence tokens.

For each  $w_1$  a sorted association list is created that contains every pair  $(w_1, \cdot)$  together with its association score  $X^2$ , sorted from highest to lowest association score. Then, the  $X^2$  scores are replaced by ranks, i.e. natural numbers starting with 1. Figure 1 shows an example for the words *soup* and *tomato* that illustrates this procedure. The lists have been shortened to show only the relevant data.

If  $m$  consecutive  $w_2$  have the same association score they are assigned the same rank  $r$ , and the  $w_2$  with the next highest score is assigned rank  $r + m$ . We only consider the 1000 highest-ranked words in each list. We denote the rank of  $w_2$  in the  $X^2$  ranking of  $w_1$  as follows:

$$R(w_2|w_1) \quad (3)$$

$R$  is defined in analogy to conditional probability  $P(w_2|w_1)$  which returns the probability of seeing  $w_2$  when  $w_1$  has already appeared. Analogously,  $R(w_2|w_1)$  returns the rank of  $w_2$  in the association list of  $w_1$ . Using the information in Figure 1, the ranks for the example pair (*soup*, *tomato*) can be determined. They are  $R(\textit{tomato}|\textit{soup})$  (“tomato given soup”) = 3 and  $R(\textit{soup}|\textit{tomato})$  (“soup given tomato”) = 10. A lower rank indicates stronger association, hence the rank measure shows a stronger association for *soup* → *tomato* than for *tomato* → *soup*.

We note that the asymmetric rank measure is based on a symmetric association measure, the  $\chi^2$  test. According to the  $\chi^2$  test, the pairs (*mango*, *fruit*) and (*fruit*, *mango*) have the same association strength because the measure is not directed. But *fruit* will figure more prominently in the association list of *mango* ( $R(\textit{fruit}|\textit{mango}) = 10$ ) than vice versa ( $R(\textit{mango}|\textit{fruit}) = 47$ ). The rank measure proposed here uses this type of difference in the associational rankings to transform symmetric  $\chi^2$ -based association scores into asymmetric  $R$  measure ranks.

## 4 Methodology

We selected the *British National Corpus* (BNC) (<http://www.natcorp.ox.ac.uk/>) as a data set for calculating corpus associations. It is a large balanced corpus of approximately 100 million words, containing samples from various genres and sources such as newspapers, popular fiction and scientific journals. Words and punctuation tokens have been automatically annotated with syntactic categories, using the *BNC Basic Tagset*. These annotations make it easy to apply a part-of-speech filter to the cooccurrence data.

In order to extract data that provide information about the corpus association between words, cooccurrence pairs were constructed in the following way: First, words containing special characters (e.g., é, £ or

“/”) and words starting with numbers or other non-letter characters were excluded from the experiment. In addition, words shorter than three characters were ignored. Each word in the corpus was then combined with its ten predecessors as well as its ten successors. A part-of-speech filter was applied, allowing only adjectives, nouns and proper nouns in the pairs. No further linguistic processing was done except for lower-casing of sentence-initial words that were not tagged as proper nouns. Subsequently, all words that occur less than 40 times in the BNC were discarded (together with the corresponding pairs). In this way, a list of 28,149,644 word tokens and 177,913,470 cooccurrence pairs (i.e., not necessarily distinct tokens of word pairs) was obtained.

## 5 Results and evaluation

The asymmetric measures were evaluated using two different methods. First, the forward and backward associations calculated for highly asymmetric and symmetric pairs were calculated. For this purpose, the ten most asymmetric and the ten most symmetric pairs were extracted from a reference set of human “psychological” association (see Sec. 5.1). Second, the ability of the measures to predict the asymmetry or symmetry of given pairs was evaluated on a large set of 5697 word pairs from the reference database.

### 5.1 Reference data

The performance of the two asymmetric association measures is evaluated against word pairs from a database that contains the results of free association experiments. This database, the *University of South Florida Free Association Norms* [12], consists of labeled *cue-target pairs* where a *cue* is a word presented to a subject and the corresponding *target* is the word that the subject wrote down on a blank shown next to the cue. The experiment is described as follows:

Participants were asked to write the first word that came to mind that was meaningfully related or strongly associated to the presented word on the blank shown next to each item. [...] For example, if given BOOK \_\_\_\_\_, they might write READ on the blank next to it. This procedure is called a discrete association task because each participant is asked to produce only a single associate to each word.

Each of the 5,019 cue words is listed in the database together with all the targets that subjects produced for it. For every single cue-target pair, a database entry lists how many subjects were presented the cue and how many of them named each target. Figure 2 shows two abbreviated entries. The number of test persons that were presented a cue word is labeled #G. #P is the number of people that gave a particular target response. The *forward strength* (FSG) is #P divided by #G and the *backward strength* (BSG) is the forward strength of the reversed pair. The terms FSG and BSG were introduced by the creators of the *USF Free Association Norms*.

Our evaluation methodology is most similar to that of [8, 19] who evaluate corpus-derived association measures on a synonym gold standard that reflects human

$w_1$	$X^2$ score	$w_2$	$R(w_2 w_1)$
soup	14666.277	bowl	1
	14531.099	pea	2
	7681.563	tomato	3
	6082.888	kitchens	4
	4116.237	mushroom	5

$w_1$	$X^2$ score	$w_2$	$R(w_2 w_1)$
tomato	8770.224	pepper	8
	8531.046	cucumber	9
	7681.563	soup	10
	7594.471	salad	11
	7417.416	chopped	12

**Fig. 1:** Ranking applied to  $X^2$  scores of the words soup and tomato

CUE	TARGET	#G	#P	FSG	BSG
aardvark	anteater	152	9	.059	.117
anteater	aardvark	145	17	.117	.059

**Fig. 2:** Example from the Free Association Norms

understanding of synonymy. However, their gold standard has a single correct answer (out of a small number of alternatives) for each cue word, rather than a large number of target words with different degrees of (forward) association.

## 5.2 Strong asymmetric associations

The first part of the evaluation is concerned with analyzing the performance of the asymmetric measures for pairs that are highly asymmetric in the reference data set. In order to create a suitable reference list from the association database, cue-target pairs with a high difference between FSG and BSG were extracted. The absolute value of the difference had to be greater than 0.7 for a pair to be selected. In order to make the list comparable to the results that are based on the corpus data, the list had to be filtered: First, all cue-target pairs with BSG 0 were removed. In those cases, the test persons were never presented the target word as a cue word so there is not enough data to determine both FSG and BSG. The second step eliminated parts of speech that were not included during the processing of the corpus data. All pairs containing words that do not occur in the corpus (or did not pass the filters) were eliminated as well.

Figure 3 shows the ten most asymmetric cue-target pairs together with the ranks and conditional probabilities that were computed from the BNC data. Obviously, conditional probabilities correspond much better to the human ratings than the rank measure. FSG exceeds BSG roughly by a factor of 10 for all ten pairs, and the conditional probabilities mirror this relation. In eight out of ten pairs that were evaluated, the ratio between  $P(w_2|w_1)$  and  $P(w_1|w_2)$  is on the same order of magnitude as the ratio between FSG and BSG. In two cases, namely pairs 4 and 8, the results deviate slightly from this pattern. The latter shows a ratio of about 4, the former a ratio of approximately 174.

The comparison between FSG, BSG, and the rank measure is less straightforward. Two quantities have to be taken into account: The difference between  $R(w_2|w_1)$  and  $R(w_1|w_2)$ , as well as the absolute value of  $R(w_2|w_1)$ . First, in order for the rank measure to express that forward association is stronger than backward association,  $R(w_2|w_1)$  must be lower than  $R(w_1|w_2)$ . Second,  $R(w_2|w_1)$  should be small in order

to express *strong* forward association. However, only seven out of the ten pairs satisfy the first condition, and only five of them also meet the second criterion. The other two (number 4 and number 7) have forward ranks of 35 and 47, respectively, which do not indicate strong forward association. Two pairs (numbers 1 and 8) are almost symmetric according to the rank measure and pair number 3 even shows a weak backward association.

Another difficulty with the rank measure is the interpretation of the magnitude of the *rank difference*  $\delta = |R(w_2|w_1) - R(w_1|w_2)|$ . First, it is not clear how large the value of  $\delta$  has to be in order to indicate strong asymmetry and second, it can only be interpreted in combination with the absolute ranks. E.g., word pairs 1 and 9 both have a rank difference of 2, but this difference is arguably more “important” for pair 9 (rank 2 vs. rank 4) than for pair 1 (rank 7 vs. rank 9).

Conditional probabilities correctly predict the direction of the asymmetry in all 10 cases. The rank measure only predicts the correct direction in 7 out of 10 cases.

## 5.3 Strong symmetric association

In addition to the strongly asymmetric pairs discussed in the last section, the reference database contains pairs with symmetric associations, i.e., FSG and BSG are almost equal. Although the measures presented in this work aim at capturing the asymmetry in the human associations, they should also be able to predict word pairs with *symmetric* associations and distinguish them from the asymmetric ones.

Symmetric pairs were extracted from the reference data by selecting pairs with  $|FSG - BSG| < 0.1$  and  $FSG > 0.5$  (in order to remove weakly associated pairs). Again, the list was filtered based on part of speech and occurrence of the words in the BNC data (cf. 5.2). Then the ten most symmetric pairs (i.e. those with the smallest difference between FSG and BSG) were evaluated. The results are shown in Fig. 4.

Symmetry is reflected by the two measures in an entirely different manner than asymmetry. The conditional probabilities did not match FSG/BSG ratios as well as in Section 5.2, while the rank measure achieved slightly better results for strongly symmetric pairs than for asymmetric pairs.

In order for conditional probabilities to express strong symmetric association, their quotient should be close to 1. However, the only word pairs meeting this requirement to some extent are pairs 1 and 4. In all other cases there is a strong discrepancy between

<sup>2</sup> The American English *omelet* appears as *omelette* in the BNC.

No.	$w_1$	$w_2$	FSG – BSG	FSG	BSG	$R(w_2 w_1)$	$R(w_1 w_2)$	$P(w_2 w_1)$	$P(w_1 w_2)$	$\approx \frac{P(w_2 w_1)}{P(w_1 w_2)}$
1	trout	fish	0.877	0.913	0.036	9	7	0.15987	0.01042	15
2	Cheddar	cheese	0.867	0.922	0.055	2	7	0.29906	0.01331	22
3	exhausted	tired	0.82	0.895	0.075	104	87	0.01479	0.00139	10
4	crib	baby	0.81	0.842	0.032	35	69	0.10638	0.00061	174
5	omelet <sup>2</sup>	eggs	0.809	0.836	0.027	3	26	0.16513	0.00504	32
6	wick	candle	0.79	0.841	0.051	3	5	0.08823	0.00807	11
7	teller	bank	0.786	0.814	0.028	47	84	0.07438	0.00099	75
8	bank	money	0.78	0.799	0.019	11	10	0.05767	0.01449	4
9	saddle	horse	0.776	0.879	0.103	2	4	0.11467	0.00997	11
10	bouquet	flowers	0.775	0.828	0.053	1	4	0.21862	0.01108	19

**Fig. 3:** Comparison of strong asymmetric human association with associations computed from corpus data

No.	$w_1$	$w_2$	FSG – BSG	FSG	BSG	$R(w_2 w_1)$	$R(w_1 w_2)$	$P(w_2 w_1)$	$P(w_1 w_2)$	$\approx \frac{P(w_2 w_1)}{P(w_1 w_2)}$
1	boys	girls	0.003	0.500	0.503	1	1	0.17965	0.14873	1.20
2	happy	sad	0.006	0.628	0.634	9	4	0.00725	0.02412	0.30
3	pepper	salt	0.006	0.695	0.701	1	1	0.48230	0.13897	3.47
4	legs	arms	0.008	0.541	0.549	2	1	0.07842	0.04870	1.61
5	bad	good	0.008	0.750	0.758	4	2	0.11129	0.02083	5.34
6	dinner	supper	0.01	0.535	0.545	45	16	0.00455	0.02037	0.22
7	grandma	grandpa	0.015	0.538	0.553	2	3	0.03333	0.1375	0.24
8	negative	positive	0.024	0.603	0.627	1	1	0.20472	0.10928	1.87
9	closing	opening	0.027	0.480	0.507	19	9	0.02495	0.00445	5.60
10	far	near	0.032	0.503	0.535	10	6	0.00898	0.02282	0.39

**Fig. 4:** Comparison of strong symmetric human association with associations computed from corpus data

$P(w_1|w_2)$  and  $P(w_2|w_1)$ , so that the conditional probabilities fail to capture the high symmetry that the reference data suggest.

The performance of the rank measure was better in that it accurately predicted the symmetry of the pairs in six cases (pairs 1, 3, 4, 5, 7 and 8). In three cases, the ranks accurately indicated both perfect symmetry ( $R(w_2|w_1) = R(w_1|w_2) = 1$ ) and a strong association between the two words (because of the low rank). For pairs 2, 9 and 10, high ranks  $R(w_2|w_1)$  indicate that there is no strong forward association. While  $R(w_1|w_2)$  is lower in each case, the backward associations are not strong enough to conclude that the pairs are clearly identified as asymmetric. In particular, pair 10 has quite similar forward and backward ranks and may be considered near-symmetric. For pair 6, the rank measure indicates a clearly asymmetric, but overall weak association.

The rank measure predicts symmetry or near-symmetry (defined as a rank difference  $\delta \leq 5$ ) for 8 of the 10 test pairs. Conditional probabilities perform less well and only predict symmetry or near-symmetry (defined as  $0.5 \leq \frac{P(w_2|w_1)}{P(w_1|w_2)} \leq 2$ ) for 3 out of 10 pairs.

## 5.4 Automated evaluation

To evaluate the two asymmetric measures on a larger scale, we extracted all pairs  $(w_1, w_2)$  that occur in both directions in the USF data set. We then selected the direction with  $FSG > BSG$ . The resulting set was randomly split into a training set consisting of 3000 pairs and a test set consisting of 2697 pairs. We then determined the median of the FSG–BSG values (0.049) and evaluated the asymmetric measures on their ability to predict whether FSG–BSG was  $\geq 0.049$  (intuitively understood as asymmetric pairs) or  $< 0.049$  (under-

stood as symmetric pairs).

We used logistic regression in  $R^3$  for predictive analysis. The response variable is  $FSG - BSG \geq 0.049 / < 0.049$ . Initially, we intended to use either the two ranks or the two conditional probabilities as predictive variables. In preliminary experiments on the training data we found that a log transformation of the predictor variables improved the model. We therefore used the logs of ranks / conditional probabilities as predictors instead of the original variables.

When applied to the test set, accuracies of predicting symmetry ( $FSG - BSG < 0.049$ ) vs. asymmetry ( $FSG - BSG \geq 0.049$ ) were 59% for ranks, 61% for conditional probabilities and 62% for a combination of ranks and conditional probabilities. All three results are significantly different from the baseline accuracy of 50% ( $p < 0.001$ ,  $\chi^2$  test). The three results were not significantly different from each other (e.g.,  $p = 0.4243$  for ranks vs. conditional probabilities,  $\chi^2$  test).

We conclude the following from this evaluation: (i) Both measures contain information about “psychological” asymmetry. (ii) There is no significant difference in accuracy of prediction between the two measures. (iii) Overall accuracy is low. This is partly due to the general difficulty of modeling human judgments with corpus data, but it may also indicate that there are more effective measures of asymmetry than the ones we have investigated here. The data sets are available at <http://ifnlp.org/ranlp07>.

## 6 Conclusion and future work

We introduced two asymmetric statistical association measures that aim to capture the asymmetry of human

<sup>3</sup> <http://www.r-project.org/>



word associations, one based on conditional probabilities and the other on ranks according to an established association measure. Both measures were implemented and applied to a large data set of co-occurrences extracted from the *British National Corpus*. The resulting directed association scores were evaluated against norms obtained from free association tasks with human subjects (the *USF Free Association Norms* database).

We found that the new measures are able to distinguish between symmetric and asymmetric word pairs to some extent, but with a relatively high error rate (62% accuracy vs. 50% baseline). Additional experiments with a small number of highly symmetric and highly asymmetric pairs showed that the measure based on conditional probabilities works well for asymmetric pairs and makes reasonable predictions for the magnitude of the asymmetry. However, its scores for highly symmetric pairs were unreliable and difficult to interpret. The rank-based measure seems more suitable for identifying symmetric pairs. It is also the more robust measure overall, with  $\geq 50\%$  accuracy for both sets.

The work presented here can be extended in many ways. Our evaluation results are encouraging, but show that there is considerable room for improvement. Some extensions are concerned with the definitions of the asymmetric association measures. The maximum-likelihood estimates used by the conditional probability measure could be replaced by smoothed estimates or confidence intervals. Then rank-based measure, which currently uses rankings according to the  $X^2$  statistic, can equally well be based on any other standard association measure. Further research is also needed on the interpretation of rank differences.

Performance of the measures might be improved by working on lemmatized data, which is offered by the new XML edition of the BNC. This would help to abstract over surface forms, thus “tidying up” the association lists and increasing the significance of statistical association. Experiments with different window sizes and filtering constraints can also be performed. Finally, scaling up to much larger Web corpora would further boost statistical significance and produce more reliable association scores.

## References

- [1] K. W. Church and W. A. Gale. Concordances for parallel text. In *Proceedings of the 7th Annual Conference of the UW Center for the New OED and Text Research*, 1991. Quoted in Evert.
- [2] S. Evert. *The Statistics of Word Cooccurrences - Word Pairs and Collocations*. PhD thesis, Institut für maschinelle Sprachverarbeitung (IMS), Universität Stuttgart, 2004.
- [3] J. R. Firth. A Synopsis of Linguistic Theory, 1933-1955. In J. R. Firth, editor, *Studies in Linguistic Analysis*. Blackwell, Oxford, 1957.
- [4] M. Geffet and I. Dagan. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the ACL 2006*. Association for Computational Linguistics, 2005.
- [5] F. J. Hausmann. Le dictionnaire de collocations. In *In Wörterbücher, Dictionaries, Dictionnaires. Ein internationales Handbuch*. de Gruyter, Berlin, 1989.
- [6] J. S. Justeson and S. M. Katz. Co-occurrences of antonymous adjectives and their contexts. *Computational Linguistics*, 17(1), 1991.
- [7] G. Kjellmer. A mint of phrases. In K. Aijmer and B. Altenberg, editors, *English Corpus Linguistics*. Longman, London, 1991.
- [8] T. K. Landauer and S. T. Dumais. A solution to Plato’s problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2), 1997.
- [9] L. Lee. Measures of Distributional Similarity. In *37th Annual Meeting of the Association for Computational Linguistics*, 1999.
- [10] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.
- [11] D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. Finding predominant word senses in untagged text. In *Proceedings of the ACL 2004*. Association for Computational Linguistics, 2004.
- [12] D. L. Nelson, C. L. McEvoy, and T. A. Schreiber. The University of South Florida word association, rhyme, and word fragment norms. <http://www.usf.edu/FreeAssociation/>, 1998.
- [13] P. Pecina and P. Schlesinger. Combining association measures for collocation extraction. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 651–658, 2006.
- [14] V. Pekar. Acquisition of verb entailment from text. In *Proceedings of the Human Language Technology/North American Association for Computational Linguistics (HLT/NAACL-06)*, 2006.
- [15] E. H. Rosch. Natural Categories. *Cognitive Psychology*, 4, 1973.
- [16] J. Sinclair. *Corpus, Concordance, Collocation*. Oxford University Press, Oxford, 1991. Quoted in Evert.
- [17] J. Sinclair, S. Jones, R. Daley, and R. Krishnamurthy. *English Collocation Studies: The OSTI Report*. Continuum Books, London and New York, 2004.
- [18] M. Stubbs. Collocations and semantic profiles: On the cause of the trouble with quantitative studies. *Functions of Language*, 2(1), 1995. Quoted in Evert.
- [19] P. D. Turney. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. *Lecture Notes in Computer Science*, 2167, 2001.

# A Semantic-less Approach for the Textual Entailment Recognition Task

Daniel Micol, Óscar Ferrández, Rafael Muñoz, and Manuel Palomar  
Natural Language Processing and Information Systems Group  
Department of Computing Languages and Systems  
University of Alicante  
San Vicente del Raspeig, Alicante 03690, Spain  
{*dmicol, ofe, rafael, mpalomar*}@*dlsi.ua.es*

## Abstract

In this paper we describe the system we have developed to overcome the textual entailment recognition task without any kind of semantic knowledge. For this purpose we have designed and implemented two modules. The first one analyzes the lexical information extracted from the phrases, while the second studies them from a syntactic perspective. The main goal of this research is to allow us to acknowledge the maximum accuracy that we can achieve without a semantic analysis. To evaluate our system we used the test corpus sets from *Second and Third PASCAL Recognising Textual Entailment Challenges*, obtaining accuracy rates of 62.12% and 65.63%, respectively.

## Keywords

Textual Entailment, Lexical information, Syntactic information.

## 1 Introduction

The field of Natural Language Processing is an essential part of Artificial Intelligence that studies the communication and interaction between human beings and computers. Within this area, Textual Entailment has been defined as a generic framework for modeling semantic variability that appears when a concrete meaning is described in different manners. Throughout this paper we will follow the guidelines proposed in *PASCAL Recognising Textual Entailment*<sup>1</sup> (RTE-1, RTE-2 and RTE-3) [1, 2], which establishes that the meaning of a text snippet (termed hypothesis) should be inferred from the meaning of another one (namely text).

This paper discusses an approach that attempts to detect when the entailment is produced, and focuses on determining if such relation appears due to lexical or syntactic implications between the texts. We propose several methods that mainly rely on lexical and syntactic inferences in order to address the entailment recognition task. The reason why we have decided not to use semantic knowledge is because we would like

to acknowledge the maximum amount of information that the mentioned two perspectives can provide, so that we will be able to combine them later on with a semantic module in an optimal way.

The remainder of this paper is structured as follows. The second section details the aforementioned methods, and the third one illustrates the performed experiments and includes a discussion about the results. Finally, the last section presents the conclusions of our research and proposes possible future work.

## 2 Methods

As we previously mentioned, our system is composed of two modules, which we will now explain.

### 2.1 Lexical approach

This method relies on the computation of a wide variety of lexical measures that basically consist of overlap metrics. Some researchers have already used this kind of metrics [9]. However, the main novelty of our approach is that it does not use semantic knowledge.

Prior to the calculation of the measures, all texts and hypothesis are tokenized and lemmatized. Later on, a morphological analysis is performed as well as a stemming. Once these steps are completed, we create several data structures that contain the corresponding tokens, stems, lemmas, functional<sup>2</sup> words and the most relevant<sup>3</sup> ones corresponding to the text and the hypothesis. The lexical measures will be applied to these structures and will allow us to determine which of them are more suitable for recognizing entailment situations, depending on the similarity rates that they provide.

We will now describe the lexical measures included in our system. Each of them calculates a similarity value between text and hypothesis that will allow us to determine if there is entailment between both of them or not.

<sup>1</sup> <http://www.pascal-network.org/Challenges/RTE/>.

<sup>2</sup> As functional words we consider nouns, verbs, adjectives, adverbs and figures (number, dates, etc).

<sup>3</sup> Considering only nouns and verbs.

### 2.1.1 Simple matching

Word overlapping between text and hypothesis is initialized to zero. If a word of the hypothesis appears also in the text, an increment of one unit is added to the similarity value. Finally, the weight is normalized dividing it by the length of the hypothesis measured as the number of words.

### 2.1.2 Levenshtein distance

This distance is similar to simple matching. However, in this case we calculate the value of the function that represents the occurrences in the text of each element that belongs to the hypothesis, denoted by  $m(i)$ , as defined in Equation 1.

$$m(i) = \begin{cases} 1 & \text{if } \exists j \in T / Lv(i, j) = 0, \\ 0.9 & \text{if } \nexists j \in T / Lv(i, j) = 0 \\ & \wedge \exists k \in T / Lv(i, k) = 1, \\ \max \left( \frac{1}{Lv(i, j)} \forall j \in T \right) & \text{otherwise.} \end{cases} \quad (1)$$

where  $Lv(i, j)$  represents the Levenshtein distance [4] between  $i$  and  $j$ . In our implementation, the cost of an insertion, deletion or substitution is equal to one and the weight assigned to  $m(i)$  when  $Lv(i, j) = 1$  has been obtained empirically.

### 2.1.3 Consecutive subsequence matching

This measure assigns the highest relevance to the appearance of consecutive subsequences. In order to perform this, we have generated all possible sets of consecutive subsequences, from length two until the length in words, from the text and the hypothesis. If we proceed as mentioned, the sets of length two extracted from the hypothesis will be compared to the ones of the same length from the text. If the same element is present in both the text and the hypothesis set, then a unit is added to the accumulated weight. This procedure is applied to all sets of different length extracted from the hypothesis. Finally, the sum of the weight obtained from each set of a specific length is normalized by the number of sets corresponding to such length, and the final accumulated weight is also normalized dividing it by the length of the hypothesis in words minus one. This measure is defined as shown in Equation 2.

$$CSmatch = \frac{\sum_{i=2}^{|H|} f(SH_i)}{|H| - 1} \quad (2)$$

where  $SH_i$  contains the hypothesis' subsequences of length  $i$ , and  $f(SH_i)$  is defined as follows:

$$f(SH_i) = \frac{\sum_{j \in SH_i} m(j)}{|H| - i + 1} \quad (3)$$

being  $m(j)$  equal to one if there exists an element  $k$  that belongs to the set that contains the text's subsequences of length  $i$ , such that  $k = j$ .

We would like to point out that this measure does not consider non-consecutive subsequences. In addition, it assigns the same relevance to all consecutive subsequences with the same length. Furthermore, the longer the subsequence is, the more relevant it will be considered in our system.

### 2.1.4 Tri-grams

Two sets containing tri-grams of letters that belong to the text and the hypothesis were created. All the occurrences of the hypothesis' tri-grams set that also appear in the text's will increase the accumulated weight by a factor of one unit. The calculated weight is then normalized dividing it by the total number of tri-grams within the hypothesis.

### 2.1.5 ROUGE measures

ROUGE measures have already been tested for automatic evaluation of summaries and machine translation [5, 6]. For this reason, and considering the impact of n-gram overlap metrics in textual entailment, we believe that the idea of integrating these measures<sup>4</sup> in our system is very appealing. We have implemented them as defined in [5].

Within the entire set of measures, each one of them is considered as a feature for the training and test stages of a machine learning algorithm. The selected one was a Support Vector Machine [11] due to the fact that its behavior is suitable for recognizing entailment relations.

Next, we present a true entailment text-hypothesis pair example, and show how the lexical approach calculates the corresponding similarity rate.

**Text:** *The destruction of the ozone layer was first noticed in the late 1980s as a hole over Antarctica.*

**Hypothesis:** *The ozone hole was first noticed in the late 1980s.*

The average values obtained for each measure considering tokens, lemmas and content words are the followings:

- Simple matching = 1
- Levenshtein distance = 1
- Consecutive subsequence matching = 0.34
- Tri-grams = 1
- ROUGE measures = from 0.4 using ROUGE-S to 0.66 using ROUGE-L

As it can be observed in the previous example, simple matching, Levenshtein distance and tri-grams achieve the highest possible score, due to the fact that all word occurrences in the hypothesis also appear in the text. However, regarding the consecutive subsequence matching measure, there are some hypothesis' consecutive subsequences that do not appear in the text, but the appearance of the subsequence "was

<sup>4</sup> The considered measures were ROUGE-N with n=2 and n=3, ROUGE-L, ROUGE-W and ROUGE-S with s=2 and s=3.

first noticed in the late 1980s” produces that this measure achieves a relatively high score. Finally, ROUGE measures have a similar behavior to the previous one, achieving different scores depending on the type of measures used.

## 2.2 Syntactic approach

This approach aims to provide a good accuracy rate by using few modules that are based on syntactic knowledge. These include tree construction, filtering, tree embedding detection and tree node matching.

### 2.2.1 Tree generation

The first module constructs the corresponding syntactic dependency trees. For this purpose, *MINIPAR* [7] output is generated and afterwards parsed for each text and hypothesis of our corpus. Phrase tokens, along with their grammatical information, are stored in an on-memory data structure that represents a tree.

### 2.2.2 Tree filtering

Once the tree has been constructed, we may want to discard irrelevant data in order to reduce our system’s response time and noise. For this purpose we have generated a list of relevant grammatical categories (shown in Table 1) that will allow us to remove from the tree all those tokens whose category does not belong to such list. The resulting tree will have the same structure as the original, but will not contain any stop words nor tokens with minor relevance, such as determinants or auxiliary verbs.

### 2.2.3 Tree embedding detection

The next step of the syntactic approach consists in determining whether the hypothesis’ syntactic dependency tree is embedded into the text’s. A tree,  $T_1$ , is embedded into another one,  $T_2$ , if all nodes and branches of  $T_1$  appear in  $T_2$  as well [3]. Therefore, in this module we attempt to find a match of the hypothesis’ syntactic structure within the text’s. Since this is a very strict matching process, we will believe that there is entailment if we are able to find a coincidence. Otherwise we will not be able to assure this and will execute the next module of our system, which is described in the following subsection.

### 2.2.4 Tree node matching

In this stage we proceed to perform a tree node matching process, termed alignment, between both the text and the hypothesis. This operation consists in finding pairs of tokens in both trees whose lemmas are identical, no matter whether they are in the same position within the tree. Some authors have already designed similar matching techniques, such as the ones described in [8, 10]. However, these include semantic constraints that we have decided not to consider. The reason of this decision is that we desired to overcome the textual entailment recognition task from an exclusively syntactic perspective.

Let  $\tau$  and  $\lambda$  represent the text’s and hypothesis’ syntactic dependency trees, respectively. We assume we have found a word, namely  $\beta$ , present in both  $\tau$  and  $\lambda$ . Now let  $\gamma$  be the weight assigned to  $\beta$ ’s grammatical category (Table 1),  $\sigma$  the weight of  $\beta$ ’s grammatical relationship (Table 2),  $\mu$  an empirically calculated value that represents the weight difference between tree levels, and  $\delta_\beta$  the depth of the node that contains the word  $\beta$  in  $\lambda$ . We define the function that provides the relevance of a word as follows:

$$\phi(\beta) = \gamma \cdot \sigma \cdot \mu^{-\delta_\beta} \quad (4)$$

The value obtained by calculating this expression would represent the relevance of a word in our system. The experiments performed reveal that the optimal value for  $\mu$  is 1.1.

Grammatical category	Weight
Verbs, verbs with one argument, verbs with two arguments, verbs taking clause as complement	1.0
Nouns, numbers	0.75
<i>Be</i> used as a linking verb	0.7
Adjectives, adverbs, noun-noun modifiers	0.5
Verbs <i>Have</i> and <i>Be</i>	0.3

**Table 1:** *Weights assigned to the relevant grammatical categories (empirically calculated).*

Grammatical relationship	Weight
Subject of verbs, surface subject, object of verbs, second object of ditransitive verbs	1.0
The rest	0.5

**Table 2:** *Weights assigned to the grammatical relationships (empirically calculated).*

For a given pair  $(\tau, \lambda)$ , we define the set  $\xi$  as the one that contains all words present in both trees, being  $\xi = \tau \cap \lambda \ \forall \alpha \in \tau, \beta \in \lambda$ . Therefore, the similarity rate between  $\tau$  and  $\lambda$ , denoted by the symbol  $\psi$ , would be as defined in Equation 5.

$$\psi(\tau, \lambda) = \sum_{\nu \in \xi} \phi(\nu) \quad (5)$$

One should note that a requirement of our system’s similarity measure would be to be independent of the hypothesis length. Thus, we must define the normalized similarity rate, as shown in Equation 6.

$$\overline{\psi(\tau, \lambda)} = \frac{\sum_{\nu \in \xi} \phi(\nu)}{\sum_{\beta \in \lambda} \phi(\beta)} \quad (6)$$

Once the similarity value has been calculated, it will be provided to the user together with the corresponding text-hypothesis pair identifier. It will be his responsibility to choose an appropriate threshold that will represent the minimum similarity rate to be considered as entailment between text and hypothesis. All

values that are under such a threshold will be marked as not entailed.

We will now show the behavior of the syntactic module for the text-hypothesis pair example shown at the end of section 2.1. For this purpose, we will first generate the corresponding syntactic dependency trees that are shown in Figures 1 and 2.



Fig. 1: *The destruction of the ozone layer was first noticed in the late 1980s as a hole over Antarctica.*

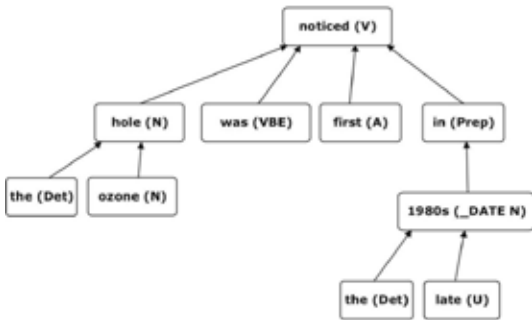


Fig. 2: *The ozone hole was first noticed in the late 1980s.*

The next step would be to perform a tree filtering over the text's and the hypothesis' trees. After this process has been completed, we will try to determine whether there is an entailment relation by calculating the value of function  $\phi$  for each remaining word, based on the values shown in Tables 1 and 2.

$$\begin{aligned}\phi(\text{noticed}) &= 1.0 \cdot 1.0 \cdot 1.1^{-1} = 0.91 \\ \phi(\text{hole}) &= 0.75 \cdot 1.0 \cdot 1.1^{-2} = 0.62 \\ \phi(\text{ozone}) &= 0.75 \cdot 0.5 \cdot 1.1^{-3} = 0.28 \\ \phi(\text{was}) &= 0.7 \cdot 0.5 \cdot 1.1^{-2} = 0.29 \\ \phi(\text{first}) &= 0.5 \cdot 0.5 \cdot 1.1^{-2} = 0.21 \\ \phi(1980) &= 0.75 \cdot 0.5 \cdot 1.1^{-2} = 0.31\end{aligned}$$

Combining all these values we will be able to obtain the similarity rate of the exposed text-hypothesis pair,

$$\text{as } \psi(\tau, \lambda) = \phi(\text{noticed}) + \phi(\text{hole}) + \phi(\text{ozone}) + \phi(\text{was}) + \phi(\text{first}) + \phi(1980) = 2.62.$$

The final step is to calculate the normalized similarity value as defined in Equation 6. However, we would like to point out that in the proposed example all words within the hypothesis also appear in the text. Therefore, the value of the denominator of the fraction from Equation 6 will be the same as the numerator, so the normalized similarity value will be the maximum possible. Since the obtained rate has a high value, we will consider the input pair as entailed.

### 3 Experimental results

The experimental results shown in this paper were obtained processing a set of text-hypothesis pairs from RTE-2 [1] and RTE-3<sup>5</sup>. The organizers of this challenge provide participants with development and test corpora, both of them with 800 sentence pairs (text and hypothesis) manually annotated for logical entailment. The judgments returned by the system will be compared to those manually assigned by the human annotators. The percentage of matching judgments will provide the *accuracy* of the system.

Table 3 shows the results obtained by both approaches individually (lexical and syntactic) and by combining them. This last approach consists in obtaining the entailment value that achieves the best performance. If both methods, lexical and syntactical, agree, then the judgement is straightforward, but if they disagree we then set the value depending on the performance of each one for true and false entailment situations. In our case, the lexical method performs better while dealing with negative examples, i.e., when there is no entailment relation, so this decision will prevail over the rest. Otherwise, the syntactical one shall decide the judgement.

As we can see in Table 3, the collaborative approach obtains the best results for the RTE-2<sup>6</sup> corpus, but using the RTE-3 corpus the approach that obtained the best performance was the lexical one. This makes us believe that an appropriate combination of these two kinds of knowledge (lexical and syntactical) would improve the entailment recognition. In addition, depending on the target task where the entailment is produced, the lexical approach performs better than the syntactical, and vice versa. For instance, the lexical method performs better when the pair belongs to the IR task. We would like to point out that, at the moment, these statements depend on the idiosyncrasies of the RTE corpora. However, these corpora are, nowadays, the most reliable source for evaluating textual entailment systems.

<sup>5</sup> The *Third PASCAL Recognising Textual Entailment Challenge* has not finished yet. Therefore, we know our individual results although we cannot compare them with the rest of the participating groups.

<sup>6</sup> If our system had participated in the *Second PASCAL Recognising Textual Entailment Challenge*, we would have obtained the fifth position out of twenty-four participating groups.

RTE-2	Development corpus	Test corpus				
	Overall	Overall	IE	IR	QA	SUM
Lexical	0.6013	0.6188	0.5300	0.6300	0.5550	0.7600
Syntactic	0.5750	0.6075	0.5050	0.6450	0.5950	0.6850
Both	<b>0.6087</b>	<b>0.6212</b>	<b>0.5100</b>	<b>0.6550</b>	<b>0.6250</b>	<b>0.6950</b>
RTE-3	Development corpus	Test corpus				
	Overall	Overall	IE	IR	QA	SUM
Lexical	<b>0.7012</b>	<b>0.6563</b>	<b>0.5150</b>	<b>0.7350</b>	<b>0.7950</b>	<b>0.5800</b>
Syntactic	0.6450	0.5925	0.5050	0.6350	0.6300	0.6000
Both	0.6900	0.6375	0.5150	0.7150	0.7400	0.5800

Table 3: Accuracy rates obtained using the RTE-2 and RTE-3 development and test corpora.

## 4 Conclusions and future work

In this paper we have presented a system for detecting textual entailment relations considering mainly lexical and syntactical information. A wide variety of lexical measures as well as syntactic structure comparisons were performed for this purpose. Decomposing the textual entailment task into subtasks allows finer analysis and high accuracy rates as well. As said before, we have been able to build a precise system without need of semantic knowledge, as a difference with most of the current state of the art approaches [1].

The separate analysis of the lexical and syntactic approaches has allowed us to study the maximum amount of knowledge that these perspectives can provide. In addition, we have been able to investigate our system's behavior when both approaches were combined. This is very useful for determining the optimal combination procedure, and will help us to couple a semantic module that does not produce conflicts with the ones described in this paper. We believe that this research line that analyzes the different kinds of knowledge separately allows a more accurate analysis of the successes and failures and the construction of a cleanly designed system.

Regarding future work, we are highly motivated in adding a semantic knowledge module to our system. Huge amounts of work in this line have been carried out by the research community within the last years. To add this kind of knowledge, we propose to extract a high amount of semantic information that would allow us to construct a characterized representation based on the input text, so that we can deduce entailment even if there is no apparent lexical nor syntactic structure similarity between text and hypothesis. This would mean to create an abstract conceptualization of the information contained in the analyzed phrases, allowing us to deduce ideas that are not explicitly mentioned in the parsed text-hypothesis pairs.

In addition, we have observed from the results shown in Table 3 that the accuracy of our system differs between tasks. Thus, we would like to apply different entailment recognition techniques based on the task of the text-hypothesis pair that is being analyzed.

Finally, due to the fact that recognizing textual entailment is a very complex task, we would like to tune the recognition by creating uncertainty thresholds. Such levels would include the situations where the system does not have enough information to determine if there is an entailment relation.

## Acknowledgments

This research has been partially funded by the QALLME consortium, contract number FP6-IST-033860, and by the Spanish Government under the project CI-CyT number TIN2006-1526-C06-01. It has also been supported by the undergraduate research fellowships financed by the Spanish Ministry of Education and Science, and the project ACOM06/90 supported by the Spanish Generalitat Valenciana.

## References

- [1] R. Bar-Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. The Second PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 1–9, Venice, Italy, April 2006.
- [2] I. Dagan, O. Glickman, and B. Magnini. The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 1–8, Southampton, UK, April 2005.
- [3] S. Katrenko and P. Adriaans. Using Maximal Embedded Syntactic Subtrees for Textual Entailment Recognition. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 33–37, Venice, Italy, April 2006.
- [4] V. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, February 1966.
- [5] C.-Y. Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In S. S. Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the Association for Computational Linguistics Workshop*, pages 74–81, Barcelona, Spain, July 2004.
- [6] C.-Y. Lin and F. J. Och. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics. In *Association for Computational Linguistics*, pages 605–612, July 2004.
- [7] D. Lin. Dependency-based Evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems*, Granada, Spain, 1998.
- [8] B. MacCartney, T. Grenager, M.-C. de Marneffe, D. Cer, and C. D. Manning. Learning to recognize features of valid textual entailments. In *Proceedings of the North American Association of Computational Linguistics*, pages 41–48, New York City, New York, United States of America, June 2006.
- [9] J. Nicholson, N. Stokes, and T. Baldwin. Detecting Entailment Using an Extended Implementation of the Basic Elements Overlap Metrics. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 122–127, Venice, Italy, April 2006.
- [10] R. Snow, L. Vanderwende, and A. Menezes. Effectively using syntax for recognizing false entailment. In *Proceedings of the North American Association of Computational Linguistics*, pages 33–40, New York City, New York, United States of America, June 2006.
- [11] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

# Sentiment Composition

Karo Moilanen and Stephen Pulman  
Oxford University Computing Laboratory  
Wolfson Building, Parks Road, Oxford, OX1 3QD, England  
{ *Karo.Moilanen* | *Stephen.Pulman* }@comlab.ox.ac.uk

## Abstract

Sentiment classification of grammatical constituents can be explained in a quasi-compositional way. The classification of a complex constituent is derived via the classification of its component constituents and operations on these that resemble the usual methods of compositional semantic analysis. This claim is illustrated with a description of sentiment propagation, polarity reversal, and polarity conflict resolution within various linguistic constituent types at various grammatical levels. We propose a theoretical composition model, evaluate a lexical dependency parsing post-process implementation, and estimate its impact on general NLP pipelines.

## Keywords

sentence-level sentiment, clause-level sentiment, entity-level sentiment, valence shifters, polarity shifters, lexical semantics

## 1 Introduction

Using lists of positive and negative keywords can give the beginnings of a sentiment classification system. However, classifying sentiment on the basis of individual words can give misleading results because atomic sentiment carriers can be modified (weakened, strengthened, or reversed) based on lexical, discursive, or paralinguistic contextual operators ([7]). Past attempts to deal with this phenomenon include writing heuristic rules to look out for negatives and other ‘changing’ words ([6]), combining the scores of individual positive and negative word frequencies ([11], [5]), and training a classifier on a set of contextual features ([10]). While statistical sentiment classifiers work well with a sufficiently large input (e.g. a 750-word movie review), smaller subsentential text units such as individual clauses or noun phrases pose a challenge. It is such low-level units that are needed for accurate entity-level sentiment analysis to assign (local) polarities to individual mentions of people, for example.

In this paper we argue that, as far as low-level (sub)sentential sentiment classification is concerned, there may be much to be gained from taking account of more linguistic structure than is usually the case. In particular we argue that it is possible to calculate in a systematic way the polarity values of larger syntactic constituents as some function of the polarities of their subconstituents, in a way almost exactly analogous to the ‘principle of compositionality’ familiar from the formal semantics literature ([2]). For if the meaning of a sentence is a function of the meanings

of its parts then the global polarity of a sentence is a function of the polarities of its parts. For example, production rules such as  $[VP_{\alpha} \rightarrow V_{\alpha} + NP]$  and  $[S_{\beta} \rightarrow NP + VP_{\beta}]$  operating on a structure like “*America invaded Iraq*” would treat the verb “*invade*” as a function from the NP meaning to the VP meaning (i.e. as combining semantically with its direct object to form a VP). The VP meaning is correspondingly a function from the NP meaning to the S meaning (i.e. as combining with a subject to form a sentence). Analogously, a ‘DECREASE’ verb like “*reduce*” (cf. [1]) should then be analysed as having a compositional sentiment property such that it reverses the polarity ( $\alpha$ ) of its object NP in forming the VP, hence  $[VP_{\beta}^{(-\alpha)} \rightarrow V_{\beta[DECREASE]} + NP^{(\alpha)}]$ . Thus the positive polarity in “*reduce the risk*” even though “*risk*” is negative in itself (cf. the negative polarity in “*reduce productivity*”). In fact, this semi-compositionality also holds at other linguistic levels: certainly amongst morphemes, and arguably also at suprasentential levels. However, this paper discusses only sentential sentiment composition. Grounded on the descriptive grammatical framework by ([4]), we propose a theoretical framework within which the sentiment of such structures can be calculated.

## 2 Composition Model

The proposed sentiment composition model combines two input (IN) constituents at a time and calculates a global polarity for the resultant composite output (OUT) constituent (cf. parent node dominance in the *modifies\_polarity* and *modified\_by\_polarity* structural features in ([10])). The two IN constituents can be of any syntactic type or size. The model assumes dominance of non-neutral (positive (+), negative (-), mixed (M)) sentiment polarity over neutral (N) polarity. The term **sentiment propagation** is used here to denote compositions in which the polarity of a neutral constituent is overridden by that of a non-neutral constituent ( $\{(+) (N)\} \rightarrow (+)$ ;  $\{(-) (N)\} \rightarrow (-)$ ). We use the term **polarity reversal** to denote compositions in which a non-neutral polarity value is changed to another non-neutral polarity value ( $(+) \rightarrow (-)$ ;  $(-) \rightarrow (+)$ ) (cf. [7]), and the term **polarity conflict** to denote compositions containing conflicting non-neutral polarities ( $\{(+)(-)\} \rightarrow (M)$ ). Polarity **conflict resolution** refers to disambiguating compositions involving a polarity conflict ( $(M) \rightarrow (+)$ ;  $(M) \rightarrow (-)$ ).

Polarity conflict resolution is achieved by ranking the IN constituents on the basis of relative weights assigned to them dictating which constituent is more

important with respect to sentiment. The stronger of the IN constituents is here denoted as SPR (superordinate) whereas the label SUB (subordinate) refers to the dominated constituent (i.e. SPR  $\gg$  SUB). Except for (N)[=] SPR constituents, it is therefore the SPR constituent and the compositional processes executed by it that determine the polarity ( $\alpha$ ) of the OUT constituent (i.e.  $\text{OUT}^{\alpha_{ij}} \rightarrow \text{SPR}^{\alpha_i} + \text{SUB}^{\alpha_j}$ ). The weights are not properties of individual IN constituents per se but are latent in specific syntactic constructions such as [Mod:Adj Head:N] (e.g. adjectival premodification of head nouns) or [Head:V Comp:NP] (e.g. direct object complements of verbs).

We tag each entry in the sentiment lexica (across all word classes) and each constituent with one of the following tags: **default** ([=]), **positive** ([+]), **negative** ([-]), and **reverse** ([-]). These tags allow us to specify at any structural level and composition stage what any given SPR constituent does *locally* to the polarity of an accompanying SUB constituent without fixed-order windows of  $n$  tokens (cf. ([7]), modification features in ([10]), change phrases in ([6])). A [=] SPR constituent combines with a SUB constituent in the default fashion. The majority of constituents are [=]. A [-] SPR constituent reverses the polarity of the SUB constituent and assigns that polarity to the OUT constituent (cf. general polarity shifters in ([10])). As SPR constituents, some carriers such as “[contaminate]<sup>(-)</sup>” or “[soothe]<sup>(+)</sup>” exhibit such strong sentiment that they can determine the OUT polarity irrespective of the SUB polarity - consider the static negativity in “[contaminated that damn disk]<sup>(-)</sup>”, “[contaminated the environment]<sup>(-)</sup>”, and “[contaminated our precious water]<sup>(-)</sup>” (vice versa for some positive carriers). Hence the [-] and [+] constants which can furthermore be used as polarity heuristics for carriers occurring prototypically with a specific polarity (e.g. “[deficiency (of sth positive)]<sup>(-)</sup>”) (cf. prepositional items in ([7]), negative and positive polarity shifters in ([10])).

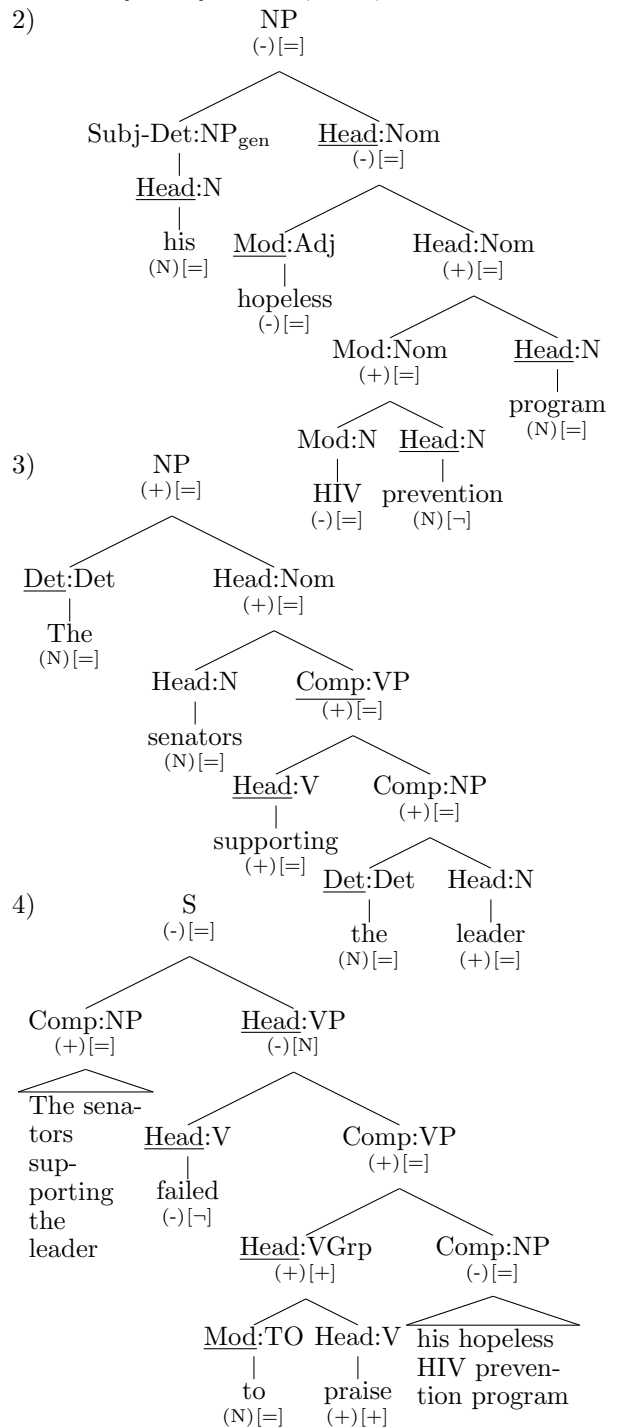
Notice that the SPR constituent operates on the SUB constituent irrespective of the polarity of the latter as a [-] SPR constituent such as the determiner “[less]<sup>(N)[-]</sup>” reverses both (+) and (-) SUB constituents (e.g. “[less tidy]<sup>(-)</sup>”, “[less ugly]<sup>(+)</sup>”), for example. However, cases in which SPR operations are required only in conjunction with a specific SUB constituent polarity do exist. The reversal potential in the degree modifier “[too]<sup>(N)[-]</sup>”, for instance, seems to operate only alongside (+) SUB constituents (i.e. “[too colourful]<sup>(-)</sup>” vs. “??[too sad]<sup>(+)</sup>”). The adjective “[effective]<sup>(+)[=]</sup>” operates similarly only with (+) or (N) SUB constituents (i.e. “[effective remedies/diagrams]<sup>(+)</sup>” vs. “[effective torture]<sup>(-)</sup>”). It is thus proposed that (?:+) and (?:-) be used as further **filters** to block specific SPR polarities as required by individual carriers.

To illustrate how the composition model operates, consider the sample sentence in Ex. 1:

- 1) *The senators supporting<sup>(+)</sup> the leader<sup>(+)</sup> failed<sup>(-)</sup> to praise<sup>(+)</sup> his hopeless<sup>(-)</sup> HIV<sup>(-)</sup> prevention program.*

Raw frequency counts, yielding three (+) and three

(-) carriers, would fail to predict the global negative polarity in the sentence. We represent the sentence as follows, starting with the direct object NP of the predicator “[praise]<sup>(+)[+]</sup>” (Ex. 2):



Through polarity reversal, the internal sentiment in “[HIV prevention]<sup>(+)[=]</sup>” is first arrived at due to the [-] status of the SPR head noun “[prevention]<sup>(N)[-]</sup>” which reverses the (-) premodifying noun “[HIV]<sup>(-)[=]</sup>”. The (N) head noun “[program]<sup>(N)[=]</sup>” is then overridden by the (+) premodifying nominal “[HIV prevention]<sup>(+)[=]</sup>”. When the resultant nominal is combined with the premodifying attributive SPR input “[hopeless]<sup>(-)[=]</sup>”, the ensuing polarity conflict can be resolved through the



dominance of the premodifier in this syntactic situation. The final combination with the SUB subject determiner “[his]<sup>(N)|[=]</sup>” is a case of propagation as the resultant NP reflects the polarity of the head nominal. Sentiment propagation can be seen throughout the subject NP (Ex. 3) as the (+) head noun “[leader]<sup>(+)|[=]</sup>”, combined with a (N) SPR determiner, results in a (+) NP (“[the leader]<sup>(+)|[=]</sup>”). When that NP is combined with a (+) SPR head participial, a (+) SPR VP is generated (“[supporting the leader]<sup>(+)|[=]</sup>”) which in turn overrides the (N) head noun “[senators]<sup>(N)|[=]</sup>”. The final (N) SPR determiner does not change the polarity any further.

The NPs thus resolved can then be combined with the two predicators to form a sentence (Ex. 4). The direct object NP “[his hopeless HIV prevention program]<sup>(-)|[=]</sup>” is reversed when it is combined with an SPR verb group outputting constant positivity (“[to praise]<sup>(+)|[+]</sup>”). When the resultant (+) VP is used as the complement of a [-] SPR head verb polarity reversal occurs once again yielding a (-) VP (“[failed to praise his hopeless HIV prevention program]<sup>(-)|[=]</sup>”). Lastly, the (+) subject NP combines with the (-) predicate, and the polarity conflict is resolved due to the predicate being the SPR constituent. Hence the global negative sentiment for the present sample sentence can be calculated from its subconstituents.

### 3 Grammatical Constructions

Within a syntactic phrase, the polarity of the phrasal head can be changed by its pre- and post-modifying dependents. In general, pre-head dependents dominate their heads. **Determiners** (e.g. “[no crime]<sup>(-)</sup>”) and **DPs** (e.g. “[too much wealth]<sup>(-)</sup>”) can be modelled as [Det:(Det|DP) >> Head:N] ([4]: 354-99, 431-2, 549, 573). Attributive **pre-head AdjPs** and simple **pre-head ING/EN Participials** are ranked similarly as [Mod:(AdjP|V) >> Head:N] to account for polarity reversals (e.g. “[trivial problem]<sup>(+)</sup>”), conflicts (e.g. “[nasty smile]<sup>(-)</sup>”), and seemingly contradictory compositions with (?:-) premodifiers (e.g. “[perfected torture]<sup>(-)</sup>”). However, mixed sentiment is possible in this construction (e.g. “[savvy liar]<sup>(M)</sup>”) ([4]: 444). We rank attributive **pre-head Adverbs** as [Mod:Adv >> Head:(Adj|Adv)] (e.g. “[decreasingly happy]<sup>(-)</sup>”, “[never graceful(ly)]<sup>(-)</sup>”) although they too can lead to unresolvable mixed sentiment (e.g. “[impressively bad(ly)]<sup>(M)</sup>”) (*idem.* 548, 572-3, 582-5). The pre-head **Negator (Neg)** “not”, which is stronger than its head in NPs (e.g. “[not a scar]<sup>(+)</sup>”), AdjPs, AdvPs, and PPs, is ranked as [Mod:Neg >> Head:(N|Adj|Adv|P)] (cf. [7]). In contrast, **pre-head Nouns and Nominals** in NPs are secondary ([Head:N >> Mod:(N|Nom)]) as seen in polarity conflicts (e.g. “[family benefit fraud]<sup>(-)</sup>”, “[abuse helpline]<sup>(+)</sup>”) and [-] head nouns (e.g. “[risk minimisation]<sup>(+)</sup>”) (*idem.* 444, 448-9). The genitive subject determiner with the clitic ‘s appears similarly weaker than its head noun or nominal ([Head:(N|Nom) >> Subj-Det:NP<sub>gen</sub>]) (e.g. “[the war’s end]<sup>(+)</sup>”), although polarity conflicts can lead to exceptions: com-

pare “[the offender’s apology]<sup>(+)</sup>” with “[the rapist’s smile]<sup>(-)</sup>” (*idem.* 467-83).

Post-head dependents’ weights are more variable. In NPs, **post-head AdjPs** generally dominate (e.g. “[my best friend angry at me]<sup>(-)</sup>”) as [Comp:AdjP >> Head:N] (*idem.* 445). **Post-head Participials** dominate their head nouns as [Comp:VP >> Head:N] (e.g. “[ugly kids smiling]<sup>(+)</sup>”, “[the cysts removed]<sup>(+)</sup>”) (*idem.* 446), but **post-head VPs** are dominated by their head prepositions ([Head:P >> Comp:VP]) (e.g. “[against helping her]<sup>(-)</sup>”) ([4]: 641). **Post-head PPs** are likewise dominated by their noun, adjective, or adverb heads. The rankings [Head:(N|Adj|Adv) >> Comp:PP] are thus proposed (e.g. “[different(ly) from those losers]<sup>(+)</sup>”, “[unhappy with success]<sup>(-)</sup>”, “[the end of the war]<sup>(+)</sup>”) ([4]: 446, 543-6). However, exceptions may surface in these constructions, especially in NPs: compare “[two morons amongst my friends]<sup>(-)</sup>” with “[cute kittens near a vicious python]<sup>(-)</sup>”. Moreover, mixed sentiment may surface (e.g. “[angry protesters against the war]<sup>(M)</sup>”). Lastly, we rank **post-head NPs** in PPs as [Head:P >> Comp:NP] (e.g. “[against racism]<sup>(+)</sup>”, “[with pleasure]<sup>(+)</sup>”) (*idem.* 635).

In clausal analysis, we treat as the clausal head the predicator (P) which is made of one verb group and compulsory (C)omplements and optional (A)djuncts. The predicator is generally stronger than its complements. We propose that internal complements (Direct Object (O<sup>D</sup>), Indirect Object (O<sup>I</sup>), Subject Predicative Complement (PC<sup>S</sup>), Object Predicative Complement (PC<sup>O</sup>), and Oblique (C)omplement) be combined with the predicator before combining the resultant predicate with the predicator’s external complements ([4]: 215-8; 236-57). In **Monotransitive Predicates (P-O<sup>D</sup>)**, the ranking [Head:P >> Comp:O<sup>D</sup>] models propagation (e.g. “[failed it]<sup>(-)</sup>”), polarity conflicts (e.g. “[spoiled the party]<sup>(-)</sup>”), and [-] predicators (e.g. “[prevent the war]<sup>(+)</sup>”) (*idem.* 244-8). **Ditransitive Predicates (P-O<sup>I</sup>-O<sup>D</sup>)**, (**P-O<sup>D</sup>-C**) behave in a similar way. Since the monotransitive “[sent junk]<sup>(-)</sup>”, pure ditransitive “[sent me junk]<sup>(-)</sup>”, and oblique ditransitive “[sent junk to me]<sup>(-)</sup>” all share a [-] P-O<sup>D</sup> core, we resolve it first before adding an O<sup>I</sup> or C to model propagation (e.g. “[baked a yummy cake for me]<sup>(+)</sup>”), and polarity conflicts (e.g. “[brought my friend sad news]<sup>(-)</sup>”) (*idem.* 244-8). Through the ranking [Head:P >> Comp:PC<sup>S</sup>], typically (N) copular verbs in **Complex Intransitive Predicates (P-PC<sup>S</sup>)** can be explained (e.g. “[seems nice]<sup>(+)</sup>”) (*idem.* 251-72). **Complex Transitive Predicates (P-O<sup>D</sup>-PC<sup>O</sup>)** resemble P-PC<sup>S</sup> predicates in that the additional direct object does not generally affect the P-PC<sup>S</sup> core (e.g. “[consider (the winner/it/the poison) ideal]<sup>(+)</sup>”). Hence the ranking [Head:P-PC<sup>O</sup> >> Comp:O<sup>D</sup>] (*ibidem.*) **(S)ubjects** are ranked as [Head:P >> Comp:S] (e.g. “[love can hurt]<sup>(-)</sup>”, “[the misery ended]<sup>(+)</sup>”) (*idem.* 235-43). Note that [-] NP complements constitute an exception calling for reverse rankings - consider “[nobody

PHRASES			
Pre-head		Post-head	
( <u>Det</u> :(Det DP) Subj-Det:NP <sub>gen</sub> <sup>[<sup>-</sup>]</sup>   <u>Mod</u> :(Neg AdjP VP))	» Head:N	Head:(N Nom)	« Comp:(AdjP VP)
( <u>Det</u> :(Det DP)  <u>Mod</u> :(Neg PP AdvP))	» Head:Adj	<u>Head</u> :Adj	» Comp:PP
( <u>Det</u> :(Det DP)  <u>Mod</u> :(Neg Adv))	» Head:Adv	<u>Head</u> :Adv	» Comp:PP
<u>Mod</u> :(Neg AdvP NP)	» Head:P	<u>Head</u> :P	» Comp:(NP VP)
(Subj-Det:NP <sub>gen</sub>  Mod:(N Nom))	« Head:N	<u>Head</u> :N	» Comp:(NP PP)
CLAUSES			
( <u>Comp</u> :(PC <sup>S</sup>  S <sup>[<sup>-</sup>]</sup>  O <sup>D</sup> <sup>[<sup>-</sup>]</sup>  O <sup>I</sup> <sup>[<sup>-</sup>]</sup> )  <u>A</u> :(AdvP AdjP PP)  <u>Mod</u> :Neg)	» Head:P	<u>Head</u> :P	» Comp:(S O <sup>D</sup> )
Comp:O <sup>D</sup>	« Head:P-PC <sup>O</sup>	<u>Head</u> :P-O <sup>D</sup>	» Comp:(O <sup>I</sup>  O <sup>C</sup> )

Table 1: Sample Construction Rankings

*died*]<sup>(+)</sup>”, “[*killed nobody*]<sup>(+)</sup>”, for example. Hence the rankings [Comp:(O<sup>D</sup><sup>[<sup>-</sup>]</sup>|S<sup>[<sup>-</sup>]</sup>) » Head:P] for these special cases. Adjuncts are generally stronger than predicates and predicators. The ranking [Comp:AdvP » Head:P] for **AdvP Adjuncts**, for example, supports propagation (e.g. “[*he moved it gently*]<sup>(+)</sup>”), and polarity conflicts (e.g. “[*greeted him insincerely*]<sup>(-)</sup>”) (*idem.* 224-5, 575, 669, 779-84).

These and other sample rankings are summarised in Table 1.

## 4 Implementation

The proposed model was implemented as a lexical parsing post-process interpreting the output of a dependency parser<sup>1</sup>. We employ a sentiment lexicon containing manually-compiled atomic core carriers<sup>2</sup> expanded semi-automatically using WordNet 2.1, all tagged with the compositional tags. A morphological unknown carrier guessing module and a missing dependency link repair module are included. Adhering to the proposed compositional processes and constituent rankings at each stage of the analysis, token dependency links and morphosyntactic token tags (e.g. word class, syntactic role, (pre-/post-)head status) are first used to construct individual syntactic phrases (NPs, VPs, AdjPs, AdvPs) and to calculate their internal polarities (**phrasal sentiment**) through stepwise chunking rules which find the rightmost subconstituent in a given phrase and expand it leftwards until a phrasal boundary is hit (see Ex. 2-3). To calculate **clausal** and **sentential sentiment**, the obtained phrasal constituents are then combined (see Ex. 4).

## 5 Experiments

To estimate the usefulness of a compositional treatment and its impact on standard NLP pipelines, we employ short headlines for sentential compositionality and NPs for phrasal compositionality. Since our implementation is fully lexical, its recall is conditioned by the coverage of the lexicon used. To estimate the (future) impact of larger lexica covering the entire WordNet, the default lexicon (at the time of writing) (DEFAULT\_LEX) was expanded with sample carriers from the test data found in WordNet 2.1 (WN\_ADD\_LEX). Polarity agreement between the gold standards and our output was measured using (*i*) all polarities (*All*

*pol*), and (*ii*) non-neutral polarities only (*Non-ntr pol*). To assess the role of sentiment intensity, results using (*i*) cases of **Any Strength** and (*ii*) those marked as **Strong** in the gold standards are given. The agreement results are shown in Table 2.

**Experiment 1: Headlines.** The sentences generated by our system were compared against 1000 news headlines in the SemEval-2007 Task #14 data set annotated for polarity (six annotators, *r* .78) ([8]). The SemEval scores [-100, 100] were collapsed into (-100 ≤ (-) < 0; 0 = (N); 0 < (+) ≤ 100) in the **Any Strength** condition, and into (-100 ≤ (-) ≤ -66; 0 = (N); 66 ≤ (+) ≤ 100) for 208 **Strong** cases. The WN\_ADD\_LEX lexicon contained 97 added carriers.

**Experiment 2: NPs.** The NPs generated by our system were compared (lax overlap) against 1541 explicit NPs in the customer review data set of 2108 product feature mentions from five home electronics products annotated for polarity (two annotators, *r* unknown) ([3]). The gold standard scores [-3, 3] were converted into (-3 ≤ (-) < 0; 0 < (+) ≤ 3) in the **Any Strength** condition, and into (-3 = (-); 3 = (+)) for 366 **Strong** cases. The WN\_ADD\_LEX lexicon contained 95 added carriers.

## Results and Error Analysis

The *All pol* figures are considerably lower than the corresponding *Non-ntr pol* ones due to the incomplete coverage of the lexica used: a (N) input into the model leads unavoidably to a (N) output and thus to an error. Since mining and tagging new carriers is a task beyond the realms of the model, we focus here on the performance in the *Non-ntr pol* conditions. Mirroring human judgements of high-intensity cases, the implementation performed noticeably better with strong cases. More interesting is the small margin between the two lexica which offers further evidence *pro* compositionality. The errors from the [WN\_ADD\_LEX *Non-ntr pol Any Strength*] condition are analysed in Table 3.

Because the model operates in the middle of the processing pipeline, the errors are classified as *pre-compositional* (i.e. erroneous input) or *post-compositional* (i.e. factors beyond the model). The performance of the model is promising as most errors (ca. 2/3) occurred earlier in the pipeline. Since full compositionality can only be achieved with a clean grammatical analysis, a heavy burden is placed on the TAGGER and PARSER which together caused ca. 28% of the errors. Hence erroneous propagation and partial compositionality due to incorrect POS tags and null dependencies, respectively. Since polarity distinctions between individual word SENSES (e.g. “[*rip*

<sup>1</sup> Connexor Machine Syntax 3.8 ([www.connexor.com](http://www.connexor.com))

<sup>2</sup> Kindly provided by Corpora Software ([www.corporasoftware.com](http://www.corporasoftware.com))

	DEFAULT_LEX		WN_ADD_LEX	
Cases	<i>All pol</i>	<i>Non-ntr pol</i>	<i>All pol</i>	<i>Non-ntr pol</i>
<b>Headlines</b>				
1000	<b>Any Strength</b>			
A	63.0	76.27	65.6	77.36
208	<b>Strong</b>			
A	81.73	89.95	86.06	91.33
<b>NPs</b>				
1541	<b>Any Strength</b>			
P	72.46	85.45	73.39	85.87
R	97.79	82.93	97.79	83.58
F	83.24	84.17	83.85	84.71
366	<b>Strong</b>			
P	79.22	89.10	80.33	89.51
R	98.63	87.70	98.63	88.52
F	87.87	88.40	88.55	89.01

**Table 2:** Agreement: (A)ccuracy, (P)recision, (R)ecall, and (F)-scores

(*a CD*)]<sup>(N)</sup>” vs. “[*rip into*]]<sup>(-)</sup>”) can have far-reaching compositional consequences, a sentiment WSD module could reduce ca. 25% of the errors. Resolving neutral ANAPHORic and CO-REFERential expressions could increase recall levels further. The errors also include supraclausal cases NOT yet IMPLEMENTED. However, even in an ideal situation with a clean input, the model would fail to solve many cases (ca. 19%) in which further WORLD knowledge is required. There are cases in which the literal/logical compositional polarity is modulated by phenomena closer to PRAGMatics than lexical semantics such as indirect speech acts (cf. logical positivity vs. implied negativity in “[*X could be better*]]<sup>(-)</sup>”). Lastly, AMBIGUOUS cases affording multiple polarity readings are always likely to be present.

	Headlines	NPs	All	
Pre-compositional errors				
ANAPHOR		13 (6.05)	13	3.23
CO-REF		4 (1.86)	4	0.99
NOT IMPL	8 (4.26)	22 (10.23)	30	7.44
PARSER	13 (6.92)	44 (20.47)	57	14.14
SENSE	58 (30.85)	46 (21.4)	104	25.81
SPELLING		2 (0.93)	2	0.5
TAGGER	24 (12.77)	32 (14.88)	56	13.9
			266	66%
Post-compositional errors				
AMBIG	35 (18.62)	4 (1.86)	39	9.68
PRAGM		21 (9.77)	21	5.21
WORLD	50 (26.6)	27 (12.56)	77	19.11
			137	34%
Total	188	215	403	100%

**Table 3:** Error distribution

## 6 Related Work

The proposed model develops further the lexical devices described in the survey of lexical and discou- r- s- a- l contextual valence shifters in ([7]). In ([6]), nega- tion and change phrases were used in a supervised learning algorithm analysing sentential polarities of clinical outcomes. A number of polarity shifters and syntactic dependencies were included as machine learning features in the phrase-level sentiment analyser

reported in ([10]). Adjectival appraisal groups comprising a head adjective with optional appraisal pre- modifiers were used in the sentiment classifier described in ([9]). ([1]) extracted and tagged words with rever- sal potential expressing a conceptual in-/decrease in magnitude, intensity or quality.

## 7 Conclusion

We have shown that sentiment exhibits quasi-compo- sitionality in noticeably many areas, and that it is possible to approach sentiment propagation, polarity re- versal, and polarity conflict resolution within different linguistic constituent types at different grammatical levels in an analytically and computationally uniform manner by relying on traditional compositional seman- tics and deep parsing. The results obtained, which are encouraging for a lexical system, point towards a crucial dependency on a wide-coverage lexicon, accurate parsing, and sentiment sense disambiguation in a com- positional approach to sentiment analysis.

## References

- [1] A. Andreevskaia and S. Bergler. Semantic tag extraction using wordnet glosses. In *Proceedings of LREC 2006*, Genoa, 2006.
- [2] D. Dowty, R. Wolf, and S. Peters. *Introduction to Montague Semantics*. D. Reidel, Dordrecht, 1981.
- [3] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2004)*, Seattle, 2004.
- [4] R. Huddleston and G. K. Pullum. *The Cambridge Grammar of the English Language*. Cambridge University Press, Cambridge, 2002.
- [5] S.-M. Kim and E. Hovy. Determining the sentiment of opinions. In *Proceedings of COLING 2004*, Geneva, 2004.
- [6] Y. Niu, X. Zhu, J. Li, and G. Hirst. Analysis of polarity information in medical text. In *Proceedings of the American Medical Informatics Association 2005 Annual Symposium (AMIA 2005)*, Washington D.C., 2005.
- [7] L. Polanyi and A. Zaenen. Contextual lexical valence shifters. In Y. Qu, J. Shanahan, and J. Wiebe, editors, *Exploring Attitude and Affect in Text: Theories and Applications: Papers from the 2004 Spring Symposium, Technical Report SS-04-07*. AAAI, 2004.
- [8] C. Strapparava and R. Mihalcea. Semeval-2007 task 14: Affective text. In *Proceedings of SemEval 2007*, Prague, 2007.
- [9] C. Whitelaw, N. Garg, and S. Argamon. Using appraisal taxonomies for sentiment analysis. In *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management*, Bremen, 2005.
- [10] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT/EMNLP 2005*, Vancouver, 2005.
- [11] H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of EMNLP 2003*, Sapporo, 2003.

# One Sense Per Discourse for Synonym Detection

Rumen Moraliyski, Gaël Dias  
Centre of Human Language Technology and Bioinformatics  
University of Beira Interior  
*rumen@penhas.di.ubi.pt, ddg@di.ubi.pt*

## Abstract

In this paper, we present a new methodology for synonym detection based on the combination of global and local distributional similarities of pairs of words. The methodology is evaluated on the noun space of the 50 multiple-choice synonym questions taken from the ESL and reaches 91.30% accuracy using a conditional probabilistic model associated with the cosine similarity measure.

## Keywords

Synonym discovery, similarity measure, discourse

## 1 Introduction

The task of recognizing synonyms can be defined as in [15]: “given a problem word and a set of alternative words, choose the member from the set of alternative words that is most similar in meaning to the problem word”. Based on this definition, many algorithms [3] [4] [6] [7] [13] [14] [15] [16] have been proposed and evaluated using multiple-choice synonym questions taken from the Test of English as Foreign Language (TOEFL).

Most of the works proposed so far explore the attributional similarity paradigm [10].

To construct attributional representation of a word, many approaches have been developed: window oriented [3], [4], [13], [14], lexicon oriented [1], syntactic oriented [2], [17], document oriented [7].

Most of the work proposed so far, independently of their categorization, have in common the fact that the word representation is built on global corpus evidence. As a consequence, all the senses of a polysemous word share a single description. This fact is clearly a drawback for any word meaning analysis. Indeed, this would mean that, to be synonyms, two words should share, as many as possible of their senses, while they usually do share just one.

A first attempt to take into account local corpus evidence is proposed in [11] who separate corpus evidences for distinct word occurrences in a corpus to build a matrix that is afterwards subjected to a SVD and analyzed to discover the major word senses. However, they do not propose any evaluation and validation of their work, neither it is reproducible on a small scale i.e. single texts.

Here, we propose a method to measure syntactic oriented attributional similarity based on the “one sense

*per discourse*” paradigm. Instead of relying exclusively on global distributions, we build words representations and compare them within documents limits. In this way, we only compare two specific senses of each word at a time.

We argue that our proposal coupled with the global approach leads to improved results. In order to test this assumption, we implemented the vector space model over term frequency, term frequency weighted by inverse document frequency, Pointwise Mutual Information [14] and conditional probability [17]. We also implemented two probabilistic similarity measures: the Ehlert model [3] and Lin model [8]. The evaluation was conducted on the subset of the 23 noun questions of a 50 multiple-choice synonym questions taken from the ESL (test for students of English as Second Language) provided by P. Turney. The best results were obtained by the vector space model over the conditional probability which scored 91% accuracy (i.e. 21 out of 23 nouns questions).

## 2 Related Work

Previous research on corpus-analytic approaches to synonymy has used the TOEFL and ESL which consist of set of multiple-choice questions. In this context, a distance function must be defined to order the correct answer word in front of then decoys.

One of the most famous work is proposed by [7] who use document distribution to measure word similarity. They show that the accuracy of Latent Semantic Analysis (LSA) is statistically indistinguishable from that of a population of non-native English speakers on the same questions.

More recent works have focused on window based vector space model. For that purpose, the word context vectors associated to all the words from the TOEFL are built on co-occurrence basis within the entire corpus. [14] studied a variety of similarity metrics and weighting schemes of contexts and achieved a statistical tie with their DR-PMI compared to the PMI-IR proposed by [15].

The PMI-IR is one of the first works to propose a hybrid approach to deal with synonym detection. Indeed, it uses a combination of evidences such as the Pointwise Mutual Information (PMI) and Information Retrieval (IR) features like the “NEAR” and “NOT” operators to measure similarity between pairs of words. This work does not follow the attributional similarity paradigm but rather proposes a heuristic to measure semantic distance. [16] refined the PMI-IR algorithm

and proposed a module combination to include new features such as LSA and thesaurus evidences.

In parallel, some works have focused on linguistic features to measure similarity. [6] give results for a number of relatively sophisticated thesaurus-based methods that looked at path length between words in the heading classifications of Roget's Thesaurus. However, this methodology does not follow the attributional similarity paradigm unlike [2], who use syntactic context relations.

Work	Best result
Landauer and Dumais 1997	64.40%
Sahlgren 2001	72.00%
Turney 2001	73.75%
Jarmasz and Szpakowicz 2003	78.75%
Terra and Clarke 2003	81.25%
Elhert 2003	82.00%
Freitag et al. 2005	84.20%
Turney et al. 2003	97.50%

**Table 1:** Accuracy on TOEFL question set.

In the syntactic attributional similarity paradigm, word context vectors associated to all target words of the test are indexed by the words they co-occur with within a given corpus for a given syntactic relation. For example, (*good*, *adjective*) and (*have*, *direct-obj*) are attributes of the noun "idea" as illustrated in [2].

Unfortunately, to our knowledge, unlike window based approaches, syntactic based methodologies have not been tested over TOEFL or ESL. Rather, they have been used to build linguistic resources. As a summary, Table 1 presents the results achieved by most of the mentioned methodologies<sup>1</sup>.

### 3 Proposal

While the attributional similarity paradigm has been used over global corpus evidence, the *ad hoc* metrics have privileged, to some extent, a closer view of the data taking advantage of the "one sense per discourse" hypothesis proposed by [5]. To our point of view discarding the corpus structure in terms of documents is a key factor for the "failure" of the attributional similarity measures based on global corpus evidence.

Our proposal consists in implementing "one sense per discourse" through comparing two words within a single document at a time and averaging over the documents in which both words were encountered. As a result words that co-occur in a document but with different meanings will rarely share contexts and will end with low similarity. On the other hand words that co-occur as synonyms will share contexts with greater probability hence will receive higher similarity estimation. The value obtained we call local similarity.

Finally, we combine the local similarity with the global one under the syntactic attributional paradigm to achieve improved performance.

<sup>1</sup> The values can not be compared directly as they may not be evaluated (1) on the same corpora or/and (2) the same set of questions. However, these results will give the reader an idea of the expected results for future methodologies. For more information about evaluation see [12].

## 4 The Corpus

### 4.1 Motivation

Any work based on the attributional similarity paradigm depends on the corpus used to calculate the values of the attributes. [14] use a terabyte of web data that contains 53 billion words and 77 million documents, [13] a 10 million words balanced corpus with a vocabulary of 94 thousand words and [3], [4] a 256 million words North American News Corpus (NANC). As mentioned in [3], [14], the size of the corpus does matter and the bigger the corpus is, the better the results are. In our case, we could also have used NANC. However our proposal demands co-occurrence of the two synonym candidates within a single document few times each. It is improbable that general purpose corpus would comprise enough documents containing pairs of our set of words four or more times each. As a result we decided to build a corpus suitable to the problem at hand thus exploring the merits and flaws of the approach as opposed to solving a problem fit to the data available. The corpus is available at <http://hultig.di.ubi.pt/>.

### 4.2 Construction

To build our corpus, we used the Google API and queried the search engine with 92 (23 questions  $\times$  4 alternatives) different pairs of words. For each ESL test case, we built 4 queries - target word and one of the proposed variants. Subsequently, we collected all of the seed results, lemmatized the text using the MontyLingua software [9] and followed a set of selected links to gather more textual information about the queried pairs. Preference to texts where only the rarest pairs occur was given. Indeed, if in the text there is one rare pair with high  $tf(.,.)idf(.)$  and many others for which we already have many examples (i.e. with low  $idf(.)$ ), then we should choose only few links for further crawling as the new textual material would bring more of the same.

One of the problems with web pages is that some of them only consist of link descriptions and do not contain meaningful sentences. In order to be sure that the processed web pages provide useful textual material as well as useful links, we assured that for each link in the page there were at least 300 characters of running text.

For our final corpus we retained those documents that contained at least one of the test pairs. Thus, the corpus consists of 39 million words and 122 thousand word types in nearly 16 thousand documents. The overall corpus was finally shallow parsed using the MontyLingua software [9] to obtain a predicate structure for each sentence.

## 5 Attributional Similarity

Theoretically, an attributional similarity measure can be defined as follows. Suppose that  $X_i = (X_{i1}, X_{i2}, X_{i3}, \dots, X_{ip})$  is a row vector of observations on  $p$  variables (or attributes) associated with a label  $i$ , the similarity between two units  $i$  and  $j$  is defined

as  $S_{ij} = f(X_i, X_j)$  where  $f$  is some function of the observed values. In our context, we must evaluate the similarity between two nouns which are represented by their respective word context vectors.

For our purpose, the attributional representation of a noun consists of tuples  $\langle r, v \rangle$  where  $r$  is an object or subject relation, and  $v$  is a given verb appearing within this relation with the target noun. For example, if the noun “brass” appears with the verb “press” within a subject relation, we will have the following triple  $\langle brass, press, subject \rangle$  and the tuple  $\langle press, subject \rangle$  will be an attribute of the word context<sup>2</sup> vector associated to the noun “brass”.

As similarity measures are based on real-value attributes, our task is two-fold. First, we must define a function which will evaluate the importance of a given attribute  $\langle v, r \rangle$  for a given noun. Our second goal is to find the appropriate function  $f$  that will accurately evaluate the similarity between two verb context vectors.

## 5.1 Weighting Attributes

In order to construct more precise representations of word meanings, numerous weighting schemas have been developed.

### 5.1.1 Word Frequency and IDF

The simplest form of the vector space model treats a noun  $n$  as a vector which attribute values are the number of occurrences of each tuple  $\langle v, r \rangle$  associated to  $n$  i.e.  $tf(n, \langle v, r \rangle)$ . However, the usual form of the vector space model introduces the inverse document frequency defined in the context of syntactic attribute similarity paradigm in Equation 1 where  $n$  is the target noun,  $\langle v, r \rangle$  a given attribute and  $N$  the set of all the nouns.

$$tf.idf(n, \langle v, r \rangle) = tf(n, \langle v, r \rangle) \times \log_2 \frac{card(N)}{card(\{n_i \in N | \exists \langle n_i, v, r \rangle\})} \quad (1)$$

### 5.1.2 Pointwise Mutual Information

The value of each attribute  $\langle r, v \rangle$  can also be seen as a measure of association with the noun being characterized. For that purpose, [15], [14] have proposed to use the Pointwise Mutual Information (PMI) as defined in Equation 2 where  $n$  is the target noun and  $\langle r, v \rangle$  a given attribute.

$$PMI(\langle n|r \rangle, \langle v|r \rangle) = \log_2 \frac{P(n, v|r)}{P(n|r)P(v|r)} \quad (2)$$

### 5.1.3 Conditional Probability

Another way to look at the relation between a noun  $n$  and a tuple  $\langle v, r \rangle$  is to estimate their conditional probability of co-occurrence. In our case, we are interested in knowing how strongly a given attribute  $\langle v, r \rangle$  may evoke the noun  $n$ .

<sup>2</sup> From now on, we will talk about verb context vectors instead of word context vectors.

$$P(n|v, r) = \frac{P(n, v, r)}{P(v, r)} \quad (3)$$

The conditional probability could also be seen as the  $\langle n, v \rangle$  distribution over the possible relations between  $n$  and  $v$ .

$$P(n, v|r) = \frac{P(n, v, r)}{P(r)} \quad (4)$$

Due to this characteristic, the model would suffer low selectivity - the similarity values calculated based on it would be within very short interval, which would result in unconfident decisions, as we tested and evidenced.

## 5.2 Similarity Measures

There exist many similarity measures in the context of the attributional similarity paradigm [17]. They can be divided into two main groups: (1) metrics in a high dimensional space also called Hyperspace Analogue to Language (HAL) [3], (2) measures which calculate the correlations between different probability distributions.

### 5.2.1 Cosine Similarity Measure

To quantify similarity between two words in a vector space model, the cosine metric measures to what extent two verb context vectors point along the same direction. It is defined in Equation 5.

$$\cos(X_i, X_j) = \frac{\sum_{k=1}^p X_{ik} X_{jk}}{\sqrt{\sum_{k=1}^p X_{ik}^2} \sqrt{\sum_{k=1}^p X_{jk}^2}} \quad (5)$$

### 5.2.2 Probabilistic Measures

Probabilistic measures can be applied to evaluate the similarity between nouns when they are represented by a probabilistic distribution. In this paper, we will employ two different measures.

**Ehler model:** Equation 6 presents proposed in [3] measure which evaluates the probability to interchange two word context vectors (i.e. what is the probability that the first noun is changed for the second one).

$$P(n_1|n_2) = \sum_{\langle v, r \rangle \in A} \frac{P(n_1|v, r)P(n_2|v, r)P(v, r)}{P(n_2)} \quad (6)$$

where  $A = \{\langle v, r \rangle | \exists \langle n_1, v, r \rangle \wedge \langle v, r \rangle | \exists \langle n_2, v, r \rangle\}$ .

**Lin model:** [8] defines similarity as the ratio between the amount of information needed to state the

commonality of the two nouns and the total information available about them.

$$Lin(n_1, n_2) = \frac{2 \times \sum_{\langle v, r \rangle \in A} \log_2 P(v, r)}{\sum_{\langle v, r \rangle \in B} \log_2 P(v, r) + \sum_{\langle v, r \rangle \in C} \log_2 P(v, r)} \quad (7)$$

where  $A = \{\langle v, r \rangle | \exists(n_1, v, r) \wedge \langle v, r \rangle | \exists(n_2, v, r)\}$ ,  
 $B = \{\langle v, r \rangle | \exists(n_1, v, r)\}$ ,  $C = \{\langle v, r \rangle | \exists(n_2, v, r)\}$ .

### 5.3 Global and Local Similarity

The common attributional similarity approach of gathering statistics from large corpora discards the information within single texts which has shown promising results as in [15]. Indeed building the verb context vectors based on the overall corpus by treating it as a single huge text implies the assumption that described words are monosemous.

The local attributional similarity approach, on the other hand, aims at introducing the document dimension to the word meaning acquisition process. As a consequence, different noun meanings are not merged together into single vector. The formal expression of the the local similarity is given in Equation 8 where  $D$  is the set of texts in the corpus where both  $n_1$  and  $n_2$  appear and  $sim(\cdot, \cdot)$  is any similarity measure described above calculated within the document and not over the entire corpus.

$$Lsim(n_1, n_2) = \frac{\sum_{d \in D} sim(n_1, n_2)}{card(D)} \quad (8)$$

This modification implies that the attribute values are calculated within the document for each member of the sum.

The global similarity works as an indicator that the words  $n_1$  and  $n_2$  are similar and the local similarity confirms that  $n_1$  and  $n_2$  are not just only similar, but instead good synonym candidates. Hence their product reaches maximal value when the words compared are synonyms. In Equation 9  $Gsim(\cdot, \cdot)$  is any similarity measure computed over the entire corpus.

$$Psim(n_1, n_2) = Gsim(n_1, n_2) \times Lsim(n_1, n_2) \quad (9)$$

## 6 Results and Discussion

The success over the ESL test does not guarantee success in real-world applications and the test also shows problematic issues [4]. However, the scores have an intuitive appeal, they are easily interpretable, and the expected performance of a random guesser (25%) and typical non-native speaker performance are both known (64.5%), thus making TOEFL-like tests a good basis for evaluation.

All the models proposed in this paper were tested on the subset of the 23 noun questions of the 50 multiple-choice synonym questions taken from ESL. Table 2 shows the different results obtained for the HAL models and the Probabilistic models.

			Global	Local	Product
HAL	tf	1	39.13%	73.91%	73.91%
		4		73.91%	69.57%
	tf.idf	1	52.17%	73.91%	65.22%
		4		69.57%	69.57%
	PMI	1	78.26%	65.22%	78.26%
		4		73.91%	78.26%
	cosPr	1	73.91%	60.87%	73.91%
		4		<b>82.61%</b>	<b>82.61%</b>
Prob	Ehlert	1	78.26%	65.22%	69.57%
		4		60.87%	73.91%
	Lin	1	60.87%	73.91%	69.57%
		4		78.26%	69.57%

Table 2: Performance for full noun vocabulary.

For the local similarity, we make a distinction between the results obtained on the set of documents which contain both words (being compared) at least once or four times (lines marked “1” and “4” in tables 2 and 3).

For the HAL models, the best results are obtained by the cosine of conditional probability reaching 82.61% accuracy (i.e. 19 correct answers out of 23). An interesting characteristic of PMI is the fact that it behaves steadily and does not gain anything by introducing our local similarity measure or the product of similarities. As it is known PMI is biased toward rare events, but here we compare pairs of words in documents where they occur more often than by chance and thus PMI can not manifest its specificity.

The Probabilistic models, likewise the HAL models, give better results for the texts with more occurrences of the examined nouns. The best results are obtained by Lin measure with 78.26% accuracy for  $Lsim$ . One interesting result is the fact that the Ehlert model gives the best results on the global similarity while it loses greatly when introducing the local similarity. In fact, the Ehlert model is an asymmetric measure, which gives an important part of its weight to the marginal probability of the examined answer word. When dealing globally, the measure shows a tendency to select the word with lowest probability. In fact, like the Pointwise Mutual Information, Ehlert is biased to rare cases. When compared to locally obtained values the figures show that indeed it does not attribute much importance to the contexts. When calculating the local Ehlert measure, the marginal probability of the answer varies from document to document but in fact turns out to be more stable when local similarities are averaged. As a consequence, it loses selectivity.

In this first analysis, we took into account all the nouns of the corpus with their respective verb context vectors. However, the same calculations can be done just by looking at the 94 nouns of the 23 noun questions<sup>3</sup>. The impact of the other nouns in the corpus is only on the marginal probabilities and on the  $idf$  values. This experiment is reasonable since we want to distinguish between just a limited set of nouns. We need factors that can point out the differences and similarities between them and as a consequence the rest of the noun vocabulary is useless. Table 3 presents the results with the 94 nouns space.

<sup>3</sup> Some of the nouns appear in more than one test case hence 94 instead of  $23 \times 5 = 115$

			Global	Local	Product
HAL	tf	1	39.13%	73.91%	73.91%
		4		73.91%	69.57%
	tf.idf	1	60.87%	69.57%	73.91%
		4		65.22%	65.22%
	PMI	1	65.22%	13.04%	30.43%
		4		26.09%	30.43%
cosPr	1	65.22%	69.57%	86.96%	
	4		<b>82.61%</b>	<b>91.30%</b>	
Prob	Ehlert	1	65.22%	60.87%	69.57%
		4		60.87%	69.57%
	Lin	1	56.52%	65.22%	69.57%
		4		78.26%	69.57%

Table 3: Performance for 94 ESL nouns.

The overall best results were again obtained by  $Psim(.,.)$  of cosine of conditional probability with 91.30% accuracy (21 correct answers over 23). However, almost all other measures loose in accuracy in all cases although they keep the same characteristics as shown in Table 2 when comparing the global, local and product figures. PMI shows a tendency to perform worse than random guesser. This observation is not a surprise since the synonyms tend to co-occur more often than by chance and so they receive lower weights by this scheme than when two unrelated words co-occur in a document. In this manner the synonymous words result with lower similarity than non-synonymous ones. Table 4 illustrates how the global similarity highlights related words yet the local similarity is the measure that selects the correct option.

stem	Global	Local	Product
a) column	0.0066	0.0370	0.0002
b) bark	0.0230	0.0225	0.0005
c) stalk	0.0278	<b>0.0577</b>	<b>0.0016</b>
d) trunk	<b>0.0288</b>	0.0151	0.0004

Table 4: Global vs. Local cosPr.

	Global	Local	Product
1	60.87%	65.22%	82.61%
4		78.26%	82.61%

Table 5: Global PMI for 94 ESL nouns.

It seems worth to investigate the combination between global association measure and local term representation thus taking advantage of more reliable association values still maintaining the context vector unambiguous. This effect is evidenced for the PMI comparing Tables 3 and 5.

## 7 Conclusions

According to [14] large enough corpora are necessary for human level performance on TOEFL synonymy test. But the common approach of gathering statistics from large corpora discards the information within single text. On the other hand, [15] shows that synonyms co-occur in texts more often than by chance. In this paper, we proposed a method which combines

both approaches by employing global and local evidence of attributional similarity into a single measure. The methodology was evaluated on the noun space of the 50 multiple-choice synonym questions taken from the ESL and reached 91.30% accuracy with the cosine of conditional probability. The results presented here encourages us to perform larger scale evaluation and experiments in word meaning acquisition.

## References

- [1] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805–810, Acapulco, 2003.
- [2] J. Curran and M. Moens. Improvements in automatic thesaurus extraction. In *Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX)*, pages 59–66, Philadelphia, USA, 2002.
- [3] B. Ehlert. Making accurate lexical semantic similarity judgments using word-context co-occurrence statistics. Master’s thesis, University of California, San Diego, 2003.
- [4] D. Freitag, M. Blume, J. Byrnes, E. Chow, S. Kapadia, R. Rohwer, and Z. Wang. New experiments in distributional representations of synonymy. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL)*, pages 25–32, Ann Arbor, Michigan, 2005.
- [5] W. Gale, K. Church, and D. Yarowsky. One sense per discourse. In *HLT ’91: Proceedings of the workshop on Speech and Natural Language*, pages 233–237, Morristown, NJ, USA, 1992.
- [6] M. Jarmasz and S. Szpakowicz. Rogets thesaurus and semantic similarity. In *Proceedings of Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 212–219, Borovets, Bulgaria, 2004.
- [7] T. Landauer and S. Dumais. Solution to plato’s problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.
- [8] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.
- [9] H. Liu. Montylingua: An end-to-end natural language processor with common sense. available at: <http://web.media.mit.edu/~hugo/montylingua>, 2004.
- [10] D. L. Medin, R. L. Goldstone, and D. Gentner. Similarity involving attributes and relations: judgments of similarity and differences are not inverses. *Psychological Science*, 1(1):64–69, 1990.
- [11] R. Rapp. Utilizing the one-sense-per-discourse constraint for fully unsupervised word sense induction and disambiguation. In *Proceedings of Forth Language Resources and Evaluation Conference, LREC*, 2004.
- [12] M. Sahlgren. Towards pertinent evaluation methodologies for word-space models. In *Proceedings of LREC 2006: Language Resources and Evaluation*, Genoa, Italy, 2006.
- [13] M. Sahlgren and J. Karlgren. Vector-based semantic analysis using random indexing for cross-lingual query expansion. In *CLEF ’01: Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, pages 169–176, London, UK, 2002.
- [14] E. Terra and C. Clarke. Frequency estimates for statistical word similarity measures. In *Proceedings of HTL/NAACL 2003*, pages 165–172, Edmonton, Canada, 2003.
- [15] P. D. Turney. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. *Lecture Notes in Computer Science*, 2167:491–502, 2001.
- [16] P. D. Turney, M. L. Littman, J. Bigham, and V. Shnayder. Combining independent modules in lexical multiple-choice problems. In *In Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003.*, pages 101–110, 2003.
- [17] J. Weeds, D. Weir, and D. McCarthy. Characterising measures of lexical distributional similarity. In *Proceedings of COLING 2004*.



# Memory-Based Semantic Role Labeling of Catalan and Spanish

Roser Morante and Antal van den Bosch  
Dept. of Language and Information Sciences  
Tilburg University, P.O.Box 90153  
NL-5000 LE Tilburg, The Netherlands  
{*R.Morante,Antal.vdnBosch*}@*uvt.nl*

## Abstract

In this paper we present a memory-based semantic role labeling (SRL) system for Catalan and Spanish. We approach the SRL task as two distinct classification problems: the assignment of semantic roles to arguments of verbs, and the assignment of semantic classes to verbs. We hypothesize that the two tasks can be solved in a uniform way, for both languages. Building on the same pool of features reported useful in earlier work, we train two classifiers for the two sub-tasks, selecting features systematically in a hill-climbing search. We use the IB1 classifier, a supervised memory-based learning algorithm based on the  $k$ -nn classifier. The system achieves overall F-scores of 85.69 for Catalan, and 84.12 for Spanish.

## Keywords

Semantic role labeling, memory-based learning, Spanish, Catalan.

## 1 Introduction

Semantic role labeling (SRL) is a sentence-level natural-language processing (NLP) task in which semantic roles are assigned to all arguments of a predicate [8]. Identifying semantic roles can be useful for several NLP applications such as information extraction [14], or in machine translation, where automatically identified predicates can be reordered as a pre-processing step to statistical MT [10]. The CoNLL-2004 and CoNLL-2005 Shared Tasks [1, 2] addressed SRL for English, providing a well-defined context for research and evaluation in this field.

In this paper we present a semantic role labeling system that is an enhanced version of an earlier system [13] developed for the task *Multilevel Semantic Annotation of Catalan and Spanish* [12] in the context of SemEval-2007. The general SRL task consists of two tasks: the assignment of semantic roles (SR) to arguments of verbs, and the prediction of the lexico-semantic class of the verb (SC). We develop systems for each of the tasks, for each of the two languages. For the SR task there are 39 classes in the Catalan training corpus and 48 in the Spanish training corpus. For the SC task there are 17 classes in both the Catalan and the Spanish corpora. The fact that verbs

belong to a certain class depends on their argument structure. For example, class *d2* covers agentive ditransitive verbs, which have a double object (patient, beneficiary), like change of possession (*dar*, ‘give’) and communication verbs (*decir*, ‘tell’).

The engine of the two systems for semantic role (SR) and semantic class (SC) prediction for both languages is a memory-based classifier. Memory-based language processing [4] is based on the idea that NLP problems can be solved by storing annotated examples of the problem in their literal form in memory, and applying similarity-based reasoning on these examples in order to solve new ones. Keeping literal forms in memory has been argued to provide a key advantage over abstracting methods in NLP that ignore exceptions and sub-regularities [5]. In general, NLP tasks aiming at aspects of semantic analysis are difficult to model by abstract rules. In SRL, it is difficult to formulate processing rules even for humans because semantic roles are inherently tied to meaning, inheriting all the ambiguity that lexical semantics is faced with – predicates with more than one possible meaning typically license different sets of semantic frames with each meaning. Since lexical word sense disambiguation is shown to be solvable at state-of-the-art levels by memory-based learning [9, 7], and since memory-based learning has also been applied to English SRL [15], we considered using memory-based learning for our present SRL experiments.

Building on a pool of features that have been successfully used in earlier work on SRL, we train two similar classifiers to predict the semantic class of the verb and the semantic roles separately, for both languages. With this study we intend to test whether individual systems could produce competitive results in both tasks, and whether they would be robust enough when applied to two languages and to the out-of-domain test sets provided. Additionally, our goal is to analyse what the most informative sets of features are in this task.

The data provided in the shared task are sentences with tokenized words annotated with lemmas, parts-of-speech, syntactic information, semantic roles, and the semantic classes of the verb (see Figure 1). Although the setting is similar to the CoNLL-2005 Shared Task, two important differences are that the corpora are smaller (500K words), and that the syntactic information is based on a manually annotated treebank carrying information on syntactic functions (i.e. direct object, indirect object, etc.).

INPUT----->					OUTPUT----->			
BASIC_INPUT_INFO----->		EXTRA_INPUT_INFO----->			NE NS----->	SR----->	SC----->	
WORD	TN	TV	LEMMA	POS	SYNTAX	NE NS	SC	PROPS----->
Las	-	-	el	daofpo	{S(sn-SUJ(espec.fp*)	*	-	*
conclusiones	*	-	conclusion	ncfp000	(grup.nom.fp*	*	05059980n	*
de	-	-	de	sps00	{sp(prepa*)	*	-	*
la	-	-	el	daofso	{sn(espec.fs*)	(ORG*	-	*
comision	*	-	comision	ncfs000	(grup.nom.fs*	*	06172564n	*
Zapatero	-	-	Zapatero	np00000	(grup.nom.*)	(PER*)	-	*
.	-	-	.	Fc	{S.F.R*	*	-	*
que	-	-	que	pr0cn00	(relatiu-SUJ*)	*	-	(Arg0-CAU*)
ampliara	-	*	ampliar	vmfjso	(gv*)	*	al	(V*)
el	-	-	el	daoms0	{sn-CD(espec.ms*)	*	-	(Arg1-PAT*)
plazo	*	-	plazo	ncms000	(grup.nom.ms*	*	10935305n	*
de	-	-	de	sps00	{sp(prepa*)	*	-	*
trabajo	*	-	trabajo	ncms000	{sn(grup.nom.ms*)}}}}}	*	00377035n	*
.	-	-	.	Fc	{*}}}}}}}	*	-	*
quedan	-	*	quedar	vmfp00	(gv*)	*	b3	(V*)
para	-	-	para	sps00	{sp-CC(prepa*)	*	-	(ArgM-TMP*)
despues_del	-	-	despues_del	spcms	{sp(prepa*)	*	-	*
verano	*	-	verano	ncms000	{sn(grup.nom.ms*)}}}}}	*	10946199n	*
.	-	-	.	Fp	{*}}}}}}}	*	-	*

Fig. 1: An example of an annotated sentence [12].

For additional information on the corpora, tagsets, and annotation manuals, we refer the reader to [12], and to the official website of the task<sup>1</sup>.

The paper is organised as follows. In Section 2 we present our double-classifier SRL system. In Section 3 the results are presented at various levels of granularity, and we report on feature selection experiments. In Section 4 we formulate our conclusions.

## 2 System description

We approach the SRL task as composed of two distinct classification problems: the assignment of semantic roles to arguments of a predicate (SR) and the assignment of semantic classes to predicates (SC). We hypothesize that the two problems can be solved uniformly for both languages. We build two similar systems that differ only in some of the features used, as outlined below.

Both the SR and the SC tasks are solved in two phases: (1) A pre-processing phase of *focus selection*, similar to the sequentialization step in [11]. Focus selection consists of identifying the potential candidates to be assigned a semantic role or a semantic verb class. (2) Classification, i.e. the actual assignment of roles and verb classes.

Regarding the focus selection process, the system starts by detecting a target verb, marked in the corpora as such. Then it identifies the complete form of the verb (which in the corpus is tagged as verb group, infinitive, gerund, etc.), and the clause boundaries in order to look for the siblings of the verb that exist within the same clause. The phrases with syntactic function *subject* are annotated in the corpora as siblings of the verb. For each sentence, the focus selection process produces two groups of focus tokens: on the one hand, the verbs, and on the other, the siblings of the verbs. These tokens will be the focal elements of the examples in each training set. Table 1 lists the number of training and test instances for each task.

<sup>1</sup> www.lsi.upc.edu/~nlp/semEval/msacs.html.

	Training 3LB		Test 3LB		Test CESS	
	Ca.	Sp.	Ca.	Sp.	Ca.	Sp.
SR	23202	24668	1335	1451	1241	1186
SC	8932	9707	510	615	463	465

Table 1: Number of instances per corpus for each task ('Ca' stands for Catalan, 'Sp' stands for Spanish).

We approach the SR and SC tasks as single-step classification tasks. We assume that all verbs belong to a class, so we generate one classification for each verb. As for the SR task, we assume that most siblings of the verb will have a class, except for those that have syntactic functions AO, ET, MOD, NEG, IMPERS, PASS, and VOC, as these never carry a semantic role in the training corpora; they are assigned the NONE tag. Because the amount of instances with a NONE class is proportionally low, we do not consider it necessary to filter these cases out.

Regarding the learning algorithm, we use the IB1 classifier as implemented in TiMBL (version 5.1) [6], a supervised inductive algorithm for learning classification tasks based on the  $k$ -nearest neighbor classification rule [3]. In IB1, similarity is defined by a feature-level distance metric between a test instance and a memorized example. The metric combines a per-feature value distance metric with global feature weights that account for relative differences in discriminative power of the features.

In our study the IB1 algorithm was parametrized by using Jeffrey Divergence as the similarity metric, gain ratio for feature weighting, using 11  $k$ -nearest neighbors, and weighting the class vote of neighbors as a function of their inverse linear distance [6].

We developed the systems by performing cross-validation experiments, iterated for every step in the feature selection process. Feature selection was performed by starting with a set of basic features (essentially the identity and the parts-of-speech tags of the head words involved, in their local context) and gradually adding new features. For training, the SR system took 8 seconds, and the SC system 4. For testing,

the systems took a total average of 9 seconds for the SC task (i.e. 64.38 instances per second), and 87 seconds for the SR task (16.04 instances per second). To evaluate the systems, two test sets were used for each language: one from the same training corpus (3LB) and one out-of-domain test set (CESS-ECE).

## 2.1 Features

We collected a pool of features that should in theory be useful for both the SC and SR tasks. Most of these features are described in earlier work as providing useful information for semantic role labeling [8, 17, 1, 2, 16]. They encode the identity and other syntactic aspects of the verb in focus and its clausal siblings. After experimenting with 323 features, we selected 98 for the SR task and 77 for the SC task. In order to select the features, we started with a basic system, the results of which were used as a baseline. Every new feature that was added to the basic system was evaluated in terms of average accuracy in a 10-fold cross-validation experiment; if it improved the performance on held-out data, it was added to the selection. One problem with this hill-climbing method is that the selection of features is determined by the order in which the features have been introduced. We selected it because it is a fast heuristics method, in comparison, for example, to genetic algorithms. We also performed experiments applying the feature selection process reported in [15], a bi-directional hill climbing process. However, experiments with this advanced method did not produce a better selection of features.

The features for the SR prediction task are the following (between parentheses we specify the number of features):

- Features of the verb in focus (6). They are shared by all the instances that represent phrases belonging to the same clause:

**VForm**; **VLemma**; **VCau**: binary features that indicate if the verb is in a causative construction with *hacer*, *fer* or if the main verb is *causar*; **VPron**, **VImp**, **VPass**: binary features that indicate if the verb is pronominal, impersonal, and in passive form respectively.

- Features of the sibling in focus (12):

**SibSynCat**: syntactic category; **SibSynFunc**: syntactic function; **SibPrep**: preposition; **SibLemW1**, **SibPOSW1**, **SibLemW2**, **SibPOSW2**, **SibLemW3**, **SibPOSW3**: lemma and POS of the first, second and third words of the sibling; **SibRelPos**: position of the sibling in relation to the verb (PRE or POST); **Sib+1RelPos**: position of the sibling next to the current phrase in relation to the verb (PRE or POST); **SibAbsPos**: absolute position of the sibling in the clause.

- Features that describe properties of the content word (CW) of the focus sibling (10): in the case of prepositional phrases, the CW is taken to be the head of the first noun phrase; in cases of coordination, we only select the first element of the coordination.

**CWord**; **CWLemma**; **CWPOS**: we take only the first character of the POS provided; **CWPOSType**:

the type of POS, second character of the POS provided; **CWGender**; **CWne**: boolean feature that indicates if the CW is a named entity; **CWtmp**, **CWloc**: boolean features that indicate if the CW is a temporal or a locative adverb respectively; **CW+2POS**, **CW+3POS**: POS of the second and third words after CW.

- Features of the clause containing the verb in focus (24):

**CCtot**: total number of siblings with function CC; **SUJRelPos**, **CAGRelPos**, **CDRelPos**, **CIRelPos**, **ATRRelPos**, **CPREDRelPos**, **CREGRelPos**: relative positions of siblings with functions SUJ, CAG, CD, CI,ATR, CPRED, and CREG in relation to verb (PRE or POST); **SEsib**: boolean feature that indicates if the clause contains a verbal *se*; **SIBtot**: total number of verb siblings in the clause; **SynFuncSib8**, **SynCatSib8**, **PrepSib8**, **W1Sib8**, **W2Sib8**, **W3Sib8**, **W4Sib8**, **SynFuncSib9**, **SynCatSib9**, **PrepSib9**, **W1Sib9**, **W2Sib9**, **W3Sib9**, **W4Sib9**: syntactic function, syntactic category, preposition, and first to fourth word of siblings 8 and 9.

- Features extracted from the verbal frames lexicon (43). The task organization provided lexicons of verbal frames for Catalan and Spanish. We access the lexicon to check if it is possible for a verb to have a certain semantic role:

The features are boolean: **Arg0-AGT**, **Arg0-CAU**, **Arg0-EXP**, **Arg0-TEM**, **Arg1-AGT**, **Arg1-PAT**, **Arg1-TEM**, **Arg2-ATR**, **Arg2-PAT**, **Arg3-ATR**, **ArgM-CAU**, **Arg2-LOC**, **Arg2-ADV**, **Arg1-LOC**, **Arg3-LOC**, **ArgM-ADV**, **ArgM-LOC**, **ArgM-MNR**, **ArgM-TMP**, **Arg0**, **Arg1**, **Arg1-EXT**, **Arg2**, **Arg2-BEN**, **Arg2-EFI**, **Arg2-EXT**, **Arg2-INS**, **Arg2-ORI**, **Arg3**, **Arg3-BEN**, **Arg3-EIN**, **Arg3-EXT**, **Arg3-FIN**, **Arg3-INS**, **Arg3-ORI**, **Arg4-DES**, **Arg4-EFI**, **ArgL**, **ArgM**, **ArgM-CAU**, **ArgM-EXT**, **ArgM-FIN**, **ArgX**.

For the SC prediction task the features are similar, but not the same. We point out the differences in both directions.

- Features exclusive to the SR system:

Verb form (**VForm**), verb lemma (**VLemma**), absolute position of the sibling in the clause (**SibAbsPos**), function of the sibling (**SibSynFunc**), preposition of the sibling (**SibPrep**), POS of the second and third words after CW (**CW+2POS**, **CW+3POS**), feature indicating whether the CW is a named entity (**CWne**, **SIBtot**), syntactic function, syntactic category, preposition and first to fourth word of siblings 8 and 9 (**SynFuncSib8**, **SynCatSib8**, **PrepSib8**, **W1Sib8**, **W2Sib8**, **W3Sib8**, **W4Sib8**, **SynFuncSib9**, **SynCatSib9**, **PrepSib9**, **W1Sib9**, **W2Sib9**, **W3Sib9**, **W4Sib9**).

- Features exclusive to the SC system:

**AllCats**: vector of the syntactic categories of the siblings in the order that they appear in the clause; **AllFuncs**: vector of the functions of the siblings in the order that they appear; **AllFuncsBin** vector with eight binary values that represent if a sibling with that function is present or not; **Sib+1Prep**, **Sib+2Prep**: prepositions of the two siblings after the verb.

### 3 Results

#### 3.1 Overall results

SR TASK	PP	Prec.	Recall	$F_{\beta=1}$
Test ca.3LB	74.32%	87.20%	86.52%	86.86
Test ca.CESS	61.62%	83.45%	78.59%	80.95
Overall ca	67.97%	85.32%	82.55%	83.90
Test sp.3LB	68.56%	83.36%	82.85%	83.10
Test sp.CESS	73.98%	85.78%	85.70%	85.74
Overall sp	71.27%	84.57%	84.27%	84.42
Overall SR	69.62%	84.95%	83.41%	84.16

SC TASK	PP	Prec.	Recall	$F_{\beta=1}$
Test ca.3LB	90.86%	90.30%	88.72%	89.50
Test ca.CESS	90.41%	90.20%	88.27%	89.22
Overall ca	90.64%	90.25%	88.50%	89.37
Test sp.3LB	84.12%	80.00%	78.44%	79.21
Test sp.CESS	90.54%	89.89%	89.89%	89.89
Overall sp	86.88%	84.30%	83.36%	83.83
Overall SC	88.67%	87.12%	85.81%	86.46

SRL TASK	PP	Prec.	Recall	$F_{\beta=1}$
Overall ca	–	86.93%	84.49%	85.69
Overall sp	–	84.38%	83.87%	84.12
Overall SRL	–	85.61%	84.17%	84.89

**Table 2:** Overall results in the SR (above), SC (middle), and general SRL tasks (‘PP’: perfect propositions; Prec.: precision; ‘ca’: Catalan; ‘sp’: Spanish).

The overall results of the system are shown in Table 2. The SC system displays a better generalization performance (overall  $F_{\beta=1} = 86.46$ ) than the SR system (overall  $F_{\beta=1} = 84.16$ ), which is also reflected in the average score in terms of correctly identified propositions (88.67% in SC, and 69.62% in SR). The two tasks are inherently different, and there are also marked differences in their example sets. There are less classes in the SC task than in the SR task (cf. Table 3), and they are more homogeneous in the SC task (cf. the entropy rate in Table 3). Additionally, the annotation process might have been different for semantic roles from the one for verb semantic classes. Finally, the verbs are easier to identify in the focus selection process because they are marked in the corpus.

	SR task		SC task	
	ca.3LB	sp.3LB	ca.3LB	sp.3LB
Classes	39	48	17	17
Entropy	3.5069	3.6231	2.5609	2.7665

**Table 3:** Number of classes and entropy rate in the train corpus (3LB) for Catalan (ca) and Spanish (sp).

The comparison of results between the 3LB test set and the out-of-domain CESS–ECE set shows that the tendency is different for Spanish and Catalan. The results for Spanish are unexpected because the sp.CESS–ECE test set yields better results: in the SR task, it is processed with  $F_{\beta=1}=85.74$ , while the sp.3LB is processed with  $F_{\beta=1}=83.10$ . On the SC task, the sp.CESS–ECE is processed with  $F_{\beta=1}=89.89$ , while sp.3LB is processed with  $F_{\beta=1}=79.21$ . The same tendency is observed in the results of the other partici-

pants in the task, suggesting that it may be relevant to investigate how the sp.3LB corpus was annotated and partitioned.

The results for Catalan follow the expectations. On the SR task, the  $F_{\beta=1}$  rate (80.95) for the out-of-domain ca.CESS–ECE test set is 6 points lower than the  $F_{\beta=1}$  rate (86.86) for the ca.3LB test set, and in the SC task, the  $F_{\beta=1}$  rate (89.22) for the ca.CESS–ECE test set is also lower than the rate (89.50) for the ca.3LB test set, although the difference is small.

With respect to the robustness of our systems, the results seem to suggest that the SC system is more robust than the SR system. Concerning the difference between the two languages, we observe that the SR system performs better for Spanish (84.42) than for Catalan (83.90), while the SC system performs better for Catalan (89.37) than for Spanish (83.83). The results suggest that the language is not the main factor of the differences in performance, confirming our hypothesis that the task can be approached with the same system for both languages.

#### 3.2 Analysis of the results on the out-of-domain test set

Next, we present detailed results on the Spanish CESS–ECE test (Tables 4 and 5). The differences in score between classes are higher in the SR task than in the SC task. The average of precision and recall is similar in each of the tasks, and both precision and recall are higher for the SC task.

SP–CESS	N	Precision	Recall	$F_{\beta=1}$
Overall	1028	85.78%	85.70%	85.74
Arg0–AGT	224	93.21%	91.96%	92.58
Arg0–CAU	6	100%	50%	66.67
Arg1	28	88.46%	82.14%	85.19
Arg1–LOC	1	0.00%	0.00%	0.00
Arg1–PAT	258	93.82%	94.19%	94.00
Arg1–TEM	98	85.71%	91.84%	88.67
Arg2	22	64.29%	81.82%	72.00
Arg2–ATR	73	91.67%	90.41%	91.03
Arg2–BEN	26	100%	100.00%	100.00
Arg2–EFI	3	0.00%	0.00%	0.00
Arg2–EXT	0	0.00%	0.00%	0.00
Arg2–LOC	4	0.00%	0.00%	0.00
Arg2–PAT	1	0.00%	0.00%	0.00
Arg3–ATR	0	0.00%	0.00%	0.00
Arg3–BEN	1	100.00%	100.00%	100.00
Arg3–EIN	0	0.00%	0.00%	0.00
Arg3–FIN	3	100.00%	33.33%	50.00
Arg3–ORI	3	0.00%	0.00%	0.00
Arg4–DES	6	80.00%	66.67%	72.73
ArgL	5	16.67%	20.00%	18.18
ArgM–ADV	69	66.20%	68.12%	67.14
ArgM–CAU	11	62.50%	45.45%	52.63
ArgM–FIN	13	64.71%	84.62%	73.33
ArgM–LOC	79	78.21%	77.22%	77.71
ArgM–MNR	7	40.00%	57.14%	47.06
ArgM–TMP	87	87.65%	81.61%	84.52
V	465	100.00%	100.00%	100.00

**Table 4:** Detailed results on the Spanish CESS–ECE test set for the SR task (N: number of appearances in the test corpus).

SP-CESS	N	Precision	Recall	$F_{\beta=1}$
Overall	465	89.89%	89.89%	89.89
a1	19	85.71%	94.74%	90.00
a2	4	80.00%	100.00%	88.89
b1	9	63.64%	77.78%	70.00
b2	1	100.00%	100.00%	100.00
c1	25	81.82%	72.00%	76.60
c3	57	84.85%	98.25%	91.06
c5	3	75.00%	100.00%	85.71
d1	14	78.57%	78.57%	78.57
d2	248	97.00%	91.13%	93.97
d3	78	91.78%	85.90%	88.74
d4	1	14.29%	100.00%	25.00
d5	1	33.33%	100.00%	50.00
e1	5	100.00%	100.00%	100.00

**Table 5:** Detailed results on the Spanish CESS-ECE test set for the SC task (N: number of appearances in the test corpus).

With the SR task, class scores are roughly correlated with the frequency of occurrence of classes in the training corpus. Some of the most frequently occurring classes in the test set (Arg0-AGT, Arg1-PAT, Arg1-TEM, Arg2-ATR) are identified at the highest accuracy rates. Aside from the fact that more training examples provide a better chance of being used as nearest neighbors in classification, the feature selection method is also naturally biased towards these classes. High scores attained for medium-frequency classes such as Arg2-BEN can typically be explained by the fact that they have overt markers: in Spanish, Arg2-BEN is always marked by the Indirect Object function and the prepositions *a* or *para*.

However, some other medium-frequency classes are identified at medium or low accuracy levels of accuracy. In particular, there seems to be a considerable group-internal confusion among the ArgM arguments. For example, ArgM-ADV is confused with ArgM-LOC in 13.0% of the cases, ArgM-MNR in 7.2%, and ArgM-TMP in 4.4% of its occurrences.

In the SC task the three most frequent classes (d2, d3, c3) are predicted at high levels of accuracy. At the same time, some of the less frequent classes also receive high scores (a2, b2, c5, e1). In contrast with the SR task, all classes receive a non-zero score.

### 3.3 Analysis of the results for all semantic roles

Table 6 shows the  $F_{\beta=1}$  rates for all individual semantic roles in the test sets. Most of the large differences between scores obtained for the same semantic role in different test sets can be explained by the fact that these semantic roles have a low frequency (Arg0-EXP, Arg1-EXT, Arg1-LOC, Arg2-EFI, Arg2-EXT, Arg2-LOC, Arg3-BEN, Arg3-FIN, Arg3-ORI, ArgL, ArgM-MNR). Some semantic roles are stable across test sets and receive a medium score (Arg0-AGT, Arg1, Arg1-PAT, Arg1-TEM, Arg2, Arg2-ATR, Arg2-BEN). This might mean that these semantic roles are frequent, that the features are expressive for these classes, and possibly that they are annotated consistently.

At the same time, some roles receive very different scores in the different test sets (Arg2-LOC, Arg2-DES, ArgM-CAU, ArgL, ArgM-MNR, ArgM-TMP). This might be caused by different frequencies of the semantic roles in the corpus, but also by inconsistent annotation.

	ca.3LB	ca.CESS	sp.3LB	sp.CESS
Arg0-AGT	93.21	91.47	90.79	92.58
Arg0-CAU	40.00	42.11	45.45	66.67
Arg0-EXP	-	0.00	50.00	-
Arg0-TEM	-	-	0.00	-
Arg1	75.68	80.00	79.17	85.19
Arg1-AGT	-	-	0.00	-
Arg1-EXT	0.00	-	100.00	-
Arg1-LOC	0.00	66.67	0.00	0.00
Arg1-PAT	94.50	93.46	92.17	94.00
Arg1-TEM	90.99	88.57	89.95	88.67
Arg2	78.38	77.14	74.07	72.00
Arg2-ATR	92.77	92.72	95.38	91.03
Arg2-BEN	100.00	100.00	94.74	100.00
Arg2-EFI	-	-	40.00	0.00
Arg2-EXT	66.67	40.00	-	0.00
Arg2-LOC	36.36	57.14	30.43	0.00
Arg2-ORI	-	0.00	-	-
Arg2-PAT	-	-	-	0.00
Arg3-ATR	0.00	0.00	0.00	-
Arg3-BEN	-	-	0.00	100.00
Arg3-EIN	0.00	0.00	-	0.00
Arg3-FIN	100.00	0.00	0.00	50.00
Arg3-ORI	25.00	0.00	61.54	0.00
Arg4-DES	72.73	54.55	50.00	72.73
Arg4-EFI	0.00	0.00	-	-
ArgL	80.00	33.33	20.00	18.18
ArgM	-	0.00	-	-
ArgM-ADV	63.79	61.07	71.89	67.14
ArgM-CAU	80.95	66.67	78.79	52.63
ArgM-EXT	0.00	0.00	-	-
ArgM-FIN	84.00	84.24	87.80	73.33
ArgM-LOC	70.31	75.24	70.24	77.71
ArgM-MNR	51.16	21.05	53.66	47.06
ArgM-PAT	-	-	0.00	-
ArgM-TMP	91.43	41.48	77.46	84.52
V	99.22	99.25	99.10	100.00

**Table 6:**  $F_{\beta=1}$  rate for all semantic roles in the four test sets.

### 3.4 Analysis of features selected for SR

Table 7 shows the twenty features with the highest gain ratio in the SR task for Catalan and Spanish. The feature SibSynFunc has the highest gain ratio (Catalan 0.7198, Spanish 0.7661). Among these twenty features, sixteen are the same in both languages. The four features exclusive to Catalan are Arg2-INS, Arg0-EXP, Arg0-TEM, and CWPOSType. Mostly these are features from the verb lexicon. For Spanish the deviating features are also from the verb lexicon: Arg2-ADV, Arg2-ORI, Arg0, and Arg3-BEN.

To sum up, features do not obtain the same gain ratio for both languages, but they show the same tendency. The top features encode information about the syntactic function, the preposition, the syntactic category, and the relative position of the focus sibling; the lemma and POS of the first word of the current

ca.roles		sp.roles	
feat.	GR	feat.	GR
SybSynFunc	0.7198	SybSynFunc	0.7661
Arg2-INS	0.4449	SibSynCat	0.4128
SibPrep	0.4179	SibPrep	0.4124
SibSynCat	0.4069	Arg2-ADV	0.3834
ATRRelPos	0.3745	SibRelPos	0.3554
SibRelPos	0.3444	ATRRelPos	0.3451
Arg0-EXP	0.3428	SibPOSW1	0.3224
SibPOSW1	0.3369	Arg3-FIN	0.3065
Arg3-FIN	0.3363	SibLemW1	0.2927
CWPOS	0.3095	Arg2-ORI	0.2871
SibLemW1	0.3082	CWPOS	0.2853
Arg1-PAT	0.3035	Arg0	0.2729
CREGRelPos	0.2653	CWtmp	0.2548
Arg0-TEM	0.2644	Arg1-PAT	0.2474
CIRelPos	0.2481	CWord	0.2466
Arg1-TEM	0.2444	CREGRelPos	0.2428
CWord	0.2422	CIRelPos	0.2372
CWtmp	0.2402	Arg1-TEM	0.2367
CWPOSType	0.2399	Arg3-BEN	0.2367
CWLemma	0.2377	CWLemma	0.2341

**Table 7:** Features with the highest Gain Ratio in the SR task.

sibling; the POS and word of the content word; the relative position of the sibling with function ATR, CREG and CI; and information from the verb lexicon.

### 3.5 Analysis of features selected for SC

ca.verbs		sp.verbs	
feat.	GR	feat.	GR
Arg0-EXP	0.7328	ATRRelPos	0.5515
ATRRelPos	0.5470	Arg3-FIN	0.4964
Arg1-TEM	0.4397	Arg3-BEN	0.4704
Arg2-EXT	0.3929	Arg1-TEM	0.4294
Arg0-AGT	0.3849	Arg0-EXP	0.3655
Arg2-LOC	0.3302	Arg2-BEN	0.3530
Arg1-PAT	0.3248	Arg0-AGT	0.3385
Arg2-BEN	0.3215	Arg2-ATR	0.3334
CIRelPos	0.3176	Arg0-CAU	0.3333
ArgX	0.3131	Arg3	0.3330
Arg3-ATR	0.2982	Arg0	0.3329
Arg4-EFI	0.2980	Arg1-PAT	0.3291
Arg3-EIN	0.2940	Arg2	0.3046
Arg2-ATR	0.2918	VCau	0.3006
Arg2-INS	0.2915	CIRelPos	0.2915
Arg2-EFI	0.2815	Arg2-EFI	0.2850
Arg2	0.2770	Arg1-EXT	0.2824
Arg3-BEN	0.2713	Arg2-PAT	0.2798
CWLemma	0.2565	CWLemma	0.2658
Arg3-EXT	0.2558	SibLemW1	0.2369

**Table 8:** Features with the highest Gain Ratio in the SC task.

Table 8 shows the twenty features with the highest gain ratio in the SC task for Catalan and Spanish. Most of the features originate from the verb lexicon. The feature with the highest gain ratio in Catalan is Arg0-EXP (0.7328), whereas in Spanish it is ATRRelPos (0.5515).

A comparison of both systems shows that in the

SC system the features with the highest gain ratio are mostly features from the verb lexicon, whereas in the SR system only some features from the lexicon are the top positions. The features CWLemma, ATRRelPos, and CWLemma are in the top positions in both systems, as well as the lexicon features Arg0-EXP, Arg1-PAT, and Arg1-TEM.

### 3.6 Analysis of the effect of removing features

Tables 9 and 10 contain information about the effects of removing features from the SR system. Table 9 focuses on the effects of removing groups of features. Removing the features that provide information about the sibling in focus ('Sibling') causes a clear decrease in the system's performance (on average 21.9 points of F-score). Removing the verb lexicon features ('Lexicon Roles') and the features of the verb in focus ('Verb') also causes a decrease in the system's performance, but much lower. Removing the features of the clause containing the verb in focus ('Clause') causes a slight decrease, and removing the features that describe properties of the content word ('CW') causes different effects in each test set, but just a slight decrease or increase. These results show that the most expressive features in this task are the features on the sibling in focus.

	ca.3LB	ca.CESS	sp.3LB	sp.CESS
With all	86.86	80.95	83.10	85.74
- Sibling	<b>-18.68</b>	<b>-19.71</b>	<b>-24.31</b>	<b>-24.90</b>
- Lexicon Roles	-1.74	-2.73	-3.79	-2.14
- Verb	-2.44	-3.29	-2.65	-1.26
- Clause	-0.70	-2.05	-0.84	-0.19
- CW	+0.09	-1.18	-0.75	+0.29

**Table 9:** Effect of removing groups of features from the SR system. (Overall  $F_{\beta=1}$ ).

	ca.3LB	ca.CESS	sp.3LB	sp.CESS
With all	86.86	80.95	83.10	85.74
- SybSynFunc	<b>-6.16</b>	<b>-2.80</b>	<b>-6.67</b>	<b>-5.80</b>
- SibPrep	-0.52	-0.52	-1.07	+0.83
- Arg0-EXP	0.00	0.00	-0.15	-0.10
- Arg3-FIN	0.00	0.00	0.00	0.00
- Arg2-ADV	0.00	0.00	0.00	0.00
- Arg2-ORI	0.00	0.00	-0.07	0.00
- Arg2-INS	0.00	0.00	-0.07	+0.20
- CWPOS	-0.09	-0.02	-0.20	+0.39
- SibPOSW1	+0.18	-0.16	-0.50	0
- SibRelPos	+0.18	-0.29	-0.46	-0.19
- SibLemW1	+0.53	-0.14	-0.69	+0.29
- SibSynCat	+0.26	-0.43	-0.06	+0.04
- ATRRelPos	-0.09	+0.15	-0.07	+0.10
- 20 feats. with highest GR	-16.90	-13.06	-26.51	-23.98

**Table 10:** Effect of removing features with high gain ratio from the SR system (overall  $F_{\beta=1}$ ).

Table 10 provides details on the effects of removing the ten individual features that have the highest gain ratio in the Catalan and Spanish training corpora (listed in Table 7). As expected, removing the feature SybSynFunc causes a clear decrease in the results (on average 5.35 points of F-score). Removing the 20

features with the highest gain ration in each of the systems (Table 7) provokes a much higher decrease.

## 4 Conclusions

We presented a memory-based semantic role labeling (SRL) system for Catalan and Spanish that makes use of full syntactic information. We approached the general SRL task as two distinct classification problems: the assignment of semantic roles to arguments of verbs, and the assignment of semantic classes to verbs. Building on a pool of features from which we selected subsets appropriate to each subtask through a hill-climbing search procedure, we trained two similar classifiers on the two subtasks using the IB1 classifier as implemented in TiMBL (version 5.1) [6]. We reported an overall performance of the system of 85.69  $F_{\beta=1}$  for Catalan, and 84.12  $F_{\beta=1}$  for Spanish.

The results show that a uniform single-classifier system can produce competitive results in both tasks. It performs slightly better on the SC task, which might be caused by several reasons: apart from the fact that the tasks are inherently different and SC may simply be easier, there are less classes in the SC task than in the SR task, with stronger predictability from the same pool of features. Other factors such as the consistency of the annotation might play a role. Results also show that the two problems can be solved in largely the same way for both languages. On the SC task the approach results in higher generalization performance for Catalan, and on the SR task the Spanish system is better. Finally, the effects of removing groups of features show that the most expressive features in the SR task are clearly the features that provide information about the sibling in focus.

## Acknowledgements

This research has been funded by the postdoctoral grant EX2005-1145 awarded by the Ministerio de Educación y Ciencia of Spain to the project *Técnicas semi-automáticas para el etiquetado de roles semánticos en corpus del español*. We are grateful to Bertjan Busser for his contribution to programming the system and to the three anonymous reviewers.

## References

- [1] X. Carreras and L. Màrquez. Introduction to the conll-2004 shared task: Semantic role labeling. In *Proceedings of CoNLL-2004*, Boston MA, USA, 2004.
- [2] X. Carreras and L. Màrquez. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of CoNLL-2005*, Ann Arbor, Michigan, June 2005.
- [3] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, 13:21-27, 1967.
- [4] W. Daelemans and A. van den Bosch. *Memory-based language processing*. Cambridge University Press, Cambridge, UK, 2005.
- [5] W. Daelemans, A. Van den Bosch, and J. Zavrel. Forgetting exceptions is harmful in language learning. *Machine Learning, Special issue on Natural Language Learning*, 34:11-41, 1999.
- [6] W. Daelemans, J. Zavrel, K. V. der Sloot, and A. V. den Bosch. TiMBL: Tilburg memory based learner, version 5.1, reference guide. Technical Report Series 04-02, ILK, Tilburg, The Netherlands, 2004.
- [7] B. Decadt, V. Hoste, W. Daelemans, and A. Van den Bosch. GAMBL, genetic algorithm optimization of memory-based WSD. In R. Mihalcea and P. Edmonds, editors, *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3)*, pages 108-112, New Brunswick, NJ, 2004. ACL.
- [8] D. Gildea and D. Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245-288, 2002.
- [9] V. Hoste, I. Hendrickx, W. Daelemans, and A. Van den Bosch. Parameter optimization for machine learning of word sense disambiguation. *Natural Language Engineering*, 8(4):311-325, 2002.
- [10] M. Komachi, Y. Matsumoto, and M. Nagata. Phrase reordering for statistical machine translation based on predicate-argument structure. In *Proceedings of the International Workshop on Spoken Language Translation: Evaluation Campaign on Spoken Language Translation*, pages 77-82, Kyoto, Japan, 2006.
- [11] L. Màrquez, P. Comas, J. Giménez, and N. Català. Semantic role labeling as sequential tagging. In *Proceedings of CoNLL-2005*, Ann Arbor, Michigan, 2005.
- [12] L. Màrquez, L. Villarejo, M. Martí, and M. Taulé. Semeval-2007 task 09: Multilevel semantic annotation of Catalan and Spanish. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 42-47, 2007.
- [13] R. Morante and B. Busser. ILK2: Semantic role labelling for Catalan and Spanish using TiMBL. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 183-186, 2007.
- [14] M. Surdeanu, S. Harabagiu, J. Williams, and P. Aarseth. Using predicate-argument structures for information extraction. In *Proceedings of the ACL 2003*, 2003.
- [15] E. Tjong Kim Sang, S. Canisius, A. van den Bosch, and T. Bogers. Applying spelling error correction techniques for improving semantic role labelling. In *Proceedings of CoNLL-2005*, pages 229-232, Ann Arbor, Michigan, 2005.
- [16] K. Toutanova, A. Haghighi, and C. Manning. Joint learning improves semantic role labeling. In *Proceedings of ACL-05*, Ann Arbor, Michigan, 2005.
- [17] N. Xue and M. Palmer. Calibrating features for semantic role labeling. In *Proceedings of 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004.

# The problems in a Question Answering system in the academic domain

P.López-Moreno, A.Ferrández, S.Roger, S.Ferrández  
Natural Language Processing and Information Systems Group  
Department of Software and Computing Systems  
University of Alicante, Spain  
{P.Lopez}@ua.es  
{antonio,sferrandez,sroger}@dlsi.es

## Abstract

In this paper we present BRUPIA, a Question Answering (QA) system in a restricted domain: the web academic environment at the University of Alicante. This system is the transformation of an open domain QA system, AliQAn. This paper focuses on explaining how an open domain system can be transformed into another one that can successfully work on a web restricted domain. We analyze the problems of carrying out this task and we also develop the necessary resources for the new system like the corpora, the questions and the set of patterns in the new domain. Finally, a new strategic approach for the improvements in the use of the terminology in web domain is proposed. The measure of evaluation is the Mean Reciprocal Rank and the final result is 32,5%.

## Keywords

Question answering system academic restricted domain web

## 1 Introduction

Different trends are used in Question Answering (QA) systems. On the one hand, the traditional QA focuses to process large amount of independent documents. Some of these systems takes part in the TREC and CLEF evaluation campaigns. On the other hand, there are systems based on the web. Most of these systems represent the restricted domain, and they use the websites to extract documents that have fixed structures and are related by means of a hierarchy of pages. The most important difference between these QA systems, is that the first one works with independent documents in a journalist style and the second one has a document collection with a web structure. The information appearing in these websites can be distributed in different texts, even in different related documents just by means of using links. It is very difficult to find the textual information as this one appears in the question. We will take an example of the queries like: *Quién es el director del DLSI? (Who is the manager of the DLSI?)*. In this case, the correct answer appears in the organization site of the Department of Software and Computing Systems (DLSI), but the sequence “manager of DLSI” is not in the text, DLSI

appears at the top of the page. The information of the page is referred to the staff working at this department, so that, the head contains the word “DLSI” and the content gives details of the different positions among them, like the manager, enclosed to their names. Another example is the following: *Qué página personal tiene Antonio Ferrández? (What is the personal website of Antonio Ferrandez?)*. Looking for the answer, we must go to the main page of DLSI department and click on “Teaching Staff”. A list of teachers is shown, where each name is a link to its personal page. But in this situation, the distance between the name and the correct answer has a main role. These examples are typical cases in a web domain.

Analyzing the restricted domain QA systems, we can make out two different tendencies. First, a restricted domain QA system has a baseline for open domain as a point of starting. This way, a general system can be transformed into a specific one. Furthermore, a QA system can be created directly for the specific domain.

We develop the first technique for our QA system in the academic domain, considering the AliQAn system as a starting point. Moreover, the specific terminology of this domain and the web structure are used in order to improve the accuracy of our system. The results obtained with the measure of evaluation Mean Reciprocal Rank (MRR) are 32,5%.

The rest of the paper is structured as follows: First, we introduce the backgrounds in a restricted domain system. Secondly, we summarize the characteristics of the restricted domain systems based on Web. Next, we explain the process to transform the baseline in the new system BRUPIA. Finally, we show the problems and the solutions found, and the main conclusions obtained.

## 2 Backgrounds

Many QA systems employed sources in order to store the specific terminology. For instance ExtrAns [5] is a QA system aimed at restricted domains, in particular terminology-rich domains. They carried out “terminological normalization”, where a term is replaced by a synset identifier when this term belongs to the category in the terminology knowledge represented by means of an ontology. The document collections in the



genomic domain was generated from Medline. In this way, although the system uses the web to extract the documents, these texts are independent.

Another example is a QA system for a home agent robot [1], which is based on templates to store the data. Each expected question topic is defined as a single query frame and each frame has a rule for SQL generation. The web crawler downloads the selected webpages from the website of the Korea Meteorological Administration and the wrapper is used to extract weather information from the webpages stored in a database.

The next study case is a system developed for the company Bell Canada to answer to client's questions in services offered by a big company [3]. They experimented with some methods of reranking with information about the domain specific language, particularly with vocabulary issues. In this case, the document collection was derived from .html and .pdf files as our system. As the structure of these files was so complicated, documents were saved as pure texts sacrificing some elements like titles, listings or tables.

In general, each restricted domain QA system uses some different techniques to the treatment of the terminology, because this one is the main role in this kind of system. In addition to this fact, there are systems that are based on an initial baseline. Some systems are based on the web and download directly the documents, while others have a document collection with independent texts.

Our approach combines some of these techniques. First, BRUPIA has a baseline system in open domain. Secondly, our system uses the web to obtain the documents and finally, we use the specific terminology of the domain in a strategic way.

### 3 Characteristics of restricted domain

The first section describes general characteristics based on [1, 2, 3].

1. Quality of responses must be higher because of the practice on the market.
2. The answers are searched in relatively small domain collections, so the redundancy is lower than in an open domain system.
3. User requirements in the quality of the answer tend to be higher in restricted domains. No answer is preferred to a wrong answer.
4. The terminology plays a central role.

Our particular contributions about web domain are represented in the following paragraphs:

1. The structure of the web documents lets to identify a symbolic structure, so it is possible to split different parts of the document in order to provide a higher score and to obtain better results.
2. Webpages contain dependent information and have a hierarchy of pages. The related information with one question can be separated in some documents, or it even can be necessary to visit different pages to find the correct answer.

## 4 Transformation of the AliQAn into BRUPIA

We propose a monolingual Spanish QA system named BRUPIA for an academic domain, particularly, the domain of the University of Alicante (UA). From our baseline AliQAn, a monolingual open domain QA system developed at the UA three years ago, we have adjusted the new system modifying the patterns and applying the necessary techniques to the treatment of the new domain knowledge. AliQAn participated in the CLEF-2005 [6] competition and last year, it participated in the CLEF-2006 [7] with a new version of our system.

We have had some problems with the new system BRUPIA. After that, we will explain a detailed description about the problems detected and the solutions proposed.

There are three large groups of problems. First of all, the generation of the corpus. We experimented with two different collections in the academic domain. Both collections were generated automatically from pages of the UA. The size of the first corpus is 102.900 documents. The second collection is more concrete than the first one. It was constituted by documents of the web but considering only the domain of the DLSI. Finally, this corpus contains 2.900 documents.

The second kind of problem is related to the system questions about an insufficient typology and several difficulties to allocate the correct type of the questions.

Finally, we analyze some problems in the baseline system regarding to the patterns.

### 4.1 Problems in the generation of the corpus

#### 4.1.1 IR<sub>n</sub> problems

IR<sub>n</sub> is a passage retrieval that returns a list of relevant documents for each question. The web structure causes that IR<sub>n</sub> returns the document in an incorrect way. For example the question: *Qué es DLSI?* (*What is DLSI?*), some documents of the corpus contain the correct answer for this question, where the word "DLSI" appears with its description, but the occurrence of this word is very low, so these documents are not returned by IR<sub>n</sub>. Nevertheless, there are documents that contain sometimes the word "DLSI", such as, the page of the staff of this department, where this word appears in the email of each person. These documents appear in the returned list by IR<sub>n</sub> but BRUPIA system cannot find the solution.

#### 4.1.2 Parsing problems

Another problem derived from the web structure is the malformation of the syntactic blocks (SB). The new documents are very different from the initial documents of the baseline system, as for as the sentences segmentation and the style of the texts. The main problem is the lack of the full-stop or period to indicate the final of the sentences. So, our parser SUPAR carries out the wrong formation of the SB because it is not able to separate correctly the different blocks. The following Figure 1 shows the malformation of the

SB caused by the absence of a point at the end of the sentences due to the automatic conversion of the text.  
*Secretario: Juan Antonio Pérez Ortiz (Secretary: Juan Antonio Perez Ortiz)*  
*Subdirector: Patricio Martínez Barco (Assistant principal: Patricio Martinez Barco)*

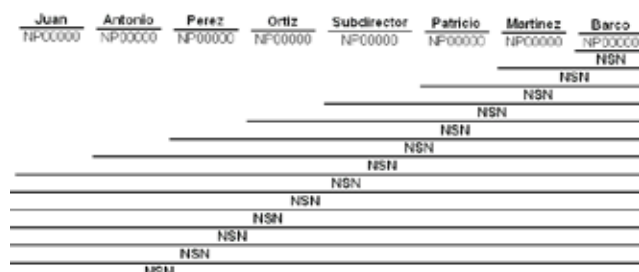


Fig. 1: Example of the segmentation of SUPAR

#### 4.1.3 Problems caused by the web structure

The information in a HTML document is semi-structured and it can be related with other pages. The style in this kind of documents is totally different from other any style. At the top, there is a page head or a title which summarizes the content of the document. If we select the webpage about the staff of this department, we can see the information represented in a table format with the names of the members of the organization. There is a question like: *Quién es el secretario del DLSI? (Who is the secretary of the DLSI?)*. In this document, the word secretary and the name appear, but the literal “DLSI” is not there, because it appears in the head.

Another problem is produced when the distance between the question words and the solution is very long. For instance, if the system looks for the projects that one teacher is carrying out: *En qué proyectos participa Patricio? (What projects does Patricio work in?)*. The name of the teacher appears at the top of the page, and then, all information appears like a list with different items, and the projects are at the end of this list. All information in the page is about the same teacher, but the name only appears in the head. So, when we look for the information there is too big distance and the score is too low.

#### 4.1.4 Problems because of the languages

This system is monolingual, so the language of the documents must be the same that the questions. These documents were downloaded automatically, so, there are some documents in different language. Sometimes, the URL indicates the language that and the documents can be leaked. However, others do not contain any indicators of the language, and when the system returns the answer is different from the question language.

## 4.2 Problems detected in questions

### 4.2.1 Problems of allocation of correct type

The system BRUPIA has a collection of the 100 questions. In accordance with the baseline typology, some questions were classified with an incorrect type. The lack of information in the question was the first reason, for example: *Cuál es el número de la centralita de la Universidad? (Which is the number of the switchboard of the University?)*. We know that this number is referred to the phone number but the system interprets the type as a quantity.

Besides, the classification patterns do not contain all the options, one example of this situation is: *Qué dirección electrónica tiene Loren Moreno Monteagudo? (What electronic direction does Loren Moreno Monteagudo have?)*. The system determines incorrectly that the type is group instead of the email type.

There are wrong cases with specific concepts for this domain, like the word “extension”, which is used for the telephone line inside of the academic domain, but as measure in the baseline system. For this situation, there is a question like: *Qué extensión tiene Jesús Peral Cortés? (What extension number does Jesus Peral Cortes have?)*.

### 4.2.2 Insufficient typology

Initially, the baseline had a typology with the following concepts: profession, first name, person, group, place country, place city, capital place, place, abbreviation, event, object, weather date, weather year, weather month, weather day, weather events, numerical economic, numerical quantity, numerical percentage, numerical measurement, numerical period, numerical age, definition, email, telephone and fax. This classification is scarce for the new domain and the system needs more concepts.

## 4.3 Problems in the baseline

### 4.3.1 Problems with the patterns

The major problems are the definitions. The journalistic style of the corpus of the baseline is composed by narrative texts, therefore the definition is more probably that appears before the term. The new domain has a different style, so in some definitions, the concept appears in the first place and afterwards the definition. So, it is more interesting to look up the definition on the right and the term on the left, modifying some parameters of the patterns.

## 5 Representation of problems

The seven types of errors detected in the adaptation to new academic domain are represented in the Figure 2. The types of problems and their percentages are represented in the graph. With regard to the colours, each type of error has one different colour, but there are two special situations like the parsing problems and the problems because of incorrect allocation of the type, which have combined colour to indicate that these failures are due to other more general problems.

	Insufficient typology	Problems in patterns	Problems of IRn	Parsing problems	Problems because of the web structure	Problems with languages	Allocations of the incorrect type
<b>Number of questions</b>	12	2	7	13	10	4	12
<b>Error percentage</b>	20 %	3 %	12 %	22 %	17 %	7 %	19 %

Table 1: Representations of problems

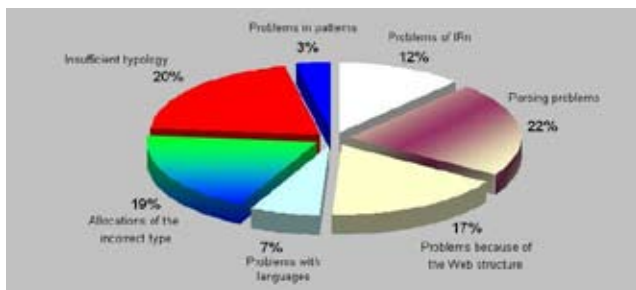


Fig. 2: Representation of problems

Concretely, the parsing problems are due to the web structure and the allocation of incorrect type is because of failures of the patterns. Most important problem are caused by the web structure that generates an error percentage of 39% (22% parsing problems caused by the web structure and 17% problems because of web structure).

In the Table 1, it is possible to distinguish the number of the questions and the error percentage of each type of error.

## 6 Proposed solutions

Once the errors are detected, we show viable solutions for the different failures presented in the previous section.

### 6.1 Solution for IRn and SUPAR

The reason of these problems consists in how the web is structured, concretely the lack of punctuation marks to indicate the end of the sentences. Along these lines, we propose to add a point at the end of the sentences to solve the segmentation of these sentences. In this way, we could solve the problem of malformation of SB carried out for SUPAR, and the problems of IRn, both caused by an incorrect segmentation of the sentences.

### 6.2 A method to improve the precision

In this point, our best innovation is presented. It consists in applying a method to use the information that pertains to different hierarchy levels in a strategic way. The related data are distributed in different texts or at least in different places of the document. Usually, the main information appears in the page head or in the title of the document. Words appearing in the head usually are not more times in the text, because these terms are general and describe all the information that

is contained in the webpage. So, we use the most important terminology that appears in a web document. We look for the words of the question which appears in the page head of the document and we remove them to the question to not look for them in the document. At the same time, we keep the definition of these words that are removed to question when the system detects that the definition question is referred to these terms. For example, for the question introduced in the initial part of this document: *Quién es el director del DLSI? (Who is the manager of the DLSI?)*. The text “DLSI” only appears in the head. To solve this question, the system removes the word “DLSI” when it detects this word in the question, and it only looks for the text “director (manager)” in the document, returning the name of the person.

Another kind of failure takes place when the information is separated by means of a high distance. For instance, a question mentioned previously: *En qué proyectos participa Patricio? (What projects does Patricio work in?)*. In the document, this name appears at the top, however the projects appear at the end of the page. In this case, we propose to remove the word “Patricio” to the question and to look for only the “proyectos (projects)”, solving the problem of the distance.

### 6.3 Solution for language problems

Two points of view are possible to solve these problems. On one hand, we could filter the language of the documents and create different corpus for each language and use the spanish corpus for the monolingual system. In the future, the other ones will be treated. So, we want to use a resource developed in this department [4], which will allow us to detect the language of the documents and to leak the spanish ones.

### 6.4 Solution for problems with the patterns

In order to solve the problems of incorrect allocation of types, the classification patterns were adapted to the new group of questions, extending the conditions.

Besides, the extraction patterns were adapted to improve the precision in our system. So that, the definition is looked after to the acronym because of the probability of appearing with this format is greater.

### 6.5 Solution for insufficient typology

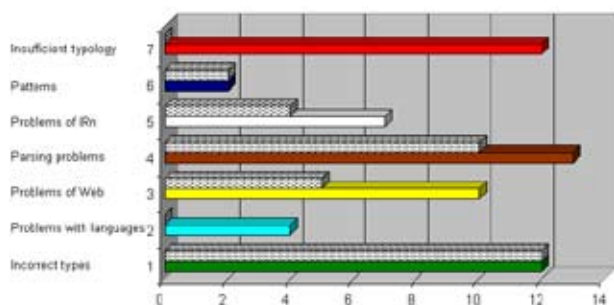
The typology must be extended considering other types like: personal page, guardianship schedule, office, subject, course and mailing dress. In addition,

1st answer	2nd answer	3rd answer
27	9	3

**Table 2:** *Results of BRUPIA*

some types were adapted for the new system BRUPIA.

## 7 Representation of solutions



**Fig. 3:** *Problems and solutions*

Two important things are represented in the Figure 3. Firstly, the detected errors that are represented in the line of down and marked with the concrete colour of each type of error and secondly, the solved errors that are situated over the detected ones. For each type, it is possible to check the obtained improvement comparing the size of both lines. The different errors detected are represented in the “y” axis, and the number of questions that have each problem is detailed in the “x” axis. The errors of classification and extraction patterns have been solved totally. Besides, some errors of the web structure have been clarified with our special contributions, ignoring the proper names of the questions that are contained in the page head, which pertain a higher hierarchic level. However, the used typology is the same than the baseline and the failures related to this concept have not been solved yet. In this way, the problem of language is future work. Even so, the final result is 32,5% of MRR.

## 8 Results

Regarding the first experiment carried out with the general training corpus for UA, the obtained results were about 5% the precision. The final result considering a set of 100 questions for the restricted domain of the DLSI is 32,5% of MRR.

In the Table 2, it is possible to distinguish the number of questions that are correct in the first, second or third position. Moreover, two answers in Valencian language returned in the second position have been considered correct. It is very interesting that the number of correct answers returned in first position is higher than the other groups.

## Acknowledgments

This research has been partially funded by the Spanish Government under project CICyT number TIN2006-15265-C06-01 and by the University of Comahue under the project 04/E062.

This work has been partially supported by the EU funded project QALL-ME (FP6 IST-033860).

## 9 Conclusion and future work

BRUPIA is a QA system for academic domain of UA. Our approach is different from the traditional QA systems, which works with independent documents. BRUPIA has a document collection with structured information and contains related data. In addition, these documents have a hierarchy of webpages that indicates how the texts are related. Our system uses the terminology of the domain in a strategic way to solve some problems with the web structure. The patterns have been adapted to new domain generalizing the conditions. Finally, we propose some alternatives to solve the specific problems in this kind of web systems. The results obtained with our experiments are 32,5% of MRR, obtaining important improvements with respect to the initial tests.

In the earliest phases, BRUPIA solved some of the problems that came up in the adaptation of the baseline system to the new domain. Nevertheless, much work is left for our future work to generate a robust system for the academic domain of UA.

## References

- [1] H. Chung, Y.-I. Song, K.-S. Han, S.-H. Kim, D.-S. Yoon, J.-Y. Lee, and H.-C. Rim. A practical QA System in Restricted Domains. *In Proceedings of the ACL Workshop*, pages 39–45, July 2004.
- [2] D. Ferrés and H. Rodríguez. Experiments Adapting an Open-Domain Question Answering System to the Geographical Domain Using Scope-Based Resources. *In Proceedings of the Multilingual Question Answering Workshop of the EACL*, pages 69–76, April 2006.
- [3] D.-N. Hai and K. Kosseim. The problem of Precision in Restricted-Domain Question-Answering. Some Proposed Methods of Improvement. *In Proceedings of the ACL Workshop*, pages 8–15, July 2004.
- [4] T. Martínez, E. Noguera, R. Muoz, and F. Llopis. Web track for CLEF2005 at ALICANTE UNIVERSITY. *In Workshop of Cross-Language Evaluation Forum (CLEF)*, September 2005.
- [5] F. Rinaldi, J. Dowdall, G. Schneider, and A. Persidis. Answering Questions in the Genomics Domain. *In Proceedings of the ACL Workshop*, pages 46–53, July 2004.
- [6] S. Roger, S. Ferrández, A. Ferrández, J. Peral, F. Llopis, A. Aguilar, and D. Tomás. AliQAn, Spanish QA System at CLEF-2005. *In Workshop of Cross-Language Evaluation Forum (CLEF)*, 2005.
- [7] S. Ferrández, P. López-Moreno, S. Roger, A. Ferrández, J. Peral, X. Alvarado, E. Noguera, and F. Llopis. AliQAn and BRILI QA systems at CLEF 2006. *In Workshop of Cross-Language Evaluation Forum (CLEF)*, 2006.

# Improved Word Alignments Using the Web as a Corpus

Preslav Nakov  
UC Berkeley  
EECS, CS division  
Berkeley, CA 94720  
nakov@cs.berkeley.edu

Svetlin Nakov  
Sofia University  
5 James Boucher Blvd.  
Sofia, Bulgaria  
nakov@fmi.uni-sofia.bg

Elena Paskaleva  
Bulgarian Academy of Sciences  
25A Acad. G. Bonchev Str.  
Sofia, Bulgaria  
hellen@lml.bas.bg

## Abstract

We propose a novel method for improving word alignments in a parallel sentence-aligned bilingual corpus based on the idea that if two words are translations of each other then so should be many words in their local contexts. The idea is formalised using the Web as a corpus, a glossary of known word translations (dynamically augmented from the Web using bootstrapping), the vector space model, linguistically motivated weighted minimum edit distance, competitive linking, and the IBM models. Evaluation results on a Bulgarian-Russian corpus show a sizable improvement both in word alignment and in translation quality.

## Keywords

Machine translation, word alignments, competitive linking, Web as a corpus, string similarity, edit distance.

## 1 Introduction

The beginning of modern *Statistical Machine Translation* (SMT) can be traced back to 1988, when Brown et al. [5] from IBM published a formalised mathematical formulation of the translation problem and proposed five word alignment models – IBM models 1, 2, 3, 4 and 5. Starting with a bilingual parallel sentence-aligned corpus, the IBM models learn how to translate individual words and the probabilities of these translations. Later, decoders like the ISI REWRITE DECODER [9] became available, which made it possible to quickly build SMT systems with decent quality.

An important shift happened in 2004, when the PHARAOH model [11] has been proposed, which uses whole *phrases* (typically of length up to 7, not necessarily representing linguistic units), rather than just words. This led to a significant improvement in translation quality, since phrases can encode local gender/number agreement, facilitate choosing the correct sense for ambiguous words, and naturally handle fixed phrases and idioms. While methods have been proposed for learning translation phrases directly [17], the most popular *alignment template approach* [23] requires bi-directional word alignments at the sentence level from which phrases consistent with those alignments are extracted. Since better word alignments can lead to better phrases<sup>1</sup>, improving word alignments remains one of the primary research problems in SMT: in

<sup>1</sup> The dependency between word alignments and translation

fact, there are more papers published yearly on word alignments than on any other SMT subproblem.

In the present paper, we describe a novel method for improving word alignments using the Web as a corpus, a glossary of known word translations (dynamically augmented from the Web using bootstrapping), the vector space model, weighted minimum edit distance, competitive linking, and the IBM models. The potential of the method is demonstrated on a Bulgarian-Russian bilingual corpus.

The rest of the paper is organised as follows: section 2 explains the method in detail, section 3 describes the corpus and the resources used, section 4 contains the evaluation, section 5 points to important related research, and section 6 concludes with some possible directions for future work.

## 2 Method

Our method combines two similarity measures which make use of different information sources. First, we define a language-specific modified minimum edit distance, based on linguistically-motivated rules targeting Bulgarian-Russian cognate pairs. Second, we define a distributional semantic similarity measure, based on the idea that if two words represent a translations pair, then the most frequently co-occurring words in their local contexts should be translations of each other as well. This intuition is formalised using the Web as a corpus, a bilingual glossary of word translation pairs used as “bridges”, and the vector space model. The two measures are combined with competitive linking [19] in order to obtain high quality word translation pairs, which are then appended to the bilingual sentence-aligned corpus in order to bias the subsequent training of the IBM word alignment models [5].

### 2.1 Orthographic Similarity

We use an orthographic similarity measure, which is based on the *minimum edit distance* (MED) or Levenshtein distance [16]. MED calculates the distance between two strings  $s_1$  and  $s_2$  as the minimum number of edit operations – INSERT, REPLACE, DELETE – needed to transform  $s_1$  into  $s_2$ . For example, the MED between *r. первый* (Russian, ‘first’) and *b. първият*

quality is indirect; improving the former does not necessarily improve the latter.

(Bulgarian, ‘*the first*’) is 4: three REPLACE operations ( $e \rightarrow \mathfrak{t}$ ,  $\mathfrak{y} \rightarrow \mathfrak{i}$ ,  $\mathfrak{h} \rightarrow \mathfrak{j}$ ) and one INSERT (of  $\mathfrak{t}$ ).

We modify the classic MED in two ways. First, we normalise the two strings, taking into account some general graphemic correlations between the phonetico-graphemic systems of the two closely-related Slavonic languages – Bulgarian and Russian:

- For Russian words, we remove the letters  $\mathfrak{b}$  and  $\mathfrak{t}$ , as their graphemic collocations are excluded in Bulgarian, e.g.  $\mathfrak{b}$  between two consonants (*r. СИЛЬНО*  $\leftrightarrow$  *b. СИЛНО*, *strongly*),  $\mathfrak{t}$  following a consonant (*r. ОБЪЯВЛЕНИЕ*  $\leftrightarrow$  *b. ОБЯВЛЕНИЕ*, *an announcement*), etc.
- For Russian words, we remove the ending  $\mathfrak{h}$ , which is the typical nominative adjective ending in Russian, but not in Bulgarian, e.g. *r. ДЕТСКИЙ*  $\leftrightarrow$  *b. ДЕТСКИ* (*children’s*).
- For Bulgarian words, we remove the definite article, e.g. *b. ГОРСКИЯТ* (*the forestal*)  $\rightarrow$  *b. ГОРСКИ* (*forestal*). The definite article is the only agglutinative morpheme in Bulgarian and has no counterpart in Russian: Bulgarian has definite, but not indefinite article, and there are no articles in Russian.
- We transliterate the Russian-specific letters (missing in the Bulgarian alphabet) or letter combinations in a regular way:  $\mathfrak{y} \leftrightarrow \mathfrak{i}$ ,  $\mathfrak{e} \leftrightarrow \mathfrak{e}$ , and  $\mathfrak{sh} \leftrightarrow \mathfrak{sh}$ , e.g. *r. ЭЛЕКТРОН*  $\leftrightarrow$  *b. ЕЛЕКТРОН* (*an electron*), *r. ВЫЛ*  $\leftrightarrow$  *b. ВИЛ* (past participle of *to howl*), *r. ШТАБ*  $\leftrightarrow$  *b. ШАБ* (*mil. a staff*), etc.
- Finally, we remove all double letters in both languages (e.g.  $\mathfrak{nn} \rightarrow \mathfrak{n}$ ;  $\mathfrak{cc} \rightarrow \mathfrak{c}$ ): While consonant and vowel doubling is very rare in Bulgarian (except at morpheme boundaries for a limited number of morphemes), it is more common in Russian, e.g. in case of words of foreign origin: *r. АССАМБЛЕЯ*  $\rightarrow$  *b. АСАМБЛЕЯ* (*an assembly*)

Second, we use different letter-pair specific costs for REPLACE. We use 0.5 for all vowel to vowel substitutions, e.g.  $\mathfrak{o} \leftrightarrow \mathfrak{e}$  as in *r. ЛИЦО*  $\leftrightarrow$  *b. ЛИЦЕ* (*a face*). We also use 0.5 for some consonant-consonant replacements, e.g.  $\mathfrak{c} \leftrightarrow \mathfrak{z}$ . Such regular phonetic changes are reflected in different ways in the orthographic systems of the two languages, Bulgarian being more conservative and sticking to morphological principles. For example, in Bulgarian the final  $\mathfrak{z}$  in prefixes like *из-* and *раз-* never change to  $\mathfrak{c}$ , while in Russian they sometimes do, e.g. *r. ИССЛЕДОВАТЕЛЬ*  $\leftrightarrow$  *b. ИЗСЛЕДОВАТЕЛ* (*an explorer*), *r. РАССКАЗ*  $\leftrightarrow$  *b. РАЗКАЗ* (*a story*), etc.

We use a cost of 1 for all other replacements.

It is easy to see that this *modified minimum edit distance* (MMED) is more adequate than MED – it is only 0.5 for *r. первый* and *b. първият*: we first normalise them to *перви* and *първи*, and then we do a single vowel-vowel REPLACE with the cost of 0.5.

We transform MMED into a similarity measure, *modified minimum edit distance ratio* (MMEDR) using the following formula ( $|s|$  is the number of letters in  $s$  before the normalisation):

$$\text{MMEDR}(s_1, s_2) = 1 - \frac{\text{MMED}(s_1, s_2)}{\max(|s_1|, |s_2|)}$$

Below we compare MMEDR with *minimum edit distance ratio* (MEDR):

$$\text{MEDR}(s_1, s_2) = 1 - \frac{\text{MED}(s_1, s_2)}{\max(|s_1|, |s_2|)}$$

and *longest common subsequence ratio* (LCSR) [18]:

$$\text{LCSR}(s_1, s_2) = \frac{|\text{LCS}(s_1, s_2)|}{\max(|s_1|, |s_2|)}$$

In the last definition,  $\text{LCS}(s_1, s_2)$  refers to the longest common subsequence of  $s_1$  and  $s_2$ , e.g.  $\text{LCS}(\text{первый}, \text{първият}) = \text{прв}$ , and therefore

$$\text{MMEDR}(\text{первый}, \text{първият}) = 3/7 \approx 0.43$$

We obtain the same score using MMED:

$$\text{MMED}(\text{первый}, \text{първият}) = 1 - 4/7 \approx 0.43$$

while with MMEDR we have:

$$\text{MMEDR}(\text{первый}, \text{първият}) = 1 - 0.5/7 \approx 0.93$$

## 2.2 Semantic Similarity

The second basic similarity measure we use is WEB-ONLY, which measures the semantic similarity between a Russian word  $w_{ru}$  and a Bulgarian word  $w_{bg}$  using the Web as a corpus and a glossary  $G$  of known Bulgarian-Russian translation pairs used as “bridges”. The basic idea is that if two words are translations of each other then many of the words in their respective local contexts should be mutual translations as well.

First, we issue a query to Google for  $w_{ru}$  or  $w_{bg}$ , limiting the language to Russian or Bulgarian, and we collect the text from the resulting 1,000 snippets. We then extract the words from the local context (two words on either side of the target word), we remove the stopwords (prepositions, pronouns, conjunctions, interjections and some adverbs), we lemmatise the remaining words, and we filter out the words that are not in  $G$ . We further replace each Russian word with its Bulgarian counter-part in  $G$ . As a result, we end up with two Bulgarian frequency vectors, corresponding to  $w_{ru}$  and  $w_{bg}$ , respectively. Finally, we TF.IDF-weight the vector coordinates [31] and we calculate the semantic similarity between  $w_{bg}$  and  $w_{ru}$  as the cosine between their corresponding vectors.

## 2.3 Combined Similarity Measures

In our experiments (see below), we have found that MMEDR yields a better precision, while WEB-ONLY has a better recall. Therefore we tried to combine the two similarity measures in different ways:

- WEB-AVG: *average* of WEB-ONLY and MMEDR;
- WEB-MAX: *maximum* of WEB-ONLY and MMEDR;
- WEB-CUT: The value of  $\text{WEB-CUT}(s_1, s_2)$  is 1, if  $\text{MMEDR}(s_1, s_2) \geq \alpha$  ( $0 < \alpha < 1$ ), and is equal to  $\text{WEB-ONLY}(s_1, s_2)$ , otherwise.



## 2.4 Competitive Linking

The above similarity measures are used in combination with *competitive linking* [19], which assumes that a source word is either translated with a single target word or is not translated at all. Given a sentence pair, the similarity between all Bulgarian-Russian word pairs is calculated<sup>2</sup>, which induces a fully-connected weighted bipartite graph. Then a greedy approximation to the maximum weighted bipartite matching in that graph is extracted as follows: First, the most similar pair of unaligned words is aligned and both words are discarded from further consideration. Then the next most similar pair of unaligned words is aligned and the two words are discarded, and so forth. The process is repeated until there are no unaligned words left or until the maximal word pair similarity falls below a pre-specified threshold  $\theta$  ( $0 \leq \theta \leq 1$ ), which could leave some words unaligned.

## 3 Resources

### 3.1 Parallel Corpus

We use a parallel sentence-aligned Bulgarian-Russian corpus: the Russian novel *Lord of the World*<sup>3</sup> by Alexander Beliaev and its Bulgarian translation<sup>4</sup>. The text has been sentence aligned automatically using the alignment tool *MARK ALISTeR* [26], which is based on the Gale-Church algorithm [8]. As a result, we obtained 5,827 parallel sentences, which we divided into *training* (4,827 sentences), *tuning* (500 sentences), and *testing set* (500 sentences).

### 3.2 Grammatical Resources

We use monolingual dictionaries for lemmatisation. For Bulgarian, we use a large morphological dictionary, containing about 1,000,000 wordforms and 70,000 lemmata [25], created at the Linguistic Modeling Department, Institute for Parallel Processing, Bulgarian Academy of Sciences. The dictionary is in DE-LAF format [30]: each entry consists of a wordform, a corresponding lemma, followed by morphological and grammatical information. There can be multiple entries for the same wordform, in case of multiple homographs. We also use a large grammatical dictionary of Russian in the same format, consisting of 1,500,000 wordforms and 100,000 lemmata, based on the Grammatical Dictionary of A. Zaliznjak [33]. Its electronic version was supplied by the Computerised fund of Russian language, Institute of Russian language, Russian Academy of Sciences.

### 3.3 Bilingual Glossary

We built a bilingual glossary from an online Bulgarian-Russian dictionary<sup>5</sup>. First, we removed all multi-word expressions. Then we combined each Rus-

sian word with each Bulgarian one – due to polysemy/homonymy some words had multiple translations. As a result, we obtained a glossary  $G$  of 3,794 word translation pairs.

Due to the modest glossary size, in our initial experiments, we were lacking translations for many of the most frequent context words. For example, when comparing  $r$ . *платье* (*a dress*) and  $b$ . *рокля* (*a dress*), we find adjectives like  $r$ . *свадебное* (*wedding*) and  $r$ . *вечернее* (*evening*) among the most frequent Russian context words, and  $b$ . *сватбена* and  $b$ . *вечерна* among the most frequent Bulgarian context words. While missing in our bilingual glossary, it is easy to see that they are orthographically similar and thus likely cognates. Therefore, we automatically extended  $G$  with possible cognate pairs. For the purpose, we collected the most frequent 30 non-stopwords  $RU_{30}$  and  $BG_{30}$  from the local contexts of  $w_{ru}$  and  $w_{bg}$ , respectively, that were missing in our glossary. We then calculated the MMEDR for every word pair  $(r, b) \in (RU_{30}, BG_{30})$ , and we added to  $G$  all pairs for which the value was above 0.90. As a result, we managed to extend  $G$  with 6,289 additional high-quality translation pairs.

## 4 Evaluation

We evaluate the similarity measures in four different ways: manual analysis of WEB-CUT, alignment quality of competitive linking, alignment quality of the IBM models for a corpus augmented with word translations from competitive linking, and translation quality of a phrase-based SMT trained on that corpus.

### 4.1 Manual Evaluation of WEB-CUT

Recall that by definition  $WEB-CUT(s_1, s_2)$  is 1, if  $MMEDR(s_1, s_2) \geq \alpha$ , and is equal to  $WEB-ONLY(s_1, s_2)$ , otherwise. To find the best value for  $\alpha$ , we tried all values  $\alpha \in \{0.01, 0.02, 0.03, \dots, 0.99\}$ . For each value, we word-aligned the training sentences from the parallel corpus using competitive linking and WEB-CUT, and we extracted a list of the distinct aligned word pairs, which we added twice as additional “sentence” pairs to the training corpus. We then calculated the perplexity of IBM model 4 for that augmented corpus. This procedure was repeated for all candidate values of  $\alpha$ , and finally  $\alpha = 0.62$  was selected as it yielded the lowest perplexity.<sup>6</sup>

The last author, a native speaker of Bulgarian who is fluent in Russian, manually examined and annotated as *correct*, *rough* or *wrong* the 14,246 distinct aligned Bulgarian-Russian word type pairs, obtained with competitive linking and WEB-CUT for  $\alpha = 0.62$ . The following groups naturally emerge:

1. **“Identical” word pairs** ( $MMEDR(s_1, s_2) = 1$ ): 1,309 or 9% of all pairs. 70% of them are completely identical, e.g. *скоро* (*soon*) is spelled the same way in both Bulgarian and Russian. The remaining 30% exhibit regular graphemic changes, which are recognised by MMEDR (See section 2.1.)

<sup>6</sup> This value is close to 0.58, which has been found to perform best for LCSR on Western-European languages [15].

<sup>2</sup> Due to their special distribution, stopwords and short words (one or two letters) are not used in competitive linking.

<sup>3</sup> <http://www.lib.ru>

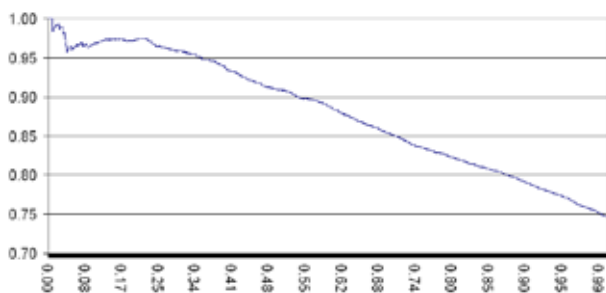
<sup>4</sup> <http://borislav.free.fr/mylib>

<sup>5</sup> <http://www.bgru.net/intr/dictionary/>

2. “**True friends**” ( $\alpha \leq \text{MMEDR}(s_1, s_2) < 1$ ): 5,289 or 37% of all pairs. This group reflects changes combining regular phonemic and morphemic (grammatical) correlations. Examples include similar but not identical affixes (e.g. the Russian prefixes **во-** and **со-** become **въ-** and **съ-** in Bulgarian), similar graphemic shapes of morpheme values (e.g. the Russian singular feminine adjective endings **-ая** and **-яя** become **-а** and **-я** in Bulgarian), etc.
3. “**Translations**” ( $\text{MMEDR}(s_1, s_2) < \alpha$ ): 7,648 or 54 % of all pairs. Here the value of  $\text{WEB-ONLY}(s_1, s_2)$  is used. We divide this group into the following sub-categories: *correct* (73%), *rough* (3%) and *wrong* (24%).

Our analysis of the *rough* and *wrong* sub-groups of the latter group exposes the inadequacy of the idea of reducing sentence translation to a sequence of word-for-word translations, even for closely related languages like Bulgarian and Russian. Laying aside the translator’s freedom of choice, the translation correspondences often link a word to a phrase, or a phrase to another phrase, often idiomatically, and sometimes involve syntactic transformations as well. For example, when aligning the Russian word *r. отвернуться* to its Bulgarian translation *b. обръщам гръб* (*to turn back*), competitive linking wrongly aligns *r. отвернуться* to *b. гръб* (*a back*). Similarly, when the Russian for *to challenge*, *r. бросать вызов* (lit. *to throw a challenge*), is aligned to its Bulgarian translation *b. хвърлям ръкавица* (lit. *to throw a glove*), this results in wrongly aligning *r. вызов* (*a challenge*) to *b. ръкавица* (*a glove*). Note however that such alignments are still helpful in the context of SMT.

Figure 1 shows the precision-recall curve for the manual evaluation of competitive linking with  $\text{WEB-CUT}$  for the third group only ( $\text{MMEDR}(s_1, s_2) < \alpha$ ), considering both *rough* and *wrong* as incorrect. We can see that the precision is 0.73 even for recall of 1.



**Fig. 1: Manual evaluation of WEB-CUT:** Precision-recall curve for competitive linking with  $\text{WEB-CUT}$  on the “translations” sub-group ( $\text{MMEDR}(s_1, s_2) < 0.62$ ).

## 4.2 Word Alignments

### 4.2.1 Gold Standard Word Alignments

The last author, a linguist, manually aligned the first 100 sentences from the training corpus, thus creating a

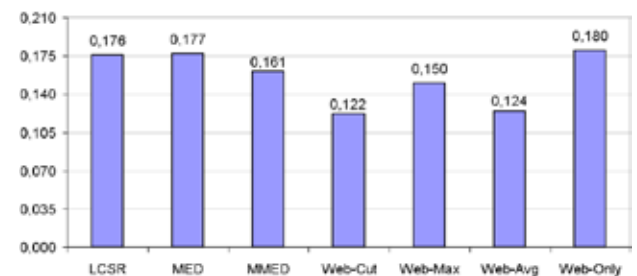
gold standard for calculating the *alignment error rate* (*AER*) for the different similarity measures.

Manual alignments typically use two kinds of links: *sure* and *possible*. As we have seen above, even for closely related languages like Russian and Bulgarian, the alignment of each source word to a target one could be impossible, unless a suitable convention is adopted. Particularly problematic are the “hanging” single words – typically stemming from syntactic differences. We prefer to align such word to the same target word to which is aligned the word it is dependent on, and to mark the link as *possible*, rather than *sure*. More formally, if the source Russian word  $x_{ru}$  is translated with a pair of target Bulgarian words  $x_{bg}$  and  $y_{bg}$ , where  $x_{ru}$  is a *sure* translation of  $x_{bg}$ , and  $y_{bg}$  is a grammatical or “empty” word ensuring the correct surface presentation of the grammatical/lexical relation, then we add a *possible* link between  $y_{bg}$  to  $x_{ru}$  as well.

For instance, the Russian genitive case is typically translated in Bulgarian with a prepositional phrase, **на+noun**, e.g. *r. звуки музыки* (*sounds of music*) is translated as *b. звучите на музиката*. Other examples include regular ellipsis/dropping of elements specific for one of the languages only, e.g. subject dropping in Bulgarian, ellipsis of Russian auxiliaries in present tense, etc. For example, *r. я знал* (*I knew*) can be translated as *b. аз знаех*, but also as *b. знаех*. On the other hand, *r. он герой* (*‘he is a hero’, lit. ‘he hero’*) is translated as *b. той е герой* (lit. *‘he is hero’*).

### 4.2.2 Competitive Linking

Figure 2 shows the AER for competitive linking with all 7 similarity measures: our orthographic and semantic measures ( $\text{MMEDR}$  and  $\text{WEB-ONLY}$ ), the three combinations ( $\text{WEB-CUT}$ ,  $\text{WEB-MAX}$  and  $\text{WEB-AVG}$ ), as well as for  $\text{LCSR}$  and  $\text{MEDR}$ . We can see an improvement of up to 6 AER points when going from  $\text{LCSR}/\text{MEDR}/\text{WEB-ONLY}$  to  $\text{WEB-CUT}/\text{WEB-AVG}$ . Note that here we calculated the AER on a modified version of the 100 gold standard sentences – the stopwords and the punctuation were removed in order to ensure a fair comparison with competitive linking, which ignores them. In addition, each of the measures has its own threshold  $\theta$  for competitive linking (see section 2.4), which we set by optimising perplexity on the training set, as we did for  $\alpha$  in the section 4.1: we tried all values of  $\theta \in \{0.05, 0.10, \dots, 1.00\}$ , and we selected the one which yielded the lowest perplexity.



**Fig. 2: AER for competitive linking:** stopwords and punctuation are not considered.



### 4.2.3 IBM Models

In our next experiment, we first extracted a list of the distinct word pairs aligned with competitive linking, and we added them twice as additional “sentence” pairs to the training corpus, as in section 4.1. We then generated two directed IBM model 4 word alignments (Bulgarian → Russian, Russian → Bulgarian) for the new corpus, and we combined them using the *intersect+grow heuristic* [22]. Table 3 shows the AER for these combined alignments. We can see that while training on the augmented corpus lowers AER by about 4 points compared to the baseline (which is trained on the original corpus), there is little difference between the similarity measures.

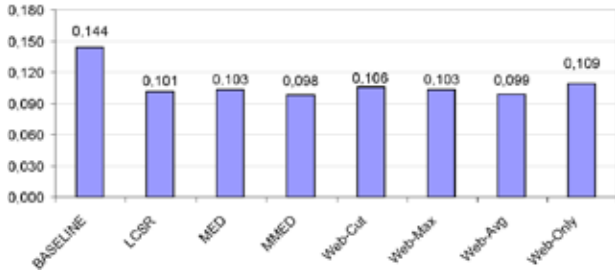


Fig. 3: AER for IBM model 4: *intersect+grow*.

## 4.3 Machine Translation

As we said in the introduction, word alignments are an important first step in the process of building a phrase-based SMT. However, as many researchers have reported, better AER does not necessarily mean improved machine translation quality [2]. Therefore, we built a full Russian → Bulgarian SMT system in order to assess the actual impact of the corpus augmentation (as described in the previous section) on the translation quality.

Starting with the symmetrised word alignments described in the previous section, we extracted phrase-level translation pairs using the *alignment template approach* [13]. We then trained a log-linear model with the standard feature functions: language model probability, word penalty, distortion cost, forward phrase translation probability, backward phrase translation probability, forward lexical weight, backward lexical weight, and phrase penalty. The feature weights, were set by maximising Bleu [24] on the development set using *minimum error rate training* [21].

Tables 4 and 5 show the evaluation on the test set in terms of Bleu and NIST scores. We can see a sizable difference between the different similarity measures: the combined measures (WEB-CUT, WEB-MAX and WEB-AVG) clearly outperforming LCSR and MEDR. MMEDR outperforms them as well, but the difference from LCSR is negligible.

## 5 Related Work

Many researchers have exploited the intuition that words in two different languages with similar or identical spelling are likely to be translations of each other.

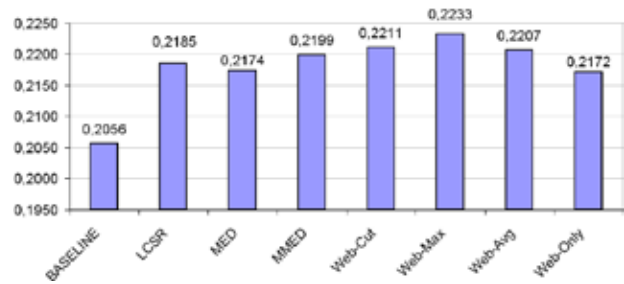


Fig. 4: Translation quality: *Bleu* score.

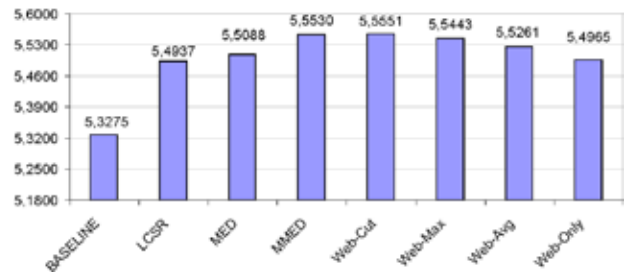


Fig. 5: Translation quality: *NIST* score.

Al-Onaizan & al. [1] create improved Czech-English word alignments using probable cognates extracted with one of the variations of LCSR [18] described in [32]. They tried to constrain the co-occurrences, to seed the parameters of IBM model 1, but their best results were achieved by simply adding the cognates to the training corpus as additional “sentences”. Using a variation of that technique, Kondrak, Marcu and Knight [15] demonstrated improved translation quality for nine European languages. We extend this work, by adding competitive linking [19], language-specific weights, and a Web-based semantic similarity measure.

Koehn & Knight [12] describe several techniques for inducing translation lexicons. Starting with unrelated German and English corpora, they look for (1) identical words, (2) cognates, (3) words with similar frequencies, (4) words with similar meanings, and (5) words with similar contexts. This is a bootstrapping process, where new translation pairs are added to the lexicon at each iteration.

Rapp [27] describes a correlation between the co-occurrences of words that are translations of each other. In particular, he shows that if in a text in one language two words *A* and *B* co-occur more often than expected by chance, then in a text in another language the translations of *A* and *B* are also likely to co-occur frequently. Based on this observation, he proposes a model for finding the most accurate cross-linguistic mapping between German and English words using non-parallel corpora. His approach differs from ours in the similarity measure, the text source, and the addressed problem. In later work on the same problem, Rapp [28] represents the context of the target word with four vectors: one for the words immediately preceding the target, another one for the ones immediately following the target, and two more for the words one more word before/after the target.

Fung and Yee [7] extract word-level translations

from non-parallel corpora. They count the number of sentence-level co-occurrences of the target word with a fixed set of “seed” words in order to rank the candidates in a vector-space model using different similarity measures, after normalisation and TF.IDF-weighting [31]. The process starts with a small initial set of seed words, which are dynamically augmented as new translation pairs are identified. We do not have a fixed set of seed words, but generate it dynamically, since finding the number of co-occurrences of the target word with each of the seed words would require prohibitively many search engine queries.

Diab & Finch [6] propose a statistical word-level translation model for comparable corpora, which finds a cross-linguistic mapping between the words in the two corpora such that the source language word-level co-occurrences are preserved as closely as possible.

Finally, there is a lot of research on string similarity which has been or potentially could be applied to cognate identification: Ristad&Yianilos’98 [29] learn the MED weights using a stochastic transducer. Tiedemann’99 [32] and Mulloni&Pekar’06 [20] learn spelling changes between two languages for LCSR and for NEDR respectively. Kondrak’05 [14] proposes longest common prefix ratio, and longest common subsequence formula, which counters LCSR’s preference for short words. Klementiev&Roth’06 [10] and Bergsma&Kondrak’07 [3] propose a discriminative frameworks for string similarity. Brill&Moore’00 [4] learn string-level substitutions.

## 6 Conclusion and Future Work

We have proposed and demonstrated the potential of a novel method for improving word alignments using linguistic knowledge and the Web as a corpus.

There are many things we plan to do in the future. First, we would like to replace competitive linking with maximum weight bipartite matching. We also want to improve MMED by adding more linguistically knowledge or by learning the NEDR or LCSR weights automatically as described in [20, 29, 32]. Even better results could be achieved with string-level substitutions [4] or a discriminative approach [3, 10].

## References

- [1] Y. Al-Onaizan, J. Curin, M. Jahr, K. Knight, J. Lafferty, D. Melamed, F. Och, D. Purdy, N. Smith, and D. Yarowsky. Statistical machine translation. Technical report, 1999.
- [2] N. Ayan and B. Dorr. Going beyond AER: an extensive analysis of word alignments and their impact on MT. In *Proceedings of ACL*, pages 9–16, 2006.
- [3] S. Bergsma and G. Kondrak. Alignment-based discriminative string similarity. In *Proceedings of ACL*, pages 656–663, 2007.
- [4] E. Brill and R. C. Moore. An improved error model for noisy channel spelling correction. In *Proceedings of ACL*, pages 286–293, 2000.
- [5] P. Brown, J. Cocke, S. D. Pietra, V. D. Pietra, F. Jelinek, R. Mercer, and P. Roossin. A statistical approach to language translation. In *Proceedings of the 12th conference on Computational linguistics*, pages 71–76, 1988.
- [6] M. Diab and S. Finch. A statistical word-level translation model for comparable corpora. In *Proceedings of RIAO*, 2000.
- [7] P. Fung and L. Yee. An IR approach for translating from non-parallel, comparable texts. In *Proceedings of ACL*, volume 1, pages 414–420, 1998.
- [8] W. Gale and K. Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102, 1993.
- [9] U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada. Fast decoding and optimal decoding for machine translation. In *Proceedings of ACL*, pages 228–235, 2001.
- [10] A. Klementiev and D. Roth. Named entity transliteration and discovery from multilingual comparable corpora. In *Proceedings of HLT-NAACL*, pages 82–88, June 2006.
- [11] P. Koehn. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proceedings of AMTA*, pages 115–124, 2004.
- [12] P. Koehn and K. Knight. Learning a translation lexicon from monolingual corpora. In *Proceedings of ACL workshop on Unsupervised Lexical Acquisition*, pages 9–16, 2002.
- [13] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of NAACL*, pages 48–54, 2003.
- [14] G. Kondrak. Cognates and word alignment in bitexts. In *Proceedings of the 10th Machine Translation Summit*, pages 305–312, 2005.
- [15] G. Kondrak, D. Marcu, and K. Knight. Cognates can improve statistical translation models. In *Proceedings of HLT-NAACL 2003 (companion volume)*, pages 44–48, 2003.
- [16] V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, (10):707–710, 1966.
- [17] D. Marcu and W. Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of EMNLP*, pages 133–139, 2002.
- [18] D. Melamed. Automatic evaluation and uniform filter cascades for inducing N-best translation lexicons. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 184–198, 1995.
- [19] D. Melamed. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249, 2000.
- [20] A. Mulloni and V. Pekar. Automatic detection of orthographic cues for cognate recognition. In *Proceedings of LREC*, pages 2387–2390, 2006.
- [21] F. J. Och. Minimum error rate training in statistical machine translation. In *Proceedings of ACL*, 2003.
- [22] F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1), 2003.
- [23] F. J. Och and H. Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449, 2004.
- [24] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318, 2002.
- [25] E. Paskaleva. Compilation and validation of morphological resources. In *Workshop on Balkan Language Resources and Tools (Balkan Conference on Informatics)*, pages 68–74, 2003.
- [26] E. Paskaleva and S. Mihov. Second language acquisition from aligned corpora. In *Proceedings of Language Technology and Language Teaching*, pages 43–52, 1998.
- [27] R. Rapp. Identifying word translations in non-parallel texts. In *Proceedings of ACL*, pages 320–322, 1995.
- [28] R. Rapp. Automatic identification of word translations from unrelated english and german corpora. In *Proceedings of ACL*, pages 519–526, 1999.
- [29] E. Ristad and P. Yianilos. Learning string-edit distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(5):522–532, 1998.
- [30] M. Silberstein. *Dictionnaires électroniques et analyse automatique de textes: le système INTEX*. Masson, Paris, 1993.
- [31] K. Sparck-Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.
- [32] J. Tiedemann. Automatic construction of weighted string similarity measures. In *Proceedings of EMNLP-VLC*, pages 213–219, 1999.
- [33] A. Zaliznyak. *Grammatical Dictionary of Russian*. Russky yazyk, Moscow, 1977 (A. Зализняк, *Грамматический словарь русского языка*. “Русский язык”, Москва, 1977).

# Do Happy Words Sound Happy?

## A study of the relation between form and meaning for English words expressing emotions

Vivi Nastase  
EML Research gGmbH  
Heidelberg, Germany  
*nastase@eml-research.de*

Marina Sokolova  
Université de Montreal  
Montreal, Quebec, Canada  
*sokolovm@iro.umontreal.ca*

Jelber Sayyad Shirabad  
University of Ottawa  
Ottawa, Ontario, Canada  
*jsayyad@site.uottawa.ca*

### Abstract

This paper presents a study of the relation between a word's form and the emotion it expresses. We analyze the possibility that the form of words expressing emotions is not completely arbitrary, but in fact, their sound evokes the emotion conveyed. We explore the relation between word form and emotions using a variety of word form representations and machine learning methods. We first show that words expressing an emotion are more similar among them than with words expressing other emotions, and then we discuss the sounds of emotions.

### Keywords

sentiment analysis, word form, meaning, machine learning

## 1 Introduction

A word has two components: **word form**, a sequence of sounds (pronunciation) and, possibly, letters/characters (written form), and **meaning**. The word form is also called **signifier**, and its meaning, or referent in the world, is called **signified**: the word form *tree* with the pronunciation  $/trE/$ <sup>1</sup> has as referent in the real world a TREE entity.<sup>2</sup> While it is usually accepted that the relation between signifier and signified is largely arbitrary [5], the idea that sounds may carry meaning has appeared at several points in time [8], and is still a matter of debate and research.

In this paper we study the relationship between signifier and signified for a class of words which can be particularly susceptible to the way a word sounds: words that express emotions – either positive or negative, or a more fine grained range (anger, disgust, fear, joy, sadness, surprise).

We work with data annotated with emotion tags: WordNet Affect and the dictionary from the Linguistic Inquiry and Word Count system. We work with the pronunciation and written form of a word. We represent the word form in various ways, using separately the written and pronunciation versions. We investigate the connection between form and emotion conveyed in two steps. We first verify, through machine learning experiments, whether such a connection exists. The results support this hypothesis, by showing that words expressing the same emotion have more in

common with each other than with words expressing other emotions. In a second step, we analyze whether the sounds of happy words are indeed happy sounding. This is a harder question to answer, as perception is subjective. We discuss the sounds of emotions based on the most salient features in our experiments and research on emotion recognition in speech.

Apart from a purely theoretical benefit, finding a relation between the way the words sound and the emotion expressed contributes to research in sentiment analysis, very much part of the highly explored areas of NLP these days, authorship analysis and other research areas. From a practical point of view, such relations could be exploited in advertising, where product names that have no literal meaning rely on their sound to catch the attention and desire of potential customers [1].

## 2 Motivation

It is a long held belief that the association between a word-form and its meaning is arbitrary [5]: there is nothing about a TREE that evokes the sequence of letters or sounds that form the English word *tree*. Support of this theory comes from language variation: a TREE is called *tree* in English, but *Baum* in German, *albero* in Italian, and numerous other variants in the languages of the world. If there was anything intrinsic to TREE that would link it to the form *tree*, it would have been called the same in all languages.

There are also onomatopoeic words, which sound like the concept they describe [2]. Onomatopoeia are language specific. In English lions *roar*, cats *purr*, flies *buzz*, snakes *hiss*, fireworks go *boom* and *bang*.

In between the two extremes of total arbitrariness of form relative to meaning and identity of the two, there are *mellifluous* words. Coming from the Latin *mellifluus* = *mel*(honey)+*fluere*(flow) – dripping with honey – mellifluous has come to refer to words whose sounds evoke the concepts they refer to. Such words were particularly exploited for effects in poetry [17]. We also use them in our everyday speech: we *hush* to make silence, we *mumble* when we speak in a low inarticulate manner.

Arbitrariness of the connection between sound and meaning is not universally accepted. The theory of *sound symbolism* or *phonosemantics*, according to which most words in a language fall into a category similar to *mellifluous* – every sound carries a certain meaning, which evoke certain aspects of a concept whose name contains this sound – has ancient roots. Plato, through his characters in the Cratylus dialogue

<sup>1</sup> From the online version of the Merriam-Webster: <http://www.m-w.com>.

<sup>2</sup> For the remainder of the paper, the signifier will be written in *italics*, and the signified in SMALLCAPS.

– Hermogenes and Socrates – discusses the provenance of words. Socrates proposes that there is a connection between the way words sound and their signifiers. As an example, he gives the Greek letter  $\rho$  (rho), which for him expresses motion. A number of (Greek) words containing  $\rho$  are brought up in support of this hypothesis, for which Hermogenes provides afterwards a plethora of counter-examples.

The idea that sounds carry meaning has reappeared throughout history. Locke’s *An Essay on Human Understanding* (1690) counters this idea. Leibniz’s book *New Essays on Human Understanding* (1765) critiques Locke’s essay. Leibniz proposes a moderate view, in which words and their referents are neither related by perfect correspondence, nor by complete arbitrariness. A detailed history of phonosemantics is presented by Genette [8], and a historical review plus recent research and developments are presented by Magnus [16].

An interesting view on the relation between sound and meaning, and the possible connection between the two, is proposed by Jakobson [11]. In Lecture VI he says: “The intimacy of connection between the sounds and the meaning of a word gives rise to the desire of speakers to add an internal relation to the external relation, resemblance to contiguity, to complement the signified by a rudimentary image”. In other words, the resemblance between sound and meaning is in the ear and mind of the beholder. This may lead to a “natural selection” of words, based on the way they sound, as suggested by Otto Jespersen: “There is no denying that there are words which we feel instinctively to be adequate to express the ideas they stand for. ... Sound symbolism, we may say, makes some words more fit to survive.” [12]. Firth [7] and Sapir [20] also share such a middle-ground view of sound symbolism. In their view, speech sounds carry meaning, but rather than being inherent to them, it is a result of what Firth called “phonetic habit”, “an attunement of the nervous system”.

### 3 Signifier and signified

We set out to investigate the connection between the signifier, or word form, and signified, or meaning, for English words that express emotions. Because we propose that words expressing emotions are mellifluous words, we do not seek a relation between form and exact meaning, but rather form and some aspect of the meaning - in our case, the emotion conveyed.

**The signifier** The signifier, in our case, can have both a written and a spoken form. A TREE is called /trE/ and written *tree* in English. The pronunciation is a sequence of sounds (phonemes). According to research in speech analysis, phonemes are not the smallest units of speech. Individual phonemes can be represented through values of a set of parameters, or features, that capture the configuration of the vocal tract that produces each sound and other acoustic features. We investigate each of these three variants of representing a word form.

**letters** : In English words are not pronounced as they are written. However, the way words are spelled may be closer to the words’ etymological roots than their pronunciation is. As an example, the word *delight*, comes from the Old French word

Phoneme	Example	Transcription
AA	alarm	AH0 L AA1 R M
AE	amorous	AE1 M ER0 AH0 S
CH	charm	CH AA1 R M
EH	enchant	EH0 N CH AE1 N T
T	tickle	T IH1 K AH0 L
Y	euphoria	Y UW0 F AO1 R IY0 AH0

**Table 1:** A sample of phonemes, words and their phonetic transcription

*delit, delitier* which in turn comes from the Latin *delectare*<sup>3</sup>. The letter *e* in *delight* is pronounced /i/ as in *bit*, while in its etymological roots, it is pronounced /e/ as in *bet*. Since texts are more readily available than word pronunciations, this type of word form is also the easiest to analyze.

**pronunciation** : Pronunciation of letters in English, especially vowels, depends on their context. Dictionaries provide a transcription of words into their phonetic equivalent. In this representation, each sound (which may correspond to one or more of a word’s letters) is represented by a special symbol. We use CMU’s pronunciation dictionary developed at the Carnegie Mellon University<sup>4</sup>, which contains approximately 125,000 words and their transcriptions. The transcriptions’ “alphabet” consists of 39 phonemes, and three extra digits for stress information (0 - no stress, 1 - primary stress, 2 - secondary stress). A sample of phonemes and word pronunciations are presented in Table 1.

**phonetic-features** : The phonemes can also be further described in terms of *phonological features* – “configurations” of the vocal tract and acoustic characteristics. From the existing phonological feature systems – [13], [9], [3] – we use the Sound Pattern of English (SPE) [3].

SPE consists of 14 binary features, which describe the tongue body position (high, back, low), tongue tip position (anterior, coronal), lips’ configuration (round), configurations affecting the air flow – by constriction, vibration of vocal folds or blocking with the tongue or lips (tensed, voiced, continuant, nasal, strident) and acoustic characteristics (vocalic, consonant, silence). Examples of phonemes (also called phones) with their SPE representation are shown in Table 2.

		v	c	h	b	l	a	c	r	t	v	c	n	s	s
		o	o	i	a	o	n	o	o	e	o	o	a	t	i
		c	n	g	c	w	t	r	u	n	i	n	s	r	l
		s	h	k				n	s	c	t	a	i	e	
ae	(bat)	+	-	-	+	+	-	-	-	+	+	+	-	-	-
b	(bee)	-	+	-	-	-	+	-	-	-	+	-	-	-	-
iy	(beet)	-	-	+	-	-	-	-	-	+	+	+	-	-	-
m	(mom)	-	+	-	-	-	+	-	-	-	+	-	+	-	-
ow	(boat)	+	-	-	+	-	-	-	+	+	+	+	-	-	-
sh	(she)	-	+	+	-	-	-	+	-	-	-	-	+	-	+

**Table 2:** Examples of sound representation using the SPE system

<sup>3</sup> From the Online Etymology Dictionary: <http://www.etymonline.com>.

<sup>4</sup> The CMU pronunciation dictionary is freely available at <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>. We have used version 0.6d.

**The signified** The signified component of our data comes from emotion tags, from two sets – a finer grained set consisting of 6 emotions, and a set consisting of 2 coarse emotion classes. Psychological research proposes the following basic emotions: {*anger, disgust, fear, joy, sadness, surprise*} [6]. We study whether analysis of word form allows us to predict whether the word expresses one of these basic emotions. Because much research in the domain of sentiment analysis works at a coarser level of emotions – *positive* and *negative* – we also study the relation between word forms and these broader emotion categories.

## 4 Emotion-tagged words

Assigning an emotion tag to words is not an easy task. Potentially, for any word one may perceive an emotional dimension, either directly from the word’s meaning, or through the word’s associations with emotionally charged words or situations.

The words we are most interested in are words that express an emotion, such as *happy, joy*. We focus on WordNet Affect [22] and LIWC [19] data because they contain words that express emotions, rather than having a semantic orientation. The word *knowledge* for example, does not have an emotion tag in WordNet Affect, but it has a positive tag in the General Inquirer data. Other resources include the General Inquirer data<sup>5</sup> and the list of positive and negative adjectives used by Hatzivassiloglou and McKeown [10]<sup>6</sup>.

**WordNet Affect** WordNet-Affect is an extension of WordNet with affective tags. Words that have an EMOTION tag, were recently more fine-grained reannotated with one of: {*joy, fear, anger, sadness, disgust, surprise*} [22]. The choice for the six emotions comes from psychological research into human (non-verbally expressed) emotions [6].

**Linguistic Inquiry and Word Count** Linguistic Inquiry and Word Count (LIWC) focuses on the analysis of text and computing statistics along 82 dimensions, such as “present”, “future”, “space”, “motion”, “occupation”, “physical”, “metaphysical”, “body” [19], based on a large dictionary that lists words under each of these dimensions. We use the words listed under “positive emotions” and “negative emotions”.

Table 3 contains information about the number of unique words for each of the WordNet affect and LIWC emotions. Column 3 shows the word count for each emotion, and column 4 shows the word count after filtering morphologically related words and after verifying that the word has an entry in the CMU dictionary. The experiments are run only using words that have a pronunciation in the dictionary, to allow for comparison of performance for the different representations. We filtered morphologically related words by performing (i) stemming (using Porter’s Stemmer), (ii) an extra step of cutting off suffixes (such as *-fully*,

Resource	class	count	filtered
WordNet Affect	anger	240	101 (25%)
	disgust	48	17 (4.2%)
	fear	134	49 (12.13%)
	joy	364	152 (37.63%)
	sadness	187	59 (14.6%)
	surprise	70	26 (6.44%)
	total	1043	404 (100%)
LIWC	negative	345	283 (59.21%)
	positive	265	195 (40.79%)
	total	610	478 (100%)

**Table 3:** Word-sentiment counts from WordNet Affect and LIWC

*-ful, -some, -ness*) to catch words with multiple suffixes, and finally (iii) word matching. We also eliminate words with the suffix “less”, because the emotion the word stem expresses and the emotion expressed by the full word are different. Words with negative prefixes (un-, in-) are kept, because it is harder to detect whether a starting sequence *un* or *in* is actually a prefix or not. Also, the bigram representation, discussed below, will cover these prefixes (as opposed to the suffix *-less* for which a 4-gram representation would be necessary).

## 5 Learning experiments

The hypothesis we explore is that word forms expressing the same emotion share sound/pronunciation characteristics – in other words, they sound similar in certain ways. The similarities may be at the smallest level – letter, sound, sound feature – or at a more complex level – letter or sound sequences, combinations of sound features. We build data representations at these three levels, and test the hypothesis using decision tree (J48, ADTree<sup>7</sup>) and memory based (IBK) algorithms in Weka [24], in 10-fold cross-validation experiments.

**Data representation** Following these considerations, we have produced a series of representations for the data, which vary along two dimensions: analyzed unit (unigrams and bigrams) and unit representation (letter, pronunciation and sound features).

We split each word into three segments – beginning segment (consisting of the first unit), ending segment (the last unit), and the middle segment which contains everything in-between. Each word is represented in terms of features for each of these three segments. For each segment, the features represent aggregated statistics for the units in this segment. For letter feature *a* in the middle segment, for example, the value is the number of occurrences of *a* in the middle segment.

An example: if we consider the word *admire* with a bigram letter representation, it will have the following segments: beginning – *ad*, middle – *dmir*, end – *re*. In its feature vector, the following features will have non-zero values: for the beginning segment – *ad*, for the middle segment – *dm, mi, ir*, for the end segment – *re*.

Table 4 shows the number of features for each data set generated for the 6 possible variations. In the table, and in the discussion that follows, we will use the abbreviations: *data sets*: WordNet Affect (W), LIWC

<sup>5</sup> The General Inquirer lexicon is freely available for research purposes from <http://www.wjh.harvard.edu/inquirer/>.

<sup>6</sup> These and other sentiment annotated resources are available from Janyce Wiebe’s web site <http://www.cs.pitt.edu/~wiebe>

<sup>7</sup> We use Weka’s MultiClassClassifier to perform multi-class classification with ADTree.

Repres.	# of Features	Repres.	# of Features
W-1Let	71	W-2Let	498
W-1P	126	W-2P	711
W-1C	42	W-2C	588
L-1Let	68	L-2Let	498
L-1P	123	L-2P	731
L-1C	42	W-2C	588

**Table 4:** Number of features used in each representation method

(L); *units*: unigram (1), bigram (2); *unit representation*: letters (Let), pronunciation/phonetic (P), SPE codes (C) levels.

For letter- and pronunciation/phonetic-based representation, the features are determined by the n-gram letter and phoneme sequences that actually appear in our list of words. For phonological features we consider the set of 14 SPE features for each word segment (beginning, middle, end). All features are numeric and their value is the number of occurrence of the feature (e.g. letter *a* or phoneme *IY*) in the corresponding segment of the word. The phonetic-based representation contains two extra features – for primary and secondary stress, as indicated in the pronunciation dictionary. These two features take as value the phoneme that was stressed (always corresponding to a vowel).

**Results for WordNet Affect data** We evaluate the quality of classification by computing the average accuracy (*Acc*) and average precision (*P*), recall (*R*) and F1 score (*F*) for each emotion class in 10-fold cross validation experiments. Our experiments have shown that dropping the features for the end of word segment has a positive impact on performance. The increase in performance may be due partly to the reduction (by approximately 33%) of the number of features.

The best results, in terms of accuracy, for the WordNet Affect data were 39.85% obtained with IB1-1Let<sup>8</sup> and 38.11% with IB1-1C, on word representations based only on the beginning and middle segment. In this 6-class learning problem the baseline accuracy is 37.63%, corresponding to classifying everything as *joy*, the majority class (this baseline maximizes accuracy).

Table 5 shows the best results in terms of F-score for each class (emotion) in the WordNet Affect data in the multi-class learning setting. For detailed results on each emotion class we use a baseline which guesses the class with a distribution that matches the one in the data set (this baseline balances precision and recall). The baseline F-score values are given by the distribution presented in Table 3, repeated here on row 2.

Method	anger	disgust	fear	joy	sadness	surprise
baseline	25%	4.2%	12.13%	37.63%	14.6%	6.44%
IB1-1Let	40.9%	33.3%	33.3%	49.3%	33%	9.8%
IB1-1C	42.2%	16.2%	29.5%	45.1%	31.5%	26.7%
highest values	44.8%	36.8%	34.5%	55%	33%	26.7%
	IB2-1C	IB2-1Let	IB2-1C	IB2-1C	IB1-1Let	IB1-1C

**Table 5:** F1 score results on 6-class classification into WordNet Affect emotions

The best recognized emotion from WordNet Affect’s emotion classes was *joy*. Despite variation in *P*, *R*, and *F* values for different representations and learning algorithms, *joy* was consistently the best classified emotion. Part of this may be due to the fact that it

<sup>8</sup> IBK, K=1, unigram letter-based representation, following the same notation convention as in Table 4.

had the most examples (37.63%). The results show statistically significant improvement over the baseline at 95% confidence level with Weka’s t-test.

**Results on LIWC data** A selection of the best results (in terms of F1 score) for the LIWC data are presented in Table 5. The baseline F1 score is equal to the distribution of the classes, as presented in Table 3. The performance increase over the baseline is statistically significant at 95% confidence level (with Weka’s t-test). For this binary classification experiment, the baseline accuracy is 56.59%, corresponding to classifying everything as *negative*, the majority class. The best results, in terms of accuracy for the LIWC data are 62.3% (IB55-1C) and 61.5% (J48-2Let).

Method	positive	negative
baseline	40.79%	59.21%
IB1-2Let	45.5%	68.2%
IB1-1P	44.1%	67.9%
highest values	45.5%	74.9%
	IB1-2Let	IB55-1C

**Table 6:** F1 score results on binary classification on LIWC

For the LIWC data, we obtained better prediction performance for words conveying a negative emotion. There are also more words expressing negative emotions in our data set.

IBK, which classifies a word based on its similarity with neighbouring words, outperforms other classifiers in finding the best results for both the binary and the 6-class learning problems. This supports the idea that words expressing the same emotion have more in common with each other than with words expressing other emotions.

## 6 The sounds of emotions

Happy words sound more like other happy words than like words expressing other emotions. But do they really sound happy?

In order to verify whether such features are indeed perceived as expressing the emotion we consider, we look into research on recognizing emotions in human speech. The type of data used in such work are recordings of (usually, multi-word) utterances, whose sound signal is represented through a variety of features (such as pitch, energy, tone contour) [21],[4], [18]. Lee et al. [14] introduce five broad phoneme classes – vowel, stop, glide, nasal, fricative – to help in classifying utterances into 4 classes – angry, happy, neutral and other. In learning experiments using Hidden Markov Models, they note that using phoneme classes in addition to the more traditional signal features leads to better emotion recognition. In particular, vowel sounds are good emotion indicators, and furthermore different vowels have different effects, possibly because of articulatory constraints: “less constricted low vowels such as /AA/ show greater effects than do high vowels like /IY/”. There are no details as to which vowels are predictive of which emotion class, but it is not just the presence or absence of a vowel that is useful for predicting the class, but also prosodic features related to its pronunciation [15].

Whissell [23] analyzed phonologically transcribed text samples from song lyrics, poetry, word lists and

advertisements) for correlations between phonemes and language emotionality. Phonemes were grouped into 8 classes, based on two dimensions – Pleasantness and Activation. Support for this grouping was given through experiments using phonemes as part of non-words. Here are the classes and a sample of their assigned phonemes: Pleasantness – /AY/-high, /DH/-this; Cheeriness – /AA/-father, /AY/-high, /CH/-chip, /F/, /V/; Softness – /TH/-thumb, /EH/-bet, /L/, /M/; Activation – /AA/-far, /OY/-voice; Nastiness – /ER/-her, /UW/-cool, /NG/; Unpleasantness – /AW/-cow, /OW/-bone; Sadness – /AW/-cow, /B/; Passivity – /AE/-hat, /K/, /L/ <sup>9</sup>.

Let us now look in a bit more detail at some of the most salient features in our data representation, as identified by the tree-based algorithms. Negative words in LIWC data are characterized by vocalic beginning and phonemes pronounced with the tongue body in back position (e.g. /CH/, /NG/, /G/, /AA/, /AH/), as in *angry*. Such phonemes appear in the Nastiness category [23]. Positive words by starting phonemes pronounced with the tongue body in high and the tip not coronal position (e.g. /IY/, /K/, /P/) and at most two phonemes pronounced with the tongue in back position in the middle segment (e.g. *improve*, *kind*). /IY/ appears in the Softness, Pleasantness and Cheerful category category, but /K/ and /P/ appear in the Unpleasantness and Passive ones. Happy words in the WordNet Affect data start with phonemes which are not continuant and the tongue tip is not in anterior position (e.g. /CH/, /NG/, /K/) and the body contains tensed phonemes (e.g. /AA/, /AW/, /EY/) (e.g. *charming*). Words expressing sadness start with non-consonantal phonemes pronounced with the tongue body in back position (e.g. /AW/, /OW/, /UH/).

We observe parallels between the features found most discriminating by the decision tree algorithms, and the phonemes previously established in the literature as having emotional connotations. We also note that it is the effect of several phonemes that gives a word its “emotion” sound. In future work we will determine a representation that captures best the interactions and relations between sounds in a word.

## 7 Conclusion

We have investigated the properties of word-forms, to learn whether we can automatically predict the emotion a word expresses based on various representations of its form. The results show that all the representations used – word spelling, pronunciation, phonetic features – are useful for determining that words expressing the same emotion are alike in certain ways.

These results answer half of the question we had set out to investigate – whether words sound like the emotion conveyed. The other half is whether what happy words have in common is what makes them sound happy. The answer to this question is harder to find, because of subjectivity of perception and bias from the meaning component. We have found interesting parallels with features used in classifying emotion words and emotional sound characteristics found in related work. Future work on larger text segments annotated with emotion and future developments in emotion recognition in speech analysis will help provide a more rigorous answer to this part of the question.

We plan to experiment with alternative word representations – such as syllables, which are considered the phonological “atoms” of words – and to determine which part of the word is most expressive from the point of view of the emotion conveyed. Research based on the words’ etymological roots may show us if the link between form and meaning gets stronger as we go back in time. Next step is to expand the study to languages other than English, and to longer text units, such as blogs. In speech emotions are detectable, and the speaker conveys these through tone and other prosodic features. It would be interesting to see whether we can identify “sub-word” level features useful for detecting emotions in blogs.

**Acknowledgments** Partial support for this work is given by RALI (Guy Lapalme et al.), Université de Montreal. We thank the three anonymous reviewers for their very good, insightful comments.

## References

- [1] S. Bedgley. Strawberry is no blackberry: Building brands using sound, 2002. <http://online.wsj.com/article/0,,SB1030310730179474675.djm,00.html>.
- [2] H. Bredin. Onomatopoeia as a figure and a linguistic principle. *New Literary History*, 27(3):555–569, 1996.
- [3] N. Chomsky and M. Halle. *The Sound Pattern of English*. MIT Press, 1968.
- [4] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kolias, W. Feltenz, and J. Taylor. Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*, 18(1):32–80, 2001.
- [5] F. de Saussure. *Cours de linguistique générale*. Harrassowitz, Wiesbaden, 1916.
- [6] P. Ekman. An argument for basic emotions. *Cognition and Emotion*, 6:169–200, 1992.
- [7] J. R. Firth. Modes and meaning. In *Papers in linguistics 1934-1951*. Oxford University Press, London, 1951.
- [8] G. Genette. *Mimologiques: voyage en Cratylie*. Seuil, Paris, 1976.
- [9] J. Harris. *English Sound Structure*. Blackwell, 1994.
- [10] V. Hatzivassiloglou and K. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th ACL/8th EACL*, pages 174–181, Madrid, Spain, 1997.
- [11] R. Jakobson. *Lectures on Sound and Meaning*. MIT Press, Cambridge, MA, 1937.
- [12] O. Jespersen. *Language - its Nature, Development and Origin*. George Allen & Unwin Ltd., London, 1922.
- [13] J. King and P. Taylor. Detection of phonological features in continuous speech using neural networks. *Computer speech and language*, 14(4):333–353, 2000.
- [14] C. M. Lee, S. Yildirim, M. Bulut, A. Kazemzadeh, C. Busso, Z. Deng, S. Lee, and S. Narayanan. Emotion recognition based on phoneme classes. In *Proc. of ICSLP*, 2004.
- [15] L. Leinonen, T. Hiltunen, I. Linnankoski, and M.-L. Laakso. Expression of emotional-motivational connotations with a one-word utterance. *Acoustical Society of America Journal*, 102:1853–1863, 1997.
- [16] M. Magnus. *What’s in a Word? Studies in Phonosemantics*. PhD thesis, University of Trondheim, Trondheim, Norway, 2001.
- [17] M. Nanny and O. Fischer. Iconicity: Literary texts. In K. Brown, editor, *Encyclopedia of Language and Linguistics*, volume 5, pages 462–472. Elsevier, Oxford, 2nd edition, 2006.
- [18] A. Nogueiras, A. Moreno, A. Bonafonte, and J. B. M. no. Speech emotion recognition using Hidden Markov Models. In *Proc. of Eurospeech*, 2001.
- [19] J. W. Pennebaker, R. J. Booth, and M. E. Francis. Linguistic Inquiry and Word Count: LIWC. [Computer software]. Mahwah, NJ: Erlbaum., 2001.
- [20] E. Sapir. A study in phonetic symbolism. *Journal of Experimental Psychology*, (12):225–239, 1929.
- [21] K. R. Scherer. Vocal communication of emotion: A review of research paradigms. *Speech Communication*, 40(1):227–256, 2003.
- [22] C. Strapparava, A. Valitutti, and O. Stock. The affective weight of the lexicon. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy, 2006.
- [23] C. Whissell. Phonoemotional profiling: a description of the emotional flavour of english texts on the basis of the phonemes employed in them. *Perceptual and Motor Skills*, 91(2):617–648, 2000.
- [24] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition edition, 2005.

<sup>9</sup> Table 1 at <http://www.trismegistos.com/IconicityInLanguage/Articles/WhisselPlath/index.html>

# Semi-automatic construction of training data for tagging non-contemporary literary texts

Costanza Navarretta  
University of Copenhagen  
Njalsgade 80  
2300 Copenhagen S  
*costanza@cst.dk*

## Abstract

In this paper we present a methodology to semi-automatically build up a PoS and lemma annotated training corpus of older Danish literary texts using contemporary annotated data. Annotated corpora of older literary texts from different periods of time are necessary to train taggers which have to deal with non-contemporary texts because these texts differ from contemporary ones in terms of spelling, vocabulary, punctuation rules and sentence structure. Because manual annotation is expensive and time-consuming, it is important to explore how far existing linguistic resources of contemporary language can be reused for this task. In the paper we describe how we have applied our methodology to build up a PoS and lemma annotated training corpus of literary texts from the 19th century. In our experiments we have used the TreeTagger[13] which has been trained on an annotated corpus and a large NLP lexicon of contemporary Danish, both adapted to the task. The texts from the 19th century we have focussed on in the training phase are fairy tales. The performance of the TreeTagger trained on the obtained material has been evaluated on texts from that period belonging to three different text types and written by different authors.

The evaluation shows that the performance of the TreeTagger varies significantly from text type to text type. The best performance is nearly as good as that obtained on contemporary texts by the same tagger trained on contemporary data. The described methodology has also been used to build up training data for texts from the beginning of the 20th century and also in this case the tagger's performance is only slightly lower than that obtained on contemporary texts.

## Keywords

annotated data, construction of non-contemporary linguistic resources, PoS tagging, literary texts

## 1 Introduction

Automatically annotating old texts with morpho-syntactic and lemma information is an important task to support language and literary studies of non-contemporary texts. Most existing annotated linguistic resources belong to contemporary language, thus

the first step for automatically annotating old literary texts is to build up appropriate resources accounting for the linguistic characteristics of these texts.

In this paper we describe how we have semi-automatically built up a PoS and lemma annotated training corpus of Danish literary texts from the 19th century using an annotated Danish corpus and a large NLP lexicon of contemporary general language. The work described was done under the MULINCO project (MultiLINGual Corpus of the University of COpenhagen), funded by the Danish Research Council of the Humanities. The project run from 2005 to the beginning of 2007 and was a cross-disciplinary cooperation between the Institute of English, German and Roman Languages and the Centre for Language Technology (CST), both from University of Copenhagen [12].

The main goals of the MULINCO project were the following:

1. to create a corpus platform for parallel aligned literary and non-literary corpora and for comparable corpora;
2. to collect parallel and comparable literary and non literary corpora from different periods of time;
3. to exploit how far language technology methods and techniques can support studies and teaching in translation and literature.

The languages represented in the project were Danish, English, French, German, Italian and Spanish. Danish played a central role as source or target language.

In the first phase of the project user requirements regarding the corpora and the platform were identified [6]. In the second phase of the project, in parallel with the corpus collection and the design and construction of a corpus platform<sup>1</sup>, available tools to automatically and/or manually annotate corpora for morphological, syntactic, semantic and structural information on the word, phrase, sentence and discourse levels were investigated together with alignment tools.

Reuse of existing resources was given high priority because of resource limitations, but the project faced the problem of applying tools trained on contemporary general language corpora and lexica to literary texts from different periods of time and with different

<sup>1</sup> The MULINCO platform uses the CQP search facilities implemented at the IMS, University of Stuttgart [5].



language usage in terms of punctuation rules, spelling, vocabulary and sentence structure.

In this paper we focus on the task of tagging non-contemporary Danish fiction texts with Part-of-Speech (PoS) and lemma information reusing a manually validated PoS and lemma annotated corpus, the so-called Danish PAROLE corpus [11] and a large NLP lexicon of contemporary Danish, STO [3].

The paper is organised as follows. In section 2 we describe the texts to be tagged in the project and discuss the differences between the language used in these texts and contemporary Danish. In section 3 we present the contemporary language resources which we have used. In section 4 we outline the methodology applied for constructing training corpora of PoS and lemma annotated fiction texts from the 19th century. In section 5 we describe the results obtained by evaluating the performance of the TreeTagger trained on the constructed material and run on 19th century texts belonging to different text types and written by different authors. We also describe the results obtained with a similarly constructed training corpus of fiction texts from the beginning of 20th century. In section 6 we discuss another strategy to construct corpora of non-contemporary data and in section 7 we make some concluding remarks and present work that still needs to be done.

## 2 The data to be tagged

A part of the MULINCO corpus consists of the following data:

- Danish fiction texts written in the 19th and in the beginning of the 20th century and their translations in one or more of the project languages
- translations into Danish (and into other languages) of fiction texts from the same periods, originally written in one of the other project languages.

A significant part of the MULINCO fiction sub-corpus had to consist of short stories because they are “complete” works useful for literature and translations studies. In particular the project focussed on Hans Christian Andersen’s Fairy Tales which were published for the first time between 1835 and 1875. Andersen’s fairy tales are extensively studied in Denmark and in other countries, have been translated in all the languages involved in the project, are available in digitalised form and are not covered by copyright restrictions because their author died more than 70 years ago<sup>2</sup>.

Automatically annotating Andersen’s fairy tales with PoS and lemma information is difficult because there only exist linguistic resources for contemporary Danish. These resources cannot be used directly because the language written in the 19th century not only differs from contemporary Danish in spelling, punctuation conventions, but also in vocabulary. Furthermore fiction texts as Andersen’s fairy tales also

differ in sentence structure from other types of written text, such as scientific papers and technical reports.

Spelling and punctuation changes occurred in the past two hundred years can be followed through the Danish spelling dictionaries and various regulations, circulars and laws about language use [8, 10]. Since 1955 spelling dictionaries have been built and published regularly by the Danish Language Council, an institution under the Danish Ministry of Culture<sup>3</sup>.

Taking into considerations the various spelling regulations and laws the following three main spelling periods can be recognised the past two hundred years:

- up to 1892
- 1892-1948
- after 1948.

These spelling periods are only indicative because in most cases spelling reforms legislate about spelling changes which have taken place for a while and/or decide on spelling tendencies and variations on the basis of political considerations. For example, while the Danish government approved the German tradition of spelling common nouns with beginning capital letters until the first half of the 20th century, they decided to abandon this tradition after the Second World War and, since then, all common nouns have been spelled with small beginning letters. Furthermore not all people follow officially imposed spelling and/or punctuation changes immediately after a spelling reform and most spelling regulations and dictionaries allow alternative spelling forms in order to cover the existing variations in language use.

In the following we give examples of the major spelling changes occurred from H.C. Andersen’s time to our days:

- Common nouns were written with initial capital letters before 1948, while in contemporary Danish they are spelled with small letters.
- The letter *aa/Aa/AA* is spelled as *å/Å/Å* after 1948. An exception to this rule are person and place names which may keep the original spelling as in *Maegaard* and *Aarhus*.
- In old Danish it was distinguished between the past tense and the infinitive of modal verbs: *kunde* ‘could’, *vilde* ‘would’, *skulde* ‘should’ and *kunne* ‘to be able to’, *ville* ‘to be willing to’, *skulle* ‘to have to’. After 1948 a unique spelling form, the infinitive one, has covered both cases: *kunne* ‘to be able to/could’, *ville* ‘to be willing to/would’, *skulle* ‘to have to/should’.
- Before 1892 there were two different spellings for singular and plural forms of verbs in present and past tense. Between 1892 and 1997 the singular form could be used instead of the plural one. For example, the plural form of the indicative present tense *løbe* ‘run’ could also be spelled as the singular correspondent *løber* and the plural form of

<sup>2</sup> However copyright restrictions hold for a large number of the translations. Copyright issues for these translations had to be dealt with in the project.

<sup>3</sup> Since 1996 the official spelling dictionary *Retskrivningsordbog*, which is published every two year, has got the same status as regulations.

the past indicative of the verb *være* (to be), *vare* 'were' could also be spelled *var* 'was/were'. In 1997 the plural forms were officially removed from the Danish spelling, but in practice they had not been used for a long time and are thus not represented in large corpora of contemporary Danish, such as the so-called *Korpus 90*<sup>4</sup>.

- Double vowels were replaced by one vowel after 1872, and since then words such *Huus* (house) and *see* (see) have been spelled as *Hus* and *se* respectively.

A general problem for the automatic treatment of non-contemporary texts is that before the 20th century there was no ideal of a spelling norm in Denmark. This can be seen in Hans Christian Andersen's fairy tales where the same word is spelled in different ways, even in the same text. For example the Danish word for "leg/bone" is both spelled *been* and *ben* in the fairy tale *Skyggen* (The Shadow), independently from the word's two meanings. Only the latter form has survived in contemporary Danish. Spelling variations must be accounted for in the training data.

In the 18th and 19th century the signs >> and << were used to start and end reported speech respectively. These signs do not exist anymore having been replaced by the quotation marks " and ".

The vocabulary used in the 19th century also differs from the contemporary one. Training material for PoS taggers and lemmatisers must be supplied with words which are not used anymore. Finally there are few words whose function has changed since the 19th century and their different uses must be accounted for in the training data.

Another problematic aspect for the automatic treatment of both contemporary and non-contemporary Danish is punctuation because more comma setting systems have co-existed in different periods of time. In an attempt to unify and simplify two at that time co-occurring comma setting systems, the Danish Spelling Council introduced a new comma regulation in 1996. This regulation was not popular and many people, including journalists and school teachers, did not follow it. Thus since 2001 two co-existing comma setting systems are again officially recognised. The two systems are the so-called grammar-based system and the new comma setting system which the Danish Spelling Council recommend.

Some problematic aspects regarding sentence structure are specific to fiction texts and especially to Andersen's Fairy Tales. Fiction is often written in a more free style than other text types such as journal articles and essays. Fiction texts can also contain reported speech which complicates the syntactic structure of sentences. This is certainly the case for Andersen's Fairy Tales which were written to be read aloud to children and thus in many respects resemble spoken language. Furthermore Andersen uses a childish language and creates new words, especially onomatopoeic ones.

<sup>4</sup> The corpus has been collected by *Det Danske Sprog- og Litteraturselskab* and consists of texts from 1988-1992. It is available on the Web together with the so-called *Korpus 2000* at <http://korpus.dsl.dk>.

### 3 The annotated data

The manually PoS and lemma annotated Danish corpus [11] which we used in our work is a subset of a larger general language balanced corpus of written Danish from the 1980ies and the beginning of the 1990ies, the so-called PAROLE corpus. The corpus is composed of extracts from texts belonging to different text types and genres including newspaper and journal articles, novels and short stories, essays, scientific papers and technical reports.

The PoS and lemma annotated sub-corpus consists of 250,000 running words [11]. The annotation of the sub-corpus was done under the European project PAROLE and the original tag set consisted of 151 tags. The tag set was reduced to 50 tags [7] under the Danish project ONTOQUERY [2] and the PAROLE corpus with the reduced tag set was used as training and test material for the Brill tagger [4].

In the MULINCO project we decided to use the TreeTagger [13] instead of the Brill tagger and train it on Danish data for the following reasons:

- Pre-trained versions of the TreeTagger were used for more languages in the project.
- The TreeTagger can tag texts with both PoS and lemma information and can mark new words, i.e. words which are not in the training lexicon, as "unknown".
- The TreeTagger recognises SGML tags in the texts. This feature was essential because texts had to be annotated in XML with different types of structural information before being PoS and lemma tagged.

Differing from the Brill tagger which only requires an annotated training corpus, the TreeTagger must be trained on a PoS annotated corpus and a lexicon containing PoS and lemma information for each word form. As training corpus we used two-third of the tagged PAROLE corpus, while the remaining part of the corpus was used as testing material. We constructed a training lexicon for the tagger extracting the PoS and lemma information encoded in the large Danish NLP lexicon STO [3] supplied with few words in the corpus, not covered by the lexicon. STO partially builds on the Danish PAROLE lexicon and contains approximately 550,000 word forms. The spelling rules accounted for in STO are those proposed by the Danish Spelling Council in 2001, thus the lexicon does not account for the changes in vocabulary and spelling occurred the last 200 years.

The precision of the TreeTagger on the subset of the PAROLE corpus was of 95.1%, which is a little lower than the precision obtained on the same material by the Brill tagger trained on the same data<sup>5</sup>.

<sup>5</sup> Dorte Haltrup, Sussi Olsen and Costanza Navarretta, all from CST, tested the Brill tagger extended with the STO lexicon on other contemporary texts and obtained results in-between 95.5 and 97.5% depending on the text type. The Brill tagger performs better than the TreeTagger also on these data because it classifies unknown words more correctly than the TreeTagger.

## 4 Building the Training Corpus

As indicated in section 2 we can recognise three main spelling periods in Danish the past two hundred years. To automatically tag texts from the three periods in a reliable way one should at least train the tagger on linguistic resources reflecting the language written in each of these periods. Unfortunately only resources for contemporary language are available. To reduce the cost of manual annotation and to take advantage of the existing linguistic resources, we decided to use the TreeTagger trained on the contemporary PAROLE corpus and STO lexicon to semi-automatically build up training data for texts from the 19th century and the first half of the 20th century respectively.

Probably the most efficient method to build up training material for texts written in these two periods is to start with texts written more recently and then moving backwards in time, because the language written 50 years ago is more similar to contemporary language than the language written 100 years ago.

Unfortunately we could not start from the most recent period for practical reasons. In fact from the beginning of the MULINCO project we had an amount of digitalised old fiction texts for which copyright did not apply, while only a few literary texts from the beginning of the 20th century were available and ready to be tagged. Furthermore most of the researchers in the project were interested in working with fairy tales from the 19th century and wanted to focus especially on H.C. Andersen's Fairy Tales and their translations in the project languages. Thus we started building a training corpus for these fairy tales, while texts from the more recent period were collected by the project<sup>6</sup>.

Our methodology for semi-automatically constructing a training corpus of non-contemporary fictive Danish texts consisted of the following steps:

1. automatically change the contemporary training data to simulate the spelling of old language and train the TreeTagger on these data;
2. tag a number of texts (in our case Andersen's Fairy Tales) with the TreeTagger trained on the modified contemporary data;
3. manually correct the automatically tagged texts;
4. use the corrected data to test the performance of the tagger;
5. add the corrected annotated texts and the still unknown words to the training data and train the tagger on the enlarged material;
6. repeat steps 2-5 until satisfactory results are obtained. In our case the ideal threshold was the precision obtained on contemporary texts by the TreeTagger trained on contemporary data.

We started tagging two fairy tales<sup>7</sup> (approx. 15,000 running words) with the TreeTagger trained on a modification of our original contemporary training material. The existing contemporary lexicon and corpus

<sup>6</sup> The sub-corpus of fiction texts from the 20th century is still limited in size due to difficulties in resolving copyright issues.

<sup>7</sup> The length of Andersen's fairy tales varies from a few hundred words to more thousands words.

were modified to account for the most general spelling differences between texts written before 1892 and contemporary texts. By this preprocess we wanted to reduce the number of errors to be corrected manually in the first cycle of our methodology. All the changes we made to the training material were automatically implemented. Some of the changes involved both training lexicon and corpus, some only involved the lexicon. The most comprehensive changes were the following:

- changing all occurrences of *å* and *Å* in the training lexicon and in the training corpus to *aa* and *Aa* or *AA* respectively;
- changing modal verbs in past tense in both lexicon and corpus;
- adding plural form variations of present and past tenses to the lexicon and, to the extent it could be done automatically, substituting generic past/present forms with plural forms in part of the corpus;
- adding old-style quotation signs to the lexicon and part of the corpus.

The texts tagged with the TreeTagger trained on the modified contemporary data were manually corrected by four project participants<sup>8</sup>.

We compared the version of the annotated texts where the annotation had been manually corrected with the version of the automatically tagged texts to calculate the precision of the TreeTagger trained on the modified contemporary material.

The precision of the tagger at this point was of 85.6%. Not surprisingly many errors were due to the incorrect classification of unknown words and to the incorrect treatment of sentences containing reported speech. A group of errors were also due to the fact that some of the corrections we made in the training lexicon introduced new ambiguities which were not accounted for in the training corpus. An example of this is the introduction of plural forms for verbs in present tense, which in many cases results in new ambiguities respect to both infinitive and imperative verb forms. Few errors were also due to the incorrect classification of a number of common nouns which were tagged as proper names being spelled with initial capital letters.

We also used the corrected annotated texts to calculate the precision of the TreeTagger trained on the original (non modified) contemporary training data run on the fairy tales used in the preceding test. The precision obtained by the "contemporary" version of the TreeTagger was of 80.1%. Thus the correction of general spelling variations resulted in an improvement of the tagger performance of 5.5%.

At this point we added the manually corrected texts to our training corpus and inserted into the lexicon the "unknown" word forms contained in these texts. We also manually added to the corpus the spelling variations which we found for very frequent words such as function words and frequently occurring verbs such as *få* (get), *være* (be) and *have* (have). The TreeTagger was trained again on these modified training data.

<sup>8</sup> In the tests that followed the first one, only one or two project participants corrected the tagged material.

Then we run the tagger on more fairy tales (approx. 10,000 running words) and repeated the correction and evaluation processes, as described above.

The precision of the TreeTagger after the second training cycle was of 90.1%.

This time some of the ambiguous words were tagged correctly, but the tagger had still classified many unknown words incorrectly. The most frequent classification errors were still singular nominal forms being classified as plural ones and vice versa, plural verbal forms in indicative present tense being classified as infinitive or imperative forms and common nouns being recognised as proper names. Some of these errors can easily be eliminated by using more general tags for the ambiguous categories. However, because many of the ambiguous cases were classified correctly, and because the project participants wanted to work with as specific PoS tags as possible, we decided not to change the tag set.

After having trained our tagger on the data enlarged with the corrected texts and with the encodings of words tagged as “unknown”, we tested the performance of the TreeTagger on more fairy tales (approx. 10,000 running words). The precision of the tagger in this test was of 91.8%. These results indicate that although the tagger performance continues to improve in new training cycles, the achieved improvements are not as impressive as those obtained in the first training phases. This is not surprising. Firstly the most frequent words in the fairy tales have been accounted for in the data in the first training/testing cycles. Secondly the most frequent errors mainly occur in relation to the still high number of unknown words and to the ambiguous forms which are not correctly classified due to the relatively little portion of the training corpus which consists of original old fiction texts. Finally some of the ambiguous cases that cause incorrect classification also occur in contemporary Danish and cannot be resolved without simplifying the tag set, and this, as previously mentioned, was not wanted by the project participants.

Given time restrictions we decided to run the training/testing cycle only once more and then to evaluate the tagger’s performance on different types of text. Therefore, for the last time, we added the corrected annotated fairy tales and more words to our training material and trained the TreeTagger on the enlarged data.

## 5 Evaluation

The tagger’s performance was tested on the following texts:

- two fairy tales written by Andersen in 1839 and 1847 respectively
- a randomly chosen extract from a novel by Andersen, *Improvisatoren* (The Improvisator) written in 1835
- a randomly chosen extract from the philosophic work *Begrebet Angest* (The concept of anxiety) by Kierkegaard published in 1844.

The texts belonging to each text type consisted of approx. 5,000 running words. The chosen texts were tagged with the TreeTagger and then they were manually corrected. The automatically tagged texts and the corrected versions of the same texts were compared to calculate the precision of the tagger. The results of the evaluation are in table 1.

text type	FAIRY TALE	NOVEL	PHIL. WORK
author	Andersen	Andersen	Kierkegaard
precision	92.8	93.7	96.2

**Table 1:** Precision of the TreeTagger on texts from the 19th century

As it can be seen in the table, the results are better than those obtained in the preceding test on all types of text, but the performance of the tagger varies significantly from text type to text type. Surprisingly the worst results were obtained on the two fairy tales, although the training material partially consisted of fairy tales written by the same author. The tagger performed better on the novel than on the fairy tales, but the best performance was obtained on the text written by Kierkegaard. The explanation of the different results can be found in both the training data and in the material used in the evaluation. As previously explained the style of Andersen’s Fairy Tales is quite informal and reported speech is quite frequent in them. Furthermore Andersen did not really care about spelling rules especially when writing non-scholar works such as the fairy tales. Another reason can be that Andersen wrote his fairy tales over a long period of time, thus his style and spelling way change during the years following the evolution of language.

The style of Andersen’s novel is much more formal and more conform to traditional writing norms than that of the fairy tales. Furthermore the extract from the novel used in the evaluation only contained one occurrence of reported speech, and numerous errors in the annotation of the fairy tales occurred in sentences containing reported speech.

Examining the language of Kierkegaard’s text on which the TreeTagger obtained the best results, we noticed that Kierkegaard’s vocabulary is quite near to that used in contemporary texts. Kierkegaard’s text does not belong to fiction, thus its language is more formal than the one used by Andersen in his fairy tales and novels. Kierkegaard was not inconsistent in his spelling, perhaps because he had an academic education. Finally most of our training data is more similar in style to Kierkegaard’s text than to the fairy tales.

The majority of the errors which occurred in our final test regard wrong classification of unknown words and of ambiguous words as it was the case in the preceding tests.

The results obtained by the tagger on Kierkegaard’s material are better than those obtained on some types of contemporary texts in tests conducted by researchers at CST. The results also indicate that the performance of taggers varies significantly from text type to text type and that taggers, not surprisingly, perform best on texts that are similar to those in the training corpus in terms not only of vocabulary, but also of sentence structure.

We have also applied the methodology described in the paper to build a training corpus of texts from the first half of the 20th century. In this case we only run three cycles of the correction/test steps to build up appropriate training material. Only the author of this paper checked the automatic annotation in this phase of the project. We evaluated the tagger on two types of text by different authors: an extract from a novel and an essay. Each text consisted of approx. 4,000 words. We obtained a precision of 93.9 and 95.1% on these two texts. The performance of the tagger in this test is similar to that obtained by the same tagger trained on texts from the 19th century when run on the novel by Andersen and the philosophic work by Kierkegaard.

## 6 An alternative strategy

An alternative strategy to that described in this paper is to use parallel corpora of different versions of the same literary work accounting for the spelling changes occurred in the period inbetween the production of the different text versions<sup>9</sup>. This strategy is interesting in cases where different versions of the same literary work are available and was taken into consideration in the beginning of the project. However we rejected it after some preliminary investigations for different reasons.

The modernised versions of Andersen's fairy tales up to the second half of the 20th century are mainly translations into Danish from German rewritings of the Danish original texts. These rewritings are quite different from the original data in that parts of text have been removed from them, while new paragraphs have been added in different places. In the worst cases part of the fairy tales, and especially their endings, have been changed because the translators found the tales too harsh or difficult to understand, and thus unsuitable for children. These texts cannot be satisfactorily aligned with the original versions of the fairy tales.

Recently some of Andersen's most famous fairy tales have been rewritten using contemporary spelling rules, but only few of them are freely available in digitalised form. A part from this practical limitation which made us to abandon the strategy, the fact that contemporary rewritings from different authors have different characteristics is also problematic. In fact some of the rewritings are only modernised in the spelling form, some are also modernised in vocabulary<sup>10</sup> and thus the texts belonging to the two groups must be handled differently, especially for what regards lemmatization.

Although we can assume that alignment on the word level of original and contemporary versions of the same text is quite reliable, in practice there would be problems related to the use of different comma settings and to differences in the way in which compound words are spelled. Andersen often splits compounds in more words, while they must be spelled in one word in contemporary Danish. E.g. Andersen spells marble balcony as *Marmor Altan* in *Den lille Havfrue* (The little

Mermaid) while the compound is spelled *marmoraltan* in the contemporary versions of the text. While split compounds are problematic for word alignment they caused seldom errors when the original fairy tales were tagged because many people incorrectly split compounds nowadays influenced by English and these incorrect spellings are accounted for in the Danish contemporary corpus.

In general it is difficult to say whether the strategy of using modernised versions of old texts would be less or more time-consuming than the strategy we have used, because it depends on the characteristics of the texts at hand. However the former strategy can be extremely useful for languages (or for periods of time) where spelling changes are not as well registered as it is the case for Danish the last two hundred years and, in some cases, the two strategies can supply each other.

## 7 Concluding Remarks and Future Work

In this paper we have described a methodology to semi-automatically build a PoS and lemma annotated training corpus for non-contemporary literary texts from the 19th century by reusing contemporary annotated data. The methodology consisted of the following steps. First the contemporary training data were automatically modified so that they followed general spelling and punctuation characteristics for texts from the relevant period. Then few fairy tales from the 19th century were tagged with the TreeTagger trained on these modified data. In the following steps we cyclically i) corrected the automatically tagged texts, ii) evaluated the results achieved by comparing the automatically tagged texts with the versions of these texts which had been manually corrected, iii) attached the corrected data to our training corpus and added new words to the training lexicon. Finally we tagged new texts with the TreeTagger trained on the enlarged training data. After four cycles we evaluated the performance of the TreeTagger on different types of text from the 19th century.

The evaluation of the results obtained so far are very promising, although the performance of the TreeTagger varies significantly from text type to text type. The worst results were achieved on fairy tales, although fairy tales by the same author were used as part of the training material. The main reason for this is that the style of the fairy tales is very informal and resembles spoken language while the training data mainly consisted of general language contemporary texts whose spelling had been modified, but whose sentence structure is still typical for written non-fictional texts. The best performance of the TreeTagger was obtained on an extract from one of Kierkegaard's philosophical works. This performance is near to that achieved by the same tagger trained and tested on contemporary texts.

We also tested our methodology to build training data for texts from the first half of the 20th century. The results achieved by the tagger trained on this constructed training data and run on texts from the first

<sup>9</sup> This strategy has also been suggested by one of the anonymous reviewers of the paper.

<sup>10</sup> This is the case for seventeen fairy tales rewritten by the contemporary Danish writer and philosopher, Villy Sørensen [1].

half of the 20th century are similar to those achieved on texts from the 19th century.

Although manually correcting automatically tagged data is time-consuming, the method we propose is less resource expensive than manually tagging old texts from scratch. Furthermore our strategy reuses existing resources, in our case the manually verified PoS and lemma annotated PAROLE corpus and the large NLP lexicon STO.

In the paper we also discuss using parallel versions of the same literary text reflecting different spelling traditions to automatically PoS tag the original versions of the texts. Although this strategy could not be used in our work because of the quantity and quality of the available modernised versions of the fairy tales, it can be very useful for languages where spelling changes are not as well described as it is the case for Danish.

In our evaluation we have only focussed on the performance of the TreeTagger in tagging PoS information. Although we corrected wrongly assigned lemmas manually, we did not calculate the precision of the lemmatization process because the TreeTagger assigns correct lemma tags to known word forms, while it only correctly assigns lemmas to unknown words if the lemma is the same as the word form that is tagged. Future work should involve improving lemmatization by using a lemmatiser such as the CST lemmatiser [9] which is language independent, but can be trained on relevant lexica and PoS tags<sup>11</sup>. It would also be interesting to test our methodology to tune taggers to run on particular types of text belonging to specific domains and to make more experiments on the use of tags of aligned parallel monolingual or multilingual texts to improve PoS annotation of non-contemporary texts.

## Acknowledgements

We want to thank all the MULINCO participants for useful discussions. A special thank goes to those who corrected the automatically annotated texts used in our work. Thanks also to Dorte Hansen and Sussi Olsen (CST) for their work on evaluating the Brill tagger's performance on contemporary Danish data and for building a golden PoS tagged corpus to be used for testing the performance of taggers on Danish contemporary data.

## References

- [1] H. C. Andersen. *H.C. Andersen / udvalgt og nyskrevet af Villy S/orensen ; illustreret af Helle Vibeke Jensen ; forord af Finn Hauberg Mortensen*. Aschehoug, København, 2002.
- [2] T. Andreasen, J. F. Nilsson, P. Paggio, B. Pedersen, and H. Thomsen. Content-based text querying with ontological descriptors. *Database and Knowledge Engineering Journal*, 48:199–219, 2004.
- [3] A. Braasch and S. Olsen. STO: A Danish Lexicon Resource - Ready for Applications. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, volume 4, pages 1079–1082, Lisboa, May 2004.
- [4] E. Brill. Transformation-Based Error-Driven Learning and Natural Language Processing. A Case Study in Part of Speech Tagging. *Computational Linguistics*, 21(4):543–565, 1995.

<sup>11</sup> The CST lemmatiser can be tested on <http://cst.dk/online/uk>.

- [5] O. Christ. A modular and flexible architecture for an integrated corpus query system. In *COMPLEX'94*, Budapest, 1994.
- [6] K. Farøe, L. Henriksen, H. Jansen, S. Jansen, X. Lepetit, B. Mægaard, C. Navarretta, L. Offersgaard, and C. Povlsen. *Behovsanalyse*. Mulinco rapport 1, University of Copenhagen, Copenhagen, 2005.
- [7] D. H. Hansen. *Træning og brug af Brill-taggeren på danske tekster*. Ontoquery technical report, Center for Sprogteknologi, Copenhagen, 2000.
- [8] H. G. Jacobsen. 1948-reformen - og før og efter. In E. Hansen and J. Lund, editors, *Det er korrekt - Dansk retskrivning 1948-98*, volume 27 of *Dansk Sprogævn Skrifter*, pages 9–45. Hans Reitzels Forlag, København, 1998.
- [9] B. Jongejan and D. Haltrup. *The CST Lemmatiser*. Technical report, Centre for Language Technology, 2001.
- [10] A. Karker. *Dansk i tusind år - Et omrids af sprogets historie*. Modersmål-Selskabets Årbog 1993. Ny revideret udgave 2001. C.A. Reitzels Forlag A/S, 2001.
- [11] B. Keson. *Vejledning til det danske morfosyntaktisk taggedde PAROLE-korpus*. PAROLE-manual, Dansk Sprog- og Litteraturselskab, København, 1997.
- [12] B. Mægaard, L. Offersgaard, L. Henriksen, H. Jansen, X. Lepetit, C. Navarretta, and C. Povlsen. The MULINCO corpus and corpus platform. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 2148–2153, Genova, 2006.
- [13] H. Schmid. Probabilistic Part-of-Speech Tagging using Decision Trees. In *Proceedings of The International Conference on New Methods in Language Processing*, Manchester, 1994.

# Identifying Cores of Semantic Classes in Unstructured Text with a Semi-supervised Learning Approach

Yun Niu and Graeme Hirst  
Department of Computer Science  
University of Toronto  
Toronto, ON, Canada M5S 3G4  
*yun,gh@cs.toronto.edu*

## Abstract

Cores of semantic classes in scenario descriptions can be extremely valuable in question-answering, information extraction, and document retrieval. We propose a semi-supervised learning approach to automatically identify and classify cores of semantic classes in unstructured text. We perform a case study on medical text. The results show that the selected features characterize the cluster structure of the data, and unlabeled data is effectively explored in the classification. Compared to a state-of-the-art supervised approach, the performance of the semi-supervised approach is much better when there is only a small amount of labeled data. The two are comparable when a large amount of labeled data is available.

## Keywords

question answering, information extraction, transductive learning, named entity identification

## 1 Introduction

While the identification of named entities (NEs) in a text is an important component of many information retrieval and knowledge management tasks, including question answering and information extraction, its benefits are constrained by its coverage. Typically, it is limited to a relatively small set of classes, such as *person*, *time*, and *location*, for which instances can be recognized with reasonable confidence by straightforward methods with a minimal amount of context. However, in sophisticated applications, such as the non-factoid medical question answering that we consider in this paper, NEs are only a small fraction of the important semantic units discussed in documents or asked about by users. In fact, many semantic roles in scenarios and events that occur often in questions and documents do not contain NEs at all. Therefore, it is imperative to extend the idea of NE identification to other kinds of semantic units. In this paper, we propose an approach to detect a more diverse set of semantic units that goes beyond simple NEs.

Our targets are *cores* of semantic classes or roles in scenario descriptions. The semantics of a scenario is defined by the role that each participant plays in it and can be expressed by a frame structure, where each slot in the frame designates a semantic class. For example, a medical treatment scenario can have three semantic classes: the patient's problem *P*, the treatment or intervention *I*, and the clinical

outcome *O*.<sup>1</sup> The slots in the corresponding frame may be filled with either *complete* or *partial* information. Consider the following example, where parentheses delimit each instance of a semantic class (a slot filler) and the labels *P, I, O* indicate its type:

### *Sentence:*

Two systematic reviews in (people with AMI)*P* investigating the use of (calcium channel blockers)*I* found a (non-significant increase in mortality of about 4% and 6%)*O*.

### *Complete slot fillers:*

*P*: people with AMI

*I*: calcium channel blockers

*O*: a non-significant increase in mortality of about 4% and 6%

### *Partial slot fillers:*

*P*: AMI

*I*: calcium channel blockers

*O*: mortality

The partial slot fillers in this example are the smallest fragments of the corresponding complete slot fillers that exhibit information rich enough for deriving a reasonably precise understanding of the scenario. We use the term *core* to refer to such a fragment of a slot filler. In this example, the cores of the patient's problem and the treatment are both NEs, whereas the core of the clinical outcome is not. Similarly, non-NE cores are common in other scenarios. For example, the *test method* in *diagnosis* scenarios, the *means* in a *shipping* event, and the *manner* in a *criticize* scenario may all have non-NE cores.

In a question answering system, keyword-based document retrieval is usually performed to find relevant documents that may contain the answer to a given question. Keywords in the retrieval are derived from the question. Cores of semantic classes can be extremely valuable in searching for such documents for complex question scenarios, as shown in this example.<sup>2</sup>

### *Question scenario:*

A physician sees a 7-year-old child with asthma in her office. She is on flovent and ventolin currently and was recently discharged from hospital following

<sup>1</sup> Readers familiar with evidence-based medicine will recognize this as a simplification of the PICO representation for the formulation of a problem-centered query [21].

<sup>2</sup> The scenario is an example used in the usability testing in the EPoCare project at the University of Toronto.

her fourth admission for asthma exacerbation. During the most recent admission, the dose of flovent was increased. Her mother is concerned about the impact of the additional dose of steroids on her daughter's growth. This is the question to which the physician wants to find the answer.

For a complex scenario description like this, the answer could be drowned in the large amount of irrelevant passages found by inappropriate keywords derived from the question. However, with the information given by cores of semantic classes, for example *P: asthma, I: steroids, O: growth*, the search can be much more effective.

Similarly, identifying cores of semantic classes in documents can facilitate the question/answer matching process. Some information relevant to the question is listed below, where boldface indicates a core:

E1: A more recent systematic review (search date 1999) found three RCTs comparing the effects of **becolmetasone** and **non-steroidal medication** on linear **growth** in children with **asthma** (200  $\mu$ g twice daily, duration up to maximum 54 weeks) suggesting a short term decrease in linear **growth** of  $-1.54$  cm a year.

E2: Two systematic reviews of studies with long term follow up and a subsequent long term RCT have found no evidence of **growth retardation** in **asthmatic children** treated with inhaled **steroids**.

The sentences here are from the book *Clinical Evidence* (CE) [3], which we are using as the base text in our project on natural-language question answering in evidence-based medicine [17]. The clinical outcomes mentioned in the evidence have very different phrasings — yet both are relevant to the question. The pieces of evidence describe two distinct outcomes. Missing either of the outcomes will lead to an incomplete answer for the physician. Here, the cores of semantic classes provide the only clue that both outcomes must be included in the answer, while complete description of semantic classes with more information could make the matching harder to find because of the different expressions of the outcomes.

In addition, semantics presented in cores of semantic classes can help filter out irrelevant information that cannot be identified by searching methods based on simple string overlaps. Consider these two questions:

In patients with **myocardial infarction**, do  **$\beta$  blockers** reduce **mortality** and **recurrent myocardial infarction** without adverse effects?

In someone with **hypertension** and **high cholesterol**, what management options will decrease his risk of **stroke** and **cardiac events**?

In the first question, the first occurrence of *myocardial infarction* is a disease but the second is part of the clinical outcome. In the second question, *stroke* is part of the clinical outcome rather than a disease to be treated as it usually is. Obviously, string matching cannot distinguish between the two cases. By identifying and classifying cores of semantic classes, the relations between these important semantic units in the scenarios are made very clear. Therefore, documents or passages that do not contain *myocardial infarction* or *stroke* as clinical outcomes can be discarded.

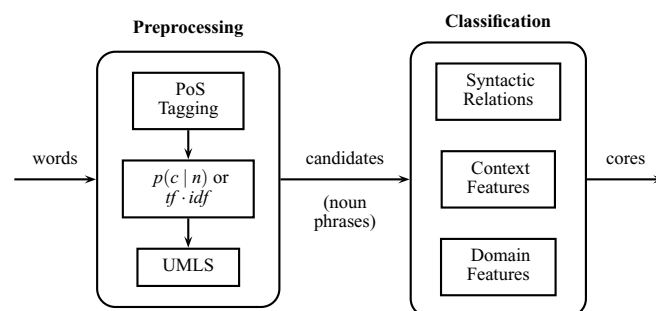


Fig. 1: Architecture of the approach to core identification.

Finally, cores of semantic classes in a scenario are connected to each other by the relations embedded in the frame structure. The frame of the treatment scenario contains a cause-effect relation: an intervention used to treat a problem results in a clinical outcome.

In the following sections, we propose a method to automatically identify and classify the cores of semantic classes according to their context in a sentence. We take the medical treatment scenario as an example, in which the goal is to identify cores of *treatments*, *problems*, and *clinical outcomes*. For ease of description, we will use the terms *intervention-core*, *disease-core*, and *outcome-core* to refer to the corresponding cores. We work at the sentence level, i.e., we identify cores in a sentence rather than a clause or paragraph. Two principles are followed in developing the method. First, complete slot fillers do not have to be extracted before core identification. Second, we aim to reduce the need for expensive manual annotation of training data by using a semi-supervised approach.

## 2 Architecture of the method

In our approach, we first collect candidates for the target cores from sentences under consideration. For each candidate, we classify it as one of the four classes: *intervention-core*, *disease-core*, *outcome-core*, or *other*. In the classification, a candidate will get a class label according to its context, its semantic types in the knowledge base Unified Medical Language System (UMLS), and the syntactic relations in which it participates. Two knowledge resources in UMLS — the Metathesaurus and the Semantic Network — are used. The Metathesaurus is the central vocabulary component of UMLS that contains information about biomedical and health-related concepts. Semantic types of concepts in the Metathesaurus are provided in the Semantic Network. Figure 1 shows the architecture of the approach.

## 3 Preprocessing

Our observation is that cores of the three types of slot fillers are usually nouns or noun phrases. In the preprocessing, all words in the data set are examined. The first two steps are to reduce noise, in which some of the words that are unlikely to be part of real cores are filtered out. Then, the rest are mapped to their corresponding concepts, and these concepts are candidates of target cores.

**PoS tagging.** Words that are not nouns are first removed from the candidate set. PoS tags are obtained by using



Brill's tagger [5].

**Filtering out some “bad” nouns.** This step is the second attempt to remove noise. Nouns that are unlikely to be part of real cores are considered to be “bad” candidates. Two different measures are considered to evaluate how good a noun is.

$tf \cdot idf$ . This is the traditional measure of informativeness of a word with regard to a document. *Clinical Evidence* text is used to obtain the  $tf \cdot idf$  value of a noun. 47 sections in *Clinical Evidence* are segmented to 143 files of about the same size. After the  $tf \cdot idf$  value of a noun is calculated in each file, the highest value is taken as its final score. Nouns with  $tf \cdot idf$  values lower than a threshold are removed from the candidate set. The threshold was set manually after observing the values of some nouns that frequently occur in the text.

**Domain specificity.** We calculate the probability  $p(c|n)$ , where  $c$  is the medical class, and  $n$  is a noun. This is the probability that a document is in the medical domain  $c$  given that it contains the noun  $n$ . Intuitively, *intervention-cores*, *disease-cores*, and *outcome-cores* are domain-specific, i.e., a document that contains them is very likely to be in the medical domain. For example, *morbidity*, *mortality*, *aspirin*, and *myocardial infarction* are very likely to occur in a medicine-related context. Therefore, we intend to retain highly medical domain-specific nouns in the candidate set. Using this measure, a noun is a better candidate if the corresponding probability is high. Text from two domains is needed in this measure: medical text, and non-medical text. In our experiment, we use the same 47 sections in *CE* as the medical class text. For the non-medical class, we use the Reuters collection, as it mainly consists of newswire stories. 1000 documents in the Reuters collection are randomly selected for the calculation. Nouns whose probability values are below a threshold (determined in the same manner as in the  $tf \cdot idf$  measure) are filtered out.

**Mapping to concepts.** In many cases, nouns are part of noun phrases that are better candidates for cores. For example, the phrase *myocardial infarction* is a better candidate for an intervention-core than *infarction*. Therefore, we use the software MetaMap [2] to map a noun to its corresponding concept (which is often a noun phrase) in the Metathesaurus of UMLS. All the concepts form the set of candidates of cores to be classified.

## 4 Representing candidates using features

Given a set of candidates, the classification task is to identify several subsets; each corresponds to a type of slot filler, or a semantic class. We expect that candidates in the same semantic class will have similar behavior, characterized by syntactic relations, context information, and semantic types. All features are binary features, i.e., a feature takes value 1 if it is present; otherwise, it takes value 0.

### 4.1 Syntactic relations

Syntactic relations have been explored in grouping similar words [14] and words of the same sense in word sense disambiguation [12]. Lin [14] inferred that *tesguino* is similar to *beer*, *wine*, etc., i.e., it is a kind of drink, by comparing

---

#### Sentence:

Thrombolysis reduces the risk of dependency, but increases the chance of death.

#### Candidates:

thrombolysis, dependency, death

#### Relations:

(thrombolysis subj-of increase), (thrombolysis subj-of reduce)

(dependency pcomp-n-of of)

(death pcomp-n-of of)

---

**Fig. 2:** Example of dependency triples extracted from output of Minipar parser.

syntactic relations in which each word participates. Kohomban and Lee [12] determined the sense of a word by observing a subset of syntactic relations of the word. The hypothesis is that different instances of the same sense will have similar relations.

We also need to group instances of the same semantic class. Such instances may participate in similar syntactic relations while those of different classes will have different relations. For example, *intervention-cores* often are subjects of sentences, while *outcome-cores* are often objects.

Candidates in our task are phrases, rather than words as in [14] and [12]. Thus, we consider all relations between a candidate noun phrase and other words in the sentence. To do that, we ignore relations between any two words in the phrase when extracting syntactic relations. Any relation between a word not in the phrase and a word in the phrase is extracted. We use the Minipar parser [13] to get the syntactic relations. In the feature construction, a relation triple containing two words and the grammatical relation between them is taken as a feature, as shown in Figure 2. The set of all distinct triples is the syntactic relation feature set in the classification.

### 4.2 Local context

The context of candidates is also important in distinguishing different classes. For example, a disease-core may often have *people with* in its left context. However, it is very unlikely that the phrase *people with mortality* (with an outcome-core) will occur in the text. We consider the two content words on both sides of a candidate. When extracting context features, all punctuation marks are removed except the sentence boundary. The window does not cross boundaries of sentences. We evaluated two representations of context: ordered and unordered. In the ordered case, local context to the left of the phrase is marked by *L-*, that to the right is marked by *R-*. Symbols *L-* and *R-* are used only to indicate the order of text. For the candidate *dependency* in Figure 2, the context features with order are: *L-reduces*, *L-risk*, *R-increases*, and *R-chance*. The context features without order are: *reduces*, *risk*, *increases*, and *chance*.

### 4.3 Domain features

Each candidate has a semantic type defined in UMLS. For example, the semantic type of *death* is **organism function** and that of *dependency* is **physical disability**. These semantic types are used as features in the classification.

**Table 1:** Number of instances of cores in the whole data set.

Intervention-core	501
Disease-core	153
Outcome-core	384
Total	1038

## 5 Data set and analysis

Two sections of *Clinical Evidence* were used in the experiments. A clinician labeled the text for intervention-cores and disease-cores. Complete clinical outcomes are also identified. Using this annotation as a basis, outcome-cores were labeled by the first author. The number of instances of each class is shown in Table 1.

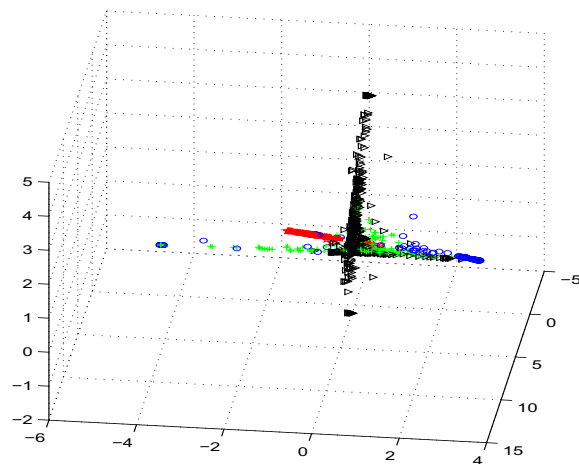
In our approach, the design of the features is intended to group similar cores together. As a first step to verify how well the intention is captured by the features, we observe the geometric structure of the data.

In the analysis, candidates are derived using the domain specificity measure  $p(c|n)$ . Each candidate is represented by a vector of dimensionality  $D$ , where each dimension corresponds to a single feature. The feature set consists of syntactic features, ordered context, and semantic types. We map the high-dimensional data space to a low-dimensional space using the locally linear embedding (LLE) algorithm [20] for easy observation. LLE maps high-dimensional data into a single global coordinate system of low dimensionality by reconstructing each data point from its neighbors. The contribution of the neighbors, summarized by the reconstruction weights, captures intrinsic geometric properties of the data. Because such properties are independent of linear transformations that are needed to map the original high-dimensional coordinates of each neighborhood to the low-dimensional coordinates, they are equally valid in the low-dimensional space. In Figure 3, the data is mapped to a 3-dimensional space (the coordinate axes in the figure do not have specific meanings as they do not represent coordinates of real data). Candidates of the four classes (intervention-core, disease-core, outcome-core, and other) are represented by (red) stars, (blue) circles, (green) crosses, and (black) triangles, respectively. We can see that candidates in the same class are close to each other, and clusters of data points are observed in the figure.

## 6 The model of classification

On the basis of the feature design and data analysis, we choose a semi-supervised learning model developed by Zhu et al. [24] that explores the clustering structure of data in classification. The general hypothesis of the approach is that similar data points will have similar labels.

Let  $x_1, \dots, x_n$  be labeled and unlabeled data. In the model, a graph  $G = (V, E)$  is constructed (it does not have to be fully connected), where the set of nodes  $V$  correspond to both labeled and unlabeled data points and  $E$  is the set of edges. The edge between two nodes  $i, j$  is weighted. Weight  $w_{ij}$  is assigned to agree with the hypothesis so that the edge between two nodes that are closer in the data space gets higher weight. This approach explores the clus-



**Fig. 3:** Manifold structure of data.

ter structure of data by propagating labels from labeled data points to unlabeled data points according to the weights on the edges. Zhu et al. formulate the intuitive label propagation approach as a problem of energy minimization in the framework of Gaussian random fields, where the Gaussian field is over a continuous state space, instead of over a discrete label set. The idea is to compute a *real-valued* function  $f : V \rightarrow \mathcal{R}$  on graph  $G$  that minimizes the energy function  $E(f) = \frac{1}{2} \sum_{i,j} w_{ij} (f(i) - f(j))^2$ . The function  $f = \operatorname{argmin}_f E(f)$  determines the labels of unlabeled data points. This solution can be efficiently computed by direct matrix calculation even for multi-label classification, in which solutions are generally computationally expensive in other frameworks. It is referred to as “SEMI” in the following description.

Label propagation explores the similarity of labeled and unlabeled data points, and thus follows closely the cluster structure of the data in prediction. We expect it to perform reasonably well on our data set. We use the SemiL [10] implementation of SEMI in the experiment.<sup>3</sup>

## 7 Results and analysis

We first evaluate the performance of the semi-supervised model on different feature sets. Then, we compare the candidate sets obtained by using  $tf \cdot idf$  with those obtained by evaluating domain specificity. Finally, we compare the semi-supervised model to a supervised approach.

In all these experiments, the data set contains all candidates of cores. Unless otherwise mentioned, the results reported are obtained using the candidate set derived by  $p(c|n)$ , the feature set of the combination of syntactic relations, ordered context, and semantic types, and the distance measure of cosine distance (as weights on the edges of the graph). The result of an experiment is the average of 20 runs. In each run, labeled data is randomly selected from the candidate set, and the rest is taken as unlabeled data whose labels need to be predicted. We make sure that all classes are present in labeled data; if any class is absent, we redo the sampling. The evaluation of the semantic classes is

<sup>3</sup> As our data is unbalanced, the parameter that handles unbalanced data set is turned on the experiment. Default values of other parameters are used unless otherwise mentioned.

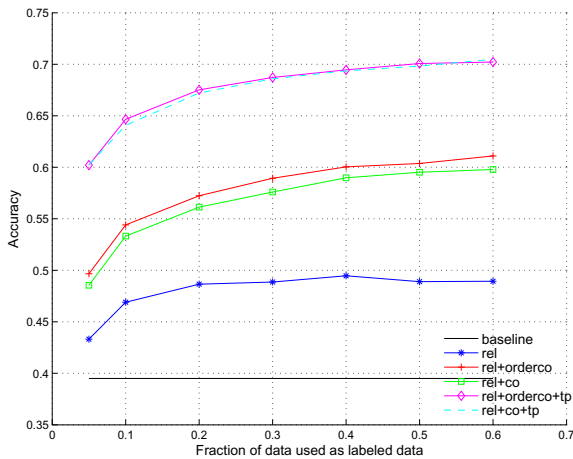


Fig. 4: Classification results of candidates.

very strict: a candidate is given credit if it gets the same label as given by the annotator and the tokens it contains are exactly the same as those marked by the annotator. Candidates that contain only some of the tokens matching the labels given by the annotators are treated as the *other* class.

## 7.1 Experiment 1: Evaluation of feature sets

This experiment evaluates different feature sets in the classification. As described in section 3, two different methods are used in the second step of preprocessing to pick up *good* candidates. Here, as our focus is on the feature set, for space reasons, we report only results on candidates selected by  $p(c|n)$ . (In section 7.2, we compare the two methods of selecting good candidates.)

Figure 4 shows the accuracy of classification using different combinations of the four feature sets: syntactic relations, ordered context, un-ordered context, and semantic types. A baseline is set by assigning labels to data points according to the prior knowledge of the distribution of the four classes, which has accuracy of 0.395. It is clear in the figure that incorporating additional kinds of features into the classification results in a large improvement in accuracy. Using only syntactic relations (*rel* in the figure) as features, the best accuracy is lower than 0.5. The addition of ordered context (*orderco*) or no-order context features (*co*) improves the accuracy by about 0.1. Adding semantic type features (*tp*) improves accuracy by a further 0.1. Combining all four kinds of features achieves the best performance. With only 5% of data as labeled data, the whole feature set achieves an accuracy of 0.6. Semantic types seems to be a very powerful feature set, as it substantially improves the performance on top of the combination of the other two kinds of features. Therefore, we took a closer look at the semantic type feature set by conducting the classification using only semantic types, but found that the result is even worse than using only syntactic relations. This observation reveals interesting relations between the feature sets. In the space defined by only one kind of features, data points may be close to each other, and hence hard to distinguish. Adding another kind sets apart data points in different classes toward a more separable position in the new space. This shows that every kind of feature is informative to the task. The feature sets characterize the candidates from different angles that are complementary in

the task.

We also see that ordered context features are only slightly better than unordered features when semantic types are not considered. This difference is not observable at all when semantic type information is considered.

## 7.2 Experiment 2: Evaluation of candidate sets

In the second step of preprocessing, one of two methods can be used to filter out some *bad* nouns – using  $tf \cdot idf$  value or the domain specificity. This experiment compares the two measures in the core identification task. A third option using neither of the measures (i.e., without filtering) is taken as the baseline. Table 2 shows numbers of instances remaining in the candidate set after preprocessing.

As shown in the table, there are much fewer instances in the *other* class in the sets derived by  $tf \cdot idf$  and the probability measure as compared to those derived by the baseline, which shows that the two measures effectively remove some of the *bad* candidates of intervention-core, disease-core, and outcome-core. At the same time, a small number of cores are removed.<sup>4</sup> Compared to the baseline method, the probability measure keeps almost the same number of intervention-cores and disease-cores in the candidate set, while omitting some outcome-cores. This indicates that outcome-cores are less domain-specific than the other two. Compared to the  $tf \cdot idf$  measure, more intervention-cores and outcome-cores are kept by the domain specificity measure, showing that the probability measuring the domain-specificity of a noun better characterizes the cores of the three semantic classes. The probability measure is also more robust than the  $tf \cdot idf$  measure, which heavily relies on the content of the text from which it is calculated. For example, if an intervention is mentioned in many documents of a document set, its  $tf \cdot idf$  value can be very low although it is a good candidate of intervention-core.

The precision, recall, and  $F$ -score of the classification shown in Table 3 confirms the above analysis. The probability measure gets substantially higher  $F$ -scores than the baseline for all the three classes that we are interested in, using different amounts of labeled data. In particular, the corresponding precision values are much higher than the baseline. Compared to  $tf \cdot idf$ , the performance of the domain specificity measure is much better on identifying intervention-cores, and slightly better on identifying outcome-cores, while the two are similar on identifying disease-cores.

## 7.3 Experiment 3: Comparison of the semi-supervised model and SVMs

In the semi-supervised model, labels propagate along high-density data trails, and settle down at low-density gaps. If the data has the desired structure, unlabeled data can be used to help learning. In contrast, a supervised approach only makes use of labeled data. This experiment compares SEMI to a state-of-the-art supervised approach; the goal is

<sup>4</sup> The first and third step in the preprocessing also results in missing cores in the candidate set. We roughly checked about one-third of the total real cores in the data set and found that 80% of lost cores occur because MetaMap either failed to find the concepts or it extracted more or fewer tokens than marked by the annotator. 10% of missing cores are caused by errors of the PoS tagger, and the rest occur because some cores are not nouns.

**Table 2:** Number of candidates in different candidate sets. Class 1: *intervention-core*, Class 2: *disease-core*, Class 3: *outcome-core*, Class 4: *other*

Measures	Class1	Class2	Class3	Class4
$tf \cdot idf$	243	108	194	785
$p(c n)$	298	106	209	801
baseline	303	108	236	1330

**Table 3:** *F*-score of classification on different candidate sets.

labeled data	1%	5%	10%	30%	60%
<i>intervention-core:</i>					
baseline	.53	.63	.66	.70	.72
$tf \cdot idf$	.51	.61	.64	.69	.71
$p(c n)$	<b>.57</b>	<b>.69</b>	<b>.72</b>	<b>.75</b>	<b>.77</b>
<i>disease-core:</i>					
baseline	.25	.36	.43	.48	.49
$tf \cdot idf$	<b>.29</b>	<b>.41</b>	.46	<b>.53</b>	<b>.55</b>
$p(c n)$	.27	<b>.41</b>	<b>.47</b>	<b>.53</b>	<b>.55</b>
<i>outcome-core:</i>					
baseline	.28	.41	.48	.53	.55
$tf \cdot idf$	.35	<b>.49</b>	.53	.59	.61
$p(c n)$	<b>.37</b>	<b>.49</b>	<b>.54</b>	<b>.60</b>	<b>.63</b>

to investigate how well unlabeled data contributes to the classification using the semi-supervised model. We compare the performance of SEMI to support-vector machines (SVMs) when different amounts of data are used as labeled data. We use OSU SVM [15] in the experiment.<sup>5</sup>

As shown in Table 4, when there is only a small amount of labeled data (less than 5% of the whole data set), which is often the case in real-world applications, SEMI achieves much better performance than SVMs in identifying all the three classes. For *intervention-core* and *outcome-core*, with 5% data as labeled data, SEMI outperforms SVMs with 10% data as labeled data. Similarly, SVMs need to have about three times the labeled data to gain the same performance achieved by SEMI using 10% data as labeled data. With less than 60% data as labeled data, the performance of SEMI is either superior to or comparable to SVMs for *intervention-core* and *outcome-core*. This shows that SEMI effectively exploits unlabeled data by following the manifold structure of the data. The promising results achieved by SEMI show the potential of exploring unlabeled data in classification.

## 8 Related work

The task of named entity (NE) identification, similar to the core-detection task, involves identifying words or word sequences in several classes, such as proper names (locations, persons, and organizations), monetary expressions, dates and times. NE identification has been an important research topic ever since it was defined in MUC [16]. In 2003, it was taken as the shared-task in CoNLL [22]. Most

<sup>5</sup> For the parameter that handles unbalanced data, we set it according to the prior knowledge of the class distribution and give larger weight to a class that contains fewer instances.

**Table 4:** *F*-score of classification using different models.

labeled data	1%	5%	10%	30%	60%
<i>intervention-core:</i>					
semi	.57	.69	.72	.75	.77
SVM	.33	.60	.68	.74	.77
<i>disease-core:</i>					
semi	.27	.41	.47	.53	.55
SVM	.21	.38	.54	.62	.65
<i>outcome-core:</i>					
semi	.37	.49	.54	.60	.63
SVM	.07	.27	.44	.56	.62

statistical approaches use supervised methods to address the problem [9, 6, 11]. Unsupervised approaches have also been tried in this task. Thelen and Riloff [23] explored a bootstrapping method to learn semantic lexicons of six categories: building, event, human, location, time, and weapon. Cucerzan and Yarowsky [8] also used a bootstrapping algorithm to learn contextual and morphological patterns iteratively. Collins and Singer [7] tested the performance of several unsupervised algorithms on the problem: modified bootstrapping (DL-CoTrain) motivated by co-training [4], an extended boosting algorithm (CoBoost), and the Expectation Maximization (EM) algorithm. The results showed that DL-CoTrain and CoBoost are superior to EM, while the two are almost the same.

Much effort in entity extraction in the biomedical domain has gene names as the target. Various supervised models including naive Bayes, support-vector machines, and hidden Markov models have been applied [1]. The work most related to our core-identification in the biomedical domain is that of Rosario and Hearst [19], which extracts *treatment* and *disease* from MEDLINE and examines seven relation types between them using generative models and a neural network. They claim that these models may be useful when only partially labeled data is available, although only supervised learning is conducted in the paper. The best *F*-score of identifying *treatment* and *disease* obtained by using the supervised method was .71. Another piece of work extracting similar semantic classes was that of Ray and Craven [18]. They report an *F*-score of about .32 for extracting *proteins* and *locations*, and an *F*-score of about .50 for *gene* and *disorder*.

## 9 Conclusion

We proposed a novel approach to automatically identify and classify cores of semantic classes in scenario descriptions. In the classification, a semi-supervised model that explores the clustering structure of the data was applied. Our experimental results show that syntactic relations, context, and semantic types are informative and complement features for this task. The features characterize the cluster structure of the data, and unlabeled data is effectively used. Compared to a state-of-the-art supervised approach, the performance of the semi-supervised approach is much better when there is only a small amount of labeled data, and performance of the two are comparable when larger amounts of labeled data are available.

Our approach does not require prior knowledge of semantic classes, and it effectively exploits unlabeled data.

The promising results achieved show the potential of semi-supervised models that explore the clustering structure of data in tasks of grouping *similar* instances. This approach can be applied to other domains as well; the syntactic relation and context features can be constructed in the same way. For domains that do not have a knowledge base like UMLS, the WordNet hierarchy may be used to get features like semantic types. In this case, the level of generalization in WordNet needs to be investigated.

A difficulty of using this approach, however, is in detecting boundaries of the targets. A segmentation step that pre-processes the text is needed. In the next step of our work, we aim to investigate approaches that perform the segmentation precisely.

**Acknowledgments.** Our work is supported by a grant from the Natural Sciences and Engineering Research Council of Canada and grants from Bell University Laboratories at the University of Toronto. We thank Xiaodan Zhu, Jianhua Li, Sharon Straus, Suzanne Stevenson, Gerald Penn and John Mylopoulos for their helpful discussion and comments on this work.

## References

- [1] S. Ananiadou and J. Tsujii, editors. *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*. Association for Computational Linguistics (ACL), PA, USA, 2003.
- [2] A. R. Aronson. Effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program. In *Proceedings of the American Medical Informatics Association Symposium*, pages 17–21, 2001.
- [3] S. Barton, editor. *Clinical evidence*. BMJ Publishing Group, London, 2002.
- [4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, 1998.
- [5] E. Brill. *A Corpus-Based Approach to Language Learning (PhD thesis)*. U of Pennsylvania, 1993.
- [6] H. L. Chieu and H. T. Ng. Named entity recognition with a maximum entropy approach. In *Proceedings of 7th Conference on Computational Natural Language Learning*, pages 160–163, 2003.
- [7] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.
- [8] S. Cucerzan and D. Yarowsky. Language independent named entity recognition combining morphological and contextual evidence. In *Proc of the 1999 Joint SIGDAT Conf on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.
- [9] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named entity recognition through classifier combination. In *Proc of 7th Conf on Computational Natural Language Learning*, pages 168–171, 2003.
- [10] T.-M. Huang, V. Kecman, and I. Kopriva. *Kernel Based Algorithms for Mining Huge Data Sets*. Springer, Berlin, Germany, 2006.
- [11] D. Klein, J. Smarr, H. Nguyen, and C. D. Manning. Named entity recognition with character-level models. In *Proc of 7th Conf on Computational Natural Language Learning*, pages 180–183, 2003.
- [12] U. S. Kohomban and W. S. Lee. Learning semantic classes for word sense disambiguation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 34–41, 2005.
- [13] D. Lin. Principar – an efficient, broad-coverage, principle-based parser. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 482–488, 1994.
- [14] D. Lin. Automatic retrieval and clustering of similar words. In *Proc of the 17th International Conf on Computational Linguistics*, pages 768 – 774, 1998.
- [15] J. Ma, Y. Zhao, S. Ahalt, and D. Eads. OSU SVM classifier Matlab toolbox. In <http://svm.sourceforge.net/docs/3.00/api/>, 2003.
- [16] MUC. Message understanding conferences. In <http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html>, 1995.
- [17] Y. Niu, X. Zhu, J. Li, and G. Hirst. Analysis of polarity information in medical text. In *Proceedings of Annual Symposium of American Medical Informatics Association*, pages 570–574, 2005.
- [18] S. Ray and M. Craven. Representing sentence structure in hidden Markov models for information extraction. In *Proc 17th International Joint Conf on Artificial Intelligence*, pages 1273–1279, 2001.
- [19] B. Rosario and M. A. Hearst. Classifying semantic relations in bioscience texts. In *Proceedings of 42nd Annual Meeting of the Association for Computational Linguistics*, pages 431–438, 2004.
- [20] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
- [21] D. L. Sackett, S. E. Straus, W. S. Richardson, W. Rosenberg, and R. B. Haynes. *Evidence-Based Medicine*. Harcourt, Edinburgh, 2000.
- [22] E. F. T. K. Sang and F. D. Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL*, pages 142–147, 2003.
- [23] M. Thelen and E. Riloff. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 214–221, 2002.
- [24] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, 2003.

# Exploring new measures for Open-Domain Question Answering Evaluation within a Time Constraint

Elisa Noguera<sup>1</sup>, Fernando Llopis<sup>1</sup>, Antonio Ferrández<sup>1</sup>, Alberto Escapa<sup>2</sup>

<sup>1</sup>GPLSI. Departamento de Lenguajes y Sistemas Informáticos  
Escuela Politécnica Superior. University of Alicante  
{elisa,llopis,antonio}@dlsi.ua.es

<sup>2</sup>Departamento de Matemática Aplicada  
Escuela Politécnica Superior. University of Alicante  
alberto.escapa@ua.es

## Abstract

Common researches on evaluating the performance of Question Answering (QA) systems are focused on the evaluation of the precision. In previous work, we studied the importance of the answer time on the evaluation of the QA systems, developing a mathematic procedure in order to explore new evaluation measures in QA systems. In this paper, we keep on with this, with the aim of improving the  $MRRT_E$  measure proposed in mentioned work. For the experiments, we evaluate the results of the participant systems in the realtime experiment carried out at CLEF-2006 with the  $MRRT_{E,r}$  measures. The main conclusion is that  $MRRT_{E,r}$  is a suitable measure for the evaluation of QA systems within a time constraint, allowing to select different evaluation measures, accordingly some prefixed criteria.

## Keywords

Question Answering, Performance, Evaluation Measures

## 1 Introduction

The goal of Question Answering (QA) systems is to locate concrete answers to questions in collections of text. These systems are very useful for the users because they do not need to read all the document or fragment to obtain a specific information. Questions as: How old is Nelson Mandela? Who is the president of the United States? When was the Second World War? can be answered by these systems. They contrast with the more conventional Information Retrieval (IR) systems, because they treat to retrieve relevant documents to a query, where the query may be a simple collection of keywords (e.g. old Nelson Mandela, president United States, Second World War, ..).

The annual Text REtrieval Conference (TREC<sup>1</sup>), Cross-Language Evaluation Forum (CLEF<sup>2</sup>) and National Institute of Informatics Test Collection for IR Systems (NTCIR<sup>3</sup>) are a serie of workshops designed to advance in the state-of-the-art in text retrieval by

<sup>1</sup> <http://trec.nist.gov>

<sup>2</sup> <http://www.clef-campaign.org>

<sup>3</sup> <http://research.nii.ac.jp/ntcir>

providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies. They have a specific QA track ([5], [3], [2]). This evaluation consists of given a large number of newspaper and newswire articles, participating systems try to answer a set of questions by analyzing the documents in the collection in a fully automated way.

The main evaluation measures used in these forums are *accuracy*, *Mean Reciprocal Rank (MRR)*, *K1 measure* and *Confident Weighted Score (CWS)* (for further information about the QA task at CLEF see [3]). The answer time of the QA systems is not considered in these evaluation measures. There are several aspects in these evaluations of QA systems that could be improved: (1) the answer time is not evaluated and this causes that the systems have a good performance but they can be very slowly also, and (2) the comparison among QA systems can be difficult if they have different answer time. Therefore, the performance analysis involves the evaluation of the speed and the effectivity of the systems. Because of that, the motivation of this work is continue studying the evaluation of QA systems within a time constraint, initiate in previous works ([1], [4]).

The remainder of this paper is organized as follows: the next section describes the  $MRRT_E$  evaluation measure and its new uniparametric family of functions. Section 3 describes the evaluation used and the results achieved. Finally, section 4 gives some conclusions and future work.

## 2 New uniparametric family of evaluation measures $MRRT_{E,r}$

From a mathematical point of view the problem of classifying QA systems accordingly to their precision and efficiency can be solved by introducing a ranking function ([1], [4]). Let us recall that with the aid of this two real variables function, it is possible to define a preorder relationship among all the QA systems, which will be identified with an ordered pair of real numbers that reflect the precision and the efficiency of the system. In this way, the system  $(c, d)$  is situated in a higher position of the classification than the system  $(a, b)$  if it is fulfilled the condition



$$(a, b) \preceq (c, d) \Leftrightarrow f(a, b) \leq f(c, d), \forall (a, b), (c, d) \in D, \quad (1)$$

being  $D$  the set of all possible values accessible to the QA systems. In our case this set is given by  $D \equiv [0, 1] \times (0, 1]$ , since we will characterize the precision by the mean reciprocal rank ( $MRR$ ), ranging from 0 to 1 and the efficiency by the effective time resulting of dividing the answer time of each system by the higher answer time register in the QA task. In this way the efficiency runs from 0 to 1, corresponding the lesser values to more efficient systems. By so doing, the best possible system result corresponds to the ideal situation with precision 1 and efficiency 0, that is to say, to the pair  $(1, 0)$  and the worst one to the pair  $(0, 1)$ .

This manner of ranking the systems is easily visualized by means of a ranking graphic. In this graphic we plot each system as a point of  $\mathbb{R}^2$  and partition the plane with the iso ranking curves ([1], [4]), which are merely the level curves of the ranking function, that is to say, the subset of all the points of  $D$  that fulfill the equation  $f(x, t) = C$ , being  $C$  a real number belonging to the image of  $f$ . By so doing, we can rank immediately all the systems of a QA task with respect to a given ranking function.

There are countless ways to define a ranking function. Each definition reflects our preferences about the criteria that we have considered suitable to classify the QA systems, given different weights to the precision and efficiency. Anyway, all these functions must share some common features ([1], [4]) like:

1. The function  $f$  must be continuous in  $D$ .
2. The supremum of  $I$  is given by  $\lim_{t \rightarrow 0} f(1, t)$ . In the case that  $I$  is not upper bound, we must have  $\lim_{t \rightarrow 0} f(1, t) = +\infty$ .
3. The infimum of  $I$  is given by  $f(0, 1)$ .

Most of the possible definitions of the ranking functions are equivalent for our purposes since the above described procedure is of an ordinal type. This means that the relevant information to classify the systems is the relative difference of the numerical values of the ranking function for different systems, being meaningless the concrete value of the ranking function for a single system.

Within this framework there have been considered different kinds of ranking functions ([4]) that take into account both the precision and the efficiency of the systems, not only the precision as it is usually done in the evaluation of QA systems. For the time being, the most suitable ranking function that we have found is the so called  $MRRT_E$  measure. This ranking function depends both on the precision and the efficiency of the system but in such a way that the efficiency has less weight than the accuracy, being its analytical expression

$$MRRT_E(x, t) = \frac{2x}{1 + e^t}, \quad (2)$$

with  $e^t$  the exponential of the effective time. This function verifies the following requirements:

1. The image of  $MRRT_E$  is the interval  $[0, 1)$ .
2. The function  $MRRT_E$  is continuous in  $D$ .
3.  $\lim_{t \rightarrow 0} MRRT_E(1, t) = 1$ .
4.  $MRRT_E(0, 1) = 0$ .

The ranking graphic corresponding to this function is sketched in figure 4. An important feature of this function is that if the systems answer instantaneously, effective time equal to 0, this ranking function coincides with the usual  $MRR$  measure. However, the dependence on time of  $MRRT_E$  modulates the value of  $x$ : if the time grows up the value of the ranking function, for a fixed value of  $MRR$ , decreases.

To control the weight of the efficiency in the evaluation of QA systems, it would be expedient to have a family of ranking functions of the same type controlled by a set of parameters. By so doing, the value of the parameters could be adjusted in any QA task allowing to design different evaluation measures, accordingly some prefixed criteria. In view of the good properties of the  $MRRT_E$  ranking function, we have constructed an uniparametric family of ranking functions of this kind whose expression is given by

$$MRRT_{E,r}(x, t) = \frac{2x}{1 + e^{rt}}, \quad (3)$$

being  $r$  the control parameter. Let us note that if  $r = 1$  we recover the expression of the ranking function  $MRRT_E$ . In addition, it is worthy to note that the usual  $MRR$  measure, which only takes into account the precision of the system, is also contained in this family of ranking functions. In particular, this is achieved by taking  $r = 0$ . In general, the real parameter  $r$  can only take values in the interval  $[0, +\infty)$ . This ensures that the family of the ranking functions  $MRRT_{E,r}$  also verifies the requirements that we have imposed to any ranking function for all the allowed values of the parameter  $r$ . Namely:

1. The image of  $MRRT_{E,r}$  is the interval  $[0, 1)$ .
2. The function  $MRRT_{E,r}$  is continuous in  $D$ .
3.  $\lim_{t \rightarrow 0} MRRT_{E,r}(1, t) = 1$ .
4.  $MRRT_{E,r}(0, 1) = 0$ .

The condition that the parameter  $r$  cannot take negative values is derived from the fact that for a fixed value of  $x$ , the resulting real function of  $t$  must be a non increasing function, since we require that the systems with small efficiency, high value of  $t$ , have a lesser ranking function value than the systems with high efficiency, small value of  $t$ . This condition is mathematically translated by imposing that

$$\frac{\partial MRRT_{E,r}(x, t)}{\partial t} \leq 0, \quad (4)$$

provided the ranking function has partial derivative with respect to time in the interior of  $D$ , as it is the case. This also allows us to give a direct meaning to the parameter  $r$ : when the value of  $r$  increases from 0

to  $+\infty$  the weight of the efficiency is also increased. In this way, a ranking function with a small value of the parameter  $r$  takes into account very little the efficiency of the systems in the evaluation of the QA task. This is clear if we observe the functional form of the ranking function family, where the  $MRR$  value is multiplied by a function that only depends on time and always take positive values equal or smaller than 1. In the figure 1 we represent this time function, which is equal to  $MRRT_{E,r}(1,t)$ , for different values of the parameter  $r$ , showing that for higher values  $r$  the value of  $MRR$  is more and more penalized as the time grows up.

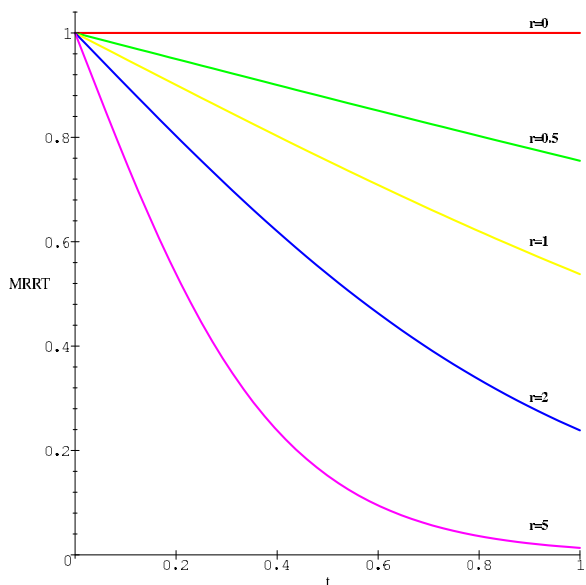


Fig. 1: Function  $MRRT_{E,r}(1,t)$

It is also convenient to interpret the meaning of the parameter  $r$  in terms of the ranking graphic or, equivalently, of the balance between the precision and efficiency of a system. To do this let us introduce a new real parameter  $p \in [0, 1)$  relates with the parameter  $r$  by means of the expression

$$r = \log \left( \frac{1+p}{1-p} \right), \quad (5)$$

being  $\log$  the natural logarithm of a positive real number. The meaning of this new parameter is derived in the following way. Let us consider the system  $A = (1, 1)$ , that is to say, the system with the highest precision and the lowest efficiency. If the evaluation measure takes into account the efficiency, this system will be tied with all the systems that are located in the same iso ranking curve, in other words, with all the systems whose ranking function had the value

$$MRRT_{E,r}(x,t) = 1 - p, \quad (6)$$

since

$$MRRT_{E,r}(1,1) = \frac{2}{1+e^r} = \frac{2}{1+e^{\log \left( \frac{1+p}{1-p} \right)}} = 1-p. \quad (7)$$

In this way, all the systems with less precision than 1 but with higher efficiency will be ranked in the same position of the classification as the system  $A$ . In particular, the smallest possible value of  $MRR$  will be associated to an ideal system that answers instantaneously, therefore the ranking function for this system will be

$$MRRT_{E,r}(x,0) = x. \quad (8)$$

Hence, we derive that this minimum value of  $MRR$  is precisely  $1 - p$ , corresponding to the system  $B = (1 - p, 0)$ . In this way, to fix the value of the parameter  $p$  is equivalent to establish in the evaluation measure that amount of precision we can balance increasing the efficiency of the system. Since our family of ranking functions has a non linear dependence this only can be done by taking a reference system that in our case and for simplify the algebra is the system  $A = (1, 1)$ . So the value of the parameter  $p$  means that all the systems  $(x,t)$  with  $1 - p \leq x \leq 1$  and  $0 \leq t \leq 1$  belonging to the iso ranking curve  $1 - p$  are tied in our rank.

### 3 Experiments

As above is mentioned, we considered the time as a fundamental part in the evaluation of QA systems. In accordance with CLEF organization, we carried out a pilot task at CLEF-2006 whose aim was to evaluate the ability of QA systems to answer within a time constraint. This was an innovative experiment and the initiative was aimed towards providing a new scenario for the evaluation of QA systems. This experiment followed the same procedure that the main task at QA@CLEF-2006, but the main difference was the consideration of the answer time. In total, five groups took part in this pilot task. The participating groups were: *daedalus* (Spain), *tokyo* (Japan), *priberam* (Portugal), *alicante* (Spain) and *inaoe* (Mexico) (for futher information about the realtime experiment see [3]).

#### 3.1 Performance Evaluation

In this section we evaluate the five systems which participated in the realtime experiment with the uniparametric family of evaluation measures  $MRRT_{E,r}$ .

In table 1, it is shown the summary of results for the used metrics (MRR,  $t$ ,  $MRRT_E$ ). Also, the ranking of each measure ( $r$ ) is shown. Namely, *daedalus1* and *daedalus2* point out the two runs that *daedalus* sent.

Graphically, we can compare the different values of  $MRRT_E$  with a ranking graphic (see figure 4). For example, *tokyo* had the second best MRR (0.38) and the worst  $t$ , and it is penalized being the last in the ranking of  $MRRT_E$ . It can be observed that the ranking graphic denote the same position in the ranking list, for example to obtain the same position in the ranking as the system  $S \equiv (0.4, 0)$  the precision needed could vary in the range from 0.36 to 0.67, corresponding to a variation of the time from 0 to 1.



Participant	MRR (r)	tsec/t (r)	$MRRT_E$ (r)
daedalus1	<b>0.41</b> (1°)	549/0.10 (4°)	<b>0.38</b> (1°)
tokyo	0.38 (2°)	5141/1.00 (6°)	0.20 (6°)
priberam	0.35 (3°)	<b>56/0.01</b> (1°)	<b>0.34</b> (2°)
daedalus2	0.33 (4°)	198/0.03 (3°)	0.32 (3°)
inaoe	0.3 (5°)	1966/0.38 (5°)	<b>0.24</b> (4°)
alicante	0.24 (6°)	76/0.02 (2°)	<b>0.23</b> (5°)

**Table 1:** Evaluation results with the different metrics ( $MRR, t, MRRT_E$ )

### 3.1.1 Evaluation results with the evaluation measures: $MRRT_{E,r}$ .

In this work, we have proposed an improvement in the metric  $MRRT_E$ . This improvement was presented in section 2. With this metric, we can give more significance to one measure than the other (MRR or t) giving different values to the parameter  $p$ .

In the figures 2, 3, 4 and 5 we have shown the ranking graphics for four values of the parameter  $p$ . As it can be seen, in the case  $p = 0$ , which correspond to  $r = 0$ , the ranking function coincides with the usual  $MRR$  measure since, with the meaning given in the section 2 to the parameter  $p$ , we cannot allow to balance the precision with the efficiency. In the other cases, it can be observed that accordingly the values of the parameter  $p$  are higher the iso ranking curves are more bended corresponding to the situation in which a low precision of a system can be compensated by increasing its efficiency. In particular, for the value  $p = 0.46$  ( $r = 1$ ) we recover the  $MRRT_E$  evaluation measure.

In this way, we can obtain different classification of the systems determined by the values of the parameter  $p$  or, equivalently the associated parameter  $r$ . For example, daedalus1 and tokyo obtain the best results of  $MRR$  (0.41 and 0.38 respectively). But, the position of tokyo goes down in the ranking accordingly we increase the values of  $p$ . On the contrary, alicante obtains the worst value of MRR (0.24), as a consequence it is the last in the ranking if we take only the MRR into account, but it goes up if we increase the parameter  $p$ . daedalus1 and priberam do not change their position in the ranking practically, although if we increase the parameter  $p$  their values bring near, because priberam has a shorter answer time than daedalus.

## Acknowledgments

This research has been partially funded by the Spanish Government under project CICyT number TIC2006-15265-C06-01 and by the Valencia Government under project number GV06-161.

## 4 Conclusions and Future Work

Mainly, the evaluation of QA systems is studied deeply in three known evaluation forums: TREC, CLEF and NTCIR. But, these forums are only focused on evaluating the precision of the systems, and they do not evaluate their efficiency (we consider the answer time of the system as measure of efficiency). Mostly, this evaluation entails accurate systems but slowly at the

same time. For this reason, we studied the evaluation of QA systems taking into account the answer time.

In previous work, we proposed a new measure ( $MRRT_E$ ) to evaluate QA systems within a time constraint. This measure is based on an exponential function and it allows to classify the systems considering the MRR and the answer time.

In this work, we have improved the  $MRRT_E$  measure building a uniparametric family of evaluation measures  $MRRT_{E,r}$ . These depend on the parameter  $p$ , so that if we give different values to it, we can give more prior to the MRR or the answer time. For the experiments, we used the results of the experiment that we carried out in the CLEF-2006 in order to evaluate QA systems within a time constraint. As conclusion, we can fix the prior of the efficiency or the precision in the evaluation of QA systems, with a uniparametric family of ranking functions:  $MRRT_{E,r}$ . The value of the parameter  $r$  can be adjusted in any QA task allowing to design different evaluation measures, accordingly some prefixed criteria.

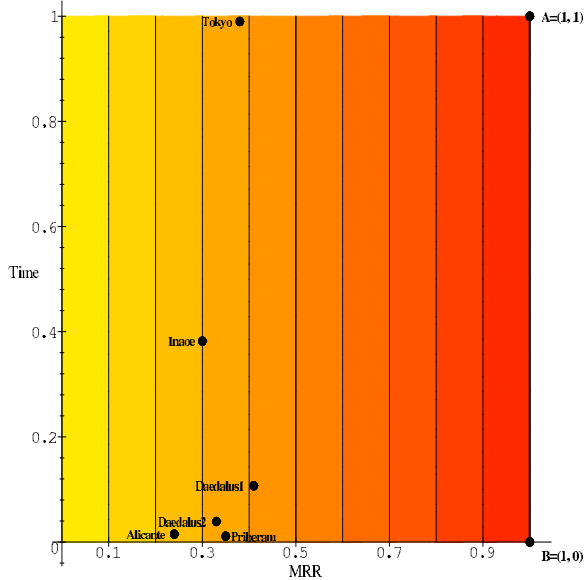
Finally, the future direction that we plan to undertake is to take into account more variables as the hardware used by the systems.

## References

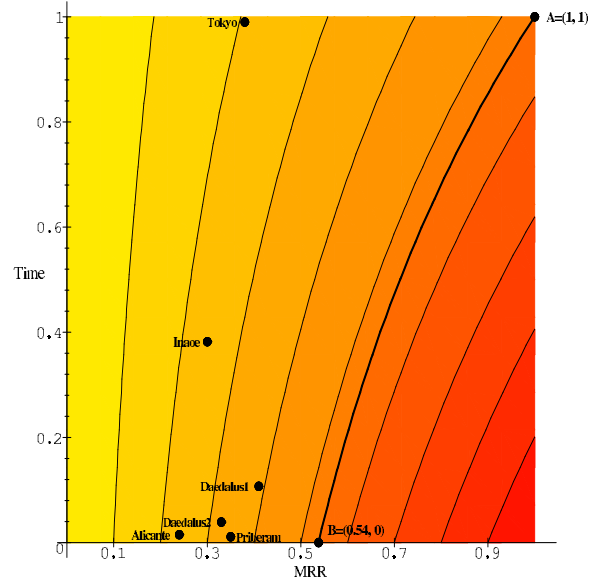
- [1] A. Escapa, E. Noguera, F. Llopis, and A. Ferrández. Ranking Functions to evaluate Open-Domain Question Answering Systems. Unpublished, 2007.
- [2] J. Fukumoto, T. Kato, , and F. Masui. Question Answering Challenge (QAC-1): An Evaluation of Question Answering Task at NTCIRWorkshop 3. In K. Oyama, E. Ishida, and N. Kando, editors, *Proceedings of the Third NTCIR Workshop on Research in Information Retrieval, Question Answering and Summarization*, volume 3, Tokyo (Japan), October 2002. National Institute of Informatics (NII).
- [3] B. Magnini, D. Giampiccolo, P. Forner, C. Ayache, P. Osenova, A. Peñas, V. Jijkoun, B. Sacaleanu, P. Rocha, and R. Sutcliffe. Overview of the CLEF 2006 Multilingual Question Answering Track. In A. Nardi, C. Peters, and J. Vicedo, editors, *WORKING NOTES CLEF 2006 Workshop, 20-22 September, Alicante, Spain. Results of the CLEF 2006 Cross-Language System Evaluation Campaign*, 2006.
- [4] E. Noguera, F. Llopis, A. Ferrández, and A. Escapa. New Evaluation Measures for Open-Domain Question Answering Systems within a time constraint. In *Text, Speech and Dialogue. Proceedings of the 10th International Conference TSD 2007, Plzen, Czech Republic, September 2007*. To appear, 2007.
- [5] E. M. Voorhees and H. T. Dang. Overview of the TREC 2005 Question Answering Track. In *TREC*, 2005.

Participant	$p=0$ (r)	$p=0.25$ (r)	$p=0.46$ (r)	$p=0.75$ (r)
daedalus1	<b>0.41</b> (1°)	0.40 (1°)	0.39 (1°)	0.37 (1°)
tokyo	0.38 (2°)	<b>0.28</b> (4°)	<b>0.19</b> (6°)	0.09 (6°)
priberam	0.35 (3°)	<b>0.35</b> (2°)	0.35 (2°)	0.35 (2°)
daedalus2	0.33 (4°)	<b>0.33</b> (3°)	0.32 (3°)	0.32 (3°)
inaoe	0.30 (5°)	0.27 (5°)	0.23 (5°)	0.19 (5°)
alicante	0.24 (6°)	0.24 (6°)	<b>0.24</b> (4°)	0.24 (4°)

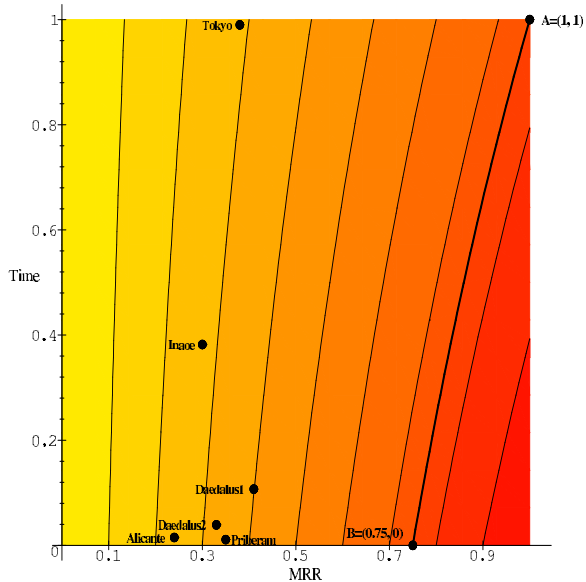
**Table 2:** Evaluation results with the different values of  $p$  in the  $MRRT_{E,r}$  metric



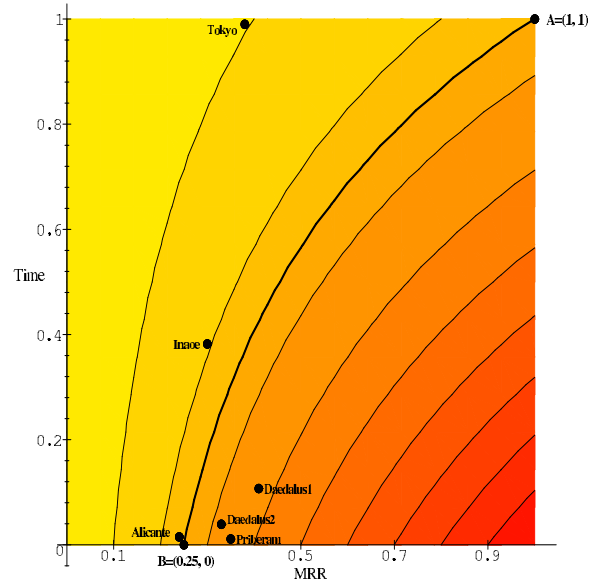
**Fig. 2:** Comparative of the results obtained by each system with the  $MRRT_{E,r}$  evaluation measure in its ranking graphic (with  $p=0$ ). It fits with MRR.



**Fig. 4:** Comparative of the results obtained by each system with the  $MRRT_{E,r}$  evaluation measure in its ranking graphic (with  $p=0.46$ ). It fits with  $MRRT_E$ .



**Fig. 3:** Comparative of the results obtained by each system with the  $MRRT_{E,r}$  evaluation measure in its ranking graphic (with  $p=0.25$ ).



**Fig. 5:** Comparative of the results obtained by each system with the  $MRRT_{E,r}$  evaluation measure in its ranking graphic (with  $p=0.75$ ).

# Pronominal anaphora resolution for text summarisation

Constantin Orăsan  
Research Group in Computational Linguistics  
University of Wolverhampton  
Stafford St.  
Wolverhampton, WV1 1SB, UK  
*C.Orasan@wlv.ac.uk*

## Abstract

The assumption of term-based summarisation method is that the importance of a sentence can be determined by the importance of the words it contains. One drawback of these methods is that they usually consider the words in isolation, ignoring relations such as anaphoric links between them. This paper investigates to what extent the integration of pronominal anaphora resolution into the summarisation process can improve the informativeness of the automatically produced summaries. Evaluation of three anaphora resolution methods plus three baselines on a corpus of journal articles shows that anaphora resolution can have a beneficial effect on informativeness. In addition, one experiment which uses a simulated anaphora resolver with predefined accuracy is performed in order to demonstrate that term-based summarisation can benefit from anaphora resolution, but that high accuracy methods are necessary.

## Keywords

automatic summarisation, evaluation, anaphora resolution

## 1 Introduction

With the current information overload experienced by researchers, it is increasingly difficult to keep up-to-date with all current developments in a field, and it has become necessary to look for a piece of information only when it is needed. Automatic summarisation can help people deal with this abundance of information by extracting the gist of it. Term-based summarisation is one of the most common components of text summarisation systems and is the focus of this paper. First proposed by Luhn [27], it is still widely used today in combination with other methods [23, 44, 25, 43] due to its lack of complexity and its high speed. The assumption of term-based summarisation is that it is possible to determine the importance of a sentence on the basis of the words it contains. The most common way of achieving this is to weigh all the words in a text and calculate the score of a sentence by adding together the weights of the words within it. In this way, a summary can be produced by extracting the sentences with the highest scores until the desired length is reached. One drawback of most implementations of term-based summarisers

is that they score words in isolation, ignoring links between words such as anaphoric relations.

This paper investigates the extent to which pronominal anaphora resolution can improve the results of a term-based summariser by resolving pronouns to their antecedents and incorporating this information in the summariser. It has to be pointed out that the purpose of this paper is not to produce a new summarisation method, but to assess whether information from an anaphora resolver can be beneficial for the summarisation process. To this end, term-based summarisation is very appropriate for this task as it depends on a limited number of parameters and any change in its performance can be justified by the additional information from the anaphora resolver.

The paper is structured as follows: Section 2 briefly presents background information about automatic summarisation, pronominal anaphora resolution and previous attempts to combine the two. Section 3 presents the term-based summarisation method employed in this paper, as well as the anaphora resolvers used to enhance the summarisation method. The corpus used in our experiments is described in Section 4, which is followed by a section on evaluation. The evaluation focuses on both the accuracy of anaphora resolution and on the performance of term-based summarisation methods, and tries to establish whether there is any correlation between the accuracy of an anaphora resolver and the increase in the accuracy of the summariser when it incorporates the resolver.<sup>1</sup> In order to further assess the influence of anaphora resolution for automatic summarisation, the evaluation also presents the effects of a resolver with controlled accuracy on term-based summarisation. The paper finishes with conclusions.

## 2 Background

Both automatic summarisation and anaphora resolution have received extensive attention from the research community. This section briefly describes the two areas with an emphasis on the aspects relevant to this paper. Due to space restrictions no attempt will be made to present a comprehensive overview of the two fields. Such an overview for automatic

---

<sup>1</sup> At this stage the *accuracy* and *performance* are used as general terms. Section 5 explains what they mean in the context of this paper.

summarisation can be found in [28] and for anaphora resolution in [31].

## 2.1 Automatic summarisation

Automatic summarisation is a field in computational linguistics concerned with the development of systems which can produce summaries automatically. These systems take one or several related documents, and summarise the most important information from them or information related to aspects chosen by a user. These systems could prove very useful for example to researchers who need to quickly find out the content of an article. Unfortunately, with the current technology, it is difficult to produce automatic summaries which replace the whole document. Instead, automatic summarisation can be used to produce summaries which indicate whether a document is relevant to one's interests, allowing researchers to quickly browse through large masses of information.

The first attempt to produce automatic summaries is presented in [27], and relies on the distribution of words in a text to identify the important sentences. The promising results obtained by Luhn encouraged other researchers to apply similar approaches, in most cases in combination with other methods [14, 8, 23, 44, 43]. Alternative summarisation methods rely on presence of certain words [14] or phrases [39], discourse structure [34, 29, 12], anaphoric or coreferential links [9, 2] or lexical repetition [7, 11] to name but a few.

## 2.2 Anaphora resolution

Haliday and Hasan [17] describe anaphora as 'cohesion which points back to some previous item'. Anaphora resolution is the process of resolving an *anaphor*, the pointing back expression, to the word or phrase it points back to, to an *antecedent*. If the antecedent and the anaphor have the same referent in the real world they are *coreferential* [31].

One of the most widespread type of anaphor and usually dealt with by computational linguists is *pronominal anaphora with NPs as antecedent*, a type of *nominal anaphora* [31]. In this case, an anaphoric expression is represented by a personal, possessive or reflexive pronoun, and the antecedent consists of one or several NPs. Due to their characteristics and high frequency of use, such anaphoric expressions pose great challenges to most of the fields in computational linguistics. The main reason for this is that these expressions do not carry much information on their own, and therefore have to be processed before they can be used.

Researchers in pronominal anaphora resolution have dedicated a great amount of effort to developing automatic resolution methods. Some of the methods rely mainly on one type of information, whereas others try to combine information from different sources. A syntax-based method is proposed in [20], whilst [21] relies mainly on semantic information for resolving personal pronouns. Centering Theory [16], a discourse theory of local coherence, is used as the only method to resolve pronouns in [10]. Methods which combine several types of information include methods employing preferences, indicators and

constraints [24, 22, 3, 30] and machine learning based methods [1, 15, 38, 40, 5]. A comprehensive discussion of these methods can be found in [31], and the methods investigated here are briefly described in Section 3.2.

## 2.3 Pronominal anaphora resolution and automatic summarisation

Even though it was hypothesised that pronominal anaphora resolution could have a beneficial influence on the summarisation process, very few researchers have employed it to produce summaries. Often, pronominal anaphora resolution is part of a larger system which employs coreference resolution or lexical relations to produce summaries [4, 7, 9, 2]. However, these approaches do not try explicitly to assess the influence of pronominal anaphora resolution on the summarisation process. A small study on how pronominal anaphora resolution can influence a Swedish summarisation system is discussed in [19]. Manual evaluation on 10 newswire texts indicates that both the average important information in a summary and summaries' coherence improve when a pronominal anaphora resolver is used.

Orasan [36] performed a series of experiments similar to the ones presented in this paper. The results reported there suggest that anaphora resolution can help the summarisation process, but due to the small size of the corpus used in the investigation, it is difficult to make any generalisation. Steinberger et. al. [41] show how anaphora resolution can be used to improve the accuracy of a summariser based on latent semantic analysis, but they do not focus only on pronominal anaphora. As in the case of term-based summarisation, this method also uses the frequency of words to identify the important terms in a text, and uses them to extract important sentences. Evaluation on the CAST corpus [18] shows that anaphora resolution improves the results of the summariser significantly at both 15% and 30% compression rates.

# 3 Method

The method employed to produce summaries in this paper relies on terms and how they occur in sentences. The way this method works is described in Section 3.1. As already mentioned, the purpose of this paper is to assess whether the accuracy of the term-based summariser used here can be improved when it integrates an anaphora resolver. The anaphora resolvers employed in this research are described in Section 3.2, followed by the approach used to enhance the term-based summariser in Section 3.3.

## 3.1 Term-based summarisation

Term-based summarisation assumes that the importance of a sentence can be determined on the basis of the importance of the words it contains. To achieve this, each word is scored using term-weighting measures and then used to determine importance of sentences. The most common measures used to score each word are term frequency and TF\*IDF. Moreover,

evaluation of several term-weighting measures on the corpus used in this paper reveals that term frequency and TF\*IDF are the most appropriate ones [37].

Term frequency (TF) assigns to each word a score equal to its frequency in order to indicate the topicality of the concept represented by it. The main drawback of this method is that it wrongly assigns high scores to frequent tokens such as prepositions and articles. For this reason, a stoplist is used to filter out such words.

$$TF(w), = \text{the frequency of word } w \quad (1)$$

The words awarded high scores by term frequency are not necessary the most indicative of the importance of a sentence. There are open class words which appear frequently in a document but are not good indicators of the topicality of a sentence. This normally happens with words that occur frequently not only in the document, but also in a collection of documents. *Inverse document frequency* addresses this problem by measuring the importance of a word in report to how many documents from a collection contain it, and assigning it a score inversely proportionate to the number of documents which include it. This means that words appearing in many documents will not be awarded a high score. Because document frequency is too weak to be used on its own as a scoring method, it is usually combined with term frequency. The formula used in this paper is:

$$TF * IDF(w) = TF(w) * \log \frac{N}{n_w} \quad (2)$$

where  $N$  is the number of documents in the collection, and  $n_w$  is the number of documents in the collection which contain the word  $w$ . As in the case of term frequency, it was noticed that the performance of a term-based summariser increases when a stoplist is used to filter out stopwords, even though these words obtain low scores.

### 3.2 Anaphora resolution

For the experiments described in this section the Anaphora Resolution Workbench “a parameter-driven environment for consistent evaluation of anaphora resolution” [6] was used. This environment implements several well-known anaphora resolution algorithms and enables comparison between them on the basis of the same preprocessing tools and data. In this section, the knowledge-poor methods and the three baselines implemented in this environment were used. These methods are briefly explained next.

**Kennedy & Boguraev (K&B):** The anaphora resolution method proposed by Kennedy and Boguraev [22] adapts the method proposed by Lappin and Leass [24] so it can be run without a parser, and extends it with several other factors. The K&B algorithm resolves third person pronouns with noun phrase antecedents by employing a set of ten salience preferences which rank candidates for antecedents. Each preference has an initial weight which is used to build coreference classes that contain pronouns and their antecedents. Kennedy and Boguraev [22]

reports that the algorithm was evaluated on a corpus containing 306 pronouns and the observed accuracy was around 75%.

**CogNIAC:** is a high precision anaphora resolution algorithm which can resolve a subset of anaphors that do not require world knowledge or sophisticated linguistic processing [3]. The algorithm relies on six highly accurate rules to select the antecedent of a pronoun. Because the rules apply to only some of the pronouns, the original version of the algorithm was extended to include two more rules which allow it to operate in robust mode (i.e. it attempts to solve every single anaphor). The robust algorithm achieved 77.9% accuracy on the MUC-6 corpus, whilst the high accuracy non-robust algorithm achieves 92% precision and 64% recall, but it resolves only some pronouns.

**MARS:** is a robust anaphora resolution method which relies on a set of boosting and impeding indicators to select the antecedent [30]. The algorithm assigns scores to each candidate using the indicators, and the candidate with the highest aggregate score is selected as the antecedent for a pronoun. The method was evaluated on technical manuals and a hand-simulated evaluation reported results over 80%. The method used in this paper implements the original algorithm which does not include the extensions proposed in [32] or a pleonastic pronoun recogniser.<sup>2</sup>

**Baselines:** In order to have a clear idea of how effective the anaphora resolution methods are, three baseline methods were used: **BLAST** selects the closest candidate which agrees in gender and number with the anaphor; **BLASTSUBJ** selects the most recent subject which agrees in gender and number with the anaphor; and **BRAND** randomly selects an antecedent which agrees in gender and number with the anaphor from the list of candidates.

### 3.3 Enhanced term-based summarisation

The term-based summariser described in Section 3.1 relies on word frequencies to calculate the score of a word. Because some of these words are referred to by pronouns, the frequencies of the concepts they represent are not correctly calculated. The enhanced term-based summarisation method takes the output of the anaphora resolver and increases the frequencies of words referred to by pronouns, thereby producing more accurate frequency counts. Section 5.2 evaluates the improved term-based summarisation method.

## 4 Corpus

For the experiments described in this paper, a corpus of journal articles published in the Journal of Artificial Intelligence Research (JAIR) was built. The corpus used here contains 65 texts with over 600,000 words in total. In order to assemble this corpus, electronic versions of the texts have been downloaded and

<sup>2</sup> This is due to the way the Anaphora Resolution Workbench implements MARS.

converted to plain text. As the conversion was not perfect due to the presence of equations, formulae and other types of special formatting in the source, the resulting files were passed through a series of filters, which cleaned wrongly converted parts of the text and marked special information such as equations, tables, figures, footnotes and headers.

For the purpose of automatic summarisation, the corpus was automatically annotated with sentence boundaries, token boundaries and part-of-speech information using the FDG tagger [42]. In order to evaluate the performance of the automatic summarisation methods the author produced abstract was identified and extracted from the article.

A third of the corpus was also annotated with coreference information in order to evaluate the anaphora resolution methods used in this paper and for the experiment presented in Section 5.3. The difficulty of the annotation task and amount of time required to annotate a text made it impossible to apply the annotation to a larger part of the corpus. The annotation guidelines used for this purpose were derived from those proposed in [33], but instead of marking full coreferential chains, only parts of the coreference chains which contain nominal anaphoric pronouns were annotated. Therefore, if a chain did not contain a pronoun it was completely ignored. The annotation was applied using PALinkA [35], a multi-purpose annotation tool.

The annotation process first involved the automatic identification of all personal, reflexive and possessive pronouns, and annotation of these pronouns as potentially anaphoric. After this, each annotated pronoun was manually checked to see whether it was really referential, and that its antecedent was one or several NPs. For referential pronouns with NP antecedents, all the antecedents from the current paragraph and the most recent heading were identified. The reason for restricting the annotation only to these antecedents was due to the fact that all the anaphora resolution methods used here identify antecedents only from the current paragraph or the most recent heading, and therefore for the current investigation annotation of full coreferential chains would have been unnecessary. The corpus contains a total of 1873 referential pronouns, the vast majority are personal pronouns (1324), followed by possessive pronouns (502), with only a negligible number of pronouns being reflexive (47). The majority of referential pronouns are represented by different forms of the *it* pronoun, followed by different forms of *they* pronouns. As expected, the pronouns *he* and *she* have a very low frequency in the corpus.

## 5 Evaluation and discussion

In order to evaluate the effectiveness of anaphora resolution for automatic summarisation, the author produced summaries were considered the gold standard and the automatic summaries were compared to them. The measure used for computing the informativeness of an automatic summary is the cosine similarity between it and the author produced summary as proposed by Donaway et. al. [13]. Before

	Average	Standard deviation
MARS	0.512	0.080
K&B	0.448	0.088
BLAST	0.307	0.077
BRAND	0.166	0.065
BLASTSUBJ	0.115	0.051
CogNIAC	0.084	0.038

**Table 1:** The average success rate obtained by different anaphora resolution methods

computing the similarity, stopwords were eliminated.

For each of the 65 texts in our corpus summaries of 2%, 3%, 5%, 6% and 10% compression rates were produced. The reason for producing summaries of so many compression rates was to determine whether anaphora resolution influences the term-based summarisation method differently when it produces summaries of different lengths. Moreover, as can be seen in Section 5.2, the two term weighting methods investigated here lead to different results depending on the compression rate used.

In the rest of this section, the accuracy of the anaphora resolution methods employed here is first assessed to find out which one leads to the best results. After that, Section 5.2 investigates whether anaphora resolution can help term-based summarisation. The section finishes with an experiment where a simulated anaphora resolution system with predefined accuracy is used. The purpose of this experiment is to get further insights into how anaphora resolution can help term-based summarisation.

### 5.1 Evaluation of anaphora resolution

All anaphora resolution methods used in this paper are robust.<sup>3</sup> For this reason, the only measure used in the evaluation is *success rate*, computed as the number of correctly resolved anaphors divided by the number of anaphors identified by the system [31]. Table 1 contains the average success rate obtained by different anaphora resolution methods on the coreferentially annotated corpus.

The success rate of all the methods evaluated here is much lower than that reported by their authors. There are two justifications for this. First, the evaluation performed by the authors was an evaluation of the algorithm and not an evaluation of a practical system. This means that the algorithms were either hand-simulated or they processed manually prepared data. In contrast, the evaluation presented here was fully automatic and the systems had to deal with errors introduced by preprocessing steps such as part-of-speech tagging and NP extraction. The second reason for obtaining lower results is that the anaphora resolution methods used here were not designed to deal with texts from the scientific domain: MARS was developed for the technical domain and includes indicators specific for this domain, CogNIAC was tested on the MUC-6 texts, and K&B was evaluated

<sup>3</sup> A system is considered robust if it tries to resolve all the pronouns which are anaphoric. Some systems such as the non-robust version of CogNIAC resolve only a part of these pronouns because of the way they were designed.

	2%	3%	5%	6%	10%
TF					
No anaphora	0.415	0.443	0.461	0.467	0.484
Blast	0.451	0.476	0.495	0.498	0.511
Blastsubj	0.458	0.481	0.493	0.501	0.514
Brand	0.457	0.478	0.494	0.499	0.512
CogNIAC	0.454	0.478	0.495	0.500	0.512
K&B	0.454	0.481	0.494	0.500	0.511
MARS	0.455	0.480	0.494	0.500	0.513
Perfect	0.512	0.525	0.551	0.555	0.561
TF*IDF					
No anaphora	0.396	0.427	0.467	0.472	0.496
Blast	0.430	0.463	0.497	0.503	0.519
Blastsubj	0.431	0.464	0.496	0.502	0.520
Brand	0.431	0.463	0.498	0.501	0.521
CogNIAC	0.433	0.463	0.497	0.503	0.520
K&B	0.428	0.461	0.498	0.502	0.520
MARS	0.428	0.463	0.499	0.503	0.520
Perfect	0.455	0.500	0.531	0.536	0.540

**Table 2:** The average informativeness of summaries produced by the improved summarisation method

on a random selection of texts, none of which seem to be from the scientific domain.

According to the results in Table 1, the best method is MARS, followed K&B and BLAST, with CogNIAC performing the worst. BLAST, the baseline which selects the most recent candidate agreeing in gender and number with the pronoun obtains the best results among baselines. For all the methods, the differences are statistically significant with a 0.01 confidence level. Next section investigates whether the differences between the accuracy of different anaphora resolution methods are reflected in differences between the informativeness of summaries produced by the term-based summariser which uses their output.

## 5.2 Evaluation of enhanced summarisation method

As already mentioned, it is assumed that term-based summarisers do not achieve a very high performance because they ignore the fact that some words are referred to by pronouns, and therefore their frequency is not accurately computed. In this section, the three anaphora resolution methods and three baselines evaluated in the previous section are incorporated into the term-based summariser.

The results of the term-based summarisation methods augmented with information from anaphora resolvers are presented in Table 2. The row labeled *No anaphora* indicates the informativeness of summaries when no information from an anaphora resolver is incorporated in the term-based summariser, whilst the row *Perfect* corresponds to an anaphora resolver with a success rate of 100%. The results in the *Perfect* row were obtained only for the texts annotated with coreference information because the manual annotation is considered to be the output of a perfect anaphora resolver.<sup>4</sup> The rest of the rows indicate the informativeness of the summary when an automatic anaphora resolver was integrated into the system and were calculated on the whole corpus.

<sup>4</sup> We acknowledge the fact that errors in the annotation can limit the degree of ‘perfectness’ of the output.

As can be seen in the table, in the cases where no anaphora resolution is used, term frequency is the best term-weighting method for 2% and 3% summaries, whereas TF\*IDF is better for 5%, 6% and 10% summaries. This pattern still holds when the anaphora resolution methods are incorporated in the summarisation program, but the differences are negligible. If the output of a perfect anaphora resolver is used by the term-weighting method the term frequency leads to the best results for all compression rates. The results in Table 2 clearly indicate that a perfect anaphora resolver significantly improves the accuracy of a summarisation method.

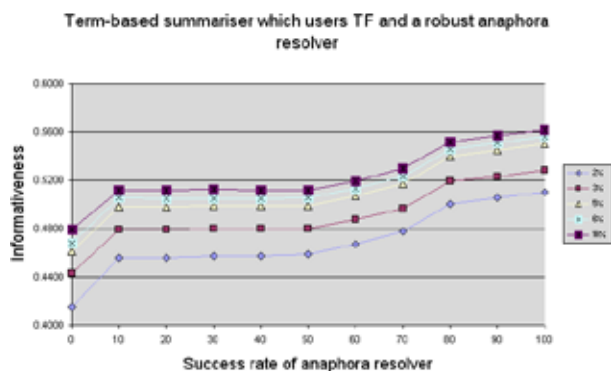
Closer investigation of the results reveals that there is no correlation between the accuracy of the automatic anaphora resolvers investigated here and the informativeness of the produced summaries. All the summaries seem to contain more or less the same amount of information, the differences between the quantity of information not being statistically significant. Moreover, the incorporation of the best anaphora resolution method (i.e. MARS) in the summariser leads to the best results in only two cases, both for TF\*IDF. CogNIAC, the worst anaphora resolver, also leads to the best results in 2 cases.

One possible explanation for this result is that the anaphora resolvers used in this research are not accurate enough to really have a beneficial effect on term-based summarisation, and that perhaps anaphora resolvers with higher accuracies would lead to clearer improvements of the informativeness of summaries produced. In light of this, an experiment with a simulated anaphora resolver is presented in the next section.

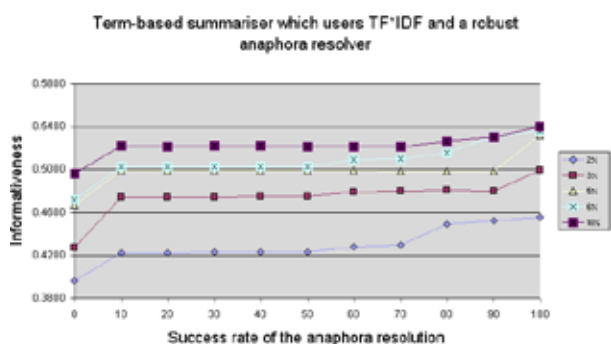
## 5.3 Robust anaphora resolver with predefined accuracy

Evaluation of the anaphora resolvers presented in Section 3.2 showed that they often resolve pronouns to the wrong antecedent. As a result, some concepts have their scores wrongly increased. The anaphora resolver simulated in this section tries to perform in the same manner as the automatic anaphora resolvers investigated in Section 3.2, by boosting the frequency scores of both correct and incorrect antecedents, but it is designed in such a way that its success rate can be controlled. For this experiment, the success rate of this resolver was increased from 10% to 100% in 10% increments. In order to achieve this, a predefined percentage of correct (pronoun, antecedent) pairs were selected from each text. For the rest of the pronouns, wrong antecedents were selected in order to introduce errors. This process was repeated 100 times for each text and for each success rate value to ensure fairness and reliability of the experiment. The manual annotation was used to simulate this anaphora resolver, and so the experiment was carried out only on the coreferentially annotated texts. Figures 1 and 2 present the results of the experiment.

The results of these experiments are in line with the results reported in Section 5.2, but still contain some unexpected features because they show that even if only 10% of the pronouns are correctly resolved, the



**Fig. 1:** The informativeness of automatic summaries produced using TF and an anaphora resolver with variable success rate



**Fig. 2:** The informativeness of automatic summaries produced using TF\*IDF and an anaphora resolver with variable success rate

results of the automatic summariser are significantly better. The next significant improvement is obtained only for an anaphora resolver which achieves at least 60% success rate and is used with term frequency. For the automatic summariser which uses TF\*IDF, it is necessary to have an anaphora resolver which achieves around 80%-90% success rate to have a noticeable improvement.

## 6 Conclusions

This paper has investigated the influence of pronominal anaphora resolution on term-based summarisation. The underlying hypothesis was that by incorporating an anaphora resolver into the term-weighting process it is possible to obtain more accurate frequency counts of concepts referred to by pronouns. To this end, three robust anaphora resolvers and three baselines were incorporated into two term-weighting measures, which were in turn used by a term-based summariser. Comparison of the informativeness of summaries produced by this improved term-based summariser revealed that there is no correlation between the informativeness of a summary and the performance of the anaphora resolver used to improve the frequency counts. Despite this, the results clearly

indicate that the summarisation process benefits from anaphora resolution.

The beneficial influence of anaphora resolution on term-based summarisation was further investigated by performing an experiment with a simulated anaphora resolver with controlled success rate. The results of the experiment show that due to the increase of scores for both correct and incorrect antecedents, a significant improvement of the summaries' informativeness is noticed only when accuracy of the resolver is between 60% and 80%, depending on the term-weighting method. This explains why no difference was observed for the relatively poor performance of anaphora resolvers investigated here.

The integration of an anaphora resolver into the term-based summariser also reveals some interesting results. Without anaphora resolution, term frequency leads to the best results only for 2% and 3% compression rates. Once an automatic anaphora resolution is integrated into the term-based summariser the differences between summaries produced using term frequency and those produced using TF\*IDF at 5%, 6% and 10% become negligible. Moreover, if a perfect anaphora resolver is used, the summariser which uses term frequency always performs significantly better than the summariser which uses TF\*IDF.

This paper has focused only on how pronominal anaphora resolvers can be used in the summarisation process. For the future it would be interesting to extend this research to other types of anaphoric expressions such as definite descriptions. Another interesting development of this paper would be to use other evaluation methods such as ROUGE [26] for measuring the informativeness of summaries in order to find out whether the findings change. The summarisation methods employed in this paper are rather simple and do not necessarily reflect the state of the art of summarisation methods. In the future it is planned to evaluate the influence of pronominal anaphora resolution on other summarisation methods which could benefit from this information.

## References

- [1] C. Aone and S. W. Bennett. Evaluating automated and manual acquisition of anaphora resolution rules. In *Proceedings of the 33th Annual Meeting of the Association for Computational Linguistics (ACL '95)*, pages 122 – 129, 1995.
- [2] S. Azzam, K. Humphrey, and R. Gaizauskas. Using coreference chains for text summarisation. In A. Bagga, B. Baldwin, and S. Shelton, editors, *Coreference and Its Applications*, pages 77 – 84, University of Maryland, College Park, Maryland, USA, June 1999.
- [3] B. Baldwin. CogNIAC: High precision coreference with limited knowledge and linguistic resources. In R. Mitkov and B. Boguraev, editors, *Proceedings of Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 38 – 45, 1997.
- [4] B. Baldwin and T. S. Morton. Dynamic coreference-based summarization. In *Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing (EMNLP-3)*, Granada, Spain, June 1998.
- [5] C. Barbu. Genetic algorithms in anaphora resolution. In T. M. Antonio Branco and R. Mitkov, editors, *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC2002)*, pages 7 – 12, Lisbon, Portugal, September, 18 – 20 2002.



- [6] C. Barbu and R. Mitkov. Evaluation tool for rule-based anaphora resolution methods. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 34 – 41, Toulouse, France, July 9 – 11 2001.
- [7] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In I. Mani and M. T. Maybury, editors, *Advances in Automated Text Summarization*, pages 111 – 121. The MIT Press, 1999.
- [8] P. B. Baxendale. Man-made index for technical literature - an experiment. *I.B.M. Journal of Research and Development*, 2(4):354 – 361, 1958.
- [9] B. Boguraev and C. Kennedy. Salience-based content characterisation of text documents. In I. Mani and M. T. Maybury, editors, *Advances in Automated Text Summarization*, pages 99 – 110. The MIT Press, 1999.
- [10] S. E. Brennan, M. W. Friedman, and C. J. Pollard. A centering approach to pronouns. In *Proceedings of the 25th Annual Meeting of the ACL*, pages 155 – 162, Stanford, California, June 1987.
- [11] M. Brunn, Y. Chali, and C. J. Pinchak. Text summarization using lexical chains. In *Proceedings of DUC2001 Conference*, New Orleans, Louisiana, USA, September 13 – 14 2001.
- [12] S. H. Corston-Oliver. Beyond string matching and cue phrases: Improving the efficiency and coverage in discourse analysis. In *AAAI Spring Symposium on Intelligent Text Summarisation*, pages 9 – 15, Stanford, California, USA, March 23-25 1998.
- [13] R. L. Donaway, K. W. Drummey, and L. A. Mather. A comparison of rankings produced by summarization evaluation measures. In *Proceedings of NAACL-ANLP 2000 Workshop on Text Summarisation*, pages 69 – 78, Seattle, Washington, April 30 2000.
- [14] H. P. Edmundson. New methods in automatic extracting. *Journal of the Association for Computing Machinery*, 16(2):264 – 285, April 1969.
- [15] N. Ge, J. Hale, and E. Charniak. A statistical approach to anaphora resolution. In *Proceedings of the Sixth Workshop on Very Large Corpora, COLING-ACL '98*, pages 161 – 170, Montreal, Canada, 1998.
- [16] B. J. Grosz, A. K. Joshi, and S. Weinstein. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, 21(2):203 – 225, 1995.
- [17] M. A. K. Halliday and R. Hasan. *Cohesion in English*. English Language Series. Longman Group Ltd, 1976.
- [18] L. Hasler, C. Orăsan, and R. Mitkov. Building better corpora for summarisation. In *Proceedings of Corpus Linguistics 2003*, pages 309 – 319, Lancaster, UK, March, 28 – 31 2003.
- [19] M. Hassel. Pronominal resolution in automatic text summarisation. Master's thesis, Department of Computer and Systems Sciences, Stockholm University, 2000.
- [20] J. Hobbs. Pronoun resolution. Research report 76-1, City College, City University of New York, 1976.
- [21] J. Hobbs. Pronoun resolution. *Lingua*, 44:339–352, 1978.
- [22] C. Kennedy and B. Boguraev. Anaphora for everyone: pronominal anaphora resolution without a parser. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING'96)*, pages 113 – 118, Copenhagen, Denmark, August 05 – 09 1996.
- [23] J. Kupiec, J. Pederson, and F. Chen. A trainable document summarizer. In *Proceedings of the 18th ACM/SIGIR Annual Conference on Research and Development in Information Retrieval*, pages 68 – 73, Seattle, July 09 – 13 1995.
- [24] S. Lappin and H. J. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535 – 562, 1994.
- [25] C.-Y. Lin. Assembly of topic extraction modules in SUMMARIST. In *AAAI Spring Symposium on Intelligent Text Summarisation*, pages 44 – 50, Stanford, California, USA, March 23 – 25 1998.
- [26] C.-Y. Lin and E. H. Hovy. Automatic evaluation of summaries using n-gram co-occurrence. In *Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003)*, pages 71 – 78, Edmonton, Canada, May 27 – June 1 2003.
- [27] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159 – 165, 1958.
- [28] I. Mani. *Automatic Summarization*. Natural Language Processing. John Benjamins Publishing Company, 2001.
- [29] D. Marcu. From discourse structures to text summaries. In I. Mani and M. Maybury, editors, *Proceedings of the ACL/EACL '97 Workshop on Intelligent Scalable Text Summarization*, pages 82 – 88, Madrid, Spain, 1997. ACL.
- [30] R. Mitkov. Robust pronoun resolution with limited knowledge. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING'98/ACL'98)*, pages 867 – 875, Montreal, Quebec, Canada, August 10 - 14 1998.
- [31] R. Mitkov. *Anaphora resolution*. Longman, 2002.
- [32] R. Mitkov, R. Evans, and C. Orăsan. A new, fully automatic version of Mitkov's knowledge-poor pronoun resolution method. In *Proceedings of CICLING-2002*, pages 168 – 186, Mexico City, Mexico, February 2002.
- [33] R. Mitkov, R. Evans, C. Orăsan, C. Barbu, L. Jones, and V. Sotirova. Coreference and anaphora: developing annotating tools, annotated resources and annotation strategies. In *Proceedings of the Discourse, Anaphora and Reference Resolution Conference (DAARC2000)*, pages 49–58, Lancaster, UK, 2000.
- [34] K. Ono, K. Sumita, and S. Miike. Abstract generation based on rhetorical structure extraction. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, pages 344 – 348, Kyoto, Japan, 1994.
- [35] C. Orăsan. PALinkA: a highly customizable tool for discourse annotation. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialog*, pages 39 – 43, Sapporo, Japan, July, 5 -6 2003.
- [36] C. Orăsan. The influence of personal pronouns for automatic summarisation of scientific articles. In *Proceedings of the 5th Discourse Anaphora and Anaphor Resolution Colloquium*, pages 127 – 132, Furnas, S. Migue, Azores, Portugal, September, 23 - 24 2004.
- [37] C. Orăsan. *Comparative evaluation of modular automatic summarisation systems using CAST*. PhD thesis, University of Wolverhampton, 2006.
- [38] C. Orăsan, R. Evans, and R. Mitkov. Enhancing preference-based anaphora resolution with genetic algorithms. In *Proceedings of Natural Language Processing - NLP2000*, pages 185 – 195. Springer, 2000.
- [39] C. D. Paice. The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases. In R. N. Oddy, C. J. Rijdsbergen, and P. W. Williams, editors, *Information Retrieval Research*, pages 172 – 191. London: Butterworths, 1981.
- [40] W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(2):521 – 544, 2001.
- [41] J. Steinberger, M. A. Kabadjov, M. Poesio, and O. Sanchez-Graillet. Improving LSA-based summarization with anaphora resolution. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 1 – 8, Vancouver, Canada, October 2005.
- [42] P. Tapanainen and T. Järvinen. A non-projective dependency parser. In *Proceedings of the 5th Conference of Applied Natural Language Processing*, pages 64 – 71, Washington D.C., USA, March 31 - April 3 1997.
- [43] S. Teufel and M. Moens. Summarizing scientific articles: Experiments with relevance and rhetorical status. *Computational linguistics*, 28(4):409–445, 2002.
- [44] K. Zechner. Fast generation of abstracts from general domain text corpora by extracting relevant sentences. In *COLING - 96, The International Conference on Computational Linguistics*, pages 986 – 989, Copenhagen, Denmark, August 1996.

# Computer-aided summarisation: How much does it really help?

Constantin Orăsan and Laura Hasler  
Research Group in Computational Linguistics  
University of Wolverhampton  
Stafford St.  
Wolverhampton, WV1 1SB, UK  
*C.Orasan@wlv.ac.uk* and *L.Hasler@wlv.ac.uk*

## Abstract

Computer-aided summarisation is a technology developed as a complement to automatic summarisation, which produces high quality summaries with less effort. To achieve this, a user-friendly environment which incorporates several well-known summarisation methods has been developed. This paper presents the main features of the computer-aided summarisation environment and evaluates the usefulness of the developed tool. Experiments showed that it is possible to reduce the time necessary to produce the summary by about 20% without any degradation in the summary's quality.

## Keywords

computer-aided summarisation, automatic summarisation, evaluation

## 1 Introduction

Automatic summarisation systems help us deal with the current information overload by reducing it. Unfortunately, despite extensive work in this field, current technology is still not capable of creating human-like summaries. Instead, it usually produces *indicative summaries*, which allow a reader to get a quick gist of the document. The drawback of indicative summaries is that they are quite brief and do not allow the user to explore the structure of the source in more detail or to be used instead of the source, as is possible with some human-produced summaries. In the cases where *informative summaries*, which can replace the source, need to be produced, automatic summarisation does not appear to offer a viable solution yet. As a result, such summaries need to be produced by humans which makes their costs high.

In light of this problem, we propose computer-aided summarisation (CAS) as a complementary approach to automatic summarisation. Whereas automatic summarisation does not require any human input to produce summaries, we argue that computer-aided summarisation is a more feasible approach as it allows the user to post-edit the automatic summaries according to their requirements, resulting in better finished products. CAS is a technology developed

at the University of Wolverhampton designed to help humans produce high quality summaries with less effort, in this way also lowering the costs.

The structure of the paper is as follows: the paper starts with a description of the computer-aided summarisation concept. Section 3 briefly presents the computer-aided summarisation tool we developed, whilst Section 4 describes the experiments carried out to prove the usefulness of the computer-aided summarisation concept. The paper finishes with a review of related work in the field of computer-aided language processing, followed by conclusions.

## 2 The computer-aided summarisation concept

The concept of computer-aided summarisation was inspired by the machine-aided translation approach suggested in 1980 by Martin Kay. Kay [8, 9] proposed the development of *cooperative man-machine systems* as a solution to the unrealistic task of fully automatic high quality translation, allowing the computer and the human translator to perform the translation tasks they are best at. CAS aims to help human summarisers by selecting the important information from a document and presenting it to them, and leaving the task of linking sentences to form a coherent abstract to the summariser. The main advantage of such an approach is that the summariser does not need to read the whole text, instead being presented with only the important parts of the document as a starting point for their summary.

The feasibility of the computer-aided approach is confirmed by research into the human summarisation process. The work of Endres-Niggemeyer [3] provides the theoretical grounding for the idea of human post-editing in computer-aided summarisation in terms of her three-stage human summarisation model of *document exploration*, *relevance assessment* and *summary production*. The first stage, document exploration, involves the summariser exploring the layout and organisation of the document to locate important information. During the next stage, relevance assessment, the summariser assesses information in the document to see if it is relevant to the summary. The final stage of summary production is where the actual creation of the summary as a

unit in itself takes place, and mainly involves cutting and pasting material from the original document using sentence patterns typical of the domain.

The first two stages of Endres-Niggemeyer's model correspond to the automatic summarisation in CAS, which uses automatic methods to identify important information in the text and present these, either in the form of a summary or as highlighted units within the full text, to the user. The third stage, which in Endres-Niggemeyer's analysis involves cutting and pasting operations and reorganising of the text, corresponds to the human summariser's post-editing of the summary, by accepting, rejecting and reorganising the information proposed by our computer-aided summarisation tool (presented in Section 3).

One could argue that for some domains, automatic summarisation methods still perform poorly and are likely to miss important information in the source. However, even when this is the case, computer-aided summarisation can still be useful. In domains where important information cannot be selected reliably, summarisers can use the automatic methods to produce a summary much larger than is actually required, and use this as a starting point, reducing it until the target length is reached. This means that the user will still need to read less text than if they produced a summary manually using the full source text, thereby speeding up the process. As an alternative, the user can choose to use automatic methods which remove unimportant sentences from the full text instead of selecting important ones. Again, this option reduces the length of the document to be read before the user can produce the summary. Because summaries are produced in a computer-aided environment rather than a fully automatic one, the user always has the option to return to the full text to get more information or to clarify uncertainties, and to over-ride the system's decisions.

### 3 The computer-aided summarisation tool (CAST)

As mentioned above, computer-aided summarisation is seen here as a complement to existing automatic summarisation techniques, as it allows human intervention in the summarisation process. However, in order to make the approach worthwhile, this intervention should be minimal, so that the effort required for a human to produce the summary using CAS is significantly less than that required to write a summary without the help of an advanced tool. To achieve this, several automatic summarisation techniques which have been extensively used were implemented in CAST. The purpose of these methods within CAST is to present to the user an extract which contains the most important sentences from a text, allowing them to post-edit it in order to improve its quality. As not all the sentences identified automatically will be worth including in a summary, the user has the option to override the program's decisions and delete irrelevant sentences, as well as to extract additional sentences.

After careful consideration of the existing automatic

summarisation methods commonly used to produce extracts, we decided to implement the following methods: term-based summarisation methods, methods based on indicating phrases, surface clues, and discourse information. The term-based summarisation methods assume that the importance of a sentence can be determined on the basis of the words it contains. Indicating phrases are phrases such as *in this paper, we conclude that* which are specific to a domain and normally indicate the important sentences [16]. Surface clues can also help the summarisation process by assuming that words in titles and headings are more important than the rest, whilst text in brackets can usually be discarded. Finally, the discourse structure of the text can also be utilised. In CAST, this information is exploited in the form of lexical chains which are used to determine links between sentences [6]. A more detailed description of these methods can be found in [15]. Given that each method depends on a host of parameters, we offer users a high level of flexibility without compromising the simplicity of the tool by giving them the option to adjust all these parameters in a user friendly way.

The automatic methods embedded in the tool are used not only to identify important sentences in a text, but also to remove sentences which do not contain important information. For example, as well as extracting sentences containing certain indicating phrases or having their term-based score above a certain threshold, it is also possible to remove sentences which contain certain indicating phrases or have a term-based score lower than a given threshold. As with the case of important sentences, the user can review the system's decisions over-riding it whenever the decision is wrong.

The results of the summarisation methods can be viewed in different ways, depending on the user's preferences. They can be viewed either in isolation, when the results are presented as an automatic extract, or the sentences extracted can be highlighted within the source text using formatting defined by the user. The advantage of highlighting the results in the text is that the user can easily see the sentences in their original context. Given the friendly graphical interface available to the user and the different styles which can be defined for each method, the user can quickly identify sentences selected by different methods. A screenshot of the tool is presented in Figure 1.

Once a user decides that a sentence is important enough to be included in a summary (either indicated by the program or on the basis of their understanding), it can be copied into the summary window at the bottom of the tool and edited. In order to facilitate the editing task further, a common set of errors such as dangling pronouns and phrases which could indicate a problem with the summary (e.g. "on the other hand", "secondly", etc.) are highlighted to draw attention to them.

## 4 Evaluation

CAST is intended to help human summarisers to produce abstracts. To assess the extent to which

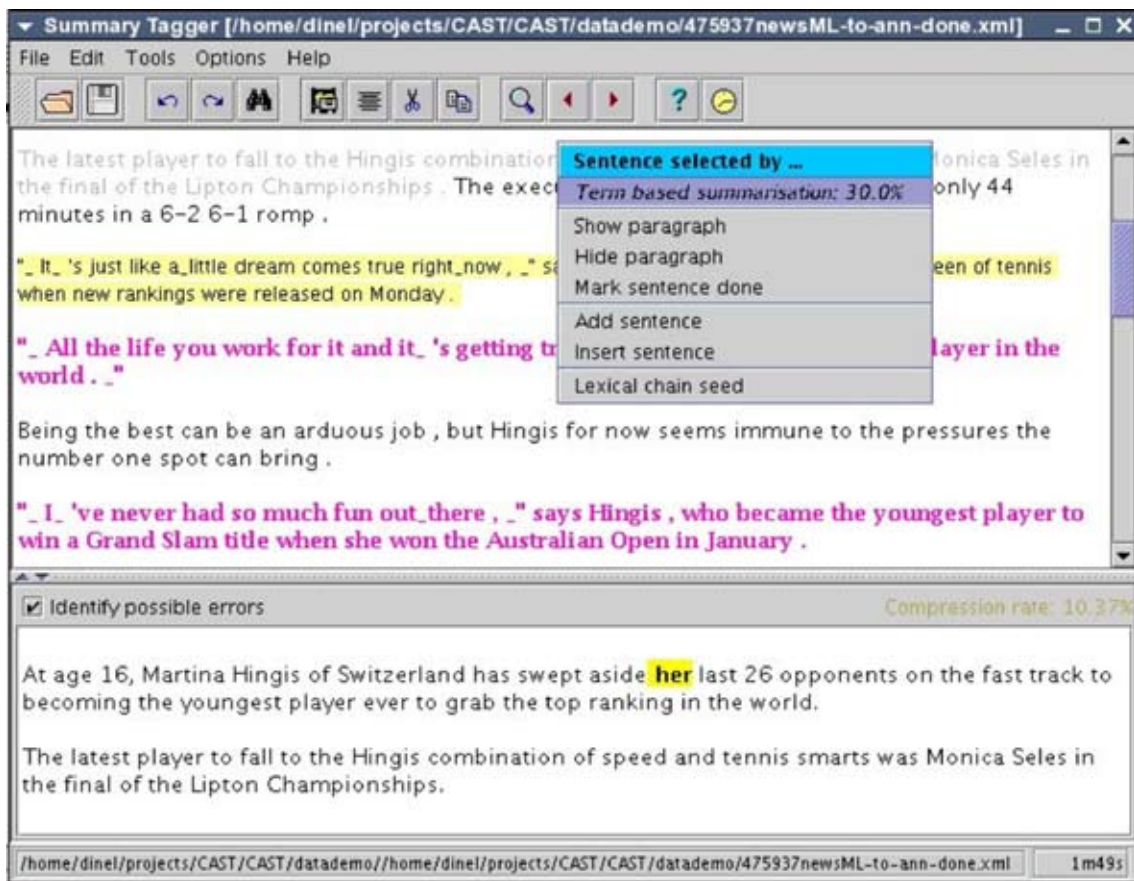


Fig. 1: Screenshot of the program

this is achieved, a professional human summariser was asked to use the tool and provide feedback about its usability. On the basis of this feedback, the tool was improved and we learnt how it is used by professional summarisers. These findings are presented in [14]. In this paper, we conjecture that summaries produced with the help of the tool will be as good as those manually produced, but that it will take less time to write them. To prove this hypothesis about the usefulness of the tool, two experiments were conducted. Their results are presented in Sections 4.1 and 4.2.

#### 4.1 Evaluation of the time

As already mentioned, our assumption in the first experiment is that CAS can reduce the time necessary to produce summaries. To this end, a two stage experiment was conducted using a professional summariser. In the first stage, he was asked to produce summaries using a simple interface which records the time necessary to produce a summary, but does not provide any help with the summarisation process. In the second stage of the experiment, the same summariser was asked to produce abstracts of the same texts using the CAST in order to see whether the time necessary to produce them was reduced. The second stage of the experiment occurred one year after the first stage so that any effect of text familiarity was

extinguished.

For this experiment, a total of 69 texts extracted from the CAST corpus [4] were summarised. Fifty four of these texts were newswire texts extracted from the Reuters corpus and fifteen were articles from New Scientist. The newswire texts contain on average 800 words whilst for the New Scientist texts the average number of words is 1750. These texts were selected for two reasons. First, they were previously annotated with information regarding the importance of sentences which allowed us to assess the accuracy of the automatic summarisation methods used by the human summariser. Secondly, the texts summarised are old enough (published around 1997) to ensure that the summariser is not very familiar with the topic discussed in the texts so that the first stage of the experiment is not unfairly helped by the summariser's background knowledge.

For both stages the professional summariser was asked to produce 20% summaries of each text. Table 1 shows the average number of seconds necessary to summarise a text with and without CAST. As expected, it takes longer to summarise the texts from New Scientist than the newswire ones due to the fact that they are almost twice in length. The table also indicates that by using CAST the time necessary to produce a summary reduces by almost 2 minutes for newswire texts and almost 2 minutes and a half for those from New Scientist. In both case the reduction

	Without CAST	With CAST	Percentage of reduction
Newswire texts	498sec	382sec	23.29%
New Scientist texts	771sec	623sec	19.19%

**Table 1:** *The time necessary to produce summaries with and without CAST*

is statistically significant.

The feedback from the summariser indicated that he first preferred to run the term-based summariser to highlight a set of sentences which could be useful for the abstract he produced. Although he was asked to produce a 20% abstract, he decided to use a 30% automatic summary as the starting point. This allowed him to see a wider selection of important sentences so that no crucial information was missed [14]. Because of the importance given to term-based summarisation by our user, we assessed its performance to establish whether there is any correlation between the accuracy of the automatic method and the time reduction. For evaluation, we used precision, recall and f-measure because the texts included in this experiment were extracted from the CAST corpus [4], a corpus which is annotated with information about the importance of sentences.

Table 2 presents the results of the evaluation. As can be seen, term-based summarisation performs significantly better on the newswire texts than on those from New Scientist. Moreover, a correlation between the reduction in the time necessary to produce the summary and the accuracy of the automatic methods was noticed. In light of this, it can be concluded that term-based summarisation methods included in CAST really help the summarisation process.

The user also at times used lexical chains to determine sentences related to those he considered important. However, this method was run only on an ad-hoc basis and therefore it is not possible to obtain figures about how useful it was. The same applies to the other summarisation methods incorporated, which were used even less often.

## 4.2 Quality of the summaries

Our second assumption was that the reduction in the time necessary to produce the summaries does not have a detrimental influence on their quality. To this end, we conducted a Turing-like test where pairs of summaries produced with and without CAST were shown to judges who were asked to select the best one in the pair. For this experiment, our hypothesis was that there are no significant differences between the two types of summaries and that human judges will not be able to make a reliable distinction between them.

For this experiment, 17 judges were shown four pairs of summaries each. The summaries were randomly selected from all the summaries produced in the first experiment. The order in which they were displayed in the pair was also random to avoid situations where one judge always selects the same element of the pair. Our judges included undergraduate and post-graduate students as well as members of staff, who were not given any instructions except that they should indicate

which summary is better on the basis of their intuition.

Analysis of the results revealed that in 41 pairs the judges preferred summaries produced using CAST, whereas in 27 those produced without CAST were considered better. In order to see whether this difference is significant we calculated chi-square between the observed judgements and the expected judgements according to our hypothesis (i.e. that the votes are equally distributed between the two classes which means that each class gets 34 votes). The chi-square test revealed that there is no statistical difference at 0.05 level which indicates that there is no difference between the quality of the two types of summaries. Despite this, the results indicate that there is a slight preference towards the summaries produced using CAST.

## 5 Related work

This section presents related work in the field of computer-aided summarisation, but does not try to review existing work in automatic summarisation because it is considered to be beyond the scope of this paper. Good sources of more information about automatic summarisation are [10, 7].

Work related to CAS is relatively sparse in comparison with computer-aided approaches used in other areas such as machine translation and computer-aided language learning. It was also proved to be useful in other areas. Mitkov and Ha has showed that the time taken to generate multiple-choice questions was reduced by 75% when a computer-aided approach was used instead of a manual one, with no decrease in quality [12]. Semi-automatic annotation methods can also speed up the production of annotated corpora [5]. Whilst the idea of some form of automated help for human summarisers may have been around for some time [11, 1, 13], the more specific notion of CAS which combines automatic extracting and human post-editing, has only recently been explored in more depth [15, 14].

Craven [2] focuses on the automatic extraction of keywords and phrases from documents which could be useful when presented to a human trying to summarise the document. He argues that even this simple automatic assistance can help humans produce summaries of a text more easily than they would have done otherwise. The abstracting tool presented by Narita [13] aims to improve summaries of research papers in the field of information engineering written in English by Japanese software engineers who are intermediate or advanced learners of English. The tool provides an organisational template for the human abstractor to flesh out with their own material, helping them in the process by providing examples from a corpus. As with Craven's work, no automatic summarisation methods are employed; instead the tool

	Precision	Recall	F-measure
Newswire texts	44.19%	48.66%	46.32%
New Scientist texts	32.26%	34.05%	33.14%

**Table 2:** *The accuracy of the automatic summarisation method*

accesses a corpus of human-produced abstracts which have been analysed for their rhetorical structure.

In a working paper in 1995, Mitkov described plans to develop a “computer-assisted and user-friendly abstracting tool” [11] which identifies and highlights sentences considered to be important in terms of content for the user. Once the computer has performed this task, the human abstractor accepts or rejects the selected sentences as they see fit, and perhaps adds new sentences, before connecting the text together into cohesive paragraphs. Mitkov terms this approach semi-automatic and argues that it will make abstracting faster and cheaper as it does not rely on fully human summarisation which is time-consuming and labour-intensive. It is Mitkov’s work which provided the basic idea for the CAST system presented in this paper.

## 6 Concluding remarks

Computer-aided summarisation was proposed as a complementary approach to automatic summarisation and a solution to producing high quality summaries at lower costs. This paper presented two experiments which prove the validity of the computer-aided summarisation concept. In the first experiment, a professional summariser produced summaries with and without the computer-aided summarisation tool. A comparison between the time necessary to produce the summaries revealed that the time is reduced by approximately 20% when CAST is used. A second experiment was carried out to determine whether the reduction in time had any negative influence on the quality of the summaries. The results of this experiment clearly indicate that there is no statistically significant difference between the two types of summaries in terms of quality. However, judges demonstrated a slight preference towards summaries produced using CAST. This is an unexpectedly good result which needs to be investigated further.

In the first experiment presented in this paper only one professional summariser was used. The reason for this is the high cost of employing such users and their limited availability. In the future, we intend to repeat the experiment using more summarisers, including non-professionals to see whether this confirms the results of our experiments.

## 7 Acknowledgments

This research was partially funded by the Arts and Humanities Research Board through the “Computer Aided Summarisation Tool - CAST” project.<sup>1</sup> We would also like to thank the professional summariser

who participated in the first experiment and all the judges involved in the second experiment.

## References

- [1] T. C. Craven. An experiment in the use of tools for computer-assisted abstracting. In *Proceedings of the ASIS 1996*, Baltimore, MD, United States, 19 - 24 October 1996.
- [2] T. C. Craven. Abstracts produced using computer assistance. *Journal of the American Society for Information Science*, 51(8):745 - 756, 2000.
- [3] B. Endres-Niggemeyer. *Summarizing information*. Springer, 1998.
- [4] L. Hasler, C. Orăsan, and R. Mitkov. Building better corpora for summarisation. In *Proceedings of Corpus Linguistics 2003*, pages 309 - 319, Lancaster, UK, March, 28 - 31 2003.
- [5] L. Hasler, C. Orăsan, and K. Naumann. NPs for Events: Experiments in Coreference Annotation. In *Proceedings of the 5th edition of the International Conference on Language Resources and Evaluation (LREC2006)*, pages 1167 - 1172, Genoa, Italy, 24 - 26 May 2006.
- [6] E. Hovy. Approaches to the planning of coherent text. In C. L. Paris, W. R. Swartout, and W. C. Mann, editors, *Natural Language in Artificial Intelligence and Computational Linguistics*. Kluwer Academic Publishers, 1990.
- [7] E. Hovy. Text summarisation. In R. Mitkov, editor, *The Oxford Handbook of computational linguistics*, pages 583 - 598. Oxford University Press, 2003.
- [8] M. Kay. The proper place of men and machines in language translation. Technical Report CSL-80-11, Xerox PARC, Palo Alto, California, 1980.
- [9] M. Kay. The proper place of man and machines in language translation. *Machine translation*, 12(1):3 - 23, 1997.
- [10] I. Mani. *Automatic Summarization*. Natural Language Processing. John Benjamins Publishing Company, 2001.
- [11] R. Mitkov. A breakthrough in automatic abstracting: the corpus-based approach. Technical report, University of Wolverhampton, 1995.
- [12] R. Mitkov and L. A. Ha. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 2003 Workshop on Building Educational Applications Using Natural Language Processing*, pages 17 - 22, Edmonton, Canada, May 2003.
- [13] M. Narita. Constructing a tagged E-J parallel corpus for assisting Japanese software engineers in writing English abstracts. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, pages 1187 - 1191, Athens, Greece, 31 May - 2 June 2000.
- [14] C. Orăsan and L. Hasler. Computer-aided summarisation - what the user really wants. In *Proceedings of the 5th edition of the International Conference on Language Resources and Evaluation (LREC2006)*, Genoa, Italy, 24 - 26 May 2006.
- [15] C. Orăsan, R. Mitkov, and L. Hasler. CAST: a Computer-Aided Summarisation Tool. In *Proceedings of EACL2003*, pages 135 - 138, Budapest, Hungary, April 2003.
- [16] C. D. Paice. The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases. In R. N. Oddy, C. J. Rijsbergen, and P. W. Williams, editors, *Information Retrieval Research*, pages 172 - 191. London: Butterworths, 1981.

<sup>1</sup> The project’s webpage is <http://clg.wlv.ac.uk/projects/CAST/>

# Multi-document Summarization Focusing on Extracting and Integrating Similarities and Differences among Documents

Shiyan Ou

Research Group in Computational Linguistics  
University of Wolverhampton  
Wolverhampton, WV1 1SB  
United Kingdom  
Shiyan.Ou@wlv.ac.uk

Christopher S. G. Khoo

Division of Information Studies  
School of Communication and Information  
Nanyang Technological University  
Singapore, 637718  
assgkhoo@ntu.edu.sg

Dion H. Goh

Division of Information Studies  
School of Communication and Information  
Nanyang Technological University  
Singapore, 637718  
ashlgoh@ntu.edu.sg

## Abstract

The study was to develop a method for automatic summarization of sets of related research abstracts. This summarization method focused on extracting and integrating similarities and differences among different abstracts. In the research studies which aim to look for relationships between research concepts, similarities and differences are mainly reflected through research concepts and relationships expressed in the text. Thus the summarization method extracts research concepts and relationships from each research abstract, integrates similar concepts and relationships across different abstracts, and incorporates them into new sentences to produce a summary. This paper reports the three main summarization steps – discourse parsing, concept extraction and integration, and relationship extraction and integration. Each step was evaluated by comparing the machine output against human codings.

## Keywords:

Automatic text summarization, concept extraction, concept clustering, relationship extraction, relationship integration

## 1. Introduction

The purpose of this study was to develop a method for automatic summarization of sets of related research abstracts. Multi-document summarization condenses a set of related documents rather than a single document into a summary. It can provide a domain overview of a topic and indicate similarities and differences among documents. However, multi-document summarization has more challenges than single-document summarization in the issues such as compression, redundancy, cohesion, coherence. Thus traditional summarization approaches, e.g. statistics-based sentence extraction, do not always work well in multi-document summarization [2].

In a multi-document environment, many of related documents are likely to contain repeated information and only differ in certain parts. An ideal multi-document summary should contain common information among most of documents, plus important unique information present among individual documents [1]. Various approaches, including shallow and deep approaches, have been used to identify and synthesize similarities and differences across documents. The shallow approaches identified and removed repeated text units (i.e. words, phrases and sentences) extracted from different documents by syntactic comparison [5]. The deep approaches synthesized text units using concept

generalization [4], summary operators [6], and rhetorical relations [9]. However, most of previous studies identified similarities and differences at a low level based on syntactic and rhetorical relations between physical elements (i.e. words, phrases and sentences). It is desirable for the similarities and differences to be identified at a more semantic level.

This study focused on semantic contents and semantic relations to identify similarities and differences across documents and integrated the similarities and differences to generate a multi-document summary. In some domains such as sociology, psychology and education, most of research adopts the traditional quantitative research paradigm of looking for relationships between concepts operationalized as variables. Thus the similarities and differences across different research studies are mainly reflected through research concepts and relationships expressed in the text. In this study, we selected dissertation abstracts in the domain of sociology as source documents to develop a new multi-document summarization method. This method focused on extracting research concepts and relationships from each dissertation abstract and integrating similar concepts and relationships across dissertation abstracts. Moreover, the research report structure was also used to identify which parts of dissertation abstracts contain desired information. The summarization method can also be extended to research abstracts in other domains employing the same research paradigm.

The summarization method includes three main parts:

- (1) Parsing documents into several sections and identify which sections contain important research information;
- (2) Extracting research concepts from each document and integrating similar concepts across different documents;
- (3) Extracting research relationships from each document and integrating the relationships associated with a cluster of similar concepts across documents.

Each summarization step was described in subsequent sections. Finally, each step was evaluated by comparing the machine output against the human codings.

## 2. Discourse Parsing

Sentences in about 85% of dissertation abstracts could be subsumed under five standard sections: 1-*background*, 2-*research objectives*, 3-*research methods*, 4-*research results* and 5-*concluding remarks* [8]. Although the remaining 15%



of the abstracts were difficult to assign into the five sections, the research objectives are still clearly discernable in most of unstructured abstracts. To parse discourse structure of dissertation abstracts and identify the *research objectives* and *research results* sections, a supervised learning method, decision tree induction, was selected to categorize each sentence into one of the above five sections. The first decision tree classifier was constructed using a well-known decision tree induction algorithm C5.0, which used high frequency non-stop word tokens (in lemma form) and normalized sentence position as features.

A random sample of 300 sociology dissertation abstracts was selected from the 2001 Dissertation Abstracts International database and 45 unstructured abstracts were removed from them. The remaining 255 structured abstracts were partitioned into a training set of 171 abstracts to construct the classifier and a test set of 84 abstracts to evaluate the accuracy of the classifier. Preliminary experiments were carried out based on the training data using 10-fold cross-validation to determine the appropriate parameters of the decision tree model. The best classifier was obtained with a word frequency threshold value of 35 and pruning severity of 90%. We then applied the classifier to the test sample of 84 abstracts and obtained an accuracy rate of 71.6% (see Table 1).

Since the dissertation abstract is a continuous discourse with relations among sentences, furthermore, we considered using the sentences which contain clear indicator words and thus are easy to classify to help identify the categories of other relevant sentences which do not contain clear indicator words. For example, the first sentence in the *research results* section often contains the indicator words “*reveal*” and “*show*”, and the subsequent sentences may amplify on the results though they do not contain clear indicator words. To test this assumption, we manually extracted indicator words from the constructed classifier above. For each sentence, we then measured the distance between the sentence and the nearest sentence (before and after) that contains each indicator word. Then we used the surrounding indicator words as additional attributes (distance as the attribute values) to construct the second classifier. The surrounding indicator words were categorized into three types – occurring in the sentences *before* the sentence being processed, *after* the sentence being processed and both *before and after* the sentence being processed.

The test results for the second classifier using 84 structured test abstracts are shown in Table 1. It was found that only “*before*” surrounding indicator words can contribute to the categorization accuracy, by obtaining the best result of 74.5%. Finally, the second classifier, using the high frequency word tokens, normalized sentence position and “*before*” surrounding indicator words, was selected to parse the macro-level discourse structure of dissertation abstracts. A set of IF-THEN categorization rules was extracted from the classifier and applied to new dissertation abstracts.

**Table 1. Percentage of correctly classified sentences using the two classifiers based on the test sample of 84 abstracts**

Sec. ID	Classifier 1	Classifier 2 (with surrounding indicator words as additional features)		
		before	after	both before & after
1	71.10%	79.77%	67.63%	80.92%
2	55.74%	52.46%	49.18%	48.63%
3	9.74%	52.38%	39.15%	52.38%
4	87.61%	91.03%	89.31%	91.03%
5	58.62%	58.62%	55.17%	58.62%
Whole	71.59%	74.47%	68.62%	73.99%

## 2. Concept Extraction and Integration

After discourse parsing, research concepts were extracted from the *research objectives* and *research results* sections of each dissertation abstract. Then similar concepts across different dissertation abstracts were clustered based on syntactic term variations. Finally, concept clusters were categorized into subjects based on a taxonomy.

### 3.1 Concept Extraction

Concepts are usually expressed as single-word or multi-word terms. In this study, we used syntactic rules to extract concept terms. A list of part-of-speech patterns, e.g. “N PREP N”, was constructed manually for recognizing n-word terms (n=1~5).

After data preprocessing, sequences of contiguous words of different lengths are extracted from each sentence to construct n-grams (n=1~5). Using the part-of-speech patterns, terms of different lengths are identified and extracted from the same part of a sentence. These terms of different lengths represent concepts at different levels of generality (narrower or broader concepts). If two terms have overlapping sentence positions, they are combined to form a full term representing a more specific full concept.

### 3.2 Concept Clustering

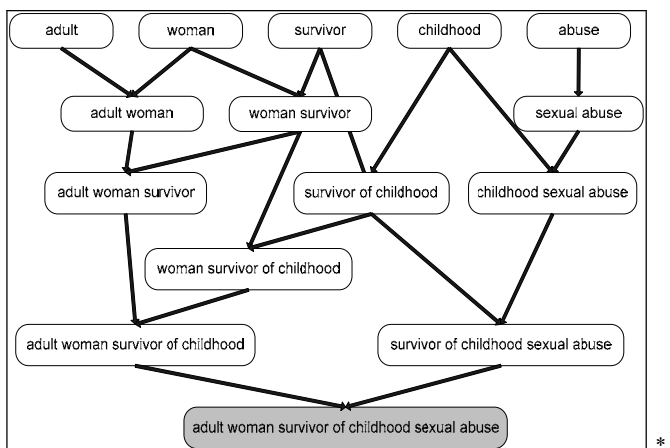
A full term, representing a specific *full concept* expressed in the text, can be segmented into shorter terms of different lengths, e.g. 1, 2, 3, 4, and 5-word terms, which are called component terms. These component terms with different number words represent *component concepts* at different levels. The shorter terms represent broader concepts whereas the longer terms represent narrower ones. There is a tangled hierarchy among the component concepts in a full concept (see Figure 1).

In Figure 1, at the top level are the broadest single-word concepts whereas the specific full concept is at the bottom. From a single-word concept to the full concept, one or more chains are created, each of which links a list of concepts of different lengths sharing a specific kind of syntactic variations. These concepts linked by the same chain have the



common head noun and are considered a group of term variants representing similar concepts at different levels of generality. The hierarchical relations among them are distinguished by their logical roles or functions. The head noun is deemed as the *main concept*. Two types of sub-level concepts are distinguished – *subclass concepts* and *facet concepts*. A subclass concept represents one of the subclasses of the parent concept whereas a facet concept specifies one of the facets of the parent concept. For example, if “abuse” is deemed as the head noun, the hierarchical relations are as follows:

- [main concept: abuse] → [subclass concept: sexual abuse] → [subclass concept: childhood sexual abuse] → [facet concept: survivor of childhood sexual abuse] → [subclass concept: adult woman survivor of childhood sexual abuse]



\* The concept in the shaded box at the bottom is a full concept.

**Figure 1. Tangled hierarchy of component concepts in a full concept**

The hierarchy of component concepts provides a way to integrate similar concepts extracted from different documents. The concepts at the higher levels can be used to generalize, to different extents, all the narrower concepts linked with them at the lower levels.

Based on this idea, a clustering algorithm was developed to construct concept hierarchies and cluster similar concepts automatically. In a set of similar dissertation abstracts, we selected high frequency nouns as head nouns in the summarization. The threshold value of the document frequency depends on the desired length of the final summary. Starting from each selected head noun, a list of term chains were constructed by linking it level by level with other multi-word terms in which the single word is used as a head noun. Each chain was constructed top down by linking the short term first, followed by longer terms containing the short term. All the chains sharing the same root node are combined to form a hierarchical cluster tree. Each cluster tree uses the root node as its cluster label. In the hierarchical cluster tree, the concepts at the top level and the second level are used to

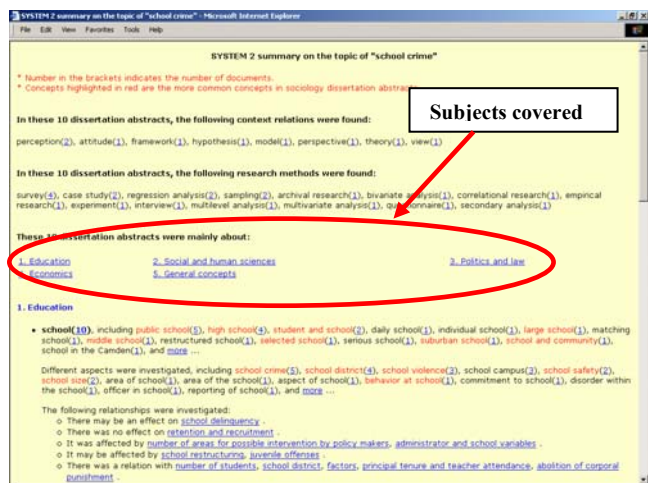
generalize all the similar concepts and integrated into a summary sentence as follows:

- *Student, including college student, undergraduate student, Latino student, ...*  
*Its different aspects are investigated, including characteristics of student, behavior of student ...*

The summary sentence is divided into two parts – the first part (*including*) giving the subclass concepts and the second part (*its different aspects*) giving the facet concepts.

### 3.3 Concept Categorization

After clustering, concept clusters are organized into subject areas based on a taxonomy (see Figure 2). The taxonomy contains important concepts in the domain of sociology, of which the main concepts (i.e. single-word concepts) were categorized into nine subjects and some sub-subjects [7]. From the subject areas covered by the concepts extracted from a set of documents, the user can get some hints on what the set of documents talks about and obtain an initial overview of the document set. Furthermore, the concept categorization can help user locate information in the subjects of interest quickly and make reading and browsing more efficient.



**Figure 2. Concept categorization into subjects in the summary**

## 4. Relationship Extraction and Integration

### 4.1 Relationship Extraction

Research relationships were extracted from the *research objectives* and *research results* sections of dissertation abstracts using pattern matching. The linguistic patterns used are regular expression patterns, each comprising a sequence of tokens. 126 relationship patterns were derived manually from the sample of 300 dissertation abstracts. An example pattern that represents one surface expression of cause-effect relationship in the text is given as follows. Each token is constrained with a part-of-speech tag.

- *<Independent variable:NP> have:V \*:DET (\*:A) effect/influence/impact:N on:PREP <Dependent variable:NP>*.

A pattern typically contains one or more slots, and the research concepts that match the slots in the pattern represent the variables linked by the relationship. A pattern matching algorithm was developed to identify the text segments that match with the relationship patterns. For example, the above pattern can match the following text segments:

- *This support the hypothesis that ontogenic variables have the greater impact on predicting risk of physical abuse.*
- *Medicaid appeared to **have a negative influence on the proportion of uninsured welfare leavers**.*

## 4.2 Relationship Normalization and Conflation

To integrate relationships, we manually analyzed 126 relationship patterns derived from the 300 sample abstracts and found that they can be categorized into nine types, including five first-order relationships (e.g. *cause-effect*, *correlation*) and four second-order relationships (e.g. *second-order cause-effect*, *second-order correlation*). The second-order relationship refers to the relationship between two or more variables influenced by a third variable. Two cause-effect relationships which are associated with the concept of “*student*” are given as follows:

- *Expected economic returns affected the college students' future career choices.*
- *School socioeconomic composition has an effect on students' academic achievement.*

The different surface expressions for the same type of relationship can be normalized using a predefined standard expression. For each standard expression, three modalities are handled – *positive*, *negative* or *hypothesized*. The relationships with the same type and modality are normalized using a standard expression. For example, the above two cause-effect relationships are normalized using the standard expression: *<dependent variable> was affected by <independent variable>*.

For a group of relationships associated with the same main concept, the normalized relationships using the same expression are conflated by combining the variables with the same roles together. For example, the above two relationships associated with “*student*” are conflated into a simple summary sentence as follows:

- *Different aspects of students were affected by expected economic returns and school socioeconomic composition.*
- The summary sentence provides an overview of all the variables that have a particular type of relationship with the given variable “*student*”.

## 5. Evaluation

In the summarization process, discourse parsing had been evaluated during the development stage of the decision tree classifier, and relationship integration was not necessary to

be evaluated since it was a simple text replacement process. Thus only three steps – *concept extraction*, *relationship extraction*, and *concept clustering* – were evaluated here.

### 5.1 Evaluation of Information Extraction

In the evaluation of information extraction, 50 structured abstracts were selected using a random table from the database. The concepts and their relationships were extracted by the machine from the *research objectives* and *research results* sections. Three human coders were asked to extract all the *important concepts* manually from each abstract, and from these to identify the *more important* concepts and then the *most important* concepts, according to the focus of the dissertation research. The machine-extracted concepts were compared against the human-extracted concepts at the three importance levels. Research relationships were extracted manually by two experts from the same 50 abstracts. From the two experts’ codings, a “gold standard” was constructed by taking the agreements in the codings. The machine-extracted relationships were compared against the human-extracted relationships.

There are four possible kinds of matches between a pair of machine-extracted and human-extracted term – *exact match*, *covered match* (a human term covers a machine term or a machine term covers a human term), and *partial match*. To obtain reasonable precision and recall, we used term similarity as weight values to reflect the degree of match between a machine term and a human term. Two key-based similarity functions were used as follows:

- Term similarity for calculating precision = 
$$\frac{\text{Number of common keywords between a machine term and a human term}}{\text{Number of keywords in a machine term}}$$
- Term similarity for calculating recall = 
$$\frac{\text{Number of common keywords between a machine term and a human term}}{\text{Number of keywords in a human term}}$$

**Table 2. Average precision, recall and F-measure for machine’s concept extraction**

Measure	For important concepts	For more important concepts	For most important concepts
Precision	49.76%	34.28%	23.60%
Recall	75.64%	78.81%	87.37%
F-measure	59.40%	47.35%	36.80%

Table 2 shows the average precision, recall and F-measure for machine’s concept extraction. The machine seldom extracted exactly the same terms as the human coders. It preferred to extract longer terms whereas the human coders preferred to extract shorter ones. For each importance level, the machine usually extracted more terms than the human coders and obtained a high recall. But it also extracted some useless terms which were ignored by the human coders and obtained a moderate level of precision. For extracting relationships, the machine obtained a high precision of 81.0% but a low recall of 54.9%.

## 5.2 Evaluation of Concept Clustering

In the evaluation of concept clustering, 15 research topics in the domain of sociology were haphazardly selected. For each topic, a set of dissertation abstracts were retrieved from the database using the topic as search query. But only five abstracts were selected from the retrieved abstracts to form a document set. For each abstract, the important concepts were automatically extracted by the machine from the *research objectives* and *research results* sections of each abstract. Human coders were asked to identify similar concepts across abstracts from the list of concepts extracted from each document set and group them into clusters. Each cluster must contain two concepts at least and was assigned a label by the human coders. For each document set, two sets of clusters were generated by two human coders and one set of clusters was generated by the machine.

To evaluate the quality of clusters, we adopted an external measure – F-measure from the field of information retrieval, employed by Larsen and Aone [3] – to compare how closely a set of machine-created clusters matches a set of known reference clusters. Two sets of human codings were each used as reference clusters. For calculating the F-measure, each machine-generated cluster is treated as the result of a query and each human-generated cluster as the desired set of concepts for a query. The recall and precision of a machine cluster (*j*) for a given human cluster (*i*) are calculated as follows:

- Precision (*i, j*) = 
$$\frac{\text{Number of common concepts between a machine cluster (j) and a human cluster (i)}}{\text{Number of concepts in a machine cluster (j)}}$$
- Recall (*i, j*) = 
$$\frac{\text{Number of common concepts between a machine cluster (j) and a human cluster (i)}}{\text{Number of concepts in a human cluster (i)}}$$

Then, the F-measure (*i, j*) is calculated as the weighted harmonic mean of precision and recall. For a given human cluster (*i*), the F-measure used is the highest one obtained among the entire set of machine clusters. Thus, the overall F-measure is calculated by taking the weighted average of the F-measures for each human cluster in the entire set:

- Overall F-measure = 
$$\frac{\sum_j \text{Number of concepts in a human cluster (i)}}{\text{Total number of concepts in the set of human clusters}} * \max \{F\text{-measure (i,j)}\}$$

**Table 3. Overall F-measures for the set of machine-created and human-created clusters**

Doc. set (N=15)	Human coding 1 as reference clusters		Human coding 2 as reference clusters	
	Machine	Coder 2	Machine	Coder 1
Avg.	51.4	47.5	67.8	54.7

Table 3 shows the overall F-measures for the set of machine-created and human-created clusters. The results suggest that the machine clustering has a higher similarity score to each of the human codings than between the human codings! This means that machine clustering can generate

reasonably good clusters of similar concepts in comparison to human clustering.

## 6. Conclusion and Discussion

In this study, we developed an automatic multi-document summarization method for research abstracts. This method parsed each dissertation abstract into five standard sections, extracted research concepts and relationships from the *research objectives* and *research results* sections of each abstract, integrated similar concepts and relationships across different abstracts, and incorporated them into new sentences to produce a summary.

Although the summarization method was developed to handle research abstracts, it can also be extended to full research articles. Research articles are much longer than abstracts and have more detailed structure in each section. Thus discourse parsing needs to be improved to handle more detailed and deeper discourse structure. Furthermore, the language used in the full research articles is probably more complex. Thus concept and relationship extraction also need to be improved, e.g. refining extracted terms and identifying more relationship patterns.

## References

- [1] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-document summarization by sentence extraction. In *Proceedings of ANLP/NAACL 2000 Workshop on Automatic Summarization* (pp.40-48). ACL, New Jersey, 2000.
- [2] U. Hahn and I. Mani. The challenges of automatic summarization. *IEEE Computer*, 33 (11), 29-36, 2000.
- [3] B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the 5<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp.16-22). ACM Press, New York, 1999.
- [4] C.-Y. Lin. Topic identification by concept generalization. In *Proceedings of the 33<sup>rd</sup> Annual Meeting of the Association for Computational Linguistics* (pp. 308-310). ACL, Morristown, 1995.
- [5] I. Mani and E. Bloedorn. Summarizing similarities and differences among related documents. *Information Retrieval*, 1(1-2), 35-67, 1999.
- [6] K. McKeown and D. Radev. Generating summaries of multiple news articles. In *Proceedings of the 18<sup>th</sup> Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp.74-82). ACM Press, New York, 1995.
- [7] S. Ou, C. Khoo, and D. Goh. Constructing a taxonomy to support multi-document summarization of dissertation abstracts. *Journal of Zhejiang University SCIENCE*, 6A (11), 1258-1267, 2005.
- [8] S. Ou, S.G. Khoo, and D. Goh. Automatic multi-document summarization of research abstracts: design and user evaluation. *Journal of the American Society for Information Science and Technology*, 58(10), 1-17, 2007.
- [9] D. Radev. A common theory of information fusion from multiple text sources step one: Cross-document structure. In *Proceedings of the 1<sup>st</sup> SIGdial Workshop on Discourse and Dialogue*, 2000.

# When word order and part-of-speech tags are not enough — Swedish dependency parsing with rich linguistic features

Lilja Øvrelid  
NLP-unit, Dept. of Swedish  
Göteborg University  
Sweden  
*lilja.ovrelid@svenska.gu.se*

Joakim Nivre  
Växjö University and  
Uppsala University  
Sweden  
*nivre@msi.vxu.se*

## Abstract

Even with high overall parsing accuracy, data-driven parsers often make errors in the assignment of core grammatical functions such as subject and object. Starting from a detailed error analysis of a state-of-the-art dependency parser for Swedish, we show that the addition of linguistically motivated features targeting specific error types may lead to substantial improvements, both for specific grammatical functions and in terms of overall parsing accuracy. In this way, we achieve the best reported results for dependency parsing of Swedish.

## Keywords

data-driven parsing, dependency parsing, error analysis, grammatical relations, linguistic features, treebanks.

## 1 Introduction

Despite the dramatic improvement in accuracy for data-driven parsers in recent years, we still have relatively little knowledge about the exact influence of data-derived features on the parsing accuracy for specific linguistic constructions. A deeper analysis of specific error sources in data-driven parsing may therefore be one of the most important steps towards a further advancement of the state of the art.

There are a number of studies that investigate the influence of different features or representational choices on overall parsing accuracy, within a variety of different frameworks, e.g., [3], [12], [10], [2] and [8]. There are also attempts at a more fine-grained analysis of accuracy, targeting specific linguistic constructions or grammatical functions, such as [4], [6], and [11]. But there are few studies that combine the two perspectives and try to tease apart the influence of different features on the analysis of specific constructions, let alone motivated by a thorough linguistic analysis.

In this paper, we present an in-depth study of the influence of certain grammatical features, such as animacy, definiteness, and finiteness, on the parsing accuracy for core grammatical functions, in particular subjects and objects. The language analyzed is Swedish, which poses special problems for the identification of subjects and objects due to limited case marking and ambiguous word order patterns. The parsing framework is deterministic classifier-based dependency parsing, more precisely the MaltParser system [13], which

achieved the highest parsing accuracy for Swedish in the CoNLL-X shared task on dependency parsing [5].

## 2 Parsing Swedish

Before we turn to a description of the treebank and the parser used in the experiments, we want to point to a few grammatical properties of Swedish that will be important in the following:

**Verb second (V2)** The finite verb always resides in second position in declarative main clauses.

**Word order variation** Pretty much any constituent may occupy the sentence-initial position. However, subjects are most common.

**Limited case marking** Nouns are only inflected for genitive case. Personal pronouns distinguish nominative and accusative case, but demonstratives and quantifying pronouns are case ambiguous (like nouns).

### 2.1 Treebank: Talbanken05

Talbanken05 is a Swedish treebank converted to dependency format, containing both written and spoken language [14].<sup>1</sup> For each token, Talbanken05 contains information on word form, part of speech, head and dependency relation, as well as various morphosyntactic and/or lexical semantic features. The nature of this additional information varies depending on part of speech:

NOUN:	<i>definiteness, animacy, case (Ø/GEN)</i>
PRO:	<i>pronoun type, animacy, case (Ø/ACC)</i>
ADJ:	<i>grade of comparison</i>
ADV:	<i>semantic class, e.g., temporal</i>
CONJ:	<i>semantic class, e.g., disjunctive</i>

### 2.2 Parser: MaltParser

We use the freely available MaltParser,<sup>2</sup> which is a language-independent system for data-driven dependency parsing. MaltParser is based on a deterministic parsing strategy, first proposed by Nivre (2003), in combination with treebank-induced classifiers for

<sup>1</sup> The written sections of the treebank consist of professional prose and student essays and amount to 197,123 running tokens, spread over 11,431 sentences.

<sup>2</sup> <http://w3.msi.vxu.se/users/nivre/research/MaltParser.html>

	FORM	POS	DEP	FEATS
S:top	+	+	+	+
S:top+1		+		
I:next	+	+		+
I:next-1	+			+
I:next+1	+	+		+
I:next+2		+		
G: head of top	+			+
G: left dep of top			+	
G: right dep of top			+	
G: left dep of next	+		+	+
G: left dep of head of top			+	
G: left sibling of right dep of top			+	
G: right sibling of left dep of top	+			+
G: right sibling of left dep of next		+	+	

**Table 1:** Baseline and extended (FEATS) feature model for Swedish; S: stack, I: input, G: graph;  $\pm n$  =  $n$  positions to the left(-) or right (+)

predicting the next parsing action. Classifiers can be trained using any machine learning approach, but the best results have so far been obtained with support vector machines, using LIBSVM [7]. MaltParser has a wide range of parameters that need to be optimized when parsing a new language. As our baseline, we use the settings optimized for Swedish in the CoNLL-X shared task [15], and the only parameter that will be varied in the later experiments is the feature model used for the prediction of the next parsing action. Hence, we need to describe the feature model in a little more detail.

MaltParser uses two main data structures, a stack (S) and an input queue (I), and builds a dependency graph (G) incrementally in a single left-to-right pass over the input. The decision that needs to be made at any point during this derivation is (a) whether to add a dependency arc (with some label) between the token on top of the stack (*top*) and the next token in the input queue (*next*), and (b) whether to pop *top* from the stack or push *next* onto the stack. The features fed to the classifier for making these decisions naturally focus on attributes of *top*, *next* and neighbouring tokens in S, I or G. In the baseline feature model, these attributes are limited to the word form (FORM), part of speech (POS), and dependency relation (DEP) of a given token, but in later experiments we will add other linguistic features (FEATS). The baseline feature model is depicted as a matrix in Table 1, where rows denote tokens in the parser configuration (defined relative to S, I and G) and columns denote attributes. Each cell containing a + corresponds to a feature of the model.

### 3 Baseline and Error Analysis

The written part of Talbanken05 was parsed employing the baseline feature model detailed above, using 10-fold cross validation for training and testing. The overall result for unlabeled and labeled dependency accuracy is 89.87 and 84.92 respectively.<sup>3</sup>

Error analysis shows that the overall most frequent errors in terms of dependency relations involve either

<sup>3</sup> Note that these results are slightly better than the official CoNLL-X shared task scores (89.50/84.58), which were obtained using a single training-test split, not cross-validation. Note also that, in both cases, the parser input contained gold standard part-of-speech tags.

Gold	Sys	before	after	Total
ss	oo	103 (23.1%)	343 (76.9%)	446 (100%)
oo	ss	103 (33.3%)	206 (66.7%)	309 (100%)

**Table 2:** Position relative to verb for confused subjects and objects

various adverbial relations (due to PP-attachment ambiguities and a large number of adverbial labels) or the core argument relations of subject and direct object. In particular, confusion of the two argument functions are among the top ten most frequent error types with respect to dependency assignment. The first three columns of Table 5 show confusion matrices for the assignment of the subject and direct object dependency relations.

The sources of errors in subject/object assignment are various. Common to all of these is that the parts of speech that realize subjects and objects are compatible with a range of dependency relations. Swedish, however, in addition exhibits ambiguities in morphology and word order which complicate the picture further. We will exemplify these factors through an analysis of the errors where subjects are assigned object status (ss\_oo) and vice versa (oo\_ss).

The confusion of subjects and objects follows from lack of sufficient formal disambiguation, i.e., simple clues such as word order, part-of-speech and word form do not clearly indicate syntactic function. The reason for this can be found in ambiguities on several levels.

With respect to word order, subjects and objects may both precede or follow their verbal head. Subjects, however, are more likely to occur preverbally (77%), whereas objects typically occupy a postverbal position (94%). Based on word order alone we would expect postverbal subjects and preverbal objects to be more dominant among the errors than in the treebank as a whole (23% and 6% respectively), since they display word order variants that depart from the canonical ordering of arguments. Table 2 shows a breakdown of the errors for confused subjects and objects and their position with respect to the verbal head. We find that postverbal subjects (after) are in clear majority among the subjects erroneously assigned the object relation. Due to the V2 property of Swedish, the subject must reside in the position directly following the finite verb whenever another constituent occupies the preverbal position, as in (1) where a direct object resides sentence-initially:

- (1) Samma erfarenhet gjorde **engelsmännen**  
 same experience made englishmen-DEF  
 ‘The same experience, the Englishmen had’

For the confused objects we find a larger proportion of preverbal elements than for subjects, which is the mirror image of the normal distribution of syntactic functions among preverbal elements. As Table 2 shows, the proportion of preverbal elements among the subject-assigned objects (33.3%) is notably higher than in the corpus as a whole, where preverbal objects account for a miniscule 6% of all objects.

In addition to the word order variation discussed above, Swedish also has limited morphological marking of syntactic function. Nouns are marked only for genitive case and only pronouns are marked for accusative case. There is also some syncretism in the pronominal paradigm where the pronoun is invariant

Gold Sys		Noun	Pro <sub>amb</sub>	Pro <sub>unamb</sub>	Other	Total
ss	oo	324 72.6%	53 11.9%	29 6.5%	40 9.0%	446 100%
oo	ss	215 69.6%	74 23.9%	9 2.9%	11 3.6%	309 100%

**Table 3:** *Parts of speech for confused subjects and objects*

for case, e.g. *det*, *den* ‘it’, *ingen/inga* ‘no’, and may, in fact, also function as a determiner. This means that, with respect to word form, only the set of unambiguous pronouns clearly indicate syntactic function. We may predict that subject/object confusion errors frequently exhibit elements whose syntactic category and/or lexical form does not disambiguate, i.e., nouns or ambiguous pronouns. Table 3 shows the distribution of nouns, functionally ambiguous and unambiguous pronouns and other parts of speech for confused subjects/objects. Indeed, we find that nouns and functionally ambiguous pronouns dominate the errors where subjects and objects are confused.

The initial error analysis shows that the confusion of subjects and objects constitutes a frequent and consistent error during parsing. It is caused by ambiguities in word order and morphological marking and we find cases that deviate from the most frequent word order patterns and are not formally disambiguated by part-of-speech information. It seems clear that we in order to resolve these ambiguities have to examine features beyond syntactic category and linear word order.

## 4 Grammatical Features for Argument Disambiguation

The core arguments themselves tend to differ along several dimensions. The property of *animacy*, a referential property of nominal elements, has been argued to play a role in argument realization in a range of languages [1], [9]. It is closely correlated with the semantic property of agentivity, hence subjects will tend to be referentially animate more often than objects. Another property which may differentiate between the argument functions of subject and object is the property of *definiteness*, which can be linked with a notion of givenness [1], [17]. This is reflected in the choice of referring expression for the various argument types in Talbanken05 – subjects are more often pronominal (49.2%), whereas objects are typically realized by an indefinite noun (67.6%). The error analysis made clear the importance of not only distinguishing between the core arguments but also between arguments and non-arguments, and in particular determiners. Both the set of case ambiguous pronouns and a group of common nouns may function as determiners. The grammatical dimensions of *person* (1st/2nd vs 3rd), as well as *case* marking for nouns (genitive) are properties which may be beneficial in this respect.

As mentioned in section 2, there are categorical constraints which are characteristic for Swedish word order. Only subjects may follow a finite verb and precede a non-finite verb and only objects may occur after a non-finite verb. Information on *finiteness* is therefore something that one might assume to be beneficial for subject/object assignment. Another property of the verb which clearly influences the assignment of core ar-

	Unlabeled	Labeled
Baseline	89.87	84.92
Pers	89.93	85.10
Def	89.87	85.02
Pro	89.91	85.04
Case	89.99	85.13
Verb	90.15	85.28
Pers&Def&Pro&Case	90.17	85.45
Pers&Def&Pro&Case&Verb	90.42	85.73
All	90.73	86.32

**Table 4:** *Overall results expressed as average unlabeled and labeled attachment scores*

gument functions is the *voice* of the verb, i.e., whether it is passive or active.

## 5 Experiments

In the following we will experiment with the addition of morphosyntactic and lexical semantic features that approximate the distinguishing properties of the core argument functions discussed in section 4. We will isolate features of the arguments and the verbal head, as well as combinations of these, and evaluate their effect on overall parsing results as well as on subject/object disambiguation specifically.

### 5.1 Experimental methodology

All parsing experiments are performed using 10-fold cross-validation for training and testing on the entire written part of Talbanken05. The feature model used throughout is the extended feature model depicted in Table 1, including all four columns.<sup>4</sup> Hence, what is varied in the experiments is only the information contained in the FEATS features (animacy, definiteness, etc.), while the tokens for which these features are defined remains constant.

Overall parsing accuracy will be reported using the standard metrics of *labeled attachment score* (LAS) and *unlabeled attachment score* (UAS).<sup>5</sup> Statistical significance is checked using Dan Bikel’s randomized parsing evaluation comparator.<sup>6</sup>

Since the main focus of this article is on the disambiguation of grammatical functions, we report accuracy for specific dependency relations, measured as a balanced F-score. We also employ two different comparative measures to compare parsers with respect to specific error types: (i) the number of errors of a certain type for the compared parsers (cf. Table 5), and (ii) the intersection of the errors of a certain type for the compared parsers.

### 5.2 Individual features

Talbanken05 explicitly distinguishes between person- and non-person referring nominal elements, a distinc-

<sup>4</sup> Preliminary experiments showed that it was better to tie FEATS features to the same tokens as FORM features (rather than POS or DEP features). Backward selection from this model was tried for several different instantiations of FEATS but with no significant improvement.

<sup>5</sup> LAS and UAS report the percentage of tokens that are assigned the correct head *with* (labeled) or *without* (unlabeled) the correct dependency label, calculated using eval.pl with default settings (<http://nextens.uvt.nl/~conll/software.html>)

<sup>6</sup> <http://www.cis.upenn.edu/~dbikel/software.html>

Confusion matrix for subjects (ss)										
sys	Baseline		Pers	Def	Pro	Case	Verb	PDPC	PDPCV	All
	#	% of tot.	# (%)	# (%)	# (%)	# (%)	# (%)	# (%)	# (%)	# (%)
OO	446	25.9	388(13.0)	425(4.7)	401(10.1)	419(6.1)	365(18.2)	361(19.1)	293(34.3)	296(33.6)
ROOT	265	15.4	270(-1.9)	284(-7.2)	275(-3.8)	277(-4.5)	260(1.9)	269(-1.5)	266(-0.4)	241(9.1)
DT	238	13.8	196(17.6)	230(3.4)	218(8.4)	205(13.9)	239(-0.4)	164(31.1)	160(32.8)	160(32.8)
SP	206	12.0	203(1.5)	187(9.2)	198(3.9)	201(2.4)	216(-4.9)	188(8.7)	187(9.2)	195(5.3)
CC	137	8.0	135(1.5)	123(10.2)	139(-1.5)	139(-1.5)	122(10.9)	120(12.4)	114(16.8)	98(28.5)
FS	133	7.7	141(-6.0)	148(-11.3)	148(-11.3)	154(-15.8)	151(-13.5)	147(-10.5)	153(-15.0)	155(-16.5)
PA	53	3.1	53(0.0)	43(18.9)	43(18.9)	37(30.2)	49(7.5)	25(52.8)	22(58.5)	26(50.9)
...	...	...	...	...	...	...	...	...	...	...

Confusion matrix for objects (oo)										
sys	Baseline		Pers	Def	Pro	Case	Verb	PDPC	PDPCV	All
	#	% of tot.	# (%)	# (%)	# (%)	# (%)	# (%)	# (%)	# (%)	# (%)
SS	309	23.8	263(14.9)	288(6.8)	280(9.4)	273(11.7)	259(16.2)	251(18.8)	215(30.4)	212(31.4)
ROOT	221	17.0	239(-8.1)	224(-1.4)	237(-7.2)	229(-3.6)	218(1.4)	251(-13.6)	245(-10.9)	241(-9.0)
PA	126	9.7	122(3.2)	129(-2.4)	123(2.4)	112(11.1)	123(2.4)	111(11.9)	109(13.5)	105(16.7)
AA	103	7.9	94(8.7)	97(5.8)	92(10.7)	106(-2.9)	102(1.0)	96(6.8)	95(7.8)	74(28.2)
DT	99	7.6	95(4.0)	94(5.1)	99(0.0)	85(14.1)	99(0.0)	81(18.2)	70(29.3)	72(27.3)
ET	58	4.5	54(6.9)	61(-5.2)	57(1.7)	59(-1.7)	64(-10.3)	49(15.5)	49(15.5)	49(15.5)
OA	57	4.4	59(-3.5)	58(-1.8)	58(-1.8)	57(0.0)	65(-14.0)	63(-10.5)	66(-15.8)	64(-12.3)
...	...	...	...	...	...	...	...	...	...	...

**Table 5:** Confusion matrices for the assignment of the subject and object dependency relations for the baseline parser (columns 2–3) and for the different extended feature models (columns 4–11). For the baseline parser, we give the absolute number of occurrences of each error type, together with the percentage of each error type out of all subject/object errors. For the extended parsers, we give absolute numbers (#) along with relative improvement compared to the baseline (%)

tion which overlaps fairly well with the traditional notion of **animacy**. As Table 4 shows, the addition of information on animacy for nominal elements causes an improvement in overall results ( $p < .0002$ ). The subject and object functions are the dependency relations whose assignment improves the most when animacy information is added. There is also an effect for a range of other functions where animacy is not directly relevant, but where the improved analysis of arguments contributes towards correct identification (e.g., adverbials and determiners). If we take a closer look at the individual error types involving subjects and objects in Table 5, we find that the addition causes a reduction of errors confusing subjects with objects (ss\_oo), determiners (ss\_dt) and subject predicatives (ss\_sp) – all functions which do not embody the same preference for person reference as subjects. The intersection of errors confusing subjects and objects shows that we improve on 23.1% of the ss\_oo errors and 28.8% of the oo\_ss errors made by the baseline parser when adding information on animacy.

Morphological **definiteness** is marked for all common nouns in Talbanken05. The addition of information on definiteness during parsing causes a slight (at the  $p < .03$  level) improvement of overall results. Most noteworthy is an improvement in the identification of subject predicatives (sp), which are often confused with subjects (cf. Table 5). Nominal predicatives in Swedish are often realized by an indefinite noun (89.4%).

The addition of information on **pronoun type**<sup>7</sup> causes a general improvement in overall parsing results ( $p < .01$ ), as we can see from Table 4. The dependency relations whose assignment improves the most are, once again, the core argument functions (ss, oo), as well as determiners (dt). We also find a general im-

provement in terms of recall for the assignment of the formal subject (fs) and object (fo) functions, which are both realized by the third person neuter pronoun *det* ‘it’, annotated as impersonal in the treebank.

Talbanken05 contains morphological **case** annotation for pronouns (null/accusative) and common nouns (null/genitive). Whereas we noted in the initial error analysis that case marking is not sufficient to disambiguate the targeted errors, we observed that core arguments were confused for determiners due to ambiguity in syntactic category and word form. When we employ case information during parsing we find a clear improvement in results ( $p < .0001$ ). However, the improvement is not first and foremost caused by improvement in assignment of subjects and objects, but rather, the assignment of determiners and prepositional objects.

Talbanken05 contains morphosyntactic information on **tense** and **voice** for all verbs. In this experiment, all information available for the verbal category is included during parsing. As Table 4 shows, the addition of morphosyntactic information for verbs causes a clear improvement in overall results ( $p < .0001$ ). The added information has a positive effect on the verbal dependency relations for finite and non-finite verbs, as well as an overall effect on the assignment of subjects and objects. Information on voice also benefits the relation expressing the demoted agent (ag) in passive constructions. The overview of individual error types typically involved in the assignment of the core argument functions (cf. confusion matrices in Table 5) indicates that the addition of information on verbal features improves on the confusion of the main argument types – subjects and objects (ss\_oo, oo\_ss), as well as subjects and expletive subjects (ss\_fs). With respect to the intersection of errors performed by the two parsers confusing subjects and objects, we observe an improvement of 33.2% (ss\_oo) and 37.2% (oo\_ss) for the parser with added verbal features.

<sup>7</sup> There are 12 pronoun types in Talbanken05 which differentiate between, e.g., local (1st/2nd) and 3rd person pronouns, reflexive, reciprocal, interrogative, impersonal pronouns, etc.



	Deprel	Freq	Baseline	PDPCV
SS	subject	0.1105	91.37	92.73
OO	object	0.0632	85.83	87.62
DT	determiner	0.1081	95.49	96.42
SP	subject predicative	0.0297	85.47	86.72
FS	formal subject	0.0050	71.81	74.57
PA	prep argument.	0.1043	95.03	95.74

**Table 6:** Comparison of balanced F-scores for the core argument relations in the combined experiment (PDPCV).

### 5.3 Feature combinations

The following experiments combine different nominal argument features, nominal argument features with verbal features, and finally all available grammatical features in Talbanken05.

The combination of the argument features of animacy, definiteness, pronoun type and case (PDPC), as well as the addition of verbal features to this feature combination (PDPCV) causes a clear improvement compared to the baseline *and* each of the individual feature experiments ( $p < .0001$ ) (cf. Table 4). Since the results are better than the individual runs, we may conclude that there is a cumulative effect of the combined information.

Table 6 shows a comparison of the balanced F-scores for the argument dependency relations in the baseline and PDPCV experiments. If we examine the counts for individual error types in Table 5, we find an error reduction for the confused subjects with objects and vice versa with 34.3% and 30.4% respectively. With respect to the specific errors performed by the baseline parser for this error type, and targeted by the experiments, we observe a reduction of 45.7% for SS\_OO and 46.6% for OO\_SS.

When we add the remaining grammatical features in Talbanken05, i.e., the features for adjectives, adverbs, conjunctions and subjunctions, we observe an improvement ( $p < .0001$ ) for the conjunct relation as well as the argument functions (SS, OO), determiners, verbal relations and adverbials. If we examine the intersected errors performed by the baseline parser in terms of confused subjects and objects, we find an improvement of 53.4% for the SS\_OO error type and 50.5% for the OO\_SS.

## 6 Conclusion

An in-depth error analysis of the best performing data-driven dependency parser for Swedish revealed consistent errors in dependency assignment, namely the confusion of core argument functions, resulting from word order ambiguity and lack of case marking. A set of experiments were designed to examine the effect of various linguistically motivated grammatical features hypothesized to target these errors.

The experiments showed that each feature individually caused an improvement in terms of overall labeled accuracy, performance for the core argument relations, and error reduction for the specific types of errors performed by the baseline parser. In particular, the final experiment (All), exhibited an error reduction of about 50% for the errors specifically targeted following the initial error analysis. In this way, we have also advanced the state of the art in Swedish dependency

parsing, increasing the labeled accuracy of the best performing parser by 1.4 percentage points.

A possible objection to the applicability of the results presented above is that the added information consists of gold standard annotation from a treebank. However, the morphosyntactic features examined here are for the most part straightforwardly derived (definiteness, case, person, tense, voice) and represent standard output from most part-of-speech taggers. The property of animacy has been shown to be fairly robustly acquired for common nouns by means of distributional features from a shallow-parsed corpus [16].

Specific plans for future work relate to further error analysis of the baseline parser, including other non-argument relations, most notably adverbials, and similar experiments to more fully understand the interplay of the various features. On a more general note, the development of methods for in-depth error analysis which relate to specific linguistic constructions constitutes an important direction for gaining further knowledge about the types of generalizations acquired through data-driven syntactic parsing.

## References

- [1] J. Aissen. Differential Object Marking: Iconicity vs. economy. *Natural Language and Linguistic Theory*, 21:435–483, 2003.
- [2] D. M. Bikel. Intricacies of Collins’ parsing model. *Computational Linguistics*, 30(4):479–511, 2004.
- [3] R. Bod. *Beyond Grammar*. CSLI Publications. University of Chicago Press, 1998.
- [4] S. Buchholz. *Memory-Based Grammatical Relation Finding*. PhD thesis, Tilburg University, 2002.
- [5] S. Buchholz and E. Marsi. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, 2006.
- [6] J. Carroll and E. Briscoe. High precision extraction of grammatical relations. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, pages 134–140, 2002.
- [7] C.-C. Chang and C.-J. Lin. Libsvm: A library for Support Vector Machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001.
- [8] E. Charniak and M. Johnson. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 173–180, 2005.
- [9] Ö. Dahl and K. Fraurud. Animacy in grammar and discourse. In T. Frøtheim and J. K. Gundel, editors, *Reference and referent accessibility*, pages 47–65. John Benjamins, Amsterdam, 1996.
- [10] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 423–430, 2003.
- [11] S. Kübler and J. Prokić. Why is German dependency parsing more reliable than constituent parsing? In *Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories (TLT)*, pages 7–18, 2006.
- [12] B. Megyesi. Shallow parsing with PoS taggers and linguistic features. *Journal of Machine Learning Research*, 2:639–668, 2002.
- [13] J. Nivre, J. Hall, and J. Nilsson. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, pages 2216–2219, 2006.
- [14] J. Nivre, J. Nilsson, and J. Hall. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC2006)*, Genoa, Italy, May 24–26 2006.
- [15] J. Nivre, J. Nilsson, J. Hall, G. Eryiğit, and S. Marinov. Labeled pseudo-projective dependency parsing with Support Vector Machines. In *Proceedings of the tenth conference on Computational Natural Language Learning (CoNLL)*, 2006.
- [16] L. Øvrelid. Towards robust animacy classification using morphosyntactic distributional features. In *Proceedings of the EACL 2006 Student Research Workshop*, Trento, Italy, 2006.
- [17] A. Weber and K. Müller. Word order variation in German main clauses: A corpus analysis. In *Proceedings of the 20th International conference on Computational Linguistics*, pages 71–77, 2004.



# Learning Selectional Preferences for Entailment or Paraphrasing Rules

Marco Pennacchiotti<sup>(\*)</sup>, Basili Roberto<sup>(†)</sup>, Diego De Cao<sup>(†)</sup>, Paolo Marocco<sup>(†)</sup>,

(<sup>(†)</sup>) DISP - University of Roma Tor Vergata  
Via del Politecnico, 1 - 00133 Roma (Italy)  
{*basili,decao,marocco*}@*info.uniroma2.it*

(<sup>(\*)</sup>) Computational Linguistics, Saarland University  
Saarbrücken, Germany.  
*pennacchiotti@coli.uni-sb.de*

## Abstract

Recent work on textual entailment or paraphrasing emphasizes the role of automatic learning of inference rules. Major weakness of these repositories is the low accuracy reachable in applying the rules in operational settings (e.g. textual entailment challenges or question answering). In this paper a robust method for automatic learning of inference rules is presented. As opposed to existing proposals, it relies on a geometrical model of similarity, based on latent semantic analysis applied to the source text collection. The result is a not merely distributional notion of lexical similarity that implies also selectional preference for the individual rule arguments. Experiments on a large data set show that selectional restrictions, applied conjunctively to all arguments in a pattern, are able to better select correct vs. incorrect cases. As the designed learning process is completely unsupervised and widely applicable, the method provides a very useful tool for different application and domains.

## Keywords

Lexical Acquisition, Inference Rules, Semantic similarity, LSA

## 1 Introduction

Textual inference is a key component of many natural language processing tasks. For example, question answering needs inference to find non-trivial answers to general questions. Given the question “*Who played the final of the World Cup?*”, the answer “*Italy*” could be retrieved from the snippet “*Italy has won the final of the World Cup*”, by knowing that the pattern “*X win Y*” entails “*X play Y*”. This type of inferences at the textual level have been successfully exploited in information extraction [13] and question answering [3], and have been recently modeled in the Recognizing Textual Entailment (RTE) challenge [2], where systems are compared on the task of recognizing if a text fragment entails another.

The challenge revealed that RTE systems critically need knowledge at the linguistic level. In particular, most useful are paraphrase and entailment resources containing lists of entailment rules such as

“*X win Y*”  $\Rightarrow$  “*X play Y*”. While these resources already exist, for example DIRT [6] and TE/ASE [14], they suffer two major limitations which make their use in inference tasks still a challenge: they lack directionality (i.e. they contain *inference rules*  $p \approx q$ , where the direction of the entailment between pattern  $p$  and  $q$  is not known) and they are not accurate enough.

Recent trends in RTE suggest that the second limitation is more critical. Indeed, most cases of textual entailment in real applications are pure paraphrases [1], and then are not much sensitive on directionality. On the contrary, the accuracy issue is fundamental: resources are both too noisy (for example DIRT has an average precision of 0.50 [6]), and too general. In particular, inference rules are often too generic to be successfully used in applications, as they do not indicate explicitly in which context they can be applied. For example, the rule “*X win Y*”  $\approx$  “*X play Y*” is useful in the previous example, but also implies the incorrect inference “*Gilmour played guitar in Pink Floyd*”  $\approx$  “*Gilmour won guitar in Pink Floyd*”.

Recently, [9] proposed a method to produce more specific rules, in which the admissible arguments for the inference rules are explicitly indicated. For this purpose they use *inferential selectional preferences* (ISP) over the DIRT rules, producing inference rules augmented with selectional preferences (SP). In the above example, the rule would be:  $\langle \textit{player} \rangle \textit{play} \langle \textit{competition} \rangle \approx \langle \textit{player} \rangle \textit{win} \langle \textit{competition} \rangle$ . We call these augmented rules *restricted inference rules* (RIR). The two SPs are inferred in [9] as the most common generalization of the  $X$  and  $Y$  slot fillers in taxonomies such as WordNet or CBC [10]. Yet, this approach suffers from three main problems:

- performance is still low for applications: the ISPs are able to filter correct/incorrect instances of a RIR with 0.59 accuracy;
- it needs pre-existing resources. This makes the method sensitive to the accuracy and the coverage of the resources themselves.
- the computational cost for building the ISPs is high.

In this paper we present a new approach to induce RIRs, based on a LSA-based geometric similarity model. Slot fillers for patterns  $p$  and  $q$  in an infer-

<sup>(\*)</sup>Formerly at University of Roma Tor Vergata

ence rule like  $p \approx q$ , are mapped into a vector representation in a reduced LSA space. Clustering is then applied as a computational model of SP's where satisfiability results in a similarity estimation problem: similar clusters are thus used to compile the valid RIRs that specialize  $p \approx q$ . This method has the following main advantages:

- the computation of similarity and the clustering in the LSA space offers an effective way to capture text semantics, as LSA is sensitive to both first and second order relations among words;
- external resources are not needed: SPs are created directly from the textual corpus, thus reducing validation costs and coverage problems;
- complexity is also kept limited, as all similarity computations are done in the reduced LSA space.

In the rest of the paper, we will report empirical evidence to support these claims. In Section 2, we analyze some previous work related to our research; in Section 3 we describe our approach, while in Section 4 we report on the acquired experimental evidence. Finally, in Section 5 we draw final conclusions and future work.

## 2 Related Work

Our work relates to three main areas: inference rule acquisition, selectional preferences and Latent Semantic Analysis (LSA).

Automatic methods for acquiring inference rules mainly use pattern distributional properties to infer a similarity score for the relation  $p \approx q$ . [6] introduce DIRT, a database of inference rules, created by first extracting patterns  $\langle X, p, Y \rangle$  from a large corpus using a dependency parser and then creating inference rules  $\langle X, p, Y \rangle \approx \langle X, q, Y \rangle$  as those pairs of patterns  $p$  and  $q$  which are distributionally similar (i.e. they have similar slot fillers for  $X$  and  $Y$ ). DIRT has an average precision of 0.50 on the task of acquiring inference rules over the top scoring 40 rules for each pattern. [14] present a scalable Web-based approach for inference rule acquisition. Starting from a verb lexicon, the method automatically acquires from the Web useful slot fillers, which are in turn used to discover distributionally similar patterns. The method achieves a precision of 0.44. Other resources for textual inference mainly focus on paraphrasing. The extraction of paraphrase patterns is usually achieved by using aligned/comparable corpora: in [8] Automata are used to extract and generate paraphrases using multiple translations of the same story; in [12] Named Entities are used to locate and drive paraphrase extraction from news on the same event.

Automatic methods for acquiring selectional preferences have been firstly introduced in [11], and have been later exploited in many NLP fields, to restrict the applicability of a given predicate to a pre-defined set of semantic classes. These classes are derived either from manually built resources (such as WordNet) or from automatically harvested ones (such as CBC). As outlined in the Introduction, [9] exploit selectional preferences to induce refinements over DIRT inference rules. To our knowledge no attempts have been made

so far to induce selectional preferences for inference rules without the use of an external taxonomy.

LSA [5], captures the essential relationships between documents and word meaning, and tries to tackle the problem of the very large number of dimensions. In [7] a LSA model is applied to explore the relationship between lexical cohesion and entailment. A generative model of entailment is formulated, in which the training consists of computing cohesion/coherence over labeled proposition-hypothesis pairs and using logistic regression to fit a supervised classifier to the data. Even if the results support the basic intuition, the model does not directly deal with the directionality of the relation.

## 3 Automatic Acquisition of Inference Rules

The goal of our model is to automatically induce *restricted inference rules* (RIRs). Similarly to [9], the approach proceeds through the following three steps.

In the first step (Section 3.2), given an inference rule  $\langle X, p, Y \rangle \approx \langle X, q, Y \rangle$ , it considers separately the two patterns  $\langle X, p, Y \rangle$  and  $\langle X, q, Y \rangle$ . The goal is to find for each slot  $X$  and  $Y$  of a pattern  $p$ , its set of *selectional preferences* SPs  $CX^p$  and  $CY^p$ , i.e. the typical semantic classes of  $X$  and  $Y$  for  $p$ . For example  $\langle X, \text{play}, Y \rangle$  will have  $CX^p = \{\text{Player}, \text{Actor}, \text{Musician}\}$  and  $CY^p = \{\text{Competition}, \text{Piece}, \text{Composition}\}$ , and  $\langle X, \text{win}, Y \rangle$  will have  $CX^q = \{\text{Football\_Player}, \text{Player}, \text{Person}\}$  and  $CY^q = \{\text{Competition}, \text{Award}\}$ . A single SP will be hereafter indicated with a pedice: e.g.  $CY_1^q \in CY^q$  is used in the example to indicate *Competition*. Our system performs this first step using clustering in the LSA space: the set of SPs are represented by clusters of slot-fillers in the reduced geometric space.

In the second step (Section 3.3), given the sets of SPs for  $p$  and  $q$ , the system finds the pairs  $\langle CX_i^p, CX_j^q \rangle$  and  $\langle CY_i^p, CY_j^q \rangle$  of *compatible SPs* for slot  $X$  and  $Y$  across the two patterns, i.e. the similar semantic classes across  $p$  and  $q$  for which the inference rule is likely to hold. In the example, the compatible SPs for the slot  $X$  are  $\langle \text{Player}, \text{Player} \rangle$  and  $\langle \text{Player}, \text{Football\_Player} \rangle$ , and for the slot  $Y$  are  $\langle \text{Competition}, \text{Competition} \rangle$ . The system performs this step by estimating compatibility between clusters of  $p$  and  $q$ , as similarity in the LSA space.

In the third step (Section 3.4), the system has to build the final set of RIRs, by leveraging the pairs of compatible clusters discovered in the previous step. For example it could discover  $\langle \text{Football\_Player}, \text{play}, \text{Competition} \rangle \approx \langle \text{Player}, \text{win}, \text{Competition} \rangle$  and  $\langle \text{Player}, \text{play}, \text{Competition} \rangle \approx \langle \text{Player}, \text{win}, \text{Competition} \rangle$ . Our system performs this last step by conjunctively applying compatibility constraints over the corresponding slot fillers in a pattern pair.

### 3.1 Using Latent Semantic Analysis for Rule Induction

LSA [5] is an extension of the vector space model based on the *Singular Value Decomposition (SVD)*, a matrix decomposition process that creates an approximation

of the original word by document matrix, and captures term semantic dependencies. In LSA, the document space is replaced by a lower dimensional document space  $M_k$ , called  $k$ -space (or LSA space) in which each dimension is a derived concept.  $M_k$  captures the same statistical information in a new  $k$ -dimensional space, where each dimension represents one of the derived LSA features (or concepts). These may be thought of as artificial concepts and represent emerging meaning components as a linear combination of many different words (or documents). Terms, on their own, are accordingly represented as combinations of the emerging concepts. The similarity between resulting vectors, as measured by the cosine of the resulting angle, has been shown to closely mimic human judgments of meaning similarity and semantic inference. LSA has two main advantages: first, the computation needed to measure similarity is drastically reduced due to the low  $k$  dimensional LSA space; secondly, unlike similarity methods in traditional vector spaces, LSA captures second order relations between words.

### 3.1.1 Leveraging LSA Similarity for discovering RIRs

As outlined in the introduction, the major limitation of inference rule resources such as DIRT, is that they do not specify for which semantic classes a rule holds. In particular the DIRT model [6] exploits the so-called *Extended Distributional Hypothesis*: “If two patterns tend to occur in similar contexts, the meanings of the patterns tend to be similar”.

In practice, this means that two patterns are similar if they have a sufficient number of slot-fillers in common, as extracted from a textual corpus. Similarity between contexts (i.e. the slot-fillers and the occurrences of the patterns in the corpus) is thus a key notion for rule induction. Yet, once rules have been induced, these originating contexts are neglected. Then, no further lexical constraint is available to decide when to apply a rule to a novel context (e.g. we don't know if the rule  $\langle X, play, Y \rangle \approx \langle X, win, Y \rangle$  can be applied to “David Gilmour plays the guitar” to derive “David Gilmour won the guitar”).

The main aim of the model proposed here is to capitalize the idea that originating contexts are very informative about the lexical conditions under which an inference rule can be triggered. The goal is then to build a DIRT-like resource in which the slot-filler information is preserved. Yet, storing such information is prohibitive, because of the huge number of lexical slot fillers observable in very large corpora. An alternative representation must be then devised. LSA gives the solution, by a synthetic way to represent slot fillers in the reduced  $M_k$  space. In  $M_k$  the semantics of the slot fillers is preserved, but the dimensionality of the problem is drastically reduced. In particular, rule induction can exploit the similarity between slot fillers of two patterns  $p$  and  $q$  in  $M_k$ , as a source of semantic information. Also, clustering in the LSA space allows to detect SPs for individual slot fillers, which can be used to decide when to apply a rule in a novel context. As clusters are expressed via an inexpensive vector representations, i.e. their centroids  $c_X$  and  $c_Y$ , satisfiability of a SP can be modeled via a simple sim-

ilarity constraint. A newly encountered word  $w$  satisfies the SP of a slot filler  $X$  iff it is enough similar to a centroid, i.e. iff  $sim(w, c_X) > r$ , where  $r$  is a positive threshold (the same stands for  $Y$ ).

The LSA space  $M_k$  used for our purpose is that obtained from the original space  $M$ , composed by the words (including the slot fillers) and the documents of the corpus from which the resource (e.g. DIRT) has been created. In particular, as requested by LSA, documents are further divided in sub-portions (e.g. few sentences in a paragraph) that constitute coherent discourse segments (e.g. a news, a full story, etc.). In the following sections, we describe how we implemented this idea.

## 3.2 Selectional Preferences as clusters in the LSA space

In the first step, the algorithm firstly performs the SVD, obtaining a reduced space  $M_k$  in which each slot filler is represented by a vector in the space. Given a pattern  $p$ , it is then possible to compute the similarity between its slot fillers using Cosine similarity.

Then, for a given pattern  $p$ , the algorithm applies a variant of the *K-means* clustering algorithm to separately cluster the  $X^p$  and  $Y^p$  slot fillers, using their vectorial representation in the LSA space. The variant is based on the *QT* (*quality threshold*) cluster algorithm [4], that does not require to specify the number of clusters a priori. The basic idea is to impose a threshold representing the maximum allowed distance from the centroid of a cluster: only if a words falls beyond this distance, a new cluster is created. Details are in [4]. (here the threshold *QT* is imposed on the similarity of vectors, which is intended as the inverse of the distance).

The produced sets of clusters, denoted as  $CX^p$  and  $CY^p$ , are the LSA representations of the SPs for the  $X^p$  and  $Y^p$  slots of the pattern  $p$ . In other terms, every cluster  $CX_i^p$  expresses a group of lexical items (i.e. slot fillers) that act as a single semantic class, and provides selectional criterion for deciding the correctness of the pattern use in future contexts. We will then assume that each cluster *is* in fact a SP, and make use of similarity between words and clusters as a selectional preference constrain. For example suppose the pattern  $\langle X, play, Y \rangle$  has the following slot fillers for the  $X^p$  slot:  $\{McEnroe, Johnny\_Depp, midfielder, Gilmour, comedian, footballer, Baggio\}$ . Ideally, three clusters should be created:  $CX_1^p = \{McEnroe, midfielder, footballer, Baggio\}$ ,  $CX_2^p = \{Johnny\_Depp, comedian\}$ ,  $CX_3^p = \{Gilmour\}$ , which should respectively represent the SPs *Player, Actor, Musician*.

Hereafter, given a generic pattern  $\langle X, p, Y \rangle$ , we denote with  $x_i$  and  $y_i$  the slot fillers of  $X$  and  $Y$ , i.e.  $x_i \in X$  and  $y_i \in Y$ .

Note that a cluster  $CX_i^p$  can be possibly made by a single element (e.g.  $CX_3^p = \{Gilmour\}$ ). Such a cluster is called **trivial**:

$$\exists! x \in X^p \text{ such that } x \in CX_i^p \quad (1)$$

i.e.  $CX_i^p = \{x\}$ . As a final operation, the algorithm computes for each cluster a degree of cohesion. The *cohesion*  $r_i$  of a cluster  $C_i^p$  is intended as the minimal similarity between the centroid  $c_i^p$ ,

and the cluster members. Given a valid similarity function  $sim$  (e.g. the cosine similarity) between two instances, the cohesion can be easily defined as  $r_i = \min_{x_j \in C_i^p} sim(x_j, c_i^p)$ . However, accordingly, the cohesion of a trivial cluster  $C_i$  would be 1, that is too restrictive to adopt for the usually more vague notion of SP. For example the SP  $CX_3^p = \{Gilmour\}$  would accept only contexts which have *Gilmour* as slot fillers; In order to *relax* the constraint, we assume that any cluster  $C_i$  must be characterized by a maximal cohesion that does not exceed a given *cohesion threshold*  $\tau \in (0, 1)$ . The lexical cohesion  $r_i$  of a generic cluster  $C_i^p$  can be thus formally defined as follows:

$$r_i = \min(\min_{x_j \in C_i^p} sim(x_j, c_i^p), \tau) \quad (2)$$

Equation 2 expresses the degree of freedom by which a cluster (i.e. a SP) can be used in future predictions, as described in the next section.

### 3.3 Discovering compatible clusters

The goal of the second step is to find compatible SPs across two patterns  $p$  and  $q$  of an inference rule. In the LSA space, the problem is then to identify pairs  $\langle CX_i^p, CX_j^q \rangle$  and  $\langle CY_i^p, CY_j^q \rangle$  of *compatible clusters*, i.e. clusters which are likely to represent the same semantic classes. We then need to define a notion of similarity between two clusters, i.e. a notion of *compatibility*.

**DEFINITION** (*Compatibility between clusters*). Given two clusters related to the same slot,  $C_i$  and  $C_j$  with centroids  $c_i$  and  $c_j$ ,  $C_i$  is **compatible** with  $C_j$ , i.e.  $C_i \simeq C_j$ , **iff** an element  $x_k \in C_j$  exists such that  $sim(x_k, c_i) \geq r_i$  or vice versa.

The notion of compatibility is leveraged in the next step to induce the RIRs. Also, it can be used to detect if a word satisfies the SP expressed by a cluster of a pattern. An incoming word  $w$  *satisfies* a SP expressed by a cluster  $C_i$ , if  $w$  is likely a *member* of  $C_i$ ,  $w \in C_i$ . This means that its similarity with the centroid  $c_i$  is *close enough* to the cluster's coherence. A tolerance factor  $\sigma(r_i)$ , can be here used for the following technical definition.

**DEFINITION** (*Satisfaction of SPs*). An incoming word  $w$  *satisfies* a SP expressed by a cluster  $C_i$ , i.e.  $w \in C_i$ , **iff**  $sim(w, c_i) > r_i - \sigma(r_i)$ , where  $\sigma(r_i)$  is a monotonic non decreasing function of the tolerance.

The higher is the tolerance the lower it should be the threshold of acceptance. A possible definition for  $\sigma(r_i)$  is  $\max(\alpha r_i^n + \beta, 0)$  with parameters  $\alpha$ ,  $\beta$  and  $n$  to be fixed empirically<sup>1</sup>. The above definition of compatibility and satisfaction for SPs allow us respectively to define a model for inducing RIRs (Section 3.4), and to decide if a text fragment is a valid pattern instance (Section 3.4.1).

### 3.4 Induction of Restricted Inference Rules

An inference rule  $\langle X, p, Y \rangle \simeq \langle X, q, Y \rangle$  states that in most contexts  $q$  can be used as a good substitute

<sup>1</sup> A setting, employed after estimation over the development set, is  $n = 1$ ,  $\alpha = 0.267$  and  $\beta = -0.0167$ .

of  $p$ . This indicates a generic *relatedness relation* between two patterns, that may eventually result in an entailment or equivalence relation (more specifically, relatedness between two patterns is a *necessary* condition for an entailment or equivalence relation).

We can here define a more precise and restrictive notion of *semantic relatedness*, which takes into consideration compatible SPs on the slot fillers. This notion is at the base of the RIRs definition.

**DEFINITION** (*Semantic relatedness between patterns*). Given two patterns  $p$  and  $q$ , they are *semantically related*, i.e.  $\langle X, p, Y \rangle \simeq \langle X, q, Y \rangle$ , if their slots are described by compatible clusters. More technically,  $\langle X, p, Y \rangle \simeq \langle X, q, Y \rangle$  holds **iff** for the slots  $X$  and  $Y$ , two cluster pairs,  $\langle CX_i^p, CX_j^q \rangle$  and  $\langle CY_i^p, CY_j^q \rangle$ , can be found such that:

$$CX_i^p \simeq CX_j^q \quad \wedge \quad CY_i^p \simeq CY_j^q \quad (3)$$

Equation 5 makes a consistent use of the geometrical constraints on selectional preferences provided by the LSA transformation for patterns  $p$  and  $q$ . This realizes an operational model of ISPs as in [9]. Yet, it has the following advantages: it does not imply any generalization of the collocational evidences from text, except the similarity estimated in the LSA space; it does not rely on external resources like WordNet or CBC; it is largely applicable. The above definition of semantic relatedness is used to induce the RIRs.

**DEFINITION** (*Restricted inference rules*). Given two patterns  $p$  and  $q$  that are semantically related, every cluster 4-tuple  $\langle CX_i^p, CX_j^q, CY_i^p, CY_j^q \rangle$  satisfying Equation 5 establishes a restricted inference rules:

$$\langle CX_i^p, p, CY_i^p \rangle \simeq \langle CX_j^q, q, CY_j^q \rangle$$

This rule justifies the semantic relatedness through the conjunctive satisfaction of all SPs, via the cluster compatibility notion.

Given two patterns  $p$  and  $q$  several restricted inference rules can be derived, as Equation 5 can be satisfied by multiple choices of cluster pairs  $\langle CX_i^p, CX_j^q \rangle$  and  $\langle CY_i^p, CY_j^q \rangle$ . These different RIRs can be studied to establish some regularities in evoking independent word senses for the support verbs of  $p$  and  $q$ . In principle, independent verb senses  $p_1, p_2$  could generate different inference rules by selecting different senses of the pattern  $q$ . We call the set of RIRs for the patterns  $p$  and  $q$  a *rule set*:

$$RIR(p, q) = \{ \langle p, q, CX_i^p, CX_j^q, CY_i^p, CY_j^q \rangle \text{ satisfying Eq. 5} \}$$

#### 3.4.1 Leveraging RIRs in Textual Inference

A restricted inference rule should predict if, given a triple like  $w_x - p - w_y$  as it is found in an incoming sentence, it is a good candidate for the substitution  $w_x - p - w_y \simeq w_x - q - w_y$ . This establishes a criteria for deciding when and why an inference rule can be used. This property can be defined as follows.

**DEFINITION** (*Relational Selectional Preference*). The inference  $w_x - p - w_y \simeq w_x - q - w_y$  is accepted **iff** a 6-tuple  $\langle p, q, CX_i^p, CX_j^q, CY_i^p, CY_j^q \rangle \in RIR(p, q)$  exists such that the following condition holds:

$$w_x \in CX_i^p \quad \wedge \quad w_y \in CY_i^p \quad (4)$$

Finally, it remained open the problem to ensure that there is a compatibility not only between  $\langle CX_i^p, p, CY_i^p \rangle$  or  $\langle CX_j^q, q, CY_j^q \rangle$  but also between the 3-tuple of the same pattern. In other words to ensure that  $\langle X, p, Y \rangle$  states in some context (i.e.: “David Gilmour plays the guitar” and not “David Gilmour plays a tennis match”). A simple solution is to add to the definition of *Semantic relatedness between patterns*, a notion of compatibility between  $X$  and  $Y$  in  $\langle X, p, Y \rangle$ .

In a LSA space, a cluster of musicians is not close to a cluster of tennis championships, whereas it is to the clusters of musical instruments. These evidences suggest that in Equation 5 also these relations must be satisfied

$$CX_i^p \simeq CY_j^p \quad \wedge \quad CX_i^q \simeq CY_j^q \quad (5)$$

## 4 Empirical Investigation

### 4.1 Experimental Set-Up and Preprocessing

The goal of the experiment is to evaluate the selective power of the acquired selectional preferences. We performed the same experiment as in [9], by using the datasets and the evaluation methodology provided by the authors.

In [9], a random set of 100 rules from DIRT are selected, and then for each of them 10 instances are extracted from the 1999 AP newswire collection (31 million words approximately), amounting to a total of 1000 instances. These are then annotated as correct or incorrect by two judges ([9] report an agreement of  $k = 0.72$ ). A third judge adjudicated in case of disagreement. The final corpus was then divided in a set of 500 development instances for parameter estimation, and a set of 500 test instances.

Systems’ filtering power is evaluated on the test set against the annotator gold standard, by using the following scores. Let  $t^+$  represent the number of positive instances correctly accepted by the system,  $t^-$  represent the number of negative instances correctly refused,  $f^+$  represent the number of accepted negative instances and  $f^-$  the number of refused positive instances. *Sensitivity* is defined as  $\frac{t^+}{t^+ + f^-}$ , i.e. the probability of accepting correct inferences. *Specificity* is defined as  $\frac{t^-}{t^- + f^+}$ , i.e. the probability of rejecting incorrect inferences. The overall *Accuracy*, i.e.  $\frac{t^+ + t^-}{t^+ + t^- + f^+ + f^-}$ , captures the quality of pointwise inference over the two classes of instances.

We compare our systems to two baseline: *accept all*, which accepts all inferences, and a *random* choice function; we then compare to the best joint and independent models presented in [9] (respectively, *ISP.IIM.or* and *ISP.JIM*).

To create the LSA space, we analyzed the corpus using Minipar and extracted nouns, verbs and adjectives, which were used to generate the LSA term by document matrix. The total number of documents is 491,384 (2.9 million tokens). After the exclusion of types occurring less than 3 times in the collection, a dictionary of 529,964 terms has been obtained. The dimension  $k$  of the LSA transformation has been set

Setting	Accuracy	Sensitivity	Specificity
<i>accept all</i>	50.00%	100%	0%
<i>random</i>	49.04%	51%	48.09%
ISP.IIM.or [9]	59%	73%	45%
ISP.JIM [9]	53%	17%	88%
$\tau = 0.7, QT = .45$	54.08%	29.9%	78.6%
$\tau = 0.8, QT = .25$	<b>61.02%</b>	61.6%	60%
$\tau = 0.8, QT = .3$	59.3%	55.8%	62.9%
$\tau = 0.9, QT = .25$	<b>60.83%</b>	62.5%	59.07%
$\tau = 0.9, QT = .45$	56.5%	60.32%	52.67%

**Table 1:** Performance Evaluation of the textual inference rules

to 100. Selectional preferences have been acquired through the selection of the fillers  $x$  of a given pattern  $p$  and its slot  $X$ , appearing in the corpus. Only nouns are clustered in this experiment. Given such lexical group  $X^p$ , the clustering algorithm results in a set of cluster  $CX^p$  given by the coherent subsets of  $X^p$ .

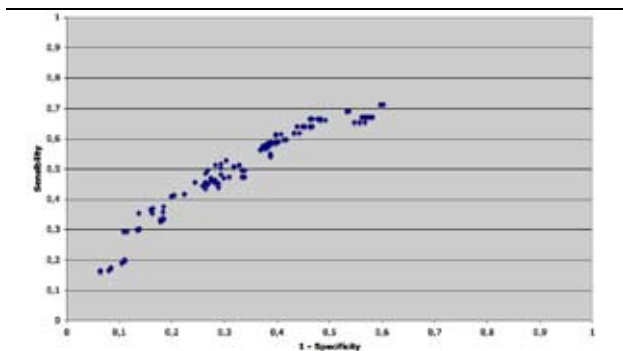
We used the development set to find the best settings of the following parameters.  $QT$ , i.e. the maximal distance allowed for cluster acceptance. We tried values ranging from 0.25 to 0.45.  $\tau$ , i.e. the coherence threshold, ranging in [0.7, 0.9].  $\sigma(r_i)$ , i.e. the tolerance function, for which two different settings have been tested but  $n = 1$  was always the best choice, with small variations for  $\beta$  and  $\alpha$ .

### 4.2 Results Analysis

Results for the best parameter settings are reported in Table 1, together with other systems’ performance. The Table shows that our system largely outperforms the baselines, also improving over the best joint and independent models in [9], of respectively 8 and 2 percentage points. In particular, our system shows a much higher balance on Sensitivity and Specificity with respect to [9], which shows pairwise highly biased values. This means that we can achieve a good level of recall on the positive instances (*Sensitivity*), while accepting a fairly low number of false positive instances (48.4%). As a major advantage, by varying the different threshold parameters, the system can be easily tuned to optimize precision or recall (this is particularly useful in those applications, such as RTE, where precise rules are required).

Figure 1 reports the complete ROC analysis of the results according to different parameter settings over the test set.

A key issue to verify the effectiveness of our method is to analyze the results of the clustering phase. With a parameter like  $QT = 0.45$  we obtained a total of **13,708** clusters (out of 56,238 analyzed slot fillers). On average this amounts to **60** cluster per slot. This ratio falls down to 26 when  $QT = 0.25$  is employed, and only 5,998 clusters are built. Results in Table 1 show that lower values of the threshold ( $QT = 0.25$ ) result in better performances. This indicates that larger clusters (low  $QT$ ) guarantee the correct level of generality for inferring selectional preference. On the contrary, small specific clusters tend to partition too much the space, thus compromising the discovery of compatible clusters and RIRs.

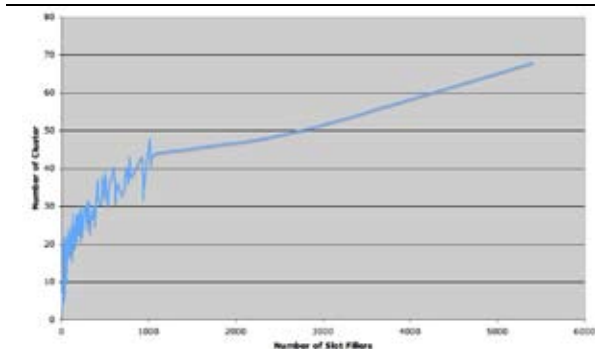


**Fig. 1:** ROC analysis over the Test Set

Pattern	Clusters (slot Y )	Coherence
bring_by	case, lawsuit, action, judge, discrimination	0.79
file_by	lawyer, attorney, prosecution, judge, counsel, defendant, Justice_Department, FBI	0.85

**Table 2:** Clusters obtained for two of the test patterns.

The plot of the average number of cluster for different numbers of slot filler as found in the corpus is reported in Fig. 2. The plot suggests that the number of clusters does not grow too much with respect to the increasing number of originating slot fillers: when thousands of different slot fillers have been found, we still have no more than 10-20 clusters. The compression factor of our method (i.e. 90% at  $QT = 0,25$ ) allows to represent a pattern by storing few representative information (i.e. cluster centroids). This compression suggests that the corpus evidence about a pattern is meaningful to the description of rules and to the modeling of selectional preferences. In fact, LSA produces high similarity values (among members of a cluster) even when a large number of fillers is considered.



**Fig. 2:** Clusters vs. different slot fillers

An example is shown in Table 2 for the Y slot of two different patterns.

## 5 Conclusions

In this paper we presented a novel method for automatically learning RIRs, relying on a geometrical model of similarity based on clustering in the reduced LSA space. Experimental evidences show that our method improves over previous approaches, also guar-

anteeing independence from existing resources and reducing computational costs. As a future work, we want to gather more empirical evidence over other collections to better estimate the effectiveness of the approach. We also plan to experiment different clustering techniques and measures. Finally, we will investigate the use of the acquired RIRs in real applications and tasks, such as RTE.

## Acknowledgments

Authors would like to thank Patrick Pantel at the ISI - University of Southern California, for having provided the experimental data and for his precious suggestions.

## References

- [1] S. Bayer, J. Burger, L. Ferro, J. Henderson, and A. Yeh. MITRE's submissions to the eu pascal rte challenge. In *Proceedings of the 1st Pascal Challenge Workshop*, Southampton, UK, 2005.
- [2] R. B. Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. In *B. Magnini and I. Dagan, editors. Proceedings of the Second PASCAL Recognizing Textual Entailment Challenge*, Venice, Italy, 2006.
- [3] S. Harabagiu and A. Hickl. Methods for using textual entailment in open-domain question answering. In *Proceedings of ACL 2006*, Sydney, Australia, 2006.
- [4] L. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research*, (9):1106-1115, 1999.
- [5] T. Landauer and S. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211-240, 1997.
- [6] D. Lin and P. Pantel. DIRT-discovery of inference rules from text. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD-01)*, San Francisco, CA, 2001.
- [7] A. Olney and Z. Cai. An orthonormal basis for entailment. In *Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference*, pages 554-559, Menlo Park, CA, May 15-17 2005. AAAI Press.
- [8] B. Pang, K. Knight, and D. Marcu. Syntax-based alignment of multiple translations: extracting paraphrases and generating new sentences. In *Proceedings of HLT/NAACL-03*, pages 49-56, Edmonton, Canada, 2003.
- [9] P. Pantel, R. Bhagat, B. Coppola, T. Chklovski, and E. Hovy. Isp: Learning inferential selectional preferences. In *Proceedings of HLT/NAACL 2007*, 2007.
- [10] P. Pantel and D. Lin. Discovering word senses from text. In *Proceedings of KDD-02*, page 613619, Edmonton, Canada, 2002.
- [11] P. Resnik. *Selection and Information: A Class-Based Approach to Lexical Relationships*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, 1993.
- [12] Y. Shinyama, S. Sekine, K. Sudo, and R. Grishman. Automatic paraphrase acquisition from news articles. In *Proceedings of HLT-02*, San Diego, CA, 2003.
- [13] K. Sudo, S. Sekine, and R. Grishman. An improved extraction pattern representation model for automatic ie pattern acquisition. In *Proceedings of ACL 2003*, 2003.
- [14] I. Szpektor, H. Tanev, I. Dagan, and B. Coppola. Scaling web-based acquisition of entailment relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004.

# Learning Shallow Semantic Rules for Textual Entailment

Marco Pennacchiotti\* & Fabio Massimo Zanzotto $\diamond$

\* Computational Linguistics, Saarland University, Saarbrücken, Germany, [penmacchiotti@coli.uni-sb.de](mailto:penmacchiotti@coli.uni-sb.de)

$\diamond$  DISP - University of Roma Tor Vergata, Roma, Italy, [zanzotto@info.uniroma2.it](mailto:zanzotto@info.uniroma2.it)

## Abstract

In this paper we present a novel technique for integrating lexical-semantic knowledge in systems for learning textual entailment recognition rules: the *typed anchors*. These describe the semantic relations between words across an entailment pair. We integrate our approach in the *cross-pair similarity* model. Experimental results show that our approach increases performance of *cross-pair similarity* learning systems.

## 1 Introduction

The Recognizing Textual Entailment (RTE) task has recently received growing attention, as a means to computationally model textual inference in Natural Language Processing (NLP) applications. Formally, given a pair of text fragments, the *Text*  $T$  and the *Hypothesis*  $H$ , the goal of an RTE system is to recognize if  $T$  entails  $H$ . Textual entailment is a key component of many NLP applications. For example, consider a Question Answering system which has to answer the question: “When did John Lennon died?”. The system could find the answer from the snippet “In 1980 Chapman killed John Lennon”, by recognizing the following implication:

$T_1$	“In 1980 Chapman killed John Lennon.”	$(E_1)$
$H_1$	“John Lennon died in 1980.”	

In the last few years, RTE challenges [1] have been organized to compare the performance of different RTE systems over a common and balanced corpus of entailment pairs  $(T, H)$ . Most strategies for RTE fall into these three categories: *lexical overlap* (e.g. [4]), *syntactic matching* (e.g. [14, 11, 9]), *entailment triggering* (e.g. [15, 5, 6]). All these approaches are plausible and effective. Also, they are fairly complementary as they recognize different set of entailment pairs [2]. Yet, today it is still not clear which approach is most appropriate for RTE; so far, only few systems successfully integrated them in a common model (e.g. [10, 5]). This lack of integration is one of the reasons of the low recognition performance (the average accuracy at the RTE-2 challenge was 0.59).

Recently, an original machine learning approach for RTE has been proposed in [16]. Its aim is to integrate *lexical overlap* and *entailment triggering*, in order to leverage complementarity and boost performance. The key idea is a similarity between pairs of texts and hypotheses, the *cross-pair similarity*, that considers the relations between words in  $T$  and  $H$ . These relations are captured using *placeholders*. This

allows the system to automatically exploit *rewrite rules*. Yet, the system suffers a major problem which highly limits its performance. Placeholders align two words if they are semantically similar, but the relation between them is not explicitly represented. This limitation can lead the learning algorithm to exploit erroneous rewrite rules.

In this paper, we present a novel method to solve the above mentioned limitation, by introducing the notion of *typed anchors*. The idea is to adopt placeholders with a semantic tag expressing the semantic relation standing between the lexicals. This intuition allows the system to exploit more semantically principled rewrite rules, which should avoid misclassifications and significantly improve performance. For example in the pair  $E_1$ , the learning algorithm would exploit the correct rule: *if the object of  $T$  aligns to the subject of  $H$ , and the verbs are in causation relation, then entailment holds*.

The paper is organized as follows. Sec. 2 reviews the cross-pair similarity model and analyzes its limits. In Sec. 3, we introduce our model for *typed anchors* aiming at integrating semantic information. Finally, in Sec. 4 we empirically assess that the use of typed anchors significantly outperforms approaches based on simple placeholders and approaches based on *lexical overlap*, *syntactic matching*, and *entailment triggering*.

## 2 Cross-pair similarity and its limits

In this section we firstly review the cross-pair similarity model used to exploit textual entailment recognition *rewrite rules*. We then analyze its limits observing how poorly defined relations among words may generate wrong rewrite rules.

### 2.1 Learning entailment rules with syntactic cross-pair similarity

The *cross-pair similarity* model [16] proposes a feature space of entailment pairs  $(T, H)$  where similarity-based learning model can exploit *rewrite rules* defined in training examples. The key idea is to define a *cross-pair similarity*  $K_S((T', H'), (T'', H''))$  that takes into account relations among words within a pair. This is done using *placeholders*. A *placeholder* co-indexes two substructures in the parse trees of  $T$  and  $H$ , indicating that such substructures are related. At the word level (i.e. leaves) placeholders link pairs of words which are highly similar: these pairs are called *anchors*. For example, the sentence pair, “All companies file annual

reports” implies “All insurance companies file annual reports”, would be represented as follows:

$$\begin{array}{l} \overline{T_2} \quad (S \text{ (NP}_{\square} \text{ (DT All) (NNS}_{\square} \text{ companies))} \\ \quad (VP_{\square} \text{ (VBP}_{\square} \text{ file) (NP}_{\square} \text{ (JJ}_{\square} \text{ annual} \\ \quad (\text{NNS}_{\square} \text{ reports})))) \\ \hline H_2 \quad (S \text{ (NP}_{\square} \text{ (DT All) (NNP Fortune) (CD} \\ \quad 50) (\text{NNS}_{\square} \text{ companies)) (VP}_{\square} \text{ (VBP}_{\square} \\ \quad \text{file) (NP}_{\square} \text{ (JJ}_{\square} \text{ annual) (NNS}_{\square} \text{ re-} \\ \quad \text{ports})))) \end{array} \quad (E_2)$$

where the placeholders  $\square$ ,  $\square$ , and  $\square$  indicate the relations between the structures of  $T$  and those of  $H$ , and *companies/companies* is an example of anchor.

Placeholders help to determine if two pairs share the same *rewrite rule* by looking at the subtrees that they have in common. For example, suppose we have to determine if “In autumn, all leaves fall” implies “In autumn, all maple leaves fall”. The related co-indexed representation is:

$$\begin{array}{l} \overline{T_3} \quad (S \text{ (PP (IN In) (NP (NN}_{\square} \text{ automm)))} \\ \quad (, ,) (\text{NP}_{\square} \text{ (DT all) (NNS}_{\square} \text{ leaves))} \\ \quad (\text{VP}_{\square} \text{ (VBP}_{\square} \text{ fall}))) \\ \hline H_3 \quad (S \text{ (PP (IN In) (NP}_{\square} \text{ (NN}_{\square} \text{ automm)))} \\ \quad (, ,) (\text{NP}_{\square} \text{ (DT all) (NN maple) (NNS}_{\square} \\ \quad \text{leaves)) (VP}_{\square} \text{ (VBP}_{\square} \text{ fall}))) \end{array} \quad (E_3)$$

$E_2$  and  $E_3$  share the following subtrees:

$$\begin{array}{l} \overline{T_4} \quad (S \text{ (NP}_{\square} \text{ (DT all) (NNS}_{\square}) (VP}_{\square} \\ \quad (\text{VBP}_{\square})) \\ \hline H_4 \quad (S \text{ (NP}_{\square} \text{ (DT all) (NN) (NNS}_{\square})} \\ \quad (\text{VP}_{\square} \text{ (VBP}_{\square})) \end{array} \quad (R_4)$$

These subtrees represent the *rewrite rule* that  $E_2$  and  $E_3$  have in common. Then,  $E_3$  can be likely classified as a valid entailment, as it shares the rule with the valid entailment  $E_2$ .

More details on the *cross-pair similarity* model can be found in [16] and an efficient algorithm for its computation is described in [13].

## 2.2 Limits of the syntactic cross-pair similarity

Learning from examples using cross-pair similarity is an attractive and effective approach, as results of the RTE-2 challenge show [1]. Yet, the cross-pair similarity strategy, as any machine learning approach, is highly sensitive on how the examples are represented in the feature space. An incorrect or inaccurate feature modelling can strongly bias the performance of the classifier.

This problem is even more evident in kernel-based methods, where the feature space is implicit, and the classifier can only rely on the syntactic structure of the examples. Then, as in the cross-pair similarity approach placeholders play an important role within the syntactic tree, the classifier can then be highly biased, if they convey incomplete or incorrect information.

Consider for example the following text-hypothesis pair, which can lead to an incorrect rule, if misused.

$$\begin{array}{l} \overline{T_5} \quad \text{“For my younger readers, Chapman} \\ \quad \text{killed John Lennon more than twenty} \\ \quad \text{years ago.”} \\ \hline H_5 \quad \text{“John Lennon died more than twenty} \\ \quad \text{years ago.”} \end{array} \quad (E_5)$$

In the basic cross-pair similarity model, the decision process can use rules like the following:

$$\begin{array}{l} \overline{T_6} \quad (S \text{ (NP}_{\square} \text{ (VP}_{\square} \text{ (VBD}_{\square} \text{ (NP}_{\square} \\ \quad (\text{ADVP}_{\square} \text{ ))))} \\ \hline H_6 \quad (S \text{ (NP}_{\square} \text{ (VP}_{\square} \text{ (VBD}_{\square} \\ \quad (\text{ADVP}_{\square} \text{ ))))} \end{array} \quad (R_6)$$

where *kill* and *die* are anchored by the  $\square$  placeholder. This rule is useful to classify examples like:

$$\begin{array}{l} \overline{T_7} \quad \text{“Cows are vegetarian but, to save} \\ \quad \text{money on mass-production, farmers fed} \\ \quad \text{cows animal extracts.”} \\ \hline H_7 \quad \text{“Cows have eaten animal extracts.”} \end{array} \quad (E_7)$$

but it will clearly fail when used for:

$$\begin{array}{l} \overline{T_8} \quad \text{“FDA warns migraine medicine makers} \\ \quad \text{that they are illegally selling migraine} \\ \quad \text{medicines without federal approval.”} \\ \hline H_8 \quad \text{“Migraine medicine makers declared} \\ \quad \text{that their medicines have been ap-} \\ \quad \text{proved.”} \end{array} \quad (E_8)$$

where *warn* and *declare* are anchored as generically similar verbs.

The limitation of the cross-pair similarity measure is then that placeholders do not convey the semantic knowledge needed in cases such as the above, where the semantic relation between connected verbs is essential.

## 3 Adding semantic information to cross-pair similarity

In the previous section we showed that the cross-pair similarity approach lacks the lexical-semantic knowledge for anchoring words. In the examples, the missed knowledge is the type of semantic relation between the main verbs. The relation that links *kill* and *die* is not a generic similarity, as a WordNet based similarity measure would suggest, but a more specific causal relation. The exploited *rewrite rule*  $R_6$  holds only for verbs in such relation. It is correctly applied in example  $E_7$ , as *feed* causes *eat*. Yet, it gives a wrong suggestion in example  $E_8$ , as *warn* and *declare* are related by a generic similarity relation.

The type of relation that links two words (*anchor type*) seems to be mandatory, in order to exploit correct rules. The problem is then to encode this information in the syntactic trees along with the placeholders.

In this section we describe how we encode the anchor types in the syntactic trees, by using two models: the *typed anchor* ( $ta$ ) and the *propagated typed anchor* ( $tap$ ) models. As anchoring words of  $H$  with words in  $T$  is the basic step, before describing the models (Sec. 3.2), we shortly revise how anchors are selected and how they are encoded in the trees (Sec. 3.1).

### 3.1 Anchors and Placeholders

As many other approaches (e.g., [4]), our anchoring model is based on a similarity measure between words  $sim_w(w_t, w_h)$ . We use a two-step greedy algorithm to anchor the content words (verbs, nouns, adjectives,



and adverbs) in the hypothesis  $W_H$  to words in the text  $W_T$ . In the first step, each word  $w_h$  in  $W_H$  is connected to all words  $w_t$  in  $W_T$  that have the highest similarity  $sim_w(w_t, w_h)$ . As result, we have a set of anchors  $A \subset W_T \times W_H$  and the subset  $W'_T \subseteq W_T$  of words in  $T$  connected with a word in  $H$ . In the second step, we select the final anchor set  $A' \subseteq A$ , as the bijective relation between  $W_H$  and  $W'_T$  that mostly satisfies a locality criterion: whenever possible, words of constituent in  $H$  should be related to words of a constituent in  $T$ . See [16] for more details on the adopted word similarity  $sim_w(w_t, w_h)$ .

Once the set  $A'$  is found, anchors are encoded in the syntactic trees with placeholders. Placeholders are put on the pre-terminal nodes of the anchored words. Then, they climb up in the tree according to this rule: *constituent nodes in the syntactic trees take the placeholder of their semantic heads*. This latter step guarantees that any subtree has the relational information. The final tree explicitly indicates how  $T$  relates to  $H$  using co-indexing (see  $E_2$ ).

### 3.2 Typing anchors and placeholders

Our goal is to augment the co-indexed syntactic trees with typed anchors. To do that, we first have to decide what type of semantic relations we want to represent in the typed anchors (Sec. 3.2.1). Then, we need to define how to encode this information in the syntactic trees (Sec. 3.2.2).

#### 3.2.1 Defining anchor types

The idea of introducing anchor types is in principle very simple. Yet, this may be not effective: attempts to introduce semantic information in RTE systems have often failed. A main reason for this failure is that any model using semantic information has the problem of dealing with ambiguity.

To investigate the validity of our idea, we then need to focus on a small set of relevant relation types. A valuable source of relation types among words is WordNet. We choose to integrate in our system three relations: *part-of*, *antonymy*, and *verb entailment*. This small set seems to be a correct choice, as it is relevant for many entailment cases such as those presented in Sec. 2.

We also define two more general anchor types: *similarity* and *surface matching*. The first type links words which are similar according to the WordNet similarity measure described in [7]. This type is intended to capture *synonymy* and *hyperonymy*. The second type is activated when words or lemmas match: it captures semantically equivalent words. The complete set of relation types used in the experiments is given in Table 1.

#### 3.2.2 Augmenting placeholders with anchor types

Once anchor types have been defined, it is necessary to decide how to integrate their information in the syntactic trees. We apply a strategy similar to that adopted for placeholders, described in Sec. 3.1. However, the main problem is then to decide how the se-

Rank	Relation Type	Symbol
1.	<i>antonymy</i>	$\leftrightarrow$
2.	<i>part-of</i>	$\subset$
3.	<i>verb entailment</i>	$\leftarrow$
4.	<i>similarity</i>	$\approx$
5.	<i>surface matching</i>	$=$

Table 1: Ranked anchor types

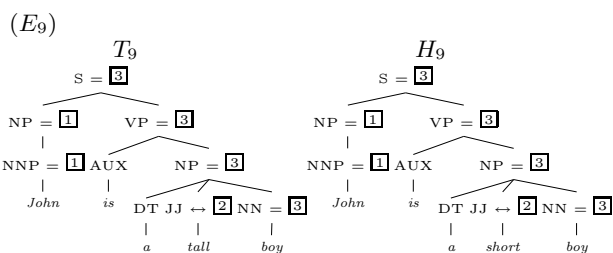
matic information should be encoded. We experiment two possible models:

**typed anchor model (ta)** : anchor types augment only the pre-terminal nodes of the syntactic tree;

**propagated typed anchor model (tap)** : anchors climb up in the syntactic tree according to some specific *climbing-up rules*, similarly to what done for placeholders.

The **ta model** is easy to implement: typed anchors simply augment the pre-terminals of anchored words. The **tap model** is apparently more suitable for our purpose. The anchor type information is repeated in several tree fragments. As tree fragments are compared in the cross-pair similarity, this guarantees that the information is used in the decision process.

Unfortunately, the *tap* model is more complex, as it depends on strategy adopted for the anchor type climbing-up. The strategy must account for how anchors that climb up to the same node should interact. We implement our strategy by using *climbing-up rules*, as done in the case of placeholders. Yet, in this case, rules must consider the semantic information of the typed anchors. The choice of correct climbing-up rules is critical, as an incorrect rule could alter completely the semantics of the tree. In the case of placeholders, the climbing-up rule states that a constituent in the syntactic tree takes the placeholder of its semantic head. It is easy to demonstrate that in the case of typed anchors this rule would have disastrous effects. For example, consider the following false entailment pair:

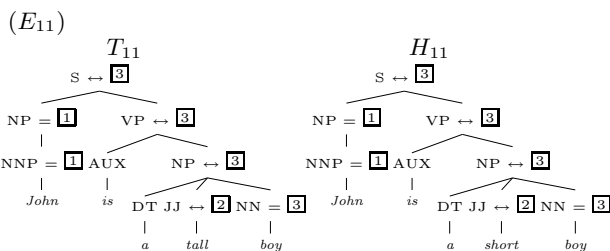


In the example, we apply the abovementioned rule: the typed anchor  $=3$  climbs up to the pre-terminal node NP, instead of the typed anchor  $\leftrightarrow 2$ , as it is the head of the constituent. If modelled in this way, this false entailment pair could generate, among others, the incorrect rewrite rule:

$$\begin{array}{l}
 \hline
 T_{10} \quad (S=3 \quad (NP=1) \quad (VP=3 \quad (AUX \quad is) \\
 \quad \quad \quad (NP=2))) \\
 \hline
 H_{10} \quad (S=3 \quad (NP=1) \quad (VP=3 \quad (AUX \quad is) \\
 \quad \quad \quad (NP=2))) \\
 \hline
 \end{array} \quad (R_{10})$$

which states: *if two fragment have the same syntactic structure  $\bar{S}(NP, VP(AUX, NP))$ , and there is a semantic equivalence ( $=$ ) on all constituents, then entailment does not hold.* This rule is wrong, as in that case entailment would hold (as all substructures are semantically equivalent).

The problem is that the wrong typed anchor climbed up the tree: we need the antonym anchor on the adjective (*tall/short*) to climb up, instead of the matching anchor on the noun (*boy/boy*), in order to exploit a correct rule. Our strategy must then implement a climbing-up rule producing these trees:



In this case the pair generates correct rewrite rules, such as:

$$\frac{T_{12} \quad (S \leftrightarrow \boxed{3} \quad (NP = \boxed{1}) \quad (VP \leftrightarrow \boxed{3}) \quad (AUX \text{ is}) \quad (NP \leftrightarrow \boxed{2}))}{H_{12} \quad (S \leftrightarrow \boxed{3} \quad (NP = \boxed{1}) \quad (VP \leftrightarrow \boxed{3}) \quad (AUX \text{ is}) \quad (NP \leftrightarrow \boxed{2}))} \quad (R_{12})$$

The rule states: *if two fragment have the same syntactic structure  $\bar{S}(NP_1, VP(AUX, NP_2))$ , and there is an antonym type ( $\leftrightarrow$ ) on the  $S$  and  $NP_2$ , then entailment does not hold.*

The above example shows that the anchor type that has to climb up depends on the structure of the constituents. This can lead to a very complex model. Luckily, this intuition can be also captured by a simpler approximation. Instead of having climbing-up rules for each constituent type, we can rely on a ranking of the anchor types (as the one reported in Tab. 1). The anchor type that climbs up is the one that has a higher rank. In the example, this strategy produces the correct solution, as *antonymy* has a higher rank than *surface match*. We then implement in our model the following climbing-up rule: *if two typed anchors climb up to the same node, give precedence to that with the highest ranking in the ordered set of types  $\mathcal{T} = (\leftrightarrow, \subset, \leftarrow, \approx, =)$ .* Our ordered set  $\mathcal{T}$  is consistent with common sense intuitions. In the next section we will empirically demonstrate its validity by reporting experiment evidences.

## 4 Experimental Results

In this section, we present empirical evidence to support the claims of the paper. In particular, we compare our **ta** and **tap** approaches with the strategies for RTE: *lexical overlap*, *syntactic matching* and *entailment triggering*. To perform the comparison we implemented these strategies in our machine learning platform for RTE, which also allows to combine them in more complex configurations.

### 4.1 Experimental Setup

For our evaluation we use the same methodology adopted at the RTE challenges [1]. The RTE task is to classify a test set of entailment pairs as true or false entailment, by relying on an annotated development set. Systems are evaluated on their prediction accuracy. We here adopt 4-fold cross validation, to obtain more reliable evidences. We use SVM-light-TK [12] as learning algorithm, which encodes the needed tree kernel functions in SVM-light [8].

We perform our experiments using the RTE-2 dataset, composed of 1600 entailment pairs from the RTE-2 challenge (800 true and 800 false entailment).

We evaluate *ta* and *tap* by comparing the performance of SVM with feature sets representing different *basic approaches*. We also experiment more complex feature spaces, representing *combined approaches*:

**tree** : the standard cross-similarity model described in Sec.2. Its comparison with *ta* and *tap* indicates the effectiveness of our approaches;

**lex** : a standard approach based on *lexical overlap*. The classifier uses as only feature the lexical overlap similarity score described in [4];

**synt** : a standard approach based on *syntactic matching*. The classifier uses as only feature a syntactic similarity score. A syntactic similarity measure  $synt(T, H)$  is used to compute the score, by comparing all the substructures of the dependency trees of  $T$  and  $H$ , in line with approaches like [14, 11, 9]. This syntactic similarity is derived using the tree kernel similarity  $K_T$  [3] as follows:  $synt(T, H) = K_T(T, H)/|H|$  where  $|H|$  is the number of subtrees in  $H$ ;

**lex+ta**, **lex+tap** : these configurations mix lexical overlap and our typed anchor approaches;

**lex+tree** : the comparison of this configuration with *lex+ta* and *lex+tap* should further support the validity of our intuition on typed anchors;

**lex+synt** : by comparing this configuration with *lex* and *synt* we aim at verifying if lexical and syntactic methods are complementary, as reported in [2];

**lex+trig** : this configuration mixes lexical overlap with basic entailment triggering features like in [15, 5, 6]. We use the following features: 1) *SVO* that tests if  $T$  and  $H$  share a similar subj-verb-obj construct; 2) *Apposition* that tests if  $H$  is a sentence headed by the verb *to be* and in  $T$  there is an apposition that states  $H$ ; 3) *Anaphora* that tests if the SVO sentence in  $H$  has a similar wh-sentence in  $T$  and the wh-pronoun may be resolved in  $T$  with a word similar to the object or the subject of  $H$ .

### 4.2 Results Analysis

Table 2 reports the 4-folds and overall accuracy of the different feature spaces. The left part of the table shows the performance of the *basic approaches*, while the right those of the *combined approaches*.

	<i>ta</i>	<i>tap</i>	<i>tree</i>	<i>lex</i>	<i>synt</i>	<i>lex + ta</i>	<i>lex + tap</i>	<i>lex + tree</i>	<i>lex + synt</i>	<i>lex + trig</i>
Mean	61.29	62.47	61.35	61.81	58.28	63.94	63.81	63.68	61.94	61.56
Std dev	± 2.54	± 2.68	± 2.32	± 1.74	± 2.48	± 1.59	± 1.24	± 1.59	± 1.65	± 2.03

**Table 2:** 4-folds accuracy using different feature sets over the RTE-2 dataset.

Results for the **basic approaches** show that *tap* outperforms all the other feature sets<sup>1</sup>. In particular, it guarantees an improvement of +1.12% accuracy over *tree*, suggesting that the addition of typed anchors to the basic cross-pair similarity model is indeed successful. This demonstrates that syntax is not enough, and that lexical-semantic knowledge, and in particular the explicit representation of word level relations, plays a key role in RTE. This is even more evident by comparing results to the pure syntactic approach *synt*, that achieves only 58.28% accuracy.

Also, *tap* outperforms *lex*, supporting a complementary conclusion: lexical-semantic knowledge does not cover alone the entailment phenomenon, but needs some syntactic evidence.

An overall analysis of basic systems further substantiate our intuition: approaches mixing syntax and lexical knowledge (*tap*) outperform method based on lexical knowledge (*lex*), which in turn outperform syntactic methods with weak lexical knowledge (*tree*) and pure syntactic methods (*synt*).

The surprisingly low performance of *ta* reveal that encoding typed anchors only at the pre-terminal level is not a sufficiently strong information for the learning algorithm. This further suggests the intuition the semantics of word relations is indeed central.

Results for **combined approaches** reveals the difficulty of integrating lexical and syntactic information. The *lex + synt* model does not substantially improves over *lex*. This suggests that a trivial integration of lexical overlap and syntactic matching between *T* and *H* is not effective. On the contrary, the use of cross-pair similarity together with lexical overlap (*lex + tree*) is successful, as accuracy improves +1.87% and +2.33% over the related basic methods (respectively *lex* and *tree*). The conclusion is then that cross-pair information across different pairs (in form of *rewrite rules*) and lexical information inside each pair are indeed both relevant. Again, our method mixed with *lex* achieve the best performance, further supporting the usefulness of typed anchors.

In general, our results also empirically confirm the manual analysis on the RTE-2 dataset performed in [2], suggesting that lexical and syntactic level are complementary for RTE, i.e. they recognize different set of entailment pairs.

## 5 Conclusions

Effectively integrating semantic knowledge in textual entailment recognition systems is one of the major problem in the area. In this paper we presented a

simple but effective model to integrate lexical semantic knowledge in a learner of rewrite rules for detecting textual entailment. Experimental results show that this is a promising model that may be used to integrate more complex semantic information.

## References

- [1] R. Bar-Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, and I. Magnini, Bernardo Szpektor. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy, 2006.
- [2] R. Bar-Haim, I. Szpektor, and O. Glickman. Definition and analysis of intermediate entailment levels. In *Proc. of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, 2005.
- [3] M. Collins and N. Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of ACL02*, 2002.
- [4] C. Corley and R. Mihalcea. Measuring the semantic similarity of texts. In *Proc. of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, 2005.
- [5] A. Hickl, J. Williams, J. Bensley, K. Roberts, B. Rink, and Y. Shi. Recognizing textual entailment with lcc's groundhog system. In B. Magnini and I. Dagan, editors, *Proc. of the Second PASCAL RTE Challenge*, Venice, Italy, 2006.
- [6] D. Inkpen, D. Kipp, and V. Nastase. Machine learning experiments for textual entailment. In B. Magnini and I. Dagan, editors, *Proc. of the Second PASCAL RTE Challenge*, Venice, Italy, 2006.
- [7] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the 10th ROCLING*, Tapei, Taiwan, 1997.
- [8] T. Joachims. Making large-scale svm learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods-Support Vector Learning*. MIT Press, 1999.
- [9] S. Katrenko and P. Adriaans. Using maximal embedded syntactic subtrees for textual entailment recognition. In B. Magnini and I. Dagan, editors, *Proc. of the Second PASCAL RTE Challenge*, Venice, Italy, 2006.
- [10] B. MacCartney, T. Grenager, M.-C. de Marneffe, D. Cer, and C. D. Manning. Learning to recognize features of valid textual entailments. In *Proc. of the HLT/NAACL*, 2006.
- [11] E. Marsi, E. Krahmer, W. Bosma, and M. Theune. Normalized alignment of dependency trees for detecting textual entailment. In B. Magnini and I. Dagan, editors, *Proceedings of the Second PASCAL RTE Challenge*, Venice, Italy, 2006.
- [12] A. Moschitti. Making tree kernels practical for natural language learning. In *Proceedings of EACL'06*, Trento, Italy, 2006.
- [13] A. Moschitti and F. M. Zanzotto. Fast and effective kernels for relational learning from texts. In *Proceedings of the International Conference of Machine Learning (ICML)*, Corvallis, Oregon, 2007.
- [14] V. Rus. Dependency-based textual entailment. In *FLAIRS Conference*, pages 110–109, 2006.
- [15] R. Snow, L. Vanderwende, and A. Menezes. Effectively using syntax for recognizing false entailment. In *Proc. of HLT/NAACL 2006*, New York, 2006.
- [16] F. M. Zanzotto and A. Moschitti. Automatic learning of textual entailments with cross-pair similarities. In *Proceedings of the 21st Coling and 44th ACL*, pages 401–408, Sydney, Australia, July 2006.

<sup>1</sup> According to the sign-test, *tap* outperforms with more than 90% of statistical significance all the other basic approaches except the *lex*. In this case, the statistical significance is lower.

# A French Interaction Grammar

Guy Perrier  
LORIA - universit  Nancy 2  
BP 239  
54506 Vandœuvre-lès-Nancy cedex - France  
*perrier@loria.fr*

## Abstract

We present a relatively large coverage French grammar written with the formalism of Interaction Grammars. This formalism combines two key ideas: the grammar is viewed as a constraint system, which is expressed through the notion of tree description, and the resource sensitivity of natural languages is used as a syntactic composition principle by means of a system of polarities. We give an outline of the expressivity of the formalism by modelling significant linguistic phenomena and we show that the grammar architecture provides for re-usability and tractability, which is crucial for building large coverage resources: a modular source grammar is distinguished from the object grammar which results from the compilation of the first one, and the lexicon is independent of the grammar. Finally, we present the results of an evaluation of the grammar achieved with the LEOPAR parser with a test suite of sentences.

## Keywords

Syntax, grammatical formalism, tree description, polarity, categorial grammar, unification grammar, interaction grammar

## 1 Introduction

The goal of our work is to model natural languages starting from linguistic knowledge and giving a central role to experimentation. For this, we need to express the linguistic knowledge by means of grammars and lexicons with the largest possible coverage: grammars have to represent all common linguistic phenomena and lexicons have to include the most frequent words with their most frequent use. As everyone knows, building such resources is a very hard task.

Firstly, we have to choose the formalism to represent the grammar. Currently, there is no leader among the formalisms used in the scientific community. Each of the most popular formalisms has its own advantages and drawbacks. We have designed a new formalism, Interaction Grammars (IG), the goal of which is to synthesize two key ideas, expressed in two kinds of formalisms up to now: using the resource sensitivity of natural languages as a principle of syntactic composition, which is a characteristic feature of Categorical Grammars (CG) [9], and viewing grammars as constraint systems, which is a feature of unification grammars such as LFG [1] or HPSG [11].

Although we use an original formalism, we are concerned with re-usability, which is expressed in two ways. Like with for programming languages, we distinguish two levels in the grammar. The *source grammar* aims at representing linguistic generalisations and it is written by a human, while the *object grammar* is directly usable by a NLP system and results from the compilation of the first one. In our case, we used XMG [2], a tool devoted to this goal. XMG provides a high level language for writing a source grammar and a compiler which translates this grammar into an operational object grammar. The grammar is also designed in such a way that it can be linked with a lexicon independent of the formalism, where entries appear as feature structures.

The goal of the article is to show that it is possible to build realistic grammatical resources, which integrate a refined linguistic knowledge with a large coverage, and for this, we have chosen an experimental approach with the construction of a French grammar.

## 2 Interaction Grammars

IG [5, 6] is a grammatical formalism which is devoted to the syntax and semantics of natural languages and which uses two notions: *tree description* and *polarity*.

### 2.1 Tree Descriptions

In a derivational view of the syntax of natural languages, the basic objects are trees and they are composed together in a more or less sophisticated way: by substitution in Context Free Grammars, by adjunction in Tree Adjoining Grammars, by application and abstraction in Categorical Grammars ... Taking our view from the Model Theory [7], we do not directly manipulate trees but properties which are used to describe them, in other words tree descriptions [10]. This approach is very flexible as it allows the expression of elementary properties in a totally independent way, as they can be freely combined.

A tree description can be viewed either as an underspecified tree, or as the specification of a tree family, each tree being a model of the specification. Figure 1 gives an example of a tree description, which is associated with the relative pronoun *qui* (who), used inside a prepositional complement. This use gives rise to the phenomenon of *pied piping* as the following example illustrates: *Jean [à la femme de qui] Pierre sait qu'on a présenté Marie □, est ingénieur (Jean [to whose wife]*

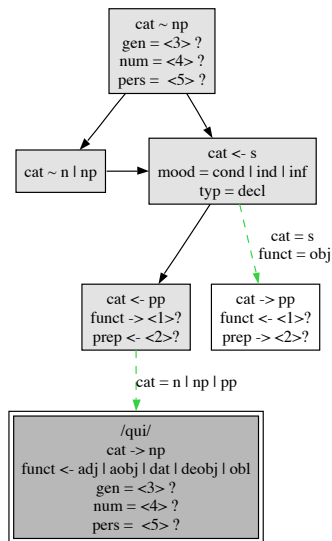


Fig. 1: Tree description associated with the relative pronoun *qui* used inside a prepositional complement

*Pierre knows someone presented Marie*  $\square$ , *is an engineer*). This example is covered by the description of figure 1.<sup>1</sup>

A tree description is a finite set of nodes structured by two kinds of relations: *dominance* and *precedence*. Dominance relations can be immediate or large (respectively solid and dashed down arrows in figure 1). Constraints can be put on intermediate nodes for large dominance relations. Precedence relations (horizontal arrows in figure 1) can also be immediate or large.

Nodes, which represent constituents, are labelled with features describing their morpho-syntactic properties. Feature values are atoms or atom disjunctions and they can be shared with the help of a co-indexation mechanism.<sup>2</sup> Nodes can be *Empty* (the white box in figure 1) or *Full*, according to whether they have an empty phonological form or not. Full nodes can be *Anchors* (the dark box in figure 1), if they anchor a word of the language.

## 2.2 Polarities

Polarities are used to express the saturation state of syntactic trees. They are attached to features that label description nodes with the following meaning:

- a positive feature  $t \rightarrow v$  expresses an available resource, which must be consumed;
- a negative feature  $t \leftarrow v$  expresses an expected resource, which must be provided; it is the dual of a positive feature;
- a neutral feature  $t = v$  expresses a linguistic property that is not a consumable resource.

<sup>1</sup> The extracted prepositional phrase is put between square brackets and its trace in the relative clause is represented by the  $\square$  symbol.

<sup>2</sup> When two features share the same value, a common index  $\langle n \rangle$  is put before their values. When a feature value is the disjunction of all elements of a domain, this value is denoted with "??".

- a virtual feature  $t \sim v$  expresses a linguistic property that needs to be realised by combining with an actual feature (an actual feature is a positive, negative or neutral feature).

In figure 1, the empty node representing the trace of the prepositional phrase extracted from the relative clause carries a positive feature  $cat \rightarrow pp$  and a negative feature  $funct \leftarrow \langle 1 \rangle?$ , which means that this node provides a prepositional phrase that needs to receive a syntactic function. The tree root carries a virtual feature  $cat \sim np$  which means that the node represents a virtual noun phrase which has to combine with an actual noun phrase.

The descriptions labelled with polarised feature structures are called *polarised tree descriptions (PTDs)* in the rest of the article.

## 2.3 Grammars as constraint systems

A particular interaction grammar is defined by a finite set of elementary PTDs, which generates a tree language. A tree belongs to the language if it is a model of a finite set of elementary PTDs with two properties:

- It is *saturated*: every positive feature  $t \rightarrow v$  is matched with its dual feature  $t \leftarrow v$  in the model and vice versa. Moreover, every virtual feature has to find an actual corresponding feature in the model.
- It is *minimal*: the model has to add a minimum of information to the initial descriptions (it cannot add immediate dominance relations or features that do not exist in the initial descriptions).

Then, parsing reduces to the resolution of a constraint system. It consists of building all saturated and minimal models of a finite set of elementary PTDs. In practice, our grammar is totally lexicalized: each elementary PTD has a unique anchor, which is used for linking the description with a word of the language. In this way, in the parsing of a sentence, it is possible to select the only PTDs that are anchored by words of the sentence. The set of PTDs being selected, the building of a saturated and minimal model is performed step by step by means of a merging operation between nodes, which is guided by one of the following constraints:

- neutralise a positive feature with a negative feature having the same name and carrying a value unifiable with the value of the first feature;
- realise a virtual feature by combining it with an actual feature (a positive, negative or neutral feature) having the same name and carrying a value unifying with the value of the first feature.

The constraints of the description interact with node merging to entail a partial superposition of their contexts represented by the tree fragments in which they are situated. To summarise, IG combine the strong points of two families of formalisms: the flexibility of *Unification Grammars* and the saturation control of *Categorial Grammars*.

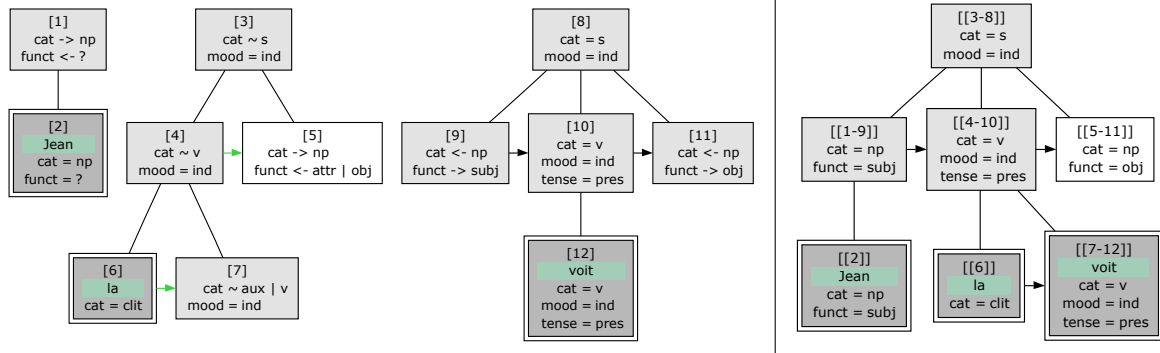


Fig. 2: PTD associated with the sentence *Jean la voit* and its minimal saturated model

Figure 2 presents an example of parsing for the sentence *Jean la voit* (*Jean sees her*).<sup>3</sup> The left side shows the set of initial PTDs associated with the sentence by the grammar. The grammar being lexicalized, each PTD is anchored by a word of the sentence and it has been extracted from a lexicon. These PTDs have been gathered in a unique PTD and precedence relations between anchors have been added to express word order in the sentence. These relations do not appear in figure 2.

The computation of the model shown on the right side of figure 2 from the initial description shown on the left side is performed by a sequence of 3 node mergings.<sup>4</sup> The interaction of tree constraints with these mergings entails two other mergings and a partial tree superposition.

### 3 The expressivity of Interaction Grammars

In the limits of this article, we have chosen to illustrate three aspects which are especially significant.

#### 3.1 Unbounded dependencies and underspecified dominance relations

Underspecified dominance relations are used to represent unbounded dependencies and the feature structures that can be associated with these relations allow the expression of constraints on these dependencies: barriers to extraction for instance.

Relative pronouns, such as *qui* or *lequel*, give rise to pied piping as the following sentence shows: *Jean [dans l'entreprise de **qui**] Marie sait que l'ingénieur travaille □, est malade (Jean [in whose firm] Marie knows that the engineer works □, is ill):*

- There is a first unbounded dependency between the verb *travaille* and its extracted complement *dans l'entreprise de qui*. The trace of the extracted complement is denoted by the □ symbol. The dependency is modelled in the PTD associated with the *qui* relative pronoun represented

in figure 1 by means of an underspecified dominance relation. The constraint linked to this dominance relation expresses that the dependency of the prepositional phrase on the verb of which it is the complement can only cross an unspecified sequence of embedded object clauses.

- Inside the prepositional phrase, there is a second unbounded dependency between the head of the constituent and the *qui* relative pronoun, which can be embedded arbitrarily deeply. This dependency is also represented in figure 1 with an underspecified dominance relation and the linked constraint expresses that all embedded constituents from the prepositional phrase to the *qui* relative pronoun are common nouns, noun phrases or prepositional phrases.

#### 3.2 Polarities used for modelling negation

In French, negation can be expressed with the help of the particle *ne* paired with a specific determiner, pronoun or adverb. The position of the particle *ne* is fixed before an inflected verb but the second component of the pair, if it is a determiner like *aucun* or a pronoun like *personne*, can have a relatively free position in the sentence, as illustrated by the following examples:

- Jean ne parle à aucun collègue* (*Jean speaks to no colleague*).
- Jean ne parle à la femme d'aucun collègue* (*Jean speaks to the wife of no colleague*).
- Aucun collègue de Jean ne parle à sa femme* (*No colleague of John's speaks to his wife*).

As figure 3 shows, the pairing of *ne* with *aucun* is expressed with a *neg* polarised feature attached to the node representing the maximal projection of the verbal kernel: *aucun* is waiting for such a feature, which will be provided by *ne*. The relatively free position of *aucun* is expressed by an underspecified dominance relation of the node representing the clause on the noun phrase that it introduces. The constraint linked to this dominance relation expresses the fact that *aucun* can only introduce arguments of the verbal head of the sentence or complements of these arguments.

<sup>3</sup> We have simplified the figure by ignoring agreement features.

<sup>4</sup> The head of each node includes the numbers of the nodes from the initial PTD which have been merged.



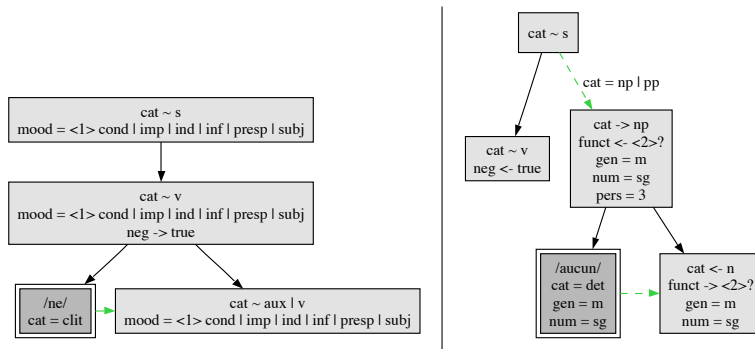


Fig. 3: PTDs respectively associated with the particle *ne* and the determiner *aucun*

### 3.3 The adjunction of modifiers by means of virtual polarities

In French, the position of adjuncts in the sentence is relatively free, as illustrated by the following example. In the sentence  $\square$  *Jean*  $\square$  *va*  $\square$  *rendre visite*  $\square$  *à Marie*  $\square$  (*Jean is going to visit Marie*), the sentence modifier *le soir* (*tonight*) can appear at any position marked with a  $\square$  symbol, according to different communicative goals.

The virtual polarity  $f \sim v$  did not exist in the previous version of IG [6]. Modifier adjunction was performed by addition of a new level in the syntactic tree of the constituent being modified. Sometimes, introducing an additional level is justified linguistically, but in most cases it introduces artificial complexity and ambiguity. Taking again an idea of [4], with his system of black and white polarities, we have introduced virtual polarities. This allows a modifier to be added as a new daughter of the node that it modifies without changing the rest of the syntactic tree, in which the modified node is situated. This operation is called *sister adjunction* and it is used in some formalisms: dependency grammars, description substitution grammars [8]. This way of modelling modifiers is more flexible and it allows the previous examples to be treated without difficulty, including parenthetical clauses.

## 4 The architecture of the grammar

### 4.1 The modular organisation of the grammar

The grammar has been built with the XMG tool [2], which allows grammars to be written with a high level of abstraction in a modular setting and to be compiled into low level grammars, usable by NLP systems.

A grammar is organised as a class hierarchy by means of two composition operations: *conjunction* and *disjunction*. It is also structured according to several dimensions, which are present in all classes. Our grammar uses only two dimensions: the first one is the syntactic dimension, where objects are PTDs, and the second one is the dimension of the interface with the lexicon, where objects are feature structures.

To define the conjunction of two classes one needs to

specify the way of combining the components of each dimension: for the syntactic dimension, PTD union is performed; for the lexicon interface dimension, it is realised as unification between feature structures.

The current grammar is composed of 448 classes, including 121 terminal classes, which are compiled into 2059 PTDs. These classes are ranked by family. Some classes from a family can be used in the definition of classes belonging to another family. This is the case for instance for the *Complement* family, which include classes related to complements of predicative structures. It is used by three other families: *Adjective*, *Noun* and *VerbDiathese*, which respectively refer to adjectives, nouns and various verbal diatheses.

### 4.2 The link with a lexicon independent of the formalism

The grammar, in its current setting, is totally lexicalised: each elementary PTD of the grammar has a unique anchor node intended to be linked with a word of the language. Each PTD is associated to a feature structure, which describes a syntactic frame corresponding to words able to anchor the PTD, the description being independent of the formalism. This feature structure constitutes the PTD interface with the lexicon.

The set of features used in the interfaces differs from that used in PTDs because they do not play the same role: they do not aim at describing syntactic structures but they are used for describing the morpho-syntactic properties of the words of the language in a way independent of the formalism.

The left side of figure 4 shows a non anchored PTD describing the syntactic behaviour of a transitive verb in the active voice. The PTD is accompanied by its interface, which is a two level feature structure.

The lexicon associates words of the language to syntactic frames in a form identical to the PTD interfaces. For instance, the central part of figure 4 shows a lexical entry for the verb *voit* in its transitive use.

The PTD anchoring is then performed by unification of the PTD interfaces with the compatible entries of the lexicon. Figure 4 on its right side shows a PTD anchored by the transitive verb *voit*. This PTD comes from the unification between the lexical entry for *voit* presented in the center of the figure and the interface of the non anchored PTD on the left side of the figure.

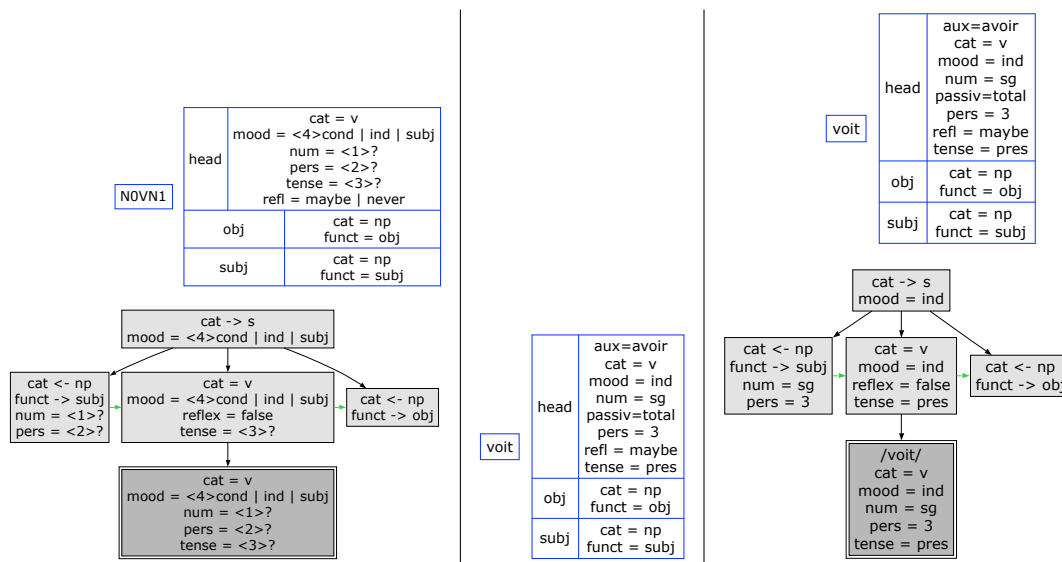


Fig. 4: From left to right, a non anchored PTD describing the syntactic behaviour of a transitive verb in the active voice, a lexical entry for the transitive verb *voit* and the PTD after anchoring with the verb *voit*

## 5 Evaluation on a sentence test suite

Our goal is to evaluate the coverage of our grammar in the most detailed manner. The least costly way of doing this is to use the grammar for parsing a sentence test suite illustrating most rules of French grammar. It is important that the suite includes not only positive examples but also negative examples to test the overgeneration of the grammar.

There are not many corpora of this type for French. We have chosen the TSNLP [3], which includes 1690 positive sentences and 1935 negative sentences. It is far from covering all of French grammar; in particular, it includes very few complex sentences but it stresses some phenomena such as coordination or the position in the sentence of the adverbial complements. On the other hand, our grammar covers phenomena that are ignored by the TSNLP: the passive and middle voice of verbs, the subcategorisation of predicative nouns and adjectives, the control of the subject of infinitive complements, the relative and interrogative clauses. . .

For the parsing, we used LEOPAR<sup>5</sup>, which is a parser devoted to IG. With the current grammar, the parser accepts 88% of the 1690 positive TSNLP sentences and rejects 85% of the 1935 negative sentences. The 15% of accepted negative sentences are due to the fact that the grammar ignores phonological rules and semantics. The 12% of unanalysed positive sentences are due to various reasons: speech sentences, frozen or semi-frozen expressions, phenomena that are not yet taken into account (causatives, superlatives. . .).

## 6 Prospects

The next step is to use our French grammar to parse raw corpora. It is already possible to use LEOPAR

<sup>5</sup> <http://www.loria.fr/equipes/calligramme/leopar>

with a large lexicon for such a task. It is necessary to enrich the grammar because some common linguistic phenomena are not yet taken into account. We also need to improve the efficiency of the parser to contain the possible explosion resulting from the increase of the grammar size in combination with the increased sentence length.

## References

- [1] J. Bresnan. *Lexical-Functional Syntax*. Blackwell Publishers, Oxford, 2001.
- [2] D. Duchier, J. Le Roux, and Y. Parmentier. XMG : Un compilateur de méta-grammaires extensible. In *TALN 2005, Dourdan, France*, 2005.
- [3] S. Lehmann, S. Oepen, S. Regnier-Prost, K. Netter, V. Lux, J. Klein, K. Falkedal, F. Fouvry, D. Estival, E. Dauphin, H. Compagnion, J. Baur, L. Balkan, and D. Arnold. TSNLP — Test Suites for Natural Language Processing. In *Proceedings of COLING 1996, Copenhagen*, 1996.
- [4] A. Nasr. A formalism and a parser for lexicalised dependency grammars. In *4th International Workshop on Parsing Technologies (IWPT)*, 1995.
- [5] G. Perrier. Interaction grammars. In *CoLing '2000, Sarrebrücken*, pages 600–606, 2000.
- [6] G. Perrier. La sémantique dans les grammaires d'interaction. *Traitement Automatique des Langues*, 45(3):123–144, 2004.
- [7] G. K. Pullum and B. C. Scholz. On the Distinction between Model-Theoretic and Generative-Enumerative Syntactic Frameworks. In *LACL 2001, Le Croisic, France*, volume 2099 of *Lecture Notes in Computer Science*, pages 17–43, 2001.
- [8] O. Rambow, K. Vijay-Shanker, and D. Weir. D-tree substitution grammars. *Computational Linguistics*, 27(1):87–121, 2001.
- [9] C. Retoré. *The Logic of Categorical Grammars*, 2000. *ESSLI'2000, Birmingham*.
- [10] J. Rogers and K. Vijay-Shanker. Obtaining trees from their descriptions: an application to tree-adjointing grammars. *Computational Intelligence*, 10(4):401–421, 1994.
- [11] I. A. Sag, T. Wasow, and E. M. Bender. *Syntactic Theory: a Formal Introduction*. Center for the Study of Language and INF, 2003.



# Exploiting Role-Identifying Nouns and Expressions for Information Extraction

William Phillips and Ellen Riloff  
School of Computing  
University of Utah  
Salt Lake City, UT 84112  
{*phillips,riloff*}@*cs.utah.edu*

## Abstract

We present a new approach for extraction pattern learning that exploits *role-identifying nouns*, which are nouns whose semantics reveal the role that they play in an event (e.g., an “assassin” is a perpetrator). Given a few seed nouns, a bootstrapping algorithm automatically learns role-identifying nouns, which are then used to learn extraction patterns. We also introduce a method to learn *role-identifying expressions*, which consist of a role-identifying verb linked to an event (e.g., “<subject> participated in the murder”). We present experimental results on the MUC-4 terrorism corpus and a disease outbreaks corpus.

## Keywords

information extraction, learning, event roles

## 1 Introduction

Our research focuses on event-based information extraction, where the task is to identify facts related to events. Event-based information extraction systems have been developed for many domains, including terrorism [8, 3, 10, 13], management succession [17], corporate acquisitions [5, 6], and disease outbreaks [7]. Many IE systems rely on extraction patterns or rules, such as CRYSTAL [13], AutoSlog/AutoSlog-TS [9, 10], RAPIER [2], WHISK [12], Ex-DISCO [17], Snowball [1], (LP)<sup>2</sup> [4], Subtree patterns [14], and predicate-argument rules [16].

Our work presents a new approach for IE pattern learning that takes advantage of *role-identifying nouns*, *role-identifying verbs*, and *role-identifying expressions*. We will refer to a word or phrase as being *role-identifying* if it reveals the role that an entity or object plays in an event. For example, the word *assassin* is a role-identifying noun because an assassin is the perpetrator of an event, by definition. Similarly, the verb *participated* is a role-identifying verb because it means that someone played the role of actor (agent) in an activity. When a role-identifying verb is explicitly linked to an event noun, we have a role-identifying expression. For example, “<subject> participated in the murder” means that the subject of “participated” is a perpetrator of the murder event.

We have developed a new approach to IE pattern learning that exploits role-identifying nouns. We em-

ploy the Basilisk bootstrapping algorithm [15] to learn role-identifying nouns, and then use them to rank extraction patterns. We also describe a learning process that creates a new type of extraction pattern that captures role-identifying expressions. This process begins by automatically inducing event nouns from a corpus via bootstrapping. We then generate patterns that extract an event noun as a syntactic argument. Finally, we match these event patterns against a corpus and generate expanded patterns for each syntactic dependency that is linked to the pattern’s verb.

This paper is organized as follows. Section 2 gives the motivation for role-identifying nouns and expressions. Section 3 describes the extraction pattern learning process. Section 4 presents our experimental results, and Section 5 discusses related work.

## 2 Motivation

Our work is motivated by the idea that *role-identifying nouns* and *role-identifying expressions* can be beneficial for information extraction. In this section, we explain what they are and how we aim to use them.

### 2.1 Role-Identifying Nouns

Our research exploits nouns that, by definition, identify the role that the noun plays with respect to an event. For example, the word *kidnapper* is defined as the perpetrator of a kidnapping. Similarly, the word *victim* is defined as the object of a violent event. We will refer to these nouns as **Lexically Role-Identifying Nouns** because their lexical meaning identifies the role that the noun plays in some event.

We have observed that there are a surprisingly large number of role-identifying nouns. For example, the words *arsonist*, *assassin*, *kidnapper*, *robber*, and *sniper* refer to perpetrators of a crime. Similarly, the words *casualty*, *fatality*, *victim*, and *target* refer to objects of a violent event. It is important to note that in a sentence these nouns may serve in a different thematic role associated with a verb. For example, in “*The assassin was arrested*”, the assassin is the theme of the verb “arrest”, but it is also understood to be the perpetrator of an (implicit) assassination event. Our work focuses on high-level **event roles**, rather than thematic (semantic) roles that represent verb arguments.

Within a specific domain, some words can also be

inferred to serve in an event role based on their general semantic class. For example, consider disease outbreak reports. If a toddler is mentioned, one can reasonably infer that the toddler is a victim of a disease outbreak. The reason is that toddlers cannot fill any other roles commonly associated with disease outbreaks (e.g., they cannot be medical practitioners, scientists, or spokespeople). The intuition comes from Grice’s Maxim of Relevance: any reference to a child in a disease report is almost certainly a reference to a victim because the child wouldn’t be relevant to the story otherwise. As another example, if a restaurant is mentioned in a crime report, then a crime probably occurred in or around the restaurant. Of course, context can always provide another explanation (e.g., the restaurant could be the place where a suspect was arrested). But generally speaking, if a word’s semantics are compatible with only one role associated with an event, then we often infer that it is serving in that role. We will refer to nouns that strongly evoke one event role as **Semantically Role-Identifying Nouns**.

Role-identifying nouns are often not the most desirable extractions for an IE system because they are frequently referential. For example, “the assassin” may be coreferent with a proper name (e.g., “Lee Harvey Oswald”), which is a more desirable extraction. However, role-identifying nouns can be exploited for extraction pattern learning. Our intuition is that if a pattern consistently extracts role-identifying nouns associated with one event role, then the pattern is probably a good extractor for that role.

## 2.2 Role-Identifying Expressions

For event-based information extraction, the most reliable IE patterns usually depend on a word that explicitly refers to an event. For example, the pattern “<subject> was kidnapped” indicates that a kidnapping took place, and the subject of “kidnapped” is extracted as the victim. In contrast, some verbs identify a role player associated with an event without referring to the event itself. For example, consider the verb “participated”. By its definition, “participated” means that someone took part in something, so the pattern “<subject> participated” identifies the actor (agent) of an activity. However, the word “participate” does not reveal what the activity is. The activity is often specified in another argument of the verb (e.g., “John participated in the debate.”). In other cases, the event must be inferred through discourse (e.g., “The debate took place at Dartmouth. John participated.”).

Our observation is that there are many verbs whose main purpose is to identify a role player associated with an event, without defining the event itself. We will refer to them as **Role-Identifying Verbs**. Some additional examples of role-identifying verbs are “perpetrated”, “accused”, and “implicated”, which all identify the (alleged) perpetrator of an event. Often, the agent of the verb is also the agent of the (implicit) event. For example, the agents of “participated” and “perpetrated” are also the agents of the event (e.g., “John perpetrated the attack”). However, an entity or object can function in one thematic role with respect to the verb and a different role with respect to the event. For example, in the sentence “John was impli-

cated in the attack”, the theme of “implicated” is the (alleged) agent of the attack.

Our goal is to use role-identifying verbs in extraction patterns. The challenge is that these verbs are generally not reliable extractors by themselves because it is crucial to know what event they are referring to. For example, “John participated in the bombing” is relevant to a terrorism IE task, but “John participated in the meeting” is not. Our solution is to create patterns that include both a role-identifying verb and a relevant event noun as a syntactic argument to the verb. We will refer to these patterns as **Role-Identifying Expression (RIE) patterns**.

## 3 Extraction Pattern Learning

### 3.1 Overview

Our hypothesis is that role-identifying nouns can be valuable for extraction pattern learning. Throughout this work, we rely heavily on the Basilisk bootstrapping algorithm [15], which was originally designed for semantic lexicon induction (i.e., to learn which nouns belong to a general semantic category, such as ANIMAL or VEHICLE). In Section 3.2.2, we will use Basilisk as it was originally intended – to generate nouns belonging to the semantic category EVENT. However, we also use Basilisk in a new way – to learn role-identifying nouns.

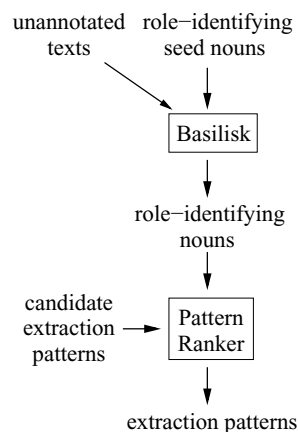


Fig. 1: The Extraction Pattern Learning Process

Fig. 1 shows the high-level process for extraction pattern learning. First, we use Basilisk to generate role-identifying nouns for an event role associated with the IE task. Next, we create a large set of *candidate patterns* by exhaustively generating all extraction patterns that occur in the training corpus. Finally, we rank the candidate patterns based on their tendency to extract the role-identifying nouns. This learning process is therefore very weakly supervised: only an unannotated corpus and a small set of role-identifying seed nouns are needed to learn extraction patterns for an event role.

In the following sections, we explain how two types of candidate patterns are generated, how Basilisk learns role-identifying nouns, and how the role-identifying nouns are used to select the best patterns.

## 3.2 Generating Candidate Patterns

Our goal is to learn two different kinds of extraction patterns. First, we generate the traditional kind of patterns which extract information from the arguments of verbs and nouns that describe an event (e.g., “<subject> was kidnapped” or “assassination of <np>”). Second, we generate a new type of extraction pattern that captures *role-identifying expressions*.

### 3.2.1 Generating Standard Patterns

We use the AutoSlog extraction pattern learner [9] to generate candidate “traditional” extraction patterns. AutoSlog applies syntactic heuristics to automatically learn lexico-syntactic patterns from annotated noun phrases. For example, consider the sentence “A turkey in Indonesia was recently infected with avian flu.” If “A turkey” is labeled as a disease victim, then AutoSlog will create the pattern “<subject> PassVP(infected)” to extract victims. This pattern matches instances of the verb “infected” in the passive voice, and extracts the verb’s subject as a victim.

We use AutoSlog in an unsupervised fashion by applying it to unannotated texts and generating a pattern to extract (literally) every noun phrase in the corpus. We will refer to the resulting set of patterns as the *candidate standard IE patterns*.

### 3.2.2 Generating RIE Patterns

Fig. 2 shows the process for generating candidate Role-Identifying Expression (RIE) patterns, which involves two steps. In Step 1, we use the Basilisk semantic lexicon learner [15] to generate *event nouns*, which are nouns that belong to the semantic category EVENT (e.g., “assassination”). This step may not be needed if a list of event nouns for the domain is already available or can be obtained from a resource such as WordNet. However, we use Basilisk to demonstrate that event nouns for a domain can be automatically generated. As input, Basilisk requires just a few seed nouns and an unannotated text corpus. We explain how the seed nouns were chosen in Section 4.1.

We ran Basilisk for 50 iterations, generating 5 event nouns per iteration. However, we are only interested in events that are relevant to the IE task. For example, for the terrorism domain we want to extract information about murder and kidnapping events, but not meetings or celebratory events. So we manually reviewed the event nouns and retained only those that are *relevant* to the IE task. Of the 250 event nouns generated for each domain, we kept 94 for terrorism and 220 for disease outbreaks.<sup>1</sup>

In Step 2, we create the role-identifying expression patterns. Each RIE pattern must be anchored by a verb phrase that has a syntactic argument that is an event noun. We begin by creating standard patterns that can extract events. We give the relevant event nouns to the AutoSlog pattern learner [9] as input,<sup>2</sup> which then creates patterns that can extract

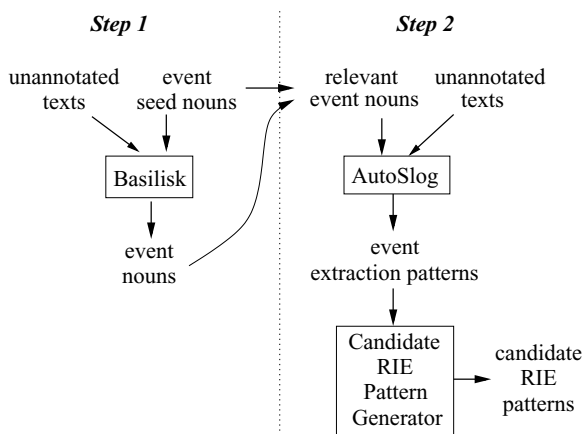


Fig. 2: Generating Candidate RIE Patterns

events. The *Candidate RIE Pattern Generator* then expands these event patterns into role-identifying expressions. For each instance of a verb pattern<sup>3</sup>, the verb’s subject, direct object, and all attached prepositional phrases are identified. For each one, an expanded pattern is spawned that includes this syntactic relation. For example, consider the event pattern “committed <EVENT>”, which matches active voice verb instances of “committed” and extracts its direct object as an event (e.g., “committed the murder”). Now suppose that this pattern is applied to the sentence: “John Smith committed the murder in November.” Two syntactic relations are associated with the verb phrase: its subject (“John Smith”) and a PP (“in November”). The following two candidate RIE patterns would then be generated: “<subject> committed <EVENT>” and “committed <EVENT> in <np>”.

## 3.3 Learning Role-Identifying Nouns

Now that we have a large set of candidate extraction patterns, we return to the high-level learning process depicted in Fig. 1. The first step is to generate role-identifying nouns for each event role associated with the IE task. We use the Basilisk bootstrapping algorithm [15], which was originally designed for semantic lexicon induction but its algorithm relies heavily on lexico-syntactic pattern matching, which also seemed well-suited for learning role-identifying nouns.

Basilisk begins with a small set of seed nouns and then iteratively induces more nouns. Each bootstrapping cycle consists of 3 steps: (1) collect a pool of patterns that tend to extract the seeds, (2) collect all nouns extracted by these patterns, (3) score each noun based on the scores of all patterns that extracted it.<sup>4</sup> We tried two different ways of selecting role-identifying seed nouns to kickstart the bootstrapping, which we will discuss in Section 4.1. Below are some of the role-identifying nouns that were learned for terrorism perpetrators and disease outbreak victims:

*Terrorism Perpetrator:* assailants, attackers, cell, culprits, extremists, hitmen, kidnappers,

<sup>1</sup> Diseases were often used to refer to outbreaks, so we included disease names as event nouns in this domain.

<sup>2</sup> Since AutoSlog is a supervised learner, the event nouns are essentially used to automatically annotate the corpus.

<sup>3</sup> AutoSlog’s noun patterns are not used.

<sup>4</sup> We made one minor change to Basilisk’s RlogF scoring function, by adding 1 inside the logarithm so that words with frequency 1 would not get a zero score.

<i>Terror PerpInd</i>	<i>Terror PerpOrg</i>	<i>Terror Target</i>	<i>Terror Victim</i>
<subj> riding was kidnapped by <np> was killed by <np> <subj> identified themselves was perpetrated by <np>	<subj> claimed responsibility <subj> is group <subj> claimed delegates of <np> was attributed to <np>	destroyed <dobj> burned <dobj> <subj> was damaged awakened with <np> blew up <dobj>	murder of <np> <subj> was killed assassination of <np> killed <dobj> <subj> was sacrificed
<i>Outbreak Victim</i>	<i>Outbreak Disease</i>	<i>Terror Weapon</i>	
brains of <np> mother of <np> disease was transmitted to <np> <subj> is unwell <subj> tests positive	outbreaks of <np> woman was diagnosed with <np> to contracted <dobj> <subj> hits to contract <dobj>	threw <dobj> hurled <dobj> confiscated <dobj> rocket <dobj> sticks of <np>	

Table 1: Top 5 Standard Patterns for Each Event Role

<i>Terror PerpInd</i>	<i>Terror PerpOrg</i>	<i>Terror Target</i>	<i>Terror Victim</i>
EV was perpetrated by <np> <subj> committed EV <subj> was involved in EV <subj> participated in EV <subj> involved in EV	<subj> carried out EV EV was perpetrated by <np> <subj> called for EV EV was attributed to <np> EV was carried out by <np>	EV destroyed <dobj> caused EV to <np> EV damaged <dobj> staged EV on <np> EV caused to <np>	<subj> was killed in EV EV including <dobj> <subj> was killed during EV EV led <dobj> identified <dobj> after EV
<i>Outbreak Victim</i>	<i>Outbreak Disease</i>	<i>Terror Weapon</i>	
<subj> was suffering from EV <subj> contracted EV EV was transmitted from <dobj> EV infect <dobj> EV killed dozens of <np>	EV known as <np> EV called <dobj> EV was known as <np> EV due to <np> <subj> was caused by EV	confiscated <dobj> during EV EV was caused by <np> EV carried out with <np> <subj> was thrown by EV <subj> caused EV	

Table 2: Top 5 RIE Patterns for Each Event Role (EV = Event Noun)

militiamen, MRTA, narco-terrorists, sniper

*Outbreak Victim:* bovines, crow, dead, eagles, fatality, pigs, swine, teenagers, toddlers, victims

Most of the perpetrator words are lexically role-identifying nouns, while most of the disease outbreak victim words are semantically role-identifying nouns.

### 3.4 Selecting Extraction Patterns

When Basilisk’s bootstrapping is done, we have a large collection of role-identifying nouns. Next, we rank all of the candidate extraction patterns based on the same RlogF metric that Basilisk uses internally, which is:  $RlogF(p_i) = \frac{f_i}{n_i} * \log_2(f_i)$ , where  $f_i$  is the number of unique role-identifying nouns extracted by pattern  $p_i$  and  $n_i$  is the total number of unique nouns extracted by  $p_i$ . The top N highest-ranking patterns are selected as the best extractors for the event role.

We used this approach to learn extraction patterns for seven event roles: five roles associated with terrorism (*individual perpetrators*, *organizational perpetrators*, *victims*, *physical targets*, and *weapons*) and two roles associated with disease outbreaks (*diseases* and *victims*). Tables 1 and 2 show the top 5 standard and RIE extraction patterns learned for each event role.

## 4 Evaluation

We evaluated our performance on two data sets: the MUC-4 terrorist events corpus [8], and a ProMed disease outbreaks corpus. The MUC-4 corpus contains 1700 stories and answer key templates for each story. We focused on five MUC-4 string slots: *perpetrator individuals*, *perpetrator organizations*, *physical targets*, *victims*, and *weapons*. We used 1400 stories for training (DEV+TST1), 100 stories for tuning (TST2), and 200 stories as a blind test set (TST3+TST4).

ProMed-mail<sup>5</sup> is an open-source, global electronic reporting system for outbreaks of infectious diseases. Our ProMed IE data set includes a training set of 4659 articles, and a test set of 120 different articles coupled with answer key templates that we manually created. We focused on extracting *diseases* and *victims*, which can be people, animals, or plants.

The complete IE task involves the creation of answer key templates, one template per incident.<sup>6</sup> Template generation is a complex process, requiring coreference resolution and discourse analysis to determine how many incidents were reported and which facts belong with each incident. Our work focuses on extraction pattern learning, so we evaluated the extractions themselves, before template generation would take place. This approach directly measures how accurately the patterns find relevant information, without confounding factors introduced by the template generation process.<sup>7</sup> We used a *head noun* scoring scheme, where an extraction is correct if its head noun matches the head noun in the answer key.<sup>8</sup>

### 4.1 Seed Word Selection

To select event seed nouns, we shallowly parsed the corpus, sorted the head nouns of NPs based on frequency, and then manually identified the first 10 nouns that represent an event.

To select role-identifying seed nouns, we experimented with two approaches. First, we collected all of the head nouns of NPs in the corpus and sorted them

<sup>5</sup> See [www.promedmail.org](http://www.promedmail.org)

<sup>6</sup> Many MUC-4 and ProMed stories mention multiple incidents.

<sup>7</sup> For example, if the coreference resolver incorrectly decides that two items are coreferent and merges them, then it will appear that only one item was extracted by the patterns when in fact both were extracted.

<sup>8</sup> This approach allows for different modifiers in an NP as long as the heads match. We also discarded pronouns because we do not perform coreference resolution.

System	PerpInd			PerpOrg			Target			Victim			Weapon		
	Rec	Pr	F	Rec	Pr	F	Rec	Pr	F	Rec	Pr	F	Rec	Pr	F
ASlogTS	.49	.35	.41	.33	.49	.40	.64	.42	.51	.52	.48	.50	.45	.39	.42
Top20	.18	.55	.27	.15	.67	.25	.46	.51	<b>.48</b>	.30	.51	.38	.40	.59	.47
Top50	.22	.48	.30	.17	.50	.25	.51	.44	.47	.35	.42	.38	.52	.48	<b>.50</b>
Top100	.36	.45	<b>.40</b>	.21	.52	.30	.59	.37	.46	.42	.37	.39	.53	.43	.48
Top200	.40	.35	.37	.34	.45	<b>.39</b>	.64	.29	.40	.48	.35	<b>.40</b>	.53	.35	.42

Table 3: MUC-4 Results for Standard Patterns

by frequency. For each event role, we then manually identified the first 10 nouns that were role-identifying nouns for that role. We will refer to these as the *high-frequency seeds*.

We also tried using seed patterns instead of seed nouns. For each event role, we manually defined 10 patterns that reliably extract NPs for that role. For example, the pattern “<subject> kidnapped” was a seed pattern to identify perpetrators. We also defined an *Other* role to capture other possible roles, using 60 seed patterns for this category in terrorism and 30 for disease outbreaks.<sup>9</sup> We then applied the patterns to the corpus and collected their extractions. For each event role ( $erole_i$ ) and each head noun of an extraction ( $n$ ), we computed the following probability:

$$Pr(erole_i | n) = \frac{|n \text{ extracted by an } erole_i \text{ pattern}|}{\sum_{k=1}^{|E|} |n \text{ extracted by an } erole_k \text{ pattern}|} \quad (1)$$

where  $E$  is the number of event roles. All nouns with probability  $> 0.50$  and frequency  $\geq 2$  were used as seeds. We will refer to these as the *pattern-generated seeds*. The advantages of this approach are that it is natural to think of seed patterns for a role, and a few patterns can yield a large set of seed nouns. The drawbacks are that these nouns may not be frequent words and they are not guaranteed to be role-specific.

Both approaches worked reasonably well, but combining the two approaches worked even better. So for all of our experiments, the seeds consist of the *high-frequency seeds* plus the *pattern-generated seeds*.

## 4.2 Experimental Results

To establish a baseline for comparison, we trained the AutoSlog-TS IE pattern learner [10] on our two data sets. AutoSlog-TS generates a ranked list of extraction patterns, which needs to be manually reviewed.<sup>10</sup> The first row of Tables 3 and 4 shows its recall, precision, and F-measure. The MUC-4 results are similar to those of ALICE and the other MUC-4 systems as reported in [3], although those results are with template generation so not exactly comparable to ours.

Next, we evaluated the standard IE patterns produced by our learning process. Tables 3 and 4 show the scores obtained for the top 20, 50, 100, and 200 patterns in the ranked list. As one would expect, the first 20 patterns yielded the highest precision. As more patterns are used, recall increases but precision drops. In most cases, the best F-measure scores were achieved with the top 100 or 200 patterns.

<sup>9</sup> We roughly wanted to balance the number of patterns for this role with all of the other roles combined.

<sup>10</sup> We reviewed patterns with score  $\geq .951$  and frequency  $\geq 3$  for terrorism, and score  $\geq 5.931$  for disease outbreaks.

System	Disease			Victim		
	Rec	Pr	F	Rec	Pr	F
ASlogTS	.51	.27	.36	.48	.36	.41
Top20	.40	.33	.36	.34	.38	.36
Top50	.44	.33	.38	.35	.38	.36
Top100	.47	.31	.37	.36	.37	<b>.37</b>
Top200	.54	.30	<b>.39</b>	.38	.33	.35

Table 4: ProMed Results for Standard Patterns

We then included the RIE patterns produced by our learning process. First, we combined the top 20 Standard patterns with the RIE patterns. Our expectation was that this set of patterns should have good precision but perhaps only moderate recall. Second, we combined the top 100 Standard patterns with the RIE patterns. We expected this set of patterns to have higher recall but lower precision. In the terrorism domain, fewer than 100 RIE patterns were learned for each event role, so we used them all. For disease outbreaks, many RIE patterns were learned so we evaluated the top 100 and the top 200.

System	Disease			Victim		
	Rec	Pr	F	Rec	Pr	F
Top20	.40	.33	.36	.34	.38	.36
Top20+100RIEs	.44	.32	.37	.36	.35	.36
Top20+200RIEs	.45	.31	.36	.40	.36	<b>.38</b>
Top100	.47	.31	.37	.36	.37	.37
Top100+100RIEs	.50	.31	<b>.38</b>	.38	.35	.36
Top100+200RIEs	.50	.30	.37	.41	.35	<b>.38</b>
ASlogTS	.51	.27	.36	.48	.36	.41

Table 5: Promed Results for All Patterns

Tables 5 and 6 show the results. The RIE patterns were most beneficial for the terrorism perpetrator roles, increasing the F score by +6 for *PerpInd* and +11 for *PerpOrg* when using 20 Standard patterns. The F score also increased by 1-2 points for the terrorism *Victim* and *Weapon* roles, but performance decreased on the *Target* role. For disease outbreaks, the RIE patterns improved the F score for both the *Disease* and *Victim* roles.

The last row of Tables 5 and 6 show the AutoSlog-TS baseline again for comparison. Our IE system is competitive with AutoSlog-TS, which required manual review of its patterns. In contrast, our IE patterns were learned automatically using only seed words and unannotated texts for training.

## 4.3 Analysis

Table 7 shows examples of RIE patterns that behaved differently from their Standard pattern counterparts. The *Pr* column shows  $Pr(erole | p)$  for each pattern  $p$ , which is the percentage of the pattern’s extractions

System	PerpInd			PerpOrg			Target			Victim			Weapon		
	Rec	Pr	F	Rec	Pr	F	Rec	Pr	F	Rec	Pr	F	Rec	Pr	F
Top20	.18	.55	.27	.15	.67	.25	.46	.51	.48	.30	.51	.38	.40	.59	.47
Top20+RIEs	.25	.48	.33	.25	.70	.36	.46	.42	.44	.32	.48	.38	.41	.60	<b>.49</b>
Top100	.36	.45	.40	.21	.52	.30	.59	.37	<b>.46</b>	.42	.37	.39	.53	.43	.48
Top100+RIEs	.40	.43	<b>.41</b>	.30	.57	<b>.40</b>	.59	.33	.42	.44	.36	<b>.40</b>	.53	.43	.48
ASlogTS	.49	.35	.41	.33	.49	.40	.64	.42	.51	.52	.48	.50	.45	.39	.42

Table 6: MUC-4 Results for All Patterns

that are role-identifying nouns. The Standard patterns in Table 7 were not learned because they did not score highly enough, but the RIE patterns were learned because they performed better. For example, “<subject> was involved in EVENT” is a more reliable pattern for identifying perpetrators than just “<subject> was involved”. In the disease outbreaks domain, “<subject> was treated for EVENT” is more reliable than just “<subject> was treated”. Overall, we found many RIE patterns that performed better than their simpler counterparts.

Pattern Type	Terrorism Perpetrator	Pr
RIE	<subj> was involved in EVENT	.65
standard	<subj> was involved	.32
RIE	<subj> staged EVENT	.27
standard	<subj> staged	.12
RIE	<subj> unleashed EVENT	.33
standard	<subj> unleashed	.17
Pattern Type	Outbreak Victim	Pr
RIE	<subj> was treated for EVENT	.65
standard	<subj> was treated	.19
RIE	<subj> was hospitalized for EVENT	.75
standard	<subj> was hospitalized	.31
RIE	spread EVENT to <np>	.44
standard	spread to <np>	.10

Table 7: RIE Patterns vs. Standard Patterns

## 5 Related Work

Many supervised learning systems have been developed for event-oriented information extraction (e.g., [13, 2, 5, 6, 4, 3]), but relatively few do not require annotated training data. AutoSlog-TS [10] requires only relevant and irrelevant training documents, and is the baseline system that we used for comparison in our experiments. The systems most similar to ours are ExDisco [17] and Meta-Bootstrapping [11], which are bootstrapping algorithms that require only relevant texts and seed words or patterns for training. However, the extraction patterns produced by Meta-Bootstrapping are general semantic class extractors and not event role extractors. The novel aspects of our work are (1) the use of role-identifying nouns in combination with a semantic bootstrapping algorithm (Basilisk) for extraction pattern learning, and (2) automatically learning a new type of extraction pattern that captures role-identifying expressions.

## 6 Summary

We have presented a new approach to IE that learns extraction patterns by exploiting role-identifying nouns. We also introduced role-identifying expressions and presented a method for learning them. Our result-

ing IE system achieved good performance on 7 event roles associated with two different domains.

## References

- [1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM International Conference on Digital Libraries*, 2000.
- [2] M. Califf and R. Mooney. Relational Learning of Pattern-matching Rules for Information Extraction. In *Proceedings of the 16th National Conference on Artificial Intelligence*, 1999.
- [3] H. Chieu, H. Ng, and Y. Lee. Closing the Gap: Learning-Based Information Extraction Rivaling Knowledge-Engineering Methods. In *ACL-03*, 2003.
- [4] F. Ciravegna. Adaptive Information Extraction from Text by Rule Induction and Generalisation. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, 2001.
- [5] D. Freitag. Toward General-Purpose Learning for Information Extraction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, 1998.
- [6] D. Freitag and A. McCallum. Information Extraction with HMM Structures Learned by Stochastic Optimization. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 584–589, Austin, TX, August 2000.
- [7] R. Grishman, S. Huttunen, and R. Yangarber. Real-Time Event Extraction for Infectious Disease Outbreaks. In *Proceedings of HLT 2002 (Human Language Technology Conference)*, 2002.
- [8] MUC-4 Proceedings. *Proceedings of the Fourth Message Understanding Conference (MUC-4)*. Morgan Kaufmann, 1992.
- [9] E. Riloff. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of the 11th National Conference on Artificial Intelligence*, 1993.
- [10] E. Riloff. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1044–1049. The AAAI Press/MIT Press, 1996.
- [11] E. Riloff and R. Jones. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 1999.
- [12] S. Soderland. Learning Information Extraction Rules for Semi-structured and Free Text. *Machine Learning*, 1999.
- [13] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. CRYSTAL: Inducing a conceptual dictionary. In *Proc. of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1314–1319, 1995.
- [14] K. Sudo, S. Sekine, and R. Grishman. An Improved Extraction Pattern Representation Model for Automatic IE Pattern Acquisition. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*, 2003.
- [15] M. Thelen and E. Riloff. A Bootstrapping Method for Learning Semantic Lexicons Using Extraction Pattern Contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 214–221, 2002.
- [16] A. Yakushiji, Y. Miyao, T. Ohta, and J. Tateisi, Y. Tsujii. Automatic construction of predicate-argument structure patterns for biomedical information extraction. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 2006.
- [17] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. Automatic Acquisition of Domain Knowledge for Information Extraction. In *Proceedings of the Eighteenth International Conference on Computational Linguistics (COLING 2000)*, 2000.

# On Some Aspects of Implementing a Pattern Engine based on Regular Expressions over Feature Structures

Jakub Piskorski

Joint Research Center of the European Commission

Web and Language Technology Group of IPSC

T.P. 267, Via Fermi 1, 21020 Ispra (VA), Italy

*Jakub.Piskorski@jrc.it*

## Abstract

Recently, we have witnessed an emergence of information extraction oriented pattern specification languages, most of which exploit finite-state devices. This paper focus on some aspects of implementing a pattern engine, whose rules are regular expressions over 'flat' feature structures. In particular, a method for efficiently processing such patterns is described.

## 1 Introduction

Information Extraction (IE) is concerned with extracting specific, structured information out of unstructured free-text documents. Most typical IE tasks focus on detecting entities, identifying relations which hold among them, and extracting events. The major process in the IE chain consists of applying a set of patterns for retrieving the sought-after information or part of it. The pattern specification languages utilize various types of formal languages, ranging from character-level regular expressions to unification-based formalisms. In order to efficiently process massive text collections, finite-state based pattern engines are the most prominent ones being used.

This paper focuses on techniques for efficient processing of patterns based on regular expressions over arbitrary non-recursive feature structures. The idea of using complex structures on the transitions of finite-state devices has been considered by several authors, e.g., [6] uses regular grammars with predicates over morphologically analyzed tokens and [7] introduces finite-state transducers with arbitrary predicates over symbols. Clearly, rich annotations on automata edges allow for compact descriptions, but standard finite-state optimization methods are hardly applicable.

The main motivation beyond the presented work comes from a need of an efficient pattern engine for extracting facts from vast amount of news articles collected daily with the Europe Media Monitor (EMM) system [1]. In particular, the presented solution takes the best of two recently introduced pattern engines, namely JAPE [2] and XTDL [4].

We start with some basic definitions and an overview of the two pattern engines we borrow from in sections 2 and 3 resp. The particularities and implementation of the new engine are presented in sections 4 and 5. The run-time performance is addressed in section 6. We end up with some conclusions in section 7.

## 2 Basic Definitions and Notions

A *deterministic finite-state automaton* (DFSA) is a quintuple  $M = (Q, \Sigma, \delta, q_0, F)$ , where  $Q$  is a finite set of states,  $\Sigma$  is the alphabet of  $M$ ,  $\delta : Q \times \Sigma \rightarrow Q$  is the transition function,  $q_0$  is the initial state and  $F \subseteq Q$  is the set of final states. The transition function can be extended to  $\delta^* : Q \times \Sigma^* \rightarrow Q \cup \{\perp\}$  by defining  $\delta^*(q, \epsilon) = q$ ,  $\delta^*(q, a) = \delta(q, a)$  if  $\delta(q, a)$  is defined or  $\delta^*(q, a) = \perp$  otherwise, and  $\delta^*(q, wa) = \delta(\delta^*(q, w), a)$  for  $a \in \Sigma$  and  $w \in \Sigma^*$ . The language accepted by a DFSA  $M$  is defined as  $L(M) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}$ . Languages accepted by finite-state automata are also called *regular*. A path in a DFSA  $M$  is a sequence of triples  $\langle (p_0, a_0, p_1), \dots, (p_{k-1}, a_{k-1}, p_k) \rangle$ , where  $(p_{i-1}, a_{i-1}, p_i) \in Q \times \Sigma \times Q$  and  $\delta(p_i, a_i) = p_{i+1}$  for  $1 \leq i < k$ . The string  $a_0 a_1 \dots a_k$  is the label of the path. Among all DFSAs recognizing the same language, there is always one which has the minimal number of states. We call such an automaton *minimal* (MDFSA). The definition of *nondeterministic finite-state automata* (NFSA) is analogous, with the difference that transition function is set-valued.

Next, we define flat feature structures. A type  $\alpha$  is a tuple  $[f_1^\alpha, \dots, f_k^\alpha]$  ( $k \geq 1$ ), where each  $f_i$  is a feature with  $f_i \in \Sigma_T$  (feature set), and  $f_i^\alpha \neq f_j^\alpha$  for  $i \neq j$ . Two types  $\alpha = [f_1^\alpha, \dots, f_k^\alpha]$  and  $\sigma = [f_1^\sigma, \dots, f_l^\sigma]$  are equal if and only if: (a)  $k = l$  and (b)  $\forall i \in \{1, \dots, k\} : f_i^\alpha = f_i^\sigma$ . Let  $|\alpha|$  be the number of features in  $\alpha$ . A flat feature structure (FFS)  $s$  is a tuple  $[\alpha, v_1, \dots, v_k]$ , where  $\alpha$  is a type,  $|\alpha| = k$ ,  $v_i \in \Sigma_V^+ \cup \{\top\}$  ( $1 \leq i \leq |\alpha|$ ) and  $\Sigma_V$  is a finite set of symbols. Further, let  $f_i(s)$  and  $\tau(s)$  denote the value of the  $i$ -th feature for  $s$  and the type of  $s$  resp. The symbol  $\top$  is used to denote unspecified feature, i.e.,  $f_i(s) = \top$  means that  $f_i(s)$  is unspecified. We say, that two FFSs  $s_1$  and  $s_2$  match if and only if: (a)  $\tau(s_1) = \tau(s_2)$ , (b)  $\forall 1 \leq i \leq |\tau(s_1)| : f_i(s_1) = f_i(s_2)$  or  $f_i(s_1) = \top$  or  $f_i(s_2) = \top$ .

## 3 IE-oriented Pattern Engines

In the last decade, several high-level specification languages for creating extraction patterns have been developed. The widely-known GATE platform comes with JAPE (Java Annotation Pattern Engine) [2]. A JAPE grammar consists of pattern-action rules. The left-hand side (LHS) of a rule is a regular expression over arbitrary atomic feature-value constraints, while

the right-hand side (RHS) constitutes a so-called *annotation manipulation statement* which specifies the output structures to be produced once the pattern matches. Additionally, the RHS may call native code, which on the one side provides a gateway to the outer world, but on the other side makes pattern writing difficult for non-programmers. Although JAPE is well established in the IE community, all its publicly available implementations face some efficiency problems while processing even moderate-size grammars.

A similar, but more declarative and linguistically-oriented pattern specification formalism called XTDL is used in SPROUT [4]. It is a blend of finite-state and unification-based grammar formalisms. In XTDL the LHS of a rule is a regular expression over typed feature structures<sup>1</sup> (TFS) with functional operators and coreferences, and the RHS is a TFS, specifying the output production. Coreferences in XTDL rules express structural identity, create dynamic value assignments, and serve as means of data transfer from LHS to RHS of a rule. Functional operators are primarily utilized for forming the slot values in the output structures and, secondly, they can act as Boolean-valued predicates, which allows for introducing complex constraints in the rules. Additionally, XTDL supports rule embedding which means a context-free descriptive power. The aforementioned features make XTDL more amenable formalism than JAPE since writing 'native code' is eliminated and coreferencing allows for compact description of linguistic phenomena. Nevertheless, processing XTDL patterns involves unification, a rather expensive operation. Although speed-up techniques for processing of such grammars have been developed [3], processing vast amount of textual data with XTDL grammars remains a bottle-neck.

JAPE and XTDL are very generic in that they make a clear distinction between rules and specification of components which produce information on top of which rules are applied. Contrary to the latter two, many other pattern languages, e.g., most of the ones surveyed in [5], are bound to a specific type of information (e.g, syntactic trees, grammatical functions) and exhibit somewhat black-box character.

## 4 EXPRESS

For EMM, we have developed EXPRESS (Extraction Pattern Recognition Engine and Specification Suite) – a new grammar formalism, which is similar in spirit to JAPE, but also encompasses some features and syntax borrowed from XTDL. The LHS of a rule is a regular expression over FFS (see sec. 2), i.e., non-recursive TFSs without coreferencing, where types are not ordered in a hierarchy. Unlike JAPE, variables can be tailored to string-valued attributes on the LHS of a rule in order to facilitate information transport into the RHS. Further, like in XTDL, functional operators are allowed on the RHSs for manipulating slot values and for establishing contact with the 'outer world'. Finally, we adapted the JAPE's feature of associating patterns with multiple actions, i.e., producing more

than one annotation (eventually nested) for a given text fragment. The following rule for matching information concerning violent events, where one person is killed by another, illustrates the syntax.

```

killing :- ((person & [FULL-NAME: #n1]):killed
            key-phrase & [METHOD: #m, FORM: "passive"]
            (person & [FULL-NAME: #n2]):killer):event
-> killed: victim & [NAME: #n1],
    killer: actor & [NAME: #n2],
    event: violence & [TYPE: "killing",
                      METHOD: #m,
                      ACTOR: #n2,
                      VICTIM: #n1,
                      IN_EVENTS: inHowManyEvents(#n2)]

```

The rule matches a sequence consisting of: a structure of type **person** representing a human(s) who is (are) the *victim* of the event, followed by a phrase in passive form, which triggers a 'killing' event, and another structure of type **person** representing the *actor*. The symbol & links a type name of the structure type with a list of feature-value pairs representing the constraints which have to be fulfilled. The variables #n1 and #n2 establish bindings to the names of both humans involved in the event. Analogously, the variable #m establishes a binding to the method of killing delivered by the **key-phrase** structure. Further, the labels **killed**, **killer**, and **event** on the LHS specify the start/end position of the annotation actions defined on the RHS of the rule. The first two actions produce structures of type **victim** and **actor** resp., where the value of the **NAME** slot is created via accessing the variables #n1 and #n2. Finally, the third action (**event**) produces an output structure of type **violence** which spans over the other two output structures. The value of the **IN\_EVENTS** slot is computed via a call to a functional operator **inHowManyEvents()** which contacts some knowledge base to find out the number of events the current actor was involved in the past. The rule described above matches the text fragment *Five Iraqi were shot by the Americans* and produces three structures: **victim & [NAME: Five Iraqi]**, **actor & [NAME: Americans]**, and a structure of type **violence** with **[TYPE: "killing", METHOD: "shooting", ACTOR: "Americans" ...]**.

The handling of Kleene constructions has to be clarified. If a structure containing a variable within a Kleene construction is matched more than once, then we create a local instances of the variable for each such submatch, and accumulate the local bindings into a concatenation thereof. This resembles the weak unidirectional coreferences in XTDL [3]. Further, labels are not allowed within Kleene constructions.

The grammars can be cascaded and each grammar can be associated with arbitrary processing resources which are integrated with the engine via implementing an appropriate programming interface. Further, for each grammar a different search strategy (e.g., longest match vs. 'all matches') and a different output production configuration (e.g., return only structures produced via grammar application or additionally also feature structures produced by other processing modules, which were not consumed by the grammar application) can be chosen. Finally, patterns may be assigned priorities which are used for resolving conflicts or for other purposes, e.g., encoding negation

<sup>1</sup> Typed feature structures are related to record structures in programming languages and are widely used as a data structure for NLP. Their formalizations include multiple inheritance and subtyping, which allow for terser descriptions.



```

FIND-MATCHES( $M = (Q, \Sigma, \delta, q_0, F), InputFS$ )
1   $node \leftarrow \text{GETFIRSTNODE}(InputFS)$ 
2   $lastNode \leftarrow \text{GETLASTNODE}(InputFS)$ 
3  while  $node \neq lastNode$ 
4  do  $Active \leftarrow \{(q_0, \epsilon, node)\}$ 
5      $Accepting \leftarrow \emptyset$ 
6     while  $Active \neq \emptyset$ 
7     do  $Next \leftarrow \emptyset$ 
8        for  $(q, \pi, v) \in Active$ 
9        do if  $q \in F$ 
10           then  $Accepting \leftarrow Accepting \cup \{(q, \pi, v)\}$ 
11           for  $(v, a, u) \in InputFS$ 
12           do for  $a' \in \Sigma : \delta(q, a') \neq \perp$ 
13           do if  $\text{MATCHES}(a, a')$ 
14              then  $Next \leftarrow Next \cup \{(\delta(q, a'), \pi \cdot a', u)\}$ 
15            $Active \leftarrow Next$ 
16     if  $Accepting \neq \emptyset$ 
17     then  $(q, \pi, v) \leftarrow \text{SELECTACCEPTINGCONFIG}(Accepting)$ 
18            $\text{EXECUTEACTION}(M, q, \pi)$ 
19            $node \leftarrow v$ 
20     else  $node \leftarrow \text{GETNEXTNODE}(InputFS, node)$ 
21 return

```

Fig. 1: Pattern matching algorithm

(not provided as such). Contrary to JAPE, priorities are encoded in a separate file. Thus, experimenting with different set-ups is more elegant and does not involve modifying the grammar file itself.

## 5 Implementation

Implementing an interpreter for processing a grammar consisting of regular expressions over arbitrary feature structures via encoding them into a single optimized finite-state network is not straightforward for two reasons. First of all, finite-state devices require finite alphabets, whereas the reservoir of linguistic-based feature structures used in extraction patterns is potentially infinite. Secondly, semantics of feature structures does not allow to turn such regular expressions into DFSAs via application of standard finite-state techniques. There are different ways of tackling these problems. We describe here the commonly used technique and introduce some enhancements. Let  $G$  be a grammar consisting of regular patterns  $r_1 \dots r_n$  over FFSs, where each pattern  $r_i$  is represented by a regular expression  $R_i$ . FFSs are replaced in each  $R_i$  by symbols representing references to these FFSs. Next, we construct a DFSA  $M$  (representing the whole grammar) which accepts the language  $R_1 \cdot \{\$1\} \dots R_n \cdot \{\$n\}$ , where  $\$1 \dots \$n$  are unique symbols representing rule identifiers. Additionally, we turn each state  $q$  into a final state if it has an outgoing transition labeled with one of the symbols in  $\{\$1, \dots, \$n\}$ . All other states are non-final. Further, let us assume, that the stream of input FFSs is represented as a directed labeled graph  $InputFS = (V, E)$ , where all nodes in  $V$  correspond to start/end positions of text spans associated with the input FFSs. An edge in  $E$  is a 3-tuple  $(v, a, u)$ , where  $v$  and  $u$  are source/target nodes, and  $a$  is the label which points to some FFS.

An algorithm that takes automaton  $M$  and finds all matches in  $InputFS$  (an input stream of feature structures) is presented in figure 1. Please note that although  $M$  is deterministic in a strict sense, it clearly is not deterministic when we consider the real seman-

tics of its transition labels. The variable  $node$  (initialized in line 1) points to the current node in  $InputFS$ , i.e., the node from which the algorithm tries to find the next potential match. The main **while** loop of the algorithm (lines 3-20) is executed until the current node is the last node in  $InputFS$ . Since there is potentially more than one path from the node  $u$  in  $InputFS$  which matches with the automaton  $M$  and due to the fact that even one single path in  $InputFS$  might match with different paths in  $M$ , we store in the set  $Active$  all 'current' configurations of  $M$ . A single configuration of  $M$  is a triple  $(q, \pi, v)$ , where  $q$  denotes the current state of  $M$ ,  $\pi$  is a sequence of input FFSs which match a path in  $M$  from  $q_0$  to  $q$ , and  $v$  denotes the next node in  $InputFS$  from which subsequent matches in the input stream will be sought. Analogously, in  $Accepting$  we store all accepting configurations of  $M$  (ones whose current state is final). Initially this set is empty (line 5). In the **while** loop in lines 6-15 all possible configurations of  $M$  that match some path in  $InputFS$  starting in the node  $node$  are computed. This process resembles breadth-first-search in graphs. In particular, in the inner loop (lines 8-14) for each  $(q, \pi, v) \in Active$  we compute all 'subsequent' configurations, i.e., the ones being the result of matching some input FFS  $a$  starting in node  $v$  with a FFS  $a'$  in the set of transitions for state  $q$ , so that  $\delta(q, a') = \dots$ . Matching test is done via a call to the function  $\text{MATCHES}$  (line 13). Note that for a single input FFS there might be potentially more than one matching transition in  $M$  (**for** loop in lines 12-13). Once all 'new' configuration have been computed, we select from the set of accepting configurations one which fulfills selection criteria (line 17). Selection criteria may vary, depending on the search strategy. For instance, in the *longest-match strategy*, one simply takes the configuration which covers the longest text span. If more than one such configuration exists, then the one being a result of application of a rule with highest priority is chosen, etc.<sup>2</sup> Once an accepting configuration is chosen, an appropriate action is performed (line 18), e.g., output structure(s) is produced. We can restore the rules that matched via inspecting transition labels from final states. Finally, the value of the current node in the input graph is then modified accordingly in the line 19. If no accepting configurations were found, the current node is set to the closest node in  $InputFS$  that has an outgoing edge (line 20).

The most time-consuming part of the algorithm in figure 1 is the **for** loop in lines 12-14. In the naive implementation we have to inspect all outgoing transitions from the state  $q$  whether their label ( $a'$ ) matches with the current input FFS ( $a$ ). Since distinct FFSs (even pairs of matching FFSs) are represented as different symbols, some states of the automaton  $M$ , being the result of merging the elementary rule automata into one DFSA, might have a quite high number of outgoing transitions. This applies in particular for the initial state and in its direct proximity. Inspecting all outgoing transition each time the initial state is visited clearly deteriorates the run-time performance.

In JAPE a speed-up method is applied which ex-

<sup>2</sup> In some applications, it is convenient to select more than one accepting configuration, but the modification to the presented algorithm is straightforward so it is not discussed any further.

exploits the fact that the feature structures being labels of outgoing transitions from a given state have shared parts. In particular, all such structures are partitioned into disjoint partial feature structures, e.g., two feature structures of the type  $morph = [pos, case]$ ,  $[morph, noun, nom]$  and  $[morph, noun, dat]$  would be split into  $[morph, noun, ]$ ,  $[morph, , nom]$  and  $[morph, , dat]$ . These 'partial' feature structures are then used to filter and match the feature structures in the input stream. In this way, redundant computations (both original feature structures share the same part-of-speech) are avoided. The proper ordering of partial feature structures reflecting dependencies might further reduce the number of time-consuming matching operations (in our example  $[morph, noun, ]$  would be checked before the other two partial feature structures are inspected).

In XTDL, where the recognition part of the rules consists of TFSs, a somewhat similar technique for ordering the outgoing transitions of a given state is used. It resembles topological sorting of acyclic graphs and consists of computing a transition hierarchy under TFS subsumption for all outgoing transitions of a given state. In the process of traversing the grammar automaton, these transition hierarchies are utilized for inspecting outgoing transitions from a given state, starting with the least specific transition(s) first, and moving downwards in the hierarchy, if necessary. If a less specific TFS does not match, the inspection of the more specific ones is discarded. Since the transition hierarchies, created solely from the TFSs in the grammar, might exhibit a somewhat flat character, artificial transitions are added for deepening the transition hierarchy. Although the transition sorting technique has been reported to give a speed-up of factor 3-5, the number of transitions which have to be inspected when computing new automaton configurations might be on an average relatively high due to the low degree of feature-value sharing. Needless to say, matching two TFSs involves unifiability test, which is less time consuming than unification, but needs more time than 'string-like' matching in JAPE.

In EXPRESS, we apply a technique which consists of flattening input FFSs into strings and converting all transitions labels of a given state into a single DFSA, so that computing new automaton configurations (line 12-14 in the algorithm) boils down to performing simple automaton look-up. Generally speaking, the process of finding a match at a given position in the input stream is split into three steps: (1) selection of the sequence(s) of input FFSs which is (are) covered by some rule(s) according to predefined selection strategy, (2) performing a fully-fledged match of the selected rule(s) against the selected input sequence of FFSs, which includes variable and label binding, and (3) producing and merging output structures. The advantage of splitting the process into 3 steps is two-fold. Firstly, postponing variable and label binding allows for efficient implementation of step (1) which, as we will see, involves only some basic string matching. Further, once we have selected an input sequence and the rules (or more) that match this sequence, performing full matching in step (2) can be done quickly due to the limited number of applicable rules. Thus, step (1) can be seen as a prefiltering of applicable rules. Since there

are potentially several paths in the automaton for the rule(s) selected in step (2), step (3) is necessary for merging and/or filtering out some output structures, but we do not address this issue in this paper.

We now turn to implementing step (1), i.e., in particular lines 12-14 of the algorithm. Firstly, let us observe that only a finite number of feature-value pairs are used in the grammar rules. We can compute for all FFSs of type  $\alpha = [f_1^\alpha, \dots, f_k^\alpha]$  appearing in the rules the respective value sets  $\Sigma_1, \dots, \Sigma_k$ . An input FFS  $s$  can be then encoded as a string  $id(\tau(s)) \cdot v_1 \cdot \$ \dots \$ \cdot v_{|\tau(s)|}$ , where  $id$  maps types to unique symbols representing their identifiers,  $\$$  is a unique symbol /  $\Sigma_i \{ \}$  (for all  $1 \leq i \leq |\tau(s)|$ ) which represents a separator and  $v_i \in \Sigma_i \{ \}$  are defined as follows:

$$v_i = \begin{cases} f_i(s) & : f_i(s) \in \Sigma_i \\ \% & : f_i(s) \notin \Sigma_i \end{cases} \quad f_i(s) =$$

In order to illustrate the idea, let us consider a FFS of the type  $morph = [pos, case, gender]$ , where  $\Sigma_{pos} = \{noun, adj\}$ ,  $\Sigma_{gender} = \{m, f\}$ , and  $\Sigma_{case} = \{n, a, d, g\}$ . The FFS  $[morph, noun, l, f]$  would be then represented as  $id(morph) \cdot noun \cdot \$ \cdot \% \cdot \$ \cdot f$ .

The FFSs in the extraction patterns are represented similarly. Let us consider the outgoing transitions  $t_1, \dots, t_n$  from the state  $q$ . We can represent a FFS  $a_k$  being the label of the transition  $t_k$  as a regular expression of the form:  $id(\tau(a_k)) \cdot v_1 \cdot \$ \dots \$ \cdot v_{|\tau(a_k)|} \cdot \% \cdot target(t_k)$ , where  $id$  and  $\$$  are defined as previously,  $\% \in \Sigma_i \{ \}$  (for all  $1 \leq i \leq |\tau(a_k)|$ ) is unique separator,  $target(t_k)$  is a symbol representing the target state of the transition  $t_k$ , and  $v_i \in (\Sigma_i \{ \})^*$  is regular expression defined as follows:<sup>3</sup>

$$v_i = \begin{cases} f_i(a_k) & : f_i(a_k) \in \Sigma_i \\ \% & : f_i(a_k) \notin \Sigma_i \end{cases} \quad f_i(a_k) =$$

Let  $T_{a_1}, \dots, T_{a_n}$  be the regular expressions representing the transitions  $t_1, \dots, t_n$  resp. Let  $M_T$  be a MDFSA which accepts the language  $T_{a_1} \dots T_{a_n}$ . We can compute the set of possible target states for state  $q$  and an input FFS  $a$  that is represented as a string  $w$  simply via computing a target state  $p = \delta(q, w)$  in  $M_T$  and inspecting all outgoing paths from  $p$ , whose labels start with  $\%$  in order to retrieve the target state identifiers in the grammar automaton  $M$ . In this way, the steps 12-14 in the algorithm in figure 1 are reduced to a simple string matching with a DFSA. We give an example to clarify the technique. Let us assume that there are two outgoing transitions from a given state,  $t_1$  and  $t_2$  which are labeled with  $[morph, noun, , ]$  and  $[morph, , a, ]$  and which lead to state 1 and 2 respectively. Turning them into corresponding regular expression representation yields  $id(morph) \cdot noun \cdot \$ \cdot \% \cdot \$ \cdot \{f, m, \} \% 1$  for  $t_1$  and analogously  $id(morph) \cdot \{noun, adj, \} \cdot \$ \cdot a \cdot \$ \cdot \{f, m, \} \% 2$  for  $t_2$ . The result of merging string representation of  $t_1$  and  $t_2$  into one DFSA  $M_T$  is shown in figure 2 in a simplified form ( $\$$  symbols were omitted).

<sup>3</sup> Note that the second part of the definition of  $v_i$  has to be a disjunction of  $\{ \top \}$  and  $\Sigma_i$  since we intend to merge all strings representing transitions into one DFSA, i.e., in case of encoding a feature with unspecified value, we have to consider all values seen in other patterns as well ( $\Sigma_i$ )

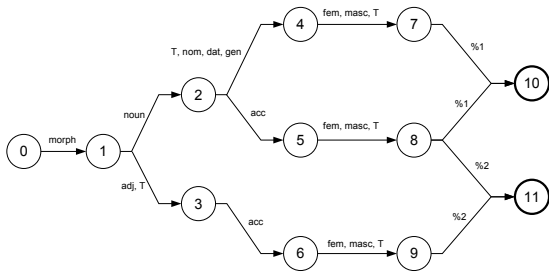


Fig. 2: Transitions labels merged into a single DFSA

## 6 Run-time Behavior

The comparison of the run-time behavior of the pattern engines discussed here is difficult due to the following reasons: (a) their expressive power is different, (b) they deploy different memory models for encoding FSAs., and (c) there were no parallel grammars available. Nevertheless, we developed a two-stage grammar in EXPRESS for the recognition of information on actors and victims in violent events. In the first stage standard named entities are recognized, e.g., persons, groups of persons, numerical expressions, whereas in the second stage, single-slot and two-slot extraction rules are applied to retrieve the sought-after information on related events, in which the entities recognized in the first stage participate. The first-stage grammar was developed by an expert, whereas the second-level grammar was obtained via encoding of ca. 3000 automatically learned patterns into EXPRESS rule format. Subsequently the grammar has been converted in almost one-to-one manner into a XTDL grammar.<sup>4</sup>

In an experiment, the grammars were applied to a 167 MB excerpt of the news on terrorism, consisting of 122 files on a PC Pentium 4 machine with 2,79 GHz. The table 1 gives figures of the average run-time (in seconds) for processing a single file (average size of 1,37 MB) at different stages. The average number of matches per document amounted to ca. 60 000. Next,

Time \ Grammar Interpreter	XTDL	EXPRESS
core linguistic components stage I	2.451	1.818
entity-pattern matching	38.212	1.923
entity-structure production	4.172	0.515
core linguistic components stage II	1.092	0.639
event-pattern matching	12.124	0.666
event-structure production	0.156	0.013
Total	58.207	5.574

Table 1: Run-time behavior: XTDL vs. EXPRESS

we have slightly 'compressed' the XTDL grammar through using coreferencing and other XTDL specific features, which resulted in deterioration of the run-time performance by the factor of two.

Converting EXPRESS grammars into JAPE format is a more laborious task. Therefore, we have only

<sup>4</sup> It turned to be a relatively simple task since the core linguistic components provided with EXPRESS have nearly identical functionality and I/O specification as those used in SPROUT. However, some rules had to be expressed as two rules in XTDL since XTDL rules do not allow for specifying more than one output structure directly.

developed a small JAPE grammar which 'resembles' the first-level grammar developed in EXPRESS. The run-time of applying it on an average document from the same test data amounted to 62,31 sec.

## 7 Conclusion and Future Work

We presented a technique for efficiently processing extraction grammars written in EXPRESS – a pattern specification language based on regular expressions over flat feature structures, which is a blend of two recently introduced IE-oriented pattern languages, namely JAPE and XTDL.<sup>5</sup> The main motivation for developing the new pattern engine comes from a need of finding a trade-off between 'compact descriptions' and efficient processing of huge text collections. An experiment revealed that modest-size grammars can be applied on MB-sized texts within seconds, which turns to be substantially faster than processing XTDL or JAPE grammars. However, a more thorough comparison in case of JAPE is indispensable.

There are a number of interesting additional speed-up techniques that can be applied. Firstly, based on an empirical analysis of the grammar and text collection, an intelligent reordering of feature-value pairs in the FFS, e.g., features which are most likely to eliminate a high number of potential target states should precede other features, would potentially yield a better run-time performance. Secondly, one could turn input FFSs from a given node into a union of their corresponding string representations and subsequently perform on-the-fly intersection thereof with the automaton representing the outgoing transitions from a given state. In this way, matching several input FFSs could be performed in a single 'intersection' step. However, it is not clear whether this would result in a speed up since intersection operation is more time-consuming than a single string acceptance check.

## References

- [1] C. Best, E. van der Goot, K. Blackler, T. Garcia, and D. Horby. Europe Media Monitor. Technical Report EUR 22173 EN, European Commission., 2005.
- [2] H. Cunningham, D. Maynard, and V. Tablan. Jape: a java annotation patterns engine (second edition). Technical Report, CS-00-10, University of Sheffield, Department of Computer Science, 2000.
- [3] W. Drożdżyński, H.-U. Krieger, J. Piskorski, U. Schäfer, and F. Xu. A Bag of Useful Techniques for Unification-Based Finite-State Transducers. In *Proceedings of 7th KONVENS Conference, Vienna, Austria*, 2004.
- [4] W. Drożdżyński, H.-U. Krieger, J. Piskorski, U. Schäfer, and F. Xu. Shallow Processing with Unification and Typed Feature Structures – Foundations and Applications. *Künstliche Intelligenz*, 2004(1):17–23, 2004.
- [5] I. Muslea. Extraction Patterns for Information Extraction Tasks: A Survey. In *Proceedings of AAAI 1999*, 1999.
- [6] G. Neumann, R. Backofen, J. Baur, M. Becker, and C. Braun. An information extraction core system for real world German text processing. In *5th International Conference of Applied Natural Language*, pages 208–215, 1997.
- [7] G. van Noord and D. Gerdemann. Finite state transducers with predicates and identity. *Grammars*, 4(3):263–286, 2001.

<sup>5</sup> The work reported here was partially supported by the Polish MEN grant no. 3 T11C 007 27

# Conditional Random Fields vs. Hidden Markov Models in a biomedical Named Entity Recognition task

Natalia Ponomareva, Paolo Rosso, Ferran Pla, Antonio Molina  
Universidad Politcnica de Valencia  
c/ Camino Vera s/n  
Valencia, Spain  
{*nponomareva, proso, fpla, amolina*}@*dsic.upv.es*

## Abstract

With a recent quick development of a molecular biology domain the Information Extraction (IE) methods become very useful. Named Entity Recognition (NER), that is considered to be the easiest task of IE, still remains very challenging in molecular biology domain because of the complex structure of biomedical entities and the lack of naming convention. In this paper we apply two popular sequence labeling approaches: Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs) to solve this task. We exploit different strategies to construct our biomedical Named Entity (NE) recognizers which take into account special properties of each approach. Although the CRF-based model has obtained much better results in the F-score, the advantage of the CRF approach remains disputable, since the HMM-based model has achieved a greater recall for some biomedical classes. This fact makes us think about a possibility of an effective combination of these models.

## Keywords

Biomedical Named Entity Recognition, Conditional Random Fields, Hidden Markov Models

## 1 Introduction

Recently the molecular biology domain has been getting a massive growth due to many discoveries that have been made during the last years and due to a great interest to know more about the origin, structure and functions of living systems. It causes to appear every year a great deal of articles where scientific groups describe their experiments and report about their achievements.

Nowadays the largest biomedical database resource is MEDLINE that contains more than 14 millions of articles of the world's biomedical journal literature and this amount is constantly increasing - about 1,500 new records per day [1]. To deal with such an enormous quantity of biomedical texts different biomedical resources as databases and ontologies have been created.

Actually NER is the first step to order and structure all the existing domain information. In molecular biology it is used to identify within the text which words or phrases refer to biomedical entities, and then to classify them into relevant biomedical concept classes.

Although NER in molecular biology domain has been receiving attention by many researchers for a decade, the task remains very challenging and the results achieved in this area are much poorer than in the newswire one.

The principal factors that have made the biomedical NER task difficult can be described as follows [11]:

(i) *Different spelling forms existing for one entity* (e.g. "N-acetylcysteine", "N-acetyl-cysteine", "NacetylCysteine").

(ii) *Very long descriptive names*. For example, in the Genia corpus (which will be described in Section 3.1) the significant part of entities has length from 1 to 7.

(iii) *Term share*. Sometimes two entities share the same words that usually are headnouns (e.g. "T and B cell lines").

(iv) *Cascaded entity problem*. There exist many cases when one entity appears inside another one (e.g.  $\langle \text{PROTEIN} \rangle \langle \text{DNA} \rangle \text{kappa3} \langle \text{DNA} \rangle \text{binding factor} \langle \text{PROTEIN} \rangle$ ) that lead to certain difficulties in a true entity identification.

(v) *Abbreviations*, that are widely used to shorten entity names, create problems of its correct classification because they carry less information and appear less times than the full forms.

This paper aims to investigate and compare a performance of two popular Natural Language Processing (NLP) approaches: HMMs and CRFs in terms of their application to the biomedical NER task. All the experiments have been realized using a JNLPBA version of Genia corpus [2].

HMMs [6] are generative models that proved to be very successful in a variety of sequence labeling tasks as Speech recognition, POS tagging, chunking, NER, etc.[5, 12]. Its purpose is to maximize the joint probability of paired observation and label sequences. If, besides a word, its context or another features are taken into account the problem might become intractable. Therefore, traditional HMMs assume an independence of each word from its context that is, evidently, a rather strict supposition and it is contrary to the fact. In spite of these shortcomings the HMM approach offers a number of advantages such as a simplicity, a quick learning and also a global maximization of the joint probability over the whole observation and label sequences. The last statement means that the deci-

sion of the best sequence of labels is made after the complete analysis of an input sequence.

CRFs [3] is a rather modern approach that has already become very popular for a great amount of NLP tasks due to its remarkable characteristics [9, 4, 8]. CRFs are indirect graphical models which belong to the discriminative class of models. The principal difference of this approach with respect to the HMM one is that it maximizes a conditional probability of labels given an observation sequence. This conditional assumption makes easy to represent any additional feature that a researcher could consider useful, but, at the same time, it automatically gets rid of the property of HMMs that any observation sequence may be generated.

This paper is organized as follows. In Section 2 a brief review of the theory of HMMs and CRFs is introduced. In Section 3 different strategies of building our HMM-based and CRF-based models are presented. Since corpus characteristics have a great influence on the performance of any supervised machine-learning model the first part of Section 3 is dedicated to a description of the corpus used in our work. In Section 4 the performances of the constructed models are compared. Finally, in Section 5 we draw our conclusions and discuss the future work.

## 2 HMMs and CRFs in sequence labeling tasks

Let  $\mathbf{x} = (x_1 x_2 \dots x_n)$  be an observation sequence of words of length  $n$ . Let  $\mathbf{S}$  be a set of states of a finite state machine each of which corresponds to a biomedical entity tag  $t \in T$ . We denote as  $\mathbf{s} = (s_1 s_2 \dots s_n)$  a sequence of states that provides for our word sequence  $\mathbf{x}$  some biomedical entity annotation  $\mathbf{t} = (t_1 t_2 \dots t_n)$ .

HMM-based classifier belongs to naive Bayes classifiers which are founded on a joint probability maximization of observation and label sequences:

$$P(\mathbf{s}, \mathbf{x}) = P(\mathbf{x}|\mathbf{s})P(\mathbf{s})$$

In order to provide a tractability of the model traditional HMM makes two simplifications. First, it supposes that each state  $s_i$  only depends on a previous one  $s_{i-1}$ . This property of stochastic sequences is also called a Markov property. Second, it assumes that each observation word  $x_i$  only depends on the current state  $s_i$ . With these two assumptions the joint probability of a state sequence  $\mathbf{s}$  with observation sequence  $\mathbf{x}$  can be represented as follows:

$$P(\mathbf{s}, \mathbf{x}) = \prod_{i=1}^n P(x_i|s_i)P(s_i|s_{i-1}) \quad (1)$$

Therefore, the training procedure is quite simple for HMM approach, there must be evaluated three probability distributions:

(1) initial probabilities  $P_0(s_i) = P(s_i|s_0)$  to begin from a state  $i$ ;

(2) transition probabilities  $P(s_i|s_{i-1})$  to pass from a state  $s_{i-1}$  to a state  $s_i$ ;

(3) observation probabilities  $P(x_i|s_i)$  of an appearance of a word  $x_i$  in a position  $s_i$ .

All these probabilities may be easily calculated using a training corpus.

The equation (1) describes a traditional HMM classifier of the first order. If a dependence of each state on two preceding ones is assumed a HMM classifier of the second order will be obtained:

$$P(\mathbf{s}, \mathbf{x}) = \prod_{i=1}^n P(x_i|s_i)P(s_i|s_{i-1}, s_{i-2}) \quad (2)$$

CRFs are undirected graphical models. Although they are very similar to HMMs they have a different nature. The principal distinction consists in the fact that CRFs are discriminative models which are trained to maximize the conditional probability of observation and state sequences  $P(\mathbf{s}|\mathbf{x})$ . This leads to a great diminution of a number of possible combinations between observation word features and their labels and, therefore, it makes possible to represent much additional knowledge in the model. In this approach the conditional probability distribution is represented as a multiplication of feature functions exponents:

$$P_\theta(\mathbf{s}|\mathbf{x}) = \frac{1}{Z_0} \exp \left( \sum_{i=1}^n \sum_{k=1}^m \lambda_k f_k(s_{i-1}, s_i, \mathbf{x}) + \sum_{i=1}^n \sum_{k=1}^m \mu_k g_k(s_i, \mathbf{x}) \right) \quad (3)$$

where  $Z_0$  is a normalization factor of all state sequences,  $f_k(s_{i-1}, s_i, \mathbf{x})$ ,  $g_k(s_i, \mathbf{x})$  are feature functions and  $\lambda_k, \mu_k$  are learning weights of each feature function. Although, in general, feature functions can belong to any family of functions, we consider the simplest case of binary functions.

Comparing equations (1) and (3) there may be seen a strong relation between HMM and CRF approaches: feature functions  $f_k$  together with its weights  $\lambda_k$  are some analogs of transition probabilities in HMMs while functions  $\mu_k f_k$  are observation probability analogs. But in contrast to the HMMs, the feature functions of CRFs may not only depend on the word itself but on any word feature, which is incorporated into the model. Moreover, transition feature functions may also take into account both a word and its features as, for instance, a word context.

A training procedure of the CRF approach consists in the weight evaluation in order to maximize a conditional log likelihood of annotated sequences for some training data set  $D = (\mathbf{x}, \mathbf{t})^{(1)}, (\mathbf{x}, \mathbf{t})^{(2)}, \dots, (\mathbf{x}, \mathbf{t})^{(|D|)}$

$$L(\theta) = \sum_{j=1}^{|D|} \log P_\theta(\mathbf{t}^{(j)}|\mathbf{x}^{(j)})$$

We have used CRF++ open source <sup>1</sup> which implemented a quasi-Newton algorithm called LBFGS for the training procedure.

<sup>1</sup> <http://www.chasen.org/taku/software/CRF++/>

### 3 Biomedical NE recognizers description

Biomedical NER task consists in the detecting in a raw text biomedical entities and assigning them to one of the existing entity classes. In this section the two biomedical NE recognizers, we constructed, based on the HMM and CRF approaches will be described.

#### 3.1 JNLPBA corpus

Any supervised machine-based model depends on a corpus that has been used to train it. The greater and the more complete the training corpus is, the more precise the model will be and, therefore, the better results can be achieved. At the moment the largest and, therefore, the most popular biomedical annotated corpus is Genia corpus v. 3.02 which contains 2,000 abstracts from the MEDLINE collection annotated with 36 biomedical entity classes. To construct our model we have used its JNLPBA version that was applied in the JNLPBA workshop in 2004 [2]. In Table 1 the main characteristics of the JNLPBA training and test corpora are illustrated.

**Table 1:** *JNLPBA corpus characteristics*

Characteristics	Training corpus	Test corpus
Number of abstracts	2,000	404
Number of sentences	18,546	3,856
Number of words	492,551	101,039
Number of biomed. tags	109,588	19,392
Size of vocabulary	22,054	9,623
Years of publication	1990-1999	1978-2001

The JNLPBA corpus is annotated with 5 classes of biomedical entities: protein, RNA, DNA, cell type and cell line. Biomedical entities are tagged using the IOB2 notation that consists of 2 parts: the first part indicates whether the corresponding word appears at the beginning of an entity (tag B) or in the middle of it (tag I); the second part refers to the biomedical entity class the word belongs to. If the word does not belong to any entity class it is annotated as "O". In Fig. 1 an extract of the JNLPBA corpus is presented in order to illustrate the corpus annotation. In Table 2 a tag distribution within the corpus is shown. It can be seen that the majority of words (about 80%) does not belong to any biomedical category. Furthermore, the biomedical entities themselves also have an irregular distribution: the most frequent class (protein) contains more than 10% of words, whereas the most rare one (RNA) only 0.5% of words. The tag irregularity may cause a confusion among different types of entities with a tendency for any word to be referred to the most numerous class.

**Table 2:** *Entity tag distribution in the training corpus*

Tag name	Protein	DNA	RNA	cell type	cell line	no-entity
Tag distr.%	11.2	5.1	0.5	3.1	2.3	77.8

IL-2	B-DNA
gene	I-DNA
expression	O
and	O
NF-kappa	B-protein
B	I-protein
activation	O
through	O
CD28	B-protein
requires	O
reactive	O
oxygen	O
production	O
by	O
5-lipoxygenase	B-protein
.	O

**Fig. 1:** *Example of the JNLPBA corpus annotation*

#### 3.2 Feature set

As it is rather difficult to represent in HMMs a rich set of features and in order to be able to compare HMM and CRF models under the same conditions we have not applied such commonly used features as orthographic or morphological ones. The only additional information we have exploited are parts-of-speech (POS) tags.

The set of POS tags was supplied by the Genia Tagger<sup>2</sup>. It is significant that this tagger was trained on the Genia corpus in order to provide better results in the biomedical texts annotation. As it has been shown by [12], the use of the POS tagger adapted to the biomedical task may greatly improve the performance of the NER system than the use of the tagger trained on any general corpus as, for instance, Penn TreeBank.

#### 3.3 Two different strategies to build HMM-based and CRF-based models

As we have already mentioned, CRFs and HMMs have principal differences and, therefore, distinct methodologies should be employed in order to construct the biomedical NE recognizers based on these models.

Due to their structure, HMMs cause certain inconveniences for feature set representation. The simplest way to add a new knowledge into the HMM model is to specialize its states. This strategy was previously applied to other NLP tasks, such as POS tagging, chunking or clause detection and proved to be very effective [5].

Thus, we have employed this methodology for the construction of our HMM-based biomedical NE recognizer. States specialization leads to the increasing of a number of states and to adjusting each of them to certain categories of observations. In other words, the idea of specialization may be formulated as a splitting of states by means of additional features which in our case are POS tags.

In our HMM-based system the specialization strategy using POS information serves both to provide an additional knowledge about entity boundaries and to diminish an entity class irregularity. As we have seen

<sup>2</sup> <http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/>

in Section 3.1, the majority of words in the corpus does not belong to any entity class. Such data irregularity can provoke errors, which are known as false negatives, and, therefore, may diminish the recall of the model. It means that many biomedical entities will be classified as non-entity. Besides, there also exists a non-uniform distribution among biomedical entity classes: e.g. class “protein” is more than 100 times larger than class “RNA” (see Table 2).

We have constructed three following models based on HMMs of the second order (2):

- (1) only the non-entity class has been splitted;
- (2) the non-entity class and two most numerous entity categories (protein and DNA) have been splitted;
- (3) all the entity classes have been splitted.

It may be observed that each following model includes the set of entity tags of the previous one. Thus, the last model has the greatest number of states.

Besides, we have carried out various experimens with a different number of boundary tags, and we have concluded that only adding two tags (E - end of an entity and S - a single word entity) to a standard set of boundary tags, supplied by the JNLPBA corpus annotation, can notably improve the performance of the HMM-based model.

Consequently, each entity tag of our models contains the following components:

- (i) entity class (protein, DNA, RNA, etc.);
- (ii) entity boundary (B - beginning of an entity, I - inside of an entity, E - end of an entity, S - a single word entity);
- (iii) POS information.

With respect to the CRF approach, the specialization strategy seems to be rather absurd, because it was exactly developed to be able to represent a rich set of features. Therefore, instead of increasing of the states number the greater quantity of feature functions corresponding to each word should be used. Our CRF-based NE recognizer along with the POS tags information employes also context features in a window of 5 words.

## 4 Experiments

The standard evaluation metrics used for classification tasks are next three measures:

- (1) Recall (R) which can described as a ratio between a number of correctly recognized terms and all the correct terms;
- (2) Precision (P) that is a ratio between a number of correctly recognized terms and all the recognized terms;
- (3) F-score (F), introduced by [10], is a weighted harmonic mean of recall and precision which is calculated as follows:

$$F_{\beta} = \frac{(1 + \beta^2) * P * R}{\beta^2 * P + R} \tag{4}$$

where  $\beta$  is a weight coefficient used to control a ratio between recall and precision. As a majority of researchers we will exploit an unbiased version of F-score -  $F_1$  which establish an equal importance of recall and precision.

The first experiments we have carried out were devoted to compare our three HMM-based models in order to analyze what entity class splitting provides the best performance. In Table 3 our baseline (i.e., the model without class balancing procedure) is compared with our three models. Although all our models have improved the baseline, there is a significant difference between the first model and the other two models, which have shown rather similar results.

**Table 3:** Comparison of the influence of different sets of POS to the HMM-based system performance

Model	Tags number	Recall, %	Precision, %	F-score
Baseline	21	63.7	60.2	61.9
Model 1	40	68.4	61.4	64.7
Model 2	95	69.1	62.5	65.6
Model 3	135	69.4	62.4	65.7

In Table 4 the results we obtained with our CRF-based system are presented. Here, the baseline model takes into account only words and their context features. Model 1 is the final model which uses also POS-tag information.

**Table 4:** The CRF-based system performance

Model	Recall, %	Precision, %	F-score
Baseline	61.9	72.2	66.7
Model 1	66.4	71.1	68.7

At first glance, if only the F-score values are compared, the CRF-based model outperforms the HMM-based one with a significant difference (3 points). However, when the recall and precision are compared their opposite behaviour may be noticed : for the HMM-based model the recall almost always is higher than the precision whereas for the CRF-based model the contrary is true.

In Tables 5, 6 recall and precision values of the detection of two biomedical entities “protein” and “cell type” for the HMM and the CRF approaches are presented. The analysis of these tables shows the higher effectiveness of HMMs in finding as many biomedical entities as possible and their failure in the correctness of this detection. CRFs are more foolproof models but, as a result, they commit a greater error of the second order: the omission of the correct entities.

**Table 5:** Recall values of a detection of “protein” and “cell type” for the HMM and the CRF medels

Method	Protein	cell type
HMM	73.4	67.5
CRF	69.8	60.9

**Table 6:** Precision values of a detection of “protein” and “cell type” for the HMM and the CRF models

Method	Protein	cell type
HMM	65.2	65.9
CRF	70.2	79.2

The certain advantage of the CRF model with respect to the HMM one could also be disputed by the fact that the best biomedical NER system [12] is principally based on the HMMs. Nevertheless, the comparison does not seem rather fair, because this system, besides exploiting a rich set of features, employs some deep knowledge resources and techniques such as biomedical databases (SwissProt and LocusLink) and a number of post-processing operations consisting of different heuristic rules in order to correct entity boundaries.

Summarizing the obtained results we can conclude that the possibility of an effective combination of CRFs and HMMs would be very beneficial. Since generative and discriminative models have different nature, it is intuitive, that their integration might allow to capture more information about the object under investigation. The example of a successful combination of these methods can be a Semi-Markov CRF approach which was developed by [7] and is a conditionally trained version of semi-Markov chains. This approach proved to obtain better results on some NER problems than CRFs.

## 5 Conclusions

In this paper we have presented two biomedical NE recognizers based on the HMM and CRF approaches. Both models have been constructed with the use of the same additional information in order to compare fairly their performance under the same conditions. Since CRFs and HMMs belong to different families of classifiers two distinct strategies have been applied to incorporate an additional knowledge into these models. For the former model a methodology of states specialization has been used whereas for the latter one all additional information has been presented in the feature functions of words.

The comparison of the results has shown a better performance of the CRF approach if only F-scores of both models are compared. If also the recall and the precision are taken into account the advantage of one method with respect to another one does not seem so evident. In order to improve the results, a combination of both approaches could be very useful. As future work we plan to apply a Semi-Markov CRF approach for the biomedical NER model construction and also investigate another possibility of the CRF-based and the HMM-based models integration.

## Acknowledgments

This work has been partially supported by MCyT TIN2006-15265-C06-04 research project.

## References

- [1] K. B. Cohen and L. Hunter. *Natural Language Processing and Systems Biology*. Springer Verlag, 2004.
- [2] J. D. Kim, T. Ohta, Y. Tsuruoka, and Y. Tateisi. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the Int. Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA 2004)*, pages 70–75, 2004.
- [3] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th International Conference on Machine Learning*, pages 282–289, 2001.
- [4] A. McCallum. Efficiently inducing features of conditional random fields. In *Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence (UAI-2003)*, 2003.
- [5] A. Molina and F. Pla. Shallow parsing using specialized hmms. *JMLR Special Issue on Machine Learning approaches to Shallow Parsing*, 2002.
- [6] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77(2), pages 257–285, 1998.
- [7] S. Sarawagi and W. W. Cohen. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing (NIPS17)*, 2004.
- [8] B. Settles. Biomedical named entity recognition using conditional random fields and novel feature sets. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA 2004)*, pages 104–107, 2004.
- [9] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT/NAACL-03)*, 2003.
- [10] J. van Rijsbergen. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.
- [11] J. Zhang, D. Shen, G. Zhou, S. Jian, and C. L. Tan. Enhancing hmm-based biomedical named entity recognition by studying special phenomena. *Journal of Biomedical Informatics*, 37(6), 2004.
- [12] G. Zhou and J. Su. Exploring deep knowledge resources in biomedical name recognition. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA 2004)*, pages 96–99, 2004.



# Kernel Methods and String Kernels for Authorship Identification: The Federalist Papers Case

Marius Popescu  
University of Bucharest  
Department of Computer Science  
Academiei 14  
010014 Bucharest, Romania,  
mpopescu@phobos.cs.unibuc.ro

Liviu P. Dinu  
University of Bucharest  
Department Computer Science  
Academiei 14  
010014 Bucharest, Romania,  
ldinu@funinf.cs.unibuc.ro

## Abstract

The main goal of this paper is to investigate the behavior of string kernels on the well known case of authorship of disputed Federalist papers. Doing this, not only that we will assess the performance of string kernels on a data set that became a benchmark in authorship identification, but also we can compare our results with results obtained by other researchers with other methods on the same data set (other kernel methods, Markov chains based methods). A second objective of the paper is to see if the performance remains the same if the kernel is used in conjunction with different kernel methods. We will also discuss why we consider that string kernels are adequate to authorship identification.

## Keywords

authorship identification, kernel methods, string kernels, Federalist papers

## 1 Introduction

Kernel methods prove to be very effective in many computational linguistics and text analysis tasks. Authorship identification is not an exception [1, 2, 6]. The success of kernel methods in this domain is due to the fact that they are state of the art techniques in machine learning and also because they allow the use of specialized kernels that can directly manipulate the text (without explicitly embedding the text in a numerical feature space) and can incorporate prior knowledge about the problem.

Recently, Sanderson and Guenter [6] used sequence kernels for authorship attribution and compared their performance with that of probabilistic methods based on Markov chains. The main goal of this paper is to investigate the behavior of string kernels on the well known case of authorship of disputed Federalist papers. Doing this, not only will we assess the performance of string kernels on a data set that since the work of Mosteller and Wallace [5] became a sort of benchmark in authorship identification, but also we can compare our results with results obtained by other researchers with other methods on the same data set, for example: results achieved by other kernel methods [2] or by Markov chains based methods [4].

A second objective of the paper is to see if the performance achieved is inherent to the kernel type, that is if the performance remains the same if the kernel is used in conjunction with different kernel methods. We tested it with Support Vector Machines (SVM) and Kernel Fisher Discriminant (KFD) [8]. We will also discuss why we consider that string kernels are adequate to authorship identification and interpret the performance of string kernels vis-a-vis of the performance of Markov chains based methods in the light of the relation that exists between string kernels and Markov models via the so called Fisher Kernels, kernels derived from probabilistic generative models [7].

In the next section we briefly describe the kernel methods we used (SVM, KFD) and string kernels. Section 3 describes the experiments on Federalist papers and the results obtained, and the last section contains discussion, interpretation of these results and suggestions for future work.

## 2 Kernel Methods and String Kernels

Kernel-based learning algorithms work by embedding the data into a feature space (a Hilbert space), and searching for linear relations in that space. The embedding is performed implicitly, that is by specifying the inner product between each pair of points rather than by giving their coordinates explicitly.

Given an input set  $\mathcal{X}$  (the space of examples), and an embedding vector space  $\mathcal{F}$  (feature space), let  $\phi : \mathcal{X} \rightarrow \mathcal{F}$  be an embedding map called feature map.

A *kernel* is a function  $k$ , such that for all  $x, z \in \mathcal{X}$ ,  $k(x, z) = \langle \phi(x), \phi(z) \rangle$ , where  $\langle \cdot, \cdot \rangle$  denotes the inner product in  $\mathcal{F}$ .

In the case of binary classification problems, kernel-based learning algorithms look for a discriminant function, a function that assigns +1 to examples belonging to one class and -1 to examples belonging to the other class. This function will be a linear function in the space  $\mathcal{F}$ , that means it will have the form:

$$f(x) = \text{sign}(\langle w, \phi(x) \rangle + b),$$

for some weight vector  $w$ . The kernel can be exploited whenever the weight vector can be expressed as a linear combination of the training points,  $\sum_{i=1}^n \alpha_i \phi(x_i)$ , im-

plying that  $f$  can be expressed as follows:

$$f(x) = \text{sign}\left(\sum_{i=1}^n \alpha_i k(x_i, x) + b\right).$$

Various kernel methods differ by the way in which they find the vector  $w$  (or equivalently the vector  $\alpha$ ). Support Vector Machines (SVM) try to find the vector  $w$  that defines the hyperplane that maximally separates the images in  $\mathcal{F}$  of the training examples belonging to the two classes. Mathematically SVMs choose the  $w$  and  $b$  that satisfy the following optimization criterion:

$$\min_{w,b} \frac{1}{n} \sum_{i=1}^n [1 - y_i (\langle w, \phi(x_i) \rangle + b)]_+ + \nu \|w\|^2$$

where  $y_i$  is the label (+1/-1) of the training example  $x_i$ ,  $\nu$  a regularization parameter and  $[x]_+ = \max(x, 0)$ .

Kernel Fisher Discriminant (KFD) selects the  $w$  that gives the direction on which the training examples should be projected such that to obtain a maximum separation between the means of the two classes scaled according to the variances of the two classes in that direction. The optimization criterion is:

$$\max_w \frac{(\mu_w^+ - \mu_w^-)^2}{(\sigma_w^+)^2 + (\sigma_w^-)^2 + \lambda \|w\|^2}$$

where  $\mu_w^+$  is the mean of the projection of positive examples onto the direction  $w$ ,  $\mu_w^-$  the mean for the negative examples,  $\sigma_w^+$  and  $\sigma_w^-$  the corresponding standard deviations and  $\lambda$  a regularization parameter. Details about SVM and KFD can be found in [8]. What is important is that above optimization problems are solved in such a way that the coordinates of the embedded points are not needed, only their pairwise inner products which in turn are given by the kernel function  $k$ .

The kernel function offers to the kernel methods the power to naturally handle input data that are not in the form of numerical vectors, such for example strings. The kernel function captures the intuitive notion of similarity between objects in a specific domain and can be any function defined on the respective domain that is symmetric and positive definite. For strings, a lot of such kernel functions exist with many applications in computational biology and computational linguistics [8].

Perhaps one of the most natural ways to measure the similarity of two strings is to count how many substrings of length  $p$  the two strings have in common. This give rise to the  $p$ -spectrum kernel. Formally, for two strings over an alphabet  $\Sigma$ ,  $s, t \in \Sigma^*$ , the  $p$ -spectrum kernel is defined as:

$$k_p(s, t) = \sum_{v \in \Sigma^p} \text{num}_v(s) \text{num}_v(t)$$

where  $\text{num}_v(s)$  is the number of occurrences of string  $v$  as a substring in  $s$ <sup>1</sup> The feature map defined by

<sup>1</sup> Note that the notion of substring requires contiguity. See [8] for discussion about the ambiguity between the terms "substring" and "subsequence" across different traditions: biology, computer science.

this kernel associate to each string a vector of dimension  $|\Sigma|^p$  containing the histogram of frequencies of all its substrings of length  $p$ . Taking into account all substrings of length less than  $p$  it will be obtained a kernel that is called the *blended spectrum kernel*:

$$k_1^p(s, t) = \sum_{q=1}^p k_q(s, t)$$

The blended spectrum kernel will be the kernel that we will use in conjunction with SM and KFD for authorship attribution. More precisely we will use a normalized version of the kernel to allow a fair comparison of strings of different length:

$$\hat{k}_1^p(s, t) = \frac{k_1^p(s, t)}{\sqrt{k_1^p(s, s) k_1^p(t, t)}}$$

### 3 Federalist Papers Experiment

The "Federalist" papers were written during the years 1787 and 1788 by Alexander Hamilton, John Jay, and James Madison. These 85 propaganda tracts were intended to help get the U.S. Constitution ratified. They were all published anonymously under the pseudonym, "Publius." The general consensus of traditional attribution scholars (although varying from time to time) is that Hamilton wrote 51 of the papers, Madison wrote 14, Jay wrote 5, while 3 papers were written jointly by Hamilton and Madison, and 12 papers have disputed authorship - either Hamilton or Madison.

Mosteller and Wallace [5] in their impressive study attribute all the 12 papers to Madison and at present, the majority of historians believe that indeed they were all written by Madison.

The Federalist Papers discuss very similar topics and are written in an almost identical style typical for political discourse of that time. It is therefore considered a very challenging task for automatic authorship attribution.

In our experiments we followed the Mosteller and Wallace setting, treating the problem as a binary classification problem. Each one of the 12 disputed papers has to be classified as being written by Hamilton (class -1) or Madison (class +1). For training we used the 51 papers written by Hamilton and the 14 papers written by Madison. The source of the texts was Project Gutenberg<sup>2</sup>. Because the string kernels work at the character level, we didn't need to split the texts in words or to do any preprocessing. The only editing done to the papers was the replacing of sequences of consecutive space characters (space, tab new line, etc.) with only one space character. This normalization was needed in order to not increase or decrease artificially the similarity between texts because of different spacing.

In all the experiments we used a normalized blended spectrum kernel of 5 characters,  $\hat{k}_1^5$ . The value of 5 proved to be good in preliminary experiments, but was chosen based on the fact that the most important style indicators in a text are function words which usually are short (2-5 characters).

<sup>2</sup> <http://www.gutenberg.org>

First we did cross validation in order to establish values for parameters  $\nu$  (for SVM) and  $\lambda$  (for KFD). Also the cross validation had the role of estimating the generalization error of learning methods used, or how reliable these methods are. The relatively small number of training examples allowed us to use leave one out cross validation which is considered an almost unbiased estimator of generalization error. For value  $\nu = 0.05$  we obtained 0% leave one out error for SVM. In the case of KFD the best value for  $\lambda$  was 0.001 and the leave one out error was 1.5%, which means that in only one case when the paper no. 23 was held out for testing it was classified as written by Madison instead of Hamilton. These values of leave one out error indicate that both SVM and KFD with the  $k_1^5$  string kernel are very reliable, the probability that their predictions are correct is very high.

Tested on disputed papers, both methods attributed all disputed papers to Madison. These results are thus consistent with results obtained by other learning methods. It is remarkable that string kernel (which works at the character level) obtained the same performance as a linear kernel [2] which used 70 function words as features, these function words being identified as good candidates by Mosteller and Wallace [5].

Compared with another method that works at the character level, string kernel performed better. Khmelev and Tweedie [4] used a Markov chain of letters and also tested it on Federalist papers case. Concerning the disputed papers the results are the same, but they reported a leave one out cross validation error of 9%. The better performance of string kernel is not surprisingly given the relation that exists between string kernels and Markov models of texts. In [7] it is proved that  $p$ -spectrum kernel can be viewed as Fisher kernel of a Markov generation Process. Fisher kernels are a principled way of combining the power of generative models (like Markov models) with that of discriminative methods (like SVM) and their performance often outperforms the performance of generative models alone or the performance of discriminative methods alone.

The fact that SVM and KFD have almost the same leave one out cross validation error (0% and 1.5% respectively) indicates that the good performance is mainly due to the string kernel and not to a particular combination of kernel and learning method.

We also tested to what author the methods will attribute the 3 papers written jointly by Hamilton and Madison. Both methods attributed these papers to Madison although the confidences<sup>3</sup> of these predictions were smaller than the confidences of the predictions for disputed papers or the confidences obtained in cross validation. Mosteller and Wallace also concluded that the most part of the joint papers was in fact written by Madison, and in some edition these papers are labeled as "Madison (with the assistance of Alexander Hamilton)" [3].

<sup>3</sup> The confidence of a prediction is the real number returned by SVM or KFD when they classify an example. The confidence is not a probability, but can be interpreted as a measure of the accuracy of the prediction.

## 4 Discussion

In this paper we have shown that string kernels are adequate for authorship identification. The reason for this in our opinion is the fact that similarity of two strings as it is measured by string kernels reflect the similarity of the two texts as it is given by the short words (2-5 characters) which usually are function words, but also take into account other morphemes like suffixes ("ing" for example) which also can be good indicators of author's style.

In future work it would be useful to test string kernels on authorship problems for other languages which are inflected to a greater extent in order to verify above hypothesis.

## Acknowledgments

Research supported by MEdC-ANCS.

## References

- [1] Joachim Diederich, Jörg Kindermann, Edda Leopold, and Gerhard Paass. 2003. Authorship attribution with support vector machines. *Applied Intelligence*, 19(1-2):109-123.
- [2] Glenn Fung. 2003. The disputed federalist papers: Svm feature selection via concave minimization. In *Richard Tapia Celebration of Diversity in Computing Conference*, pages 42-46. ACM.
- [3] Alexander Hamilton, James Madison, and John Jay. 1982. *The Federalist Papers (With an introduction and commentary by Garry Wills)*. Bantam Classic, New York, NY, USA.
- [4] Dimitri V. Khmelev and Fiona J. Tweedie. 2001. Using markov chains for identification of writers. *Literary and Linguistic Computing*, 16(3):299-307.
- [5] Frederick Mosteller and David L. Wallace. 1964. *Inference and Disputed Authorship: The Federalist*. Addison-Wesley, Massachusetts.
- [6] Conrad Sanderson and Simon Guenter. 2006. Short text authorship attribution via sequence kernels, markov chains and author unmasking: An investigation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 482-491, Sydney, Australia, July. Association for Computational Linguistics.
- [7] Craig Saunders, John Shawe-Taylor, and Alexei Vinokourov. 2003. String kernels, fisher kernels and finite state automata. In *S. Thrun S. Becker and K. Obermayer, editors, Advances in Neural Information Processing Systems 15*, pages 633-640. MIT Press, Cambridge, MA.
- [8] John S. Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA.

# Automatic Detection of Quotations in Multilingual News

Bruno Pouliquen, Ralf Steinberger, Clive Best

European Commission – Joint Research Centre

Via Enrico Fermi 1, 21020 Ispra (VA) Italy

{Bruno.Pouliquen, Ralf.Steinberger, Clive.Best}@jrc.it

## Abstract

We present fully functional software that identifies direct speech quotations as part of its automatic analysis of more than 20,000 news articles per day in currently 11 languages. The system currently identifies over 2600 quotations per day, together with the person who made the quotation and – where applicable – the persons or organisations mentioned in the quotation. The most recent quotations from and about each person are listed on this person's dedicated information page, which is updated daily. As another component of the system also identifies variants of each name, the quotes can be assigned to the same person even if his or her name is spelled differently, allowing users to view all quotations from or about any of the currently 615,000 person names in the system's database in any of these languages. This automatic news analysis system is publicly accessible at <http://press.jrc.it/NewsExplorer/>.

## Keywords

Quotation recognition, Named Entity Recognition, name variant merging, multilinguality.

## 1. Introduction

Many people and organisations are interested in finding quotations made by themselves or by other people in the world's media. The major interest groups looking for quotations are political analysts, company researchers and political actors. The motivation for the interest typically is the search for product feedback, for corporate image-relevant information, or for media feedback on political initiatives. We therefore developed an automatic tool that sieves through large quantities of media reports and extracts quotations plus the speakers and the persons referred to in the quotations. Due to the multilingual requirements in the European context, the developed quotation extraction tool had to be multilingual (it currently covers eleven languages). Due to this requirement, the applied methods needed to be simple and easy to extend to new languages.

In this paper, we first present related work (section 2) and the analysis data, i.e. the collection of media reports from which we extract quotations every day (3). We then give an overview of the method (4) and describe the details of the algorithm (5). In Sections 6 and 7, we present evaluation results and discuss them. Section 8 points to future work and draws conclusions.

## 2. Related Work

To our knowledge, there are only few online automatic systems that detect quotations by and about persons from text. Dimitrov et al. [6] developed a technique to resolve anaphora and applied it to quoted text (English only). There are a number of manually compiled websites that list famous or important quotations: *QuoteLand* (see [7]) allows to search for quotations by topic or author; *Quotation-Page* (see [8]) offers a large collection of historical quotes by known personalities; *WikiQuotes* (see [9]) is a compendium of several thousand user-collected important words in various languages, sometimes accompanied by their translation into English; *ThinkExist* (see [10]) is a large database of 300,000 English quotations, compiled over five years by more than 9,000 individuals. Most of these sites concentrate on historical quotations, and all of them are compiled manually. *DayLife* (see [11]) seems to detect recent quotes automatically in English language news. However technical details on how their system works are not known. Our own system, in comparison, automatically collects an average of over 2,600 quotations per day in eleven languages and is thus completely up-to-date. Currently, quotes are only listed on the relevant individual person pages of the JRC's NewsExplorer application (see [5]), but the plan is to make the collection searchable and display the most important quotes each day on a separate page (see Section 8 of this paper on future work).

## 3. Media Material

The JRC's *Europe Media Monitor* system (see [2]) gathers an average of 35,000 news article per day in 32 languages, by continuously monitoring about 1,000 public news sites from around the world for newly published information. The aggregated results are publicly accessible on the *EMM-NewsBrief* web site (<http://press.jrc.it>), which is updated every ten minutes.

The related *EMM-NewsExplorer* application (see [5]) clusters all articles gathered during the previous day by similarity in order to group all articles about the same subject or event. Each of these clusters is then further analysed to extract additional information, including the countries and geographical places mentioned and the references to persons and organisations. An average of 300 new person

names are automatically recognised every day together with the ‘titles’ they are associated with.

A database of all person names ever extracted by NewsExplorer is constantly updated fully automatically with the newly found information. This includes the information in which news clusters they appear, which other persons, countries and places they get mentioned with most frequently, which are the most common titles referring to them, etc. It is important to note that the whole process is automatic and that the displayed information is the result of statistics on extracted information from clusters of news.

This name repository is used to display a dedicated web page for each person, showing all the information the system was able to gather for this person (see Figure 1 as an example for information about Alexander Litvinenko).

In addition to clusters, countries and other associated persons we wanted to be able to detect automatically the quotations made by each of the persons in different languages. Moreover the quotations made by other persons about them was considered to be useful, too.

**Alexander Litvinenko**  
Information about this person was last updated on Monday, December 4, 2006.

Names	Key Titles and Phrases	External resources
Alexander Litvinenko (eu,fi)	russo (it,pt - 181)	 <p>Image obtained automatically from Wikipedia</p>
Alexandre Litvinenko (fr)	agent russe (fr - 69)	
Alexander Litvinenko (de)	ruso (es - 115)	
Aleksandr Litvinenko (fi,no)	kritikari (de - 31)	
Aleksander Litvinenko (nl,sv)	agent (en,nl - 47)	
Александр Литвиненко (ru)	russa (it - 46)	
Александра Литвиненко (ru)	russa (de,fr - 58)	
Alexandr Litvinenko (it)	agente (de,sv - 34)	
Oleksandre Litvinenko (fr)	agent secret russe (fr - 20)	
Oleksandr Litvinenko (en)	morte di (it - 27)	
Oleksander Lybynenko (de)	russi (it - 11)	
Aleksandar Litvinenko (hr)	43 ans (fr - 11)	
Александр Вальтерович Литвиненко (ru)		
Alexandr Litvinenko (cs)		
亞歷山大·利特維年科 (zh)		
Alexander Litvinenko (pl)		
Alexander Litvinenko (it)		
アレクサンダー・リトビネンコ (ja)		
Alexander Walterowitsch Litwinenko (de)		

**Figure 1.** Snapshot of part of the NewsExplorer page on the Russian spy Alexander Litvinenko, listing the automatically gathered name variants found in multilingual news and the most frequent titles and phrases that help to identify the name in running text. The example shows that different kind of information on Litvinenko (age, profession, nationality, death, etc.) was found in texts written in different languages.

## 4. Method

As it was our aim to detect quotations in many different languages, we kept the linguistic input as simple as possible. We thus rely mainly on lexical patterns with character-level regular expressions, which are easily transposable to new languages.

As mentioned previously, our material consists of news articles in various languages (currently 32 in EMM). While we are aiming at detecting quotations in all these languages, we currently detect them in only eleven of them

(Arabic, German, English, Spanish, French, Italian, Dutch, Portuguese, Romanian, Russian and Swedish).

The method used is quite simple: we look in the text of each article for quotation markers that are found close to reporting verbs (*say*, *declare* etc.) and known person names. For our purposes, known person names are those that have been found in at least five different NewsExplorer news clusters.

In most news articles, names found next to quotes are not full names consisting of first and last name. Common example types for quotations found in text are the following:

- (1) *Tony Blair said "We stand ready to support you in every way".*
- (2) *"We stand ready to support you in every way," Blair said.*
- (3) *Tony Blair visited Iraq... He said "We stand ready to support you in every way".*
- (4) *Tony Blair visited Iraq... "We stand ready to support you in every way" the British Prime Minister said.*

Our system currently only captures the first two types. Example (1) is not very common because the newspapers usually first talk about the context (*Tony Blair visiting Iraq*) and only then they introduce quotes.

Example (2) is more common and still easy to detect accurately. The issue here is that only the last name is mentioned and that we have to infer that the quote is by *British Prime Minister Tony Blair* even though there may be other persons with the name of ‘Blair’ in our database. We achieve this by first scanning the text for all occurrences of full names (consisting of first *and* last name), and by then assuming a co-reference between the full name and the name part found.

In order to recognise the person doing the quoting in the third example, we would need to identify that the pronoun *he* refers to *Tony Blair*. We do not currently attempt to resolve such cases of anaphora because it would require additional language-specific effort and state-of-the-art anaphora resolution precision is relatively low. While [12] report up to 80 or 90% precision (below 80% with light-weight methods in [6]), the results for pronoun-drop languages like Spanish (see [13]), Italian or Korean only reach up to 74%. Anaphora resolution for pro-drop languages is less successful because subject pronouns are frequently omitted so that the gender of the subject is not made explicit in text. The following Italian quotation exemplifies this. We thus decided to ignore cases of pronoun use and to aim for higher precision, obviously to the detriment of the recall.

*Luis Medina Cantalejo ha visto tutto. "La palla era altrove - \_\_ racconta in un'intervista - e l'arbitro guardava in quella direzione"*

where the subject of the verb *racconta* is not written (here indicated by \_\_).

We do not currently try to identify the co-reference between ‘British Prime Minister’ and ‘Tony Blair’ in cases like (4), but have plans to do so. See the section on future work for details.

Our tool can rely on a highly populated database of names computed and updated daily as part of the NewsExplorer system. This database contains more than 615,000 names plus their variants, although we make only use of the 50,000 names (plus their 80,000 variants) that have been found in at least five different news clusters. The system is thus able to recognise any known name variants and to identify that they all relate to the same person. For instance, we have the following variants for the Uzbek president Islam Karimov: Islam Karimow (German), Islám Karímov (Spanish), Ислам Каримов (Russian), İslam Kerimov (Turkish), Islom Karimov (Swedish) and إسلام كريموف (Arabic).

## 5. Algorithm for quote recognition

We aim to detect all quotations accompanied by a named person as we cannot think of a use for quotations for which we do not know the name of the speaker. The system will recognise quotations only if it successfully detects three parts: the speaker name, a reporting verb and the quotation.

Our analysis of quotations in the news in various languages showed that many of the quotations are similar to the two examples below, i.e. the person making the quotation is either mentioned immediately before or after the quotation:

“I don’t think Congress ought to be running the war,”  
Bush said yesterday.

Mr. Wolfowitz said yesterday “I will accept any remedies”.

What complicates matters is the use of anaphoric expressions instead of person names (‘he said’, ‘added *the President*’) and the fact that modifiers such as *yesterday* or *in a radio interview* may be found between the reporting verb and the quote. While we do not currently deal with anaphoric expressions at all, we do try to capture at least some modifiers.

### 5.1 Components for quotation recognition

Most quotations can be identified using a small number of rules. Our rules (Section 5.2) make use of the components described in paragraphs (A) to (F):

- (A) quotation marker identification (quote-characters like “, ”, «, » etc.)
- (B) reporting verbs (e.g. *confirmed, says, declared* ...)
- (C) general modifiers, which can appear close to the verb (e.g. the adverb *yesterday*)
- (D) determiners, which can appear between the verb and the person name (e.g. *the*)

(E) trigger-for-person (e.g. *British Prime Minister*)

(F) person name (e.g. *Tony Blair*)

(G) a list of matching rules (e.g. *name verb [adverb] quote-mark QUOTE quote-mark*)

We will now discuss these in detail.

#### (A) Quotation markers

In order to mark the quotation itself, we first identify and normalise the following quote-marks: ["] (two single apostrophes), [``] (two curly apostrophes), [,] (two commas, used in some Dutch newspapers), [« /.../ »] (French quotes), [“ /.../ ”] (the English curly quotes), [<< /.../ >>] (two brackets), ["/.../" ] (double single-quotes), [‘ /.../’] (single quotes)

#### (B) Reporting verbs

They define a verb or any of its inflections that express that the string between quote-marks is a quotation. Without the presence of any of these verbs, we will not recognise the quotation. Examples are English *says, said, added, commented, sums up* and Italian *ha detto, dice, diceva*.

#### (C) General modifiers

These consist of quite generous lists of strings or regular expressions that are allowed before or after the verb. These strings are generally adverbs (*often, also, today...*), but there are also some compound expressions (*on television, last month*)<sup>1</sup>. We do not make use of external dictionaries, part-of-speech taggers or syntactic patterns. Instead, the list of modifiers has been derived empirically. To avoid listing all forms of verbs (*have said, might have said, would say...*), we also included the auxiliaries in this list of modifiers (in English: *has, have, had, would, might, could, do, did, does*).

#### (D) Determiners

In some cases, determiners can precede the name of a person. In our rules, they are allowed between the verb and the person name (English: *the*, French: *le, un, l’*, German: *der, die, seine*).

#### (E) Trigger-for-person

These patterns are usually titles of persons (*Dr., Prime Minister, French President...*). However, we prefer to call them trigger-for-person because they could be more gen-

---

<sup>1</sup> The Spanish configuration includes the following regular expression (*por la |en la |a la |en*) (*mañana/tarde*) recognising *por la mañana* or *a la tarde*.

In French: *pour sa part* and even the days of the week (*lundi, mardi...*) as it is quite common to say in French: “...” *a dit lundi Jacques Chirac*.

eral expressions referring to nationality (e.g. *the Iranian*), age (*57-year-old*) or other. In a random set of 240 English quotations, we found that in nine cases (3.75%) the title of the person was found before the person name. This low number is presumably due to the fact that the titles are used when the person is first introduced while quotes are usually mentioned further down in the article.

For the detection of names in NewsExplorer, we built (semi-automatically) an extensive list of such trigger words. In English, the list currently comprises more than 1,000 items. Recognition patterns also allow for combinations of several of them (e.g. *young Spanish Ambassador*).

### (F) Person name

The most important person names are automatically detected as part of the daily process for NewsExplorer (see specifically [1]). About 50,000 person names and their variants are compiled into an automaton, which is updated every day. The person names are then marked up in each article. In order to resolve the name part co-reference resolution, we then look up in text the uppercased words that are also part of a full name found elsewhere in the text. This method can identify ‘Tony Blair’ as the author even if only the last name of the author is used in the text (e.g. [*Tony Blair*] visited Iraq yesterday. ... “I reiterate our determination to stand four-square behind you” said [Blair]).

## 5.2 Matching rules

In order to write the quotation matching rules, we first had to carry out a survey of the various ways to express a quotation across languages. We found three generic rules and a number of additional language-specific rules.

The three generic rules are:

- (1) *quote-mark QUOTE quote-mark [,] verb [modifier] [determiner] [title] name*  
e.g. "blah blah", said again the journalist John Smith.
- (2) *name [, up to 60 characters ,] verb [:|that] quote-mark QUOTE quote-mark*  
e.g. John Smith, supporting AFG, said: "blah blah".
- (3) *quote-mark QUOTE quote-mark [; or ,] [title] name [modifier] verb*  
e.g. "blah blah", Mr John Smith said.

The following format was found only in Italian and Russian articles:

- (4) *quote-mark QUOTE1 - [modifier] verb name - QUOTE2 quote-mark*  
e.g. “Ciampi – ha detto Berlusconi – ha favorito la sinistra perché era un uomo della sinistra”  
where the author (here Berlusconi) and the reporting verb (*said*) is included *inside* the quotation marks, marked by hyphens.

The Swedish writing convention for quotations includes sentences beginning with one or two hyphens “--“:

- (5) -- QUOTE, verb [adverb] [title] name  
e.g. -- Vi försökte uppmuntra samverkan, säger Urban Lundmark.

A specifically Arabic pattern is to mention the verb *before* the person name. We therefore introduced the rule:

- (6) *verb [title] name [modifier] quote-mark QUOTE quote-mark*  
[and said minister of justice Saddam Hussein to Israel radio "we don't .."]  
وقال وزير العدل صدام حسين لإذاعة إسرائيل  
"إننا نحمل عباس المسؤولية النهائية عما يحدث".

## 6. Evaluation of quotation recognition

Users can consult the quotations of each person in NewsExplorer. The process gathers an average of 2,665 quotes per day (1647 of which are found in 7000 English articles every day). As of June 2007, we have a repository of about 1,500,000 quotes, gathered during 2 years of analysis. This repository is not currently fully exploited apart from displaying quotations of/about a person as part of the NewsExplorer’s person pages. From an application-oriented point of view, this works rather well: For many persons, NewsExplorer displays recent quotes from or about the person in many different languages.

In order to evaluate the *Recall* of the quotation recognition system, we searched a random collection of news articles (documents dated 12 July 2007) for any of the quotation markers mentioned in Section 5 and carried out a manual evaluation for 55 of the quotations found. We found that a surprisingly high number of 42 examples (76%) were quotations our system does not actually try to identify. Most of these 42 quotations were by persons whose name was not mentioned at all in the article (e.g. *the officer / their neighbour*). The remaining ones were by persons that are not part of our *known persons* (i.e. persons that have been found in at least five different news clusters over the past few years). For the remaining 13 cases, i.e. those that do fall inside our mandate and that we do try to identify, seven were correct while 6 had not been found, corresponding to a Recall of 54%. However, all of the six quotations that had been missed at document level had been found in other articles, so that the Recall *within the news collection* was in fact 100%. This finding confirms that we should aim for precision rather than recall because of the data redundancy in the EMM news collection.

The reasons why the seven investigated quotations had not been found are the following (multiple counting is possible): One quote was not identified because the speaker was only represented by a pronoun (*he*). In one case, our rules did not match because the verb form was missing (*telling* – this has now been added to the rule). In one case, the speaker’s name was badly tokenised, leading to non-recognition: For UN Secretary General *Ban Ki-moon*, our



system identified *Ban Ki* as the name and the remaining string *moon* stopped the rule from recognising the quotations (The tokenisation bug has now been fixed). The largest source of errors, however, were unknown modifiers (three cases, including *in a short statement, with relief*), leading again to non-recognition. As not all possible modifiers can be captured with our simplistic rules, such cases could only be solved by making use of a full morpho-syntactic analysis of the sentence. The only erroneous quotation recognition was an incomplete quote: Only the first part of the quote was found while the second part of the quote (continued after an interruption) was missed. This case lowered the overall *Precision* for the English language evaluation to 87.5% (7/8).

In order to evaluate the *Precision for multilingual quotation detection*, we carried out a second, mixed-language evaluation: Out of the 1,500 quotations of a given day (17/12/2006), we randomly selected 120 in 10 languages (discarding two quotations of the same person in the same language). The test set contained 1 Arabic, 10 German, 41 English, 22 Spanish, 4 French, 14 Italian, 3 Dutch, 16 Portuguese, 3 Russian and 6 Swedish texts. An expert read each article where the quotation was detected and judged the quality as “correct”, “incomplete” or “wrongly assigned”. An *incomplete* quotation is when only part of the full quotation was found, i.e. the system detects the first part of the quotation, but misses its continuation, as in the example:

*“I’m really happy for Fabio,” Materazzi told the Apcom news agency Friday. “I feel part of this distinction because I think that all the Azzurri helped a great champion like Cannavaro win an important prize”.*

In this case, only “*I’m really happy for Fabio,*” was detected by the system, while the continuation was missed. A *wrongly assigned* quotation is one where the quotation was uttered by another person than the one identified by the system. An example for such a wrongly assigned quotation is the following:

*Le porte-parole du Haut représentant de l’UE pour la politique extérieur Javier Solana a jugé “condamnabile” le saccage du terminal de Rafah...[the spokesperson of the EU High Representative for external policies Javier Solana judged “reprehensible” the devastation of the Rafah terminal].*

The system detected “*condamnabile*” as a quotation, but attributed the authorship to Javier Solana, while it should have been attributed to his spokesperson.

The mixed-language evaluation yielded the following results, by category: Correct: 81.7%, incomplete: 17.5%, wrongly assigned: 0.8% (one document).

## 7. Discussion of the results

Taking into account the simplicity of the approach, we consider the overall results to be rather good. The Precision is rather high, and the relatively low Recall at document level is often compensated by the data redundancy, i.e. the same quotation will frequently be found in another news article.

Obvious restrictions of the approach are the following:

- There is no co-reference resolution for pronouns and for titles (trigger-for-person);
- There is no recognition of unknown modifiers that separate the reporting verb and the quotation (no parsers are used to recognise adverbials in the shape of adverbs, noun phrases such as *with relief* and prepositional phrases such as *in a short statement*).
- Quotes in genitive constructions are currently assigned to the wrong person (In “...” *said Blair’s spokesperson*, Blair would be identified as the author of the quote).

However, the simplicity of the system also has important advantages:

- The process is fast and can detect a high number of quotations in only a few seconds.
- Multilinguality is not an obstacle: NewsExplorer is currently handling eleven languages for quote recognition and gathers quotations of the same person in many news articles from around the world.
- The system is fully automatic. It currently runs every morning and adds new quotations of the last day to every person page.
- Time and source of the quotation are identified and displayed. The user can thus always read the full article (if it is still available on the original website) to verify the correctness of the quotation.

## 8. Future work / Conclusion

We would like to improve the accuracy of the recognition. As the evaluation showed, a full morpho-syntactic analysis of the sentences containing quotations would be beneficial, especially to deal with the wide range of adverbials that cannot all be listed as part of our simplistic rules. The cost for a full sentence analysis, however, would be that the tool would be less easily extendable to new languages because a different parser would be required for each language.

We are aware that pronoun co-reference resolution would be an important step towards increasing the recall of the system, although the error rate in anaphora resolution might lead to wrongly assigned quotations, which we want to avoid as much as possible. Instead, we may want to focus on the co-reference between titles (e.g. *Spanish Prime Minister*) and names (*José Zapatero*), by making use of the wealth of information in NewsExplorer on person names



and their frequently attributed titles. This would help to attribute quotations correctly in sentences like the following:

*[José Zapatero] visited France on Monday. "We are friends" said the [Spanish Prime Minister].*

It might not be too difficult to link multi-part quotes ("Yes we do," declared John "we will win"), using relatively simple patterns. We should investigate this.

Regarding the usage of the output of the system in NewsExplorer, we would like to offer a separate page showing the most important quotations of the day. This would require finding a criterion to rank the quotations. An idea would be to make use of multilinguality to show first the quotations by persons having made most of the quotes of the day across all languages. In this context, we also plan to develop an interface allowing users to search quotations by name or using free-text search.

We have started experimenting with detecting the sentiment of quotations and to classify them into positive and negative statements. News analysts may be rather interested in knowing the attitude of public figures towards certain themes or persons.

As part of a larger effort to extract specific relations between persons (e.g. Tanev 2007, Pouliquen et al. 2007), we plan to build a *quotation network*. The idea here is to identify a social network based on who makes reference to whom in their quotations (see Figure 2 and the prototype application at <http://langtech.jrc.it/picNews.html>).

Our system is now fully functional and identifies about 2,600 quotations per day in eleven languages. The quotations from and about a person are publicly accessible at the site <http://press.jrc.it/NewsExplorer/>.

The NewsExplorer website is very popular (getting up to 1,200,000 hits per day), among other things because it compiles information about over 615,000 persons. The quotations (from the person, or about a person or organisation) contributes to this success. The multilingual aspect presumably is a determining feature, as well. Future developments will make the quotations more visible to the end-user.

## 9. Acknowledgement

We thank the following persons for their help in developing language-specific resources for quotation recognition:

Anna Widiger (Russian, German), Camelia Ignat (Romanian), Wajdi Zaghouni (Arabic), Bart Wittebrood & Tom de Groeve (Dutch) and Ann-Charlotte Forslund & Patrik Hoglund (Swedish). We thank Jenya Belyaeva for the evaluation of the results. Our special thanks go to the entire EMM team and especially to Flavio Fuart, who helped put the results online.

## 10. References

- [1] Best, C., van der Goot, E., Blackler, K., Garcia, T., Horby, D. (2005). Europe Media Monitor – System Description. Report No. EUR 22173 EN.
- [2] DayLife (2007) <http://www.daylife.com/>, last visited 12/02/2007
- [3] Dimitrov, M., Bontcheva, K., Cunningham, H., Maynard, D., (2004) A Light-weight Approach to Coreference Resolution for Named Entities in Text, Anaphora Processing: Linguistic, Cognitive and Computational Modelling, Antonio Branco, Tony McEnery and Ruslan Mitkov (editors)
- [4] Mitkov, Ruslan (2002). Anaphora Resolution. Longman.
- [5] Pouliquen, B., Steinberger, R., Ignat, C., Temnikova, I., Widiger, A., Zaghouni, W., Žižka J. (2005). Multilingual person name recognition and transliteration. Journal CORELA - Cognition, Représentation, Langage. Numéros spéciaux, Le traitement lexicographique des noms propres..
- [6] Pouliquen Bruno, Ralf Steinberger & Jenya Belyaeva (Submitted). *Multilingual multi-document continuously-updated social networks*. Workshop *Multi-source, multilingual Information Extraction and Summarization* at RANLP'2007.
- [7] QuoteLand (2006). <http://www.quoteand.com/>, last visited on 20.12.2006.
- [8] QuotationsPage (2006). <http://www.quotationspage.com/qotd.html>, last visited on 20.12.2006.
- [9] Steinberger, R., Pouliquen, B., Ignat, C., (2005) Navigating multilingual news collections using automatically extracted information. Journal of Computing and Information Technology - CIT 13, 2005, 4, 257-264.
- [10] Tanev Hristo (2007). *Unsupervised Learning of Social Networks from a Multiple-Source News Corpus*. Proceedings of RANLP'2007.
- [11] ThinkExist (2006). <http://en.thinkexist.com/>, last visited on 20.12.2006.
- [12] WikiQuotes (2006). <http://en.wikiquote.org/>, last visited on 20.12.2006.

# Discovering Temporal Relations with TICTAC

Georgiana Puşcaşu  
Research Group in Computational Linguistics  
University of Wolverhampton  
Stafford St., Wolverhampton, WV1 1SB,  
United Kingdom  
*georgie@wlv.ac.uk*

## Abstract

Temporal information plays an important role in many NLP applications. The identification of temporal relations between temporal entities (events and temporal expressions) is indispensable in obtaining the temporal interpretation of a given text. This paper presents our approach for discovering temporal relations using the temporal annotation system we have developed. This system is called TICTAC (Syntactico-Semantic Temporal Annotation Cluster) and it comprises both knowledge based and statistical techniques. It has achieved the best performance among all systems participating at the TempEval competition organised as part of SemEval-2007, competition that evaluated temporal relation identification capabilities.

## Keywords

Temporal Processing, Temporal Information Extraction, Temporal Relation, Discourse Parsing, Evaluation

## 1 Introduction

Inferring the temporal structure of text is a crucial step toward its understanding and can lead to improvement in the performance of many NLP applications, such as Question Answering (QA), Automatic Summarisation, Topic Detection and Tracking, as well as any other NLP application involving information about temporally located events.

Natural language conveys temporal information in a wide variety of ways, including tense, aspect, narrative sequence, or expressions carrying it explicitly or implicitly. Any framework that models time and what happens or is obtained in time consists of four fundamental entities: *events*, *states*, *time expressions* and *temporal relations*. An *event* is intuitively something that happens, with a defined beginning and end ([18]). *States* pertain in reality and describe conditions that are constant throughout their duration. *Temporal expressions* (TEs) are natural language phrases carrying temporal information on their own. *Temporal relations* hold between two events, between an event and a TE or between two TEs. Temporal relations can be expressed by means of verb tense, aspect, modality, as well as temporal adverbials such as: prepositional phrases (*on Monday*), adverbs of time (*then*, *weekly*) and temporal clauses (*when the war ended*).

There is a need for tools that extract from a given natural language text these fundamental temporal entities, their discovery being an important Information Extraction task. While very good performance can be obtained in the recognition and normalisation of temporal expressions, the identification of events and temporal relations is still very challenging for researchers in the area of temporal information processing.

The present paper addresses the identification of temporal relations that can be established among events and temporal expressions (TEs). The work presented here was motivated by the TempEval evaluation exercise organised as part of the SemEval 2007 competition. TempEval has tested the capability of participating systems to relate an event and a TE located in the same sentence, an event and the TE representing the Document Creation Time (DCT), and two events located in neighbouring sentences. Our approach for discovering all these types of temporal relations combines knowledge based and statistical techniques, relying mainly upon a full syntactic analysis of the text.

The paper is organised as follows. The following section motivates our intentions to identify temporal relations and surveys related work. Section 3 describes the corpus we exploited in our experiments. Section 4 explores the methodology involved in the present study. Its subsections provide more detail on how each type of temporal relation is identified. Section 5 describes the experiments and the results obtained by TICTAC, the system implementing this approach, on the TimeBank corpus ([14]). Finally, in Section 6, conclusions are drawn and future directions of research considered.

## 2 Motivation and previous work

The logic and the automatic extraction of temporal relations between events has been a research topic for over 20 years. Allen ([2]) pioneered the field by classifying all temporal relations between pairs of temporal intervals into 13 classes. Later, Dowty ([3]) introduced the idea of "narrative convention" meaning that in a succession of two verbs in the perfect tense, the second event normally occurs after the first one. Dowty's work was then continued by Webber ([22]) who used a larger set of conventions for time stamping and ordering of phrases. Lascarides and Asher ([8]) presented a framework for calculating temporal relations on the basis of semantic content, knowledge of causation, knowledge of language use, sentential syntax and compositional

semantics, accounting for the simple past and pluperfect tenses. Hitzeman et al. ([6]) argued that such an approach is too complex, and work along those lines has been discontinued.

Recently, the automatic recognition of temporal/event expressions and of the relations between them in natural language texts has become an active area of research in computational linguistics and semantics. Therefore, a specification language for the representation of events, temporal expressions and temporal links connecting them, TimeML ([15]), has been developed. TimeML has to some extent been adopted as the interlingua of temporal markup. Much work has then focused on building a collection of TimeML annotated texts. The resulting TimeBank Corpus is a 183-document news corpus manually annotated using the TimeML markup language.

Relying on the proposed annotation scheme and on its predecessors, TIDES TIMEX2 ([4]) and STAG ([18]), many research efforts have focused on temporal expression recognition and normalisation (Mani and Wilson [10], Schilder and Habel [17], Puscasu [11], Ahn [1]). A large number of NLP evaluation efforts were also centered on TE identification and normalisation, such as the MUC 6 and 7 Named Entity Recognition tasks, the ACE-2004 Event Recognition task, the Temporal Expression Recognition and Normalisation (TERN) task. Machine Learning approaches have been found to work well in detecting the boundaries of the temporal expressions, but they are outperformed by rule-based ones at the stage of extracting the TE's temporal meaning.

In what the relative ordering of temporal entities is concerned, research is still at an exploratory stage. Filatova and Hovy ([5]) extracted TEs and their temporal values and assigned them to event instances, thus indicating their temporal anchoring and their implicit temporal ordering. They obtained 82% accuracy on time-stamping 172 clauses for a single event type. Mani and Shiffman ([9]) consider clauses as the surface realisation of events, employ clause splitting to automatically identify events, time-stamp the clauses containing temporal expressions, and finally order them using a machine learning approach. The events they order are the main events of successive clauses, not necessarily every event. Vasilakopoulos and Black ([21]) explored the use of machine learning in the automatic induction of temporal relations between temporal elements, experimenting with various subsets of the TimeBank standard set of temporal relations. The authors achieve a performance of 55.45% evaluated against the set of temporal relations included in TimeBank. Other efforts in the area of event ordering include determining intra-sentence temporal relations (Lapata and Lascarides [7]), as well as inter-sentence temporal relations (Setzer and Gaizauskas[19]).

Considering the state-of-the-art of current NLP tools, clause splitting is feasible and good performance can be achieved (Mani and Shiffman [9], Puscasu [12]). We have therefore chosen in our previous work (Puscasu [13]) the clause as the expression of one event. Our aim was to identify temporal clauses by disambiguating the subordinating conjunctions used to introduce them, and to further use this information to order them temporally. Knowing that a clause intro-

duced by a certain subordinator is temporal provides us with the temporal relation between the subordinate clause and its superordinate. The present work takes a step forward and aims at identifying temporal relations not only between certain clauses, but also among any two temporal entities in a sentence, these entities being already annotated in text according to the TimeML standard. We also aim at identifying temporal relations between the main events of any two consecutive sentences, as well as the relative ordering of events with respect to the time the document was created.

### 3 Corpus description

The corpus we experiment with throughout this paper is TimeBank. The effort to put together and annotate TimeBank started in 2002 as part of the TERQAS<sup>1</sup> project. The current version of the corpus (TimeBank 1.2) comprises 183 English newspaper articles, annotated with temporal information, adding events, times and temporal links between events and times according to TimeML.

TimeML aims to capture and represent temporal information. This is accomplished using four primary tag types: TIMEX3 for temporal expressions, EVENT for temporal events, SIGNAL for temporal signals, and LINK for representing relationships. In TimeML temporal relations are indicated using the TLINK element.

The initial temporal annotation of TimeBank is considered "preliminary", as it has been shown that systematic errors appear due to the relatively small size of the corpus and due to annotation inconsistencies and incompleteness in the case of temporal link, event classification and tense and aspect annotation.

In the case of temporal relations, the 13 relation types based on James Allen's interval logic [2] and employed as values for the TimeML TLINK tag (BEFORE, AFTER, INCLUDES, IS\_INCLUDED, HOLDS, SIMULTANEOUS, IAFTER, IBEFORE, IDENTITY, BEGINS, ENDS, BEGUN\_BY, ENDED\_BY) are considered too detailed and fine-grained, therefore a restricted set of temporal relations has been recently employed by the TempEval organisers in adding another annotation layer to TimeBank. This annotation effort has enriched TimeBank with information about temporal relations between events and time expressions situated in the same sentence, between events and the Document Creation Time (DCT), as well as between the main events of two consecutive sentences. Only a restricted set of event terms are included in the annotation. The smaller set of relation types includes only 6 values: the core relations BEFORE, AFTER and OVERLAP, the two less specific to be used in ambiguous cases BEFORE-OR-OVERLAP and OVERLAP-OR-AFTER, and finally the relation VAGUE for those cases where no particular relation can be established.

In comparison with automatically detecting the TLINKs included in TimeBank 1.2, the identification of these "simplified" temporal relations is found to be more realistic and has been targeted by the very re-

<sup>1</sup> <http://www.timeml.org/terqas/>

cent TempEval evaluation exercise. We will employ the training and test data provided by the TempEval organisers for the evaluation of the temporal ordering methods proposed in this paper.

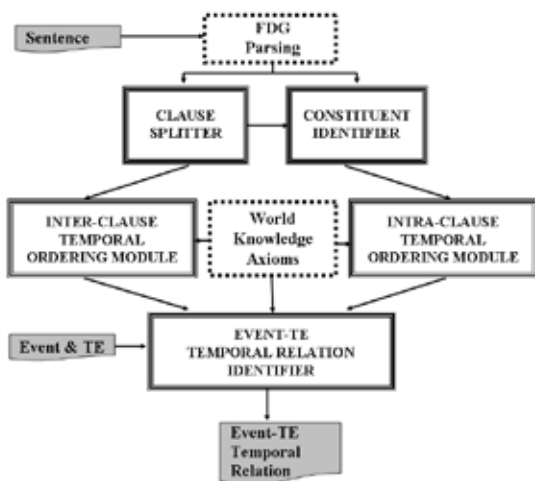
## 4 Methodology

### 4.1 Identification of intrasentential temporal relations

Our approach for discovering intrasentential temporal relations relies on sentence-level syntactic trees and on a bottom-up propagation of the temporal relations between syntactic constituents, by employing syntactical and lexical properties of the constituents and the relations between them. A temporal inference mechanism is afterwards employed to relate the two targeted temporal entities to their closest ancestor and then to each other. Conflict resolution heuristics are also applied whenever conflicts occur. Using this approach, one can discover temporal relations between any two events or between any event and any TE, whenever the two entities are situated in the same sentence.

The events and TEs are annotated in the input text in accordance with TimeML. The set of temporal relations to be predicted is: OVERLAP, BEFORE, AFTER, BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER and VAGUE.

Figure 1 depicts the processing stages involved in the identification of the temporal relation given the two temporal entities and the sentence they are in. The sentence is first annotated with morpho-syntactic and functional dependency information by employing Connexor's FDG Parser [20]. This parser reports for newspaper articles a success rate of 96.4% at morpho-syntactic level and an f-measure of 91.45% when attaching heads in a dependency relation.



**Fig. 1:** Processing stages for the discovery of intrasentential temporal relations

A clause splitter previously developed by the author is then used to detect clause boundaries and to establish the dependencies between the resulting clauses by relying on formal indicators of coordination and sub-

ordination and, in their absence, on the functional dependency relation predicted by the FDG parser. This clause splitter was evaluated on the Susanne Corpus [16] and the F-measure for the identification of complete clauses was 81.39%.

Using morpho-syntactic information we identify in each clause a set of temporally-relevant constituents (verb phrase VP, noun phrases NPs, prepositional phrases PPs, non-finite verbs and adverbial TEs).

The identified constituents and the syntactic tree of the corresponding clause are afterwards employed in a recursive bottom-up process of finding the temporal order between directly linked constituents. Each constituent is linked only with the constituent it syntactically depends on using one of the predefined temporal relations. The temporal relation is decided on the basis of heuristics that involve parameters such as: semantic properties of the two constituents' heads (whether their root forms denote reporting or aspectual start/end events - this is decided by consulting lists of reporting/aspectual start/aspectual end events extracted from TimeBank), the type of the two constituents, the syntactic relation holding between them, presence of certain temporal signals (e.g. prepositions like *before*, *after*, *until*, *since*), the tense of the clause VP and the temporal relation between any clause TE and the DCT. At the end of this recursive process there is a path of temporal relations from any clause constituent to the clause's central VP.

Each pair of clauses involved in a dependency relation are then temporally related by means of the tenses of their VPs, of the dependency relation between them and of the property of the two verbs of being reporting events or not. The underlying hypothesis is that the clause binding elements and the tenses of the two central VPs provide a natural way to establish temporal relations between two syntactically related clauses. For example, in the case of an *if*-clause, its temporal relation with the superordinate clause is BEFORE. In this way, each branch of the syntactic tree connecting a non-root node with its father gets tagged with a temporal relation.

The final stage involves retrieving the temporal relation between any two temporal entities situated in the sentence processed as above. The two entities are first tested to determine if they comply with certain world knowledge axioms that would predict their temporal relation. For example, whenever relating an event with a TE, if the TE refers to a date that is previous to the DCT, and the event is a Future tensed verb, then the event-TE temporal relation is obviously AFTER. If no axiom applies to the two entities, a temporal inference mechanism is employed to relate the two targeted temporal entities to their closest ancestor and then to each other. If conflicts occur in relating one entity to the ancestor, priority is given to the relation linked to the entity, but if the conflict is between the temporal relations of the two entities with the ancestor, the relation of the entity situated higher in the functional dependency tree with the ancestor wins.

The main advantage of this approach is the fact that the architecture and core modules are domain independent, since they mainly rely on generic correlations between syntax and temporality. At a change of domain, all we have to do is eliminate those heuristics involv-

ing the DCT or reporting events that are implicitly located on the date of the article.

## 4.2 Relating events to the document creation time

Another capability of TICTAC is the identification of temporal relations between any event and the DCT. In establishing a temporal relation between an event and the DCT, the temporal expressions directly or indirectly linked to that event are first analysed and, if no relation is detected, the temporal relation with the DCT is propagated top-down in the syntactic tree.

The processing stages for solving this task follow the course of the ones presented in Figure 1, with the only difference that the inter-clause and intra-clause temporal ordering modules no longer order clauses/constituents with respect to each other and in a bottom-up manner, but with respect to the DCT going top-down through the syntactic tree and employing the knowledge gained by identifying intrasentential temporal relations, knowledge concerning the relative ordering between same clause constituents.

Whenever establishing a temporal relation between a constituent and the DCT, the TEs directly linked to it or situated in the same clause with it are first analysed and, if no relation can be detected, the temporal relation with the DCT is propagated top-down in the syntactic tree using the father node's temporal relation with the DCT and the temporal relation between the two constituents. In the case of any clause VP, the relation with the DCT is found on the basis of the VP tense, the superordinate clause's VP tense, the syntactic relation connecting the clause with its superordinate and the relation between the superordinate clause's VP and the DCT.

## 4.3 Identification of intersentential temporal relations

Inter-sentence temporal relations are discovered by first applying several heuristics (36) that involve the temporal expressions and the tensed verbs of the two main clauses of the two sentences to be temporally related, and then by using statistical data extracted from the TimeBank corpus that revealed the most frequent temporal relation between two tensed verbs characterised by the tense information.

The task of detecting inter-sentence temporal relations is therefore reduced to relating the pair of events signalled by the main verbs of two consecutive sentences. The restricted set of temporal relations previously presented has been employed.

Figure 2 illustrates the processing flow involved in solving the task at hand.

The two sentences are first parsed using Connexor's FDG Parser and then clause boundaries are identified. We then identify the central verb of the main clause (matrix verb).

All TEs situated in the same clause with each matrix verb are investigated and if through these TEs and the relations between them and the matrix verbs we are able to predict a temporal relation then this relation represents the system output.

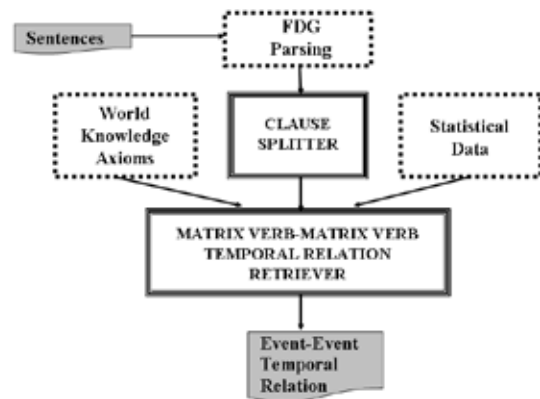


Fig. 2: Processing flow for the identification of inter-sentence temporal relations

At the next stage the semantic properties of the two matrix verbs are checked to detect whether they denote reporting events or not.

If both matrix verbs are reporting events then their tense information is used to predict a relation.

If only one matrix verb is a reporting event, then we look at the TEs linked to the other matrix verb to see if we can predict the relation to the DCT. The assumption is that a reporting event is located temporally simultaneous with the DCT and, if a relation between the other event and the DCT can be established by means of surrounding TEs, then this is the relation providing us the output. If the non-reporting event can not be positioned in time with respect to the DCT by analysing surrounding TEs, then its relation with the DCT will be the one established as described in section 4.2.

The most complicated case is the one in which both matrix verbs are non-reporting events. This case is solved by picking for each tense pair the most frequent temporal relation in the corpus, unless there is a tie or another relation with very similar frequency occurs, in which cases the two temporal relations are reconciled according to Table 1. In order to detect whether the first two most frequent temporal relations need to be reconciled, we first calculate the percentage distribution of all possible temporal relations associated to a given tense pair. Then the percentages corresponding to the two most frequent temporal relations associated to that tense pair are compared and they are considered to be very similar when the difference between them is lower than a threshold of 5%, case in which they are reconciled. In this manner a temporal relation is associated to each tense pair and, consequently, the temporal relation between the two matrix verbs is identified.

Temporal Relation	Temporal Relation	Reconciled Relation
OVERLAP	BEFORE-OR-OVERLAP	BEFORE-OR-OVERLAP
OVERLAP	BEFORE	BEFORE-OR-OVERLAP
OVERLAP	OVERLAP-OR-AFTER	OVERLAP-OR-AFTER
OVERLAP	AFTER	OVERLAP-OR-AFTER
BEFORE	BEFORE-OR-OVERLAP	BEFORE-OR-OVERLAP
AFTER	OVERLAP-OR-AFTER	OVERLAP-OR-AFTER
VAGUE	any relation	any relation

Table 1: Reconciliation between temporal relations

	BEFORE	OVERLAP	AFTER	BEFORE-OR-OVERLAP	OVERLAP-OR-AFTER	VAGUE
BEFORE	1	0	0	0.5	0	0.33
OVERLAP	0	1	0	0.5	0.5	0.33
AFTER	0	0	1	0	0.5	0.33
BEFORE-OR-OVERLAP	0.5	0.5	0	1	0.5	0.67
OVERLAP-OR-AFTER	0	0.5	0.5	0.5	1	0.67
VAGUE	0.33	0.33	0.33	0.67	0.67	1

Table 2: Relaxed scoring scheme for partial matches

## 5 Experiments

All experiments are performed on the TimeBank corpus, more specifically on the simplified annotation provided for the TempEval competition. We employ the TempEval evaluation metrics: precision, recall and f-measure, as well as the two scoring schemes: strict and relaxed. The strict scoring scheme counts only exact matches, while the relaxed one gives credit to partial semantic matches too, according to the values presented in Table 2.

The identification of temporal relations is not a straightforward task, its difficulty being also proven by the relatively low inter-annotator agreement measured on a set of TimeBank documents. In the case of annotating temporal relation types, the resulted kappa statistics<sup>2</sup> value is 0.71.

According to the TempEval evaluation results, TICTAC achieved the highest strict and relaxed f-measure scores for the tasks of intrasentential temporal ordering and event-DCT temporal relation detection.

The TempEval data was split into a set of 163 articles for training and 20 articles for testing. All 183 articles are from TimeBank. These newspaper articles are annotated with labels indicating sentence boundaries, temporal expressions and document creation times (DCT). A list of root forms of event identifying terms called the Event Target List (ETL) was employed both at the annotation and evaluation stages to define the events for which the annotation/evaluation will take place. The training data also contains for each event whose root form occurs in the ETL, its temporal relation(s) with the DCT and with the time expressions in the same sentence. Apart from this information, it also embeds temporal relations between any two consecutive sentences. The test data was first provided without the information concerning temporal relations. This information was released only after the evaluation finished.

The following tables present the detailed results corresponding to the baseline, TempEval training data, TempEval test data and the entire TimeBank corpus. For each type of temporal relation our system is able to identify, the baseline is established by the most frequent temporal relation encountered in the corresponding training data. In the case of intrasentential temporal relations, the most frequent temporal relation present in the training data is OVERLAP. For temporal relations between events and the DCT, the most prominent relation is BEFORE, and for intersentential relations OVERLAP.

Even if TICTAC is capable of recognising all intrasentential temporal relations, the existing data allowed us to evaluate only those linking an event in

<sup>2</sup> <http://timeml.org/site/timebank/documentation-1.2.html>

Intra-sentence temporal ordering	STRICT SCORE			RELAXED SCORE		
	P	R	F	P	R	F
BASELINE	0.49	0.49	<b>0.49</b>	0.51	0.51	<b>0.51</b>
TempEval-TRAIN	0.65	0.65	<b>0.65</b>	0.68	0.68	<b>0.68</b>
TempEval-TEST	0.62	0.62	<b>0.62</b>	0.63	0.63	<b>0.63</b>
TimeBank	0.65	0.65	<b>0.65</b>	0.67	0.67	<b>0.67</b>

Table 3: Results for intra-sentence temporal ordering

ETL with any TE located in the same sentence.

Event-DCT temporal ordering	STRICT SCORE			RELAXED SCORE		
	P	R	F	P	R	F
BASELINE	0.62	0.62	<b>0.62</b>	0.62	0.62	<b>0.62</b>
TempEval-TRAIN	0.80	0.80	<b>0.80</b>	0.81	0.81	<b>0.81</b>
TempEval-TEST	0.80	0.80	<b>0.80</b>	0.80	0.80	<b>0.80</b>
TimeBank	0.80	0.80	<b>0.80</b>	0.81	0.81	<b>0.81</b>

Table 4: Results for Event-DCT temporal relation detection

Our system achieves high results in the discovery of temporal relations between events and the DCT, results substantially above the baseline (18%).

Inter-sentence temporal ordering	STRICT SCORE			RELAXED SCORE		
	P	R	F	P	R	F
BASELINE	0.42	0.42	<b>0.42</b>	0.46	0.46	<b>0.46</b>
TempEval-TRAIN	0.53	0.53	<b>0.53</b>	0.63	0.63	<b>0.63</b>
TempEval-TEST	0.54	0.54	<b>0.54</b>	0.64	0.64	<b>0.64</b>
TimeBank	0.53	0.53	<b>0.53</b>	0.63	0.63	<b>0.63</b>

Table 5: Results for intersentential temporal ordering

Despite the challenges posed by intersentential temporal relation identification, our system achieved the best relaxed score among all participants at TempEval.

## 6 Conclusions

This paper presented our approach for the identification of event-time and event-event temporal relations. Our system TICTAC can relate temporal entities located in the same sentence, temporal entities with the DCT and consecutive sentences. We propose an approach mainly based on syntactical properties, combining knowledge-based and statistical techniques, all included in our automatic temporal annotation system TICTAC.

Although the system has only been evaluated on a corpus of newswire articles and despite the fact that a small number of axioms employed by the system apply only to this domain, we argue that the approach is domain independent and can be easily adapted to a new domain as long as the analysed texts are syntactically

correct. Obviously for each domain certain domain-dependent rules can improve the system's accuracy on texts belonging to that domain, but the core approach will remain unchanged.

TICTAC has been tested and evaluated not only on the TimeBank corpus, but also within the framework established by the TempEval evaluation exercise, where it achieved encouraging results. Therefore, we conclude that the proposed approach is appropriate for discovering temporal relations and we plan to find ways of improving the system's performance.

On TimeBank, TICTAC achieves according to the strict evaluation scheme a performance (f-measure) of 65% for the identification of intra-sentence temporal relations, 80% when ordering events with respect to the DCT, and 53% for the discovery of inter-sentence temporal relations.

Several future work directions emerge naturally from a first look and shallow analysis of the results. Firstly, we would like to carry out an in-depth study of other possible correlations between syntax and temporality. Secondly, we aim at exploiting apart from the syntax of the analysed text, more of its semantics.

## References

- [1] D. Ahn, S. F. Adafre, and M. de Rijke. Extracting Temporal Information from Open Domain Text. In *Journal of Digital Information Management*. 2005.
- [2] J. Allen. Towards a General Theory of Action and Time. In *Artificial Intelligence*. 1984.
- [3] D. Dowty. The effects of Aspectual Class on the Temporal Structure of Discourse: Semantics or Pragmatics? In *Linguistics and Philosophy*. 1986.
- [4] L. Ferro, I. Mani, B. Sundheim, and G. Wilson. TIDES Temporal Annotation Guidelines. Technical Report MTR 01W0000041, The MITRE Corporation, 2001.
- [5] E. Filatova and E. Hovy. Assigning Time-Stamps to Event-Clauses. In *Proceedings of the 2001 ACL Workshop on Temporal and Spatial Information Processing*, 2001.
- [6] J. Hitzeman, M. Moens, and C. Grover. Algorithms for Analyzing the Temporal Structure of Discourse. In *Proceedings of the Annual Meeting of the European Chapter of the Association of Computational Linguistics (EACL'95)*, 1995.
- [7] M. Lapata and A. Lascarides. Inferring Sentence-Internal Temporal Relations. In *Proceedings of HLT-NAACL 2004*, 2004.
- [8] A. Lascarides and N. Asher. Temporal Relations, Discourse Structure, and Commonsense Entailment. In *Linguistics and Philosophy*. 1993.
- [9] I. Mani and B. Shiffman. Temporally Anchoring and Ordering Events in News. In J. Pustejovsky and R. Gaizauskas, editors, *Time and Event Recognition in Natural Language*. John Benjamins, 2004.
- [10] I. Mani and G. Wilson. Robust temporal processing of news. In *Proceedings of ACL 2000*, 2000.
- [11] G. Puscasu. A Framework for Temporal Resolution. In *Proceedings of the LREC2004*, 2004.
- [12] G. Puscasu. A Multilingual Method for Clause Splitting. In *Proceedings of the 7th Annual Colloquium for the UK Special Interest Group for Computational Linguistics*, 2004.
- [13] G. Puscasu. On the Identification of Temporal Clauses. In *Proceedings of the Mexican International Conference on Artificial Intelligence (MICAI 2006)*, 2006.
- [14] J. Pustejovsky, P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, and M. Lazo. The TIMEBANK Corpus. In *Proceedings of Corpus Linguistics 2003*, 2003.
- [15] J. Pustejovsky, R. Sauri, A. Setzer, R. Gaizauskas, and R. Ingria. TimeML Annotation Guidelines Version 1.0. <http://www.cs.brandeis.edu/jamesp/arda/time/>, 2002.
- [16] G. Sampson. *English for the computer: the SUSANNE corpus and analytic scheme*. Oxford University Press, 1995.
- [17] F. Schilder and C. Habel. From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages. In *Proceedings of the 2001 ACL Workshop on Temporal and Spatial Information Processing*, 2001.
- [18] A. Setzer. *Temporal Information in Newswire Articles: An Annotation Scheme and Corpus Study*. PhD thesis, University of Sheffield, 2001.
- [19] A. Setzer and R. Gaizauskas. On the Importance of Annotating Event-Event Temporal Relations in Text. In *Proceedings of the LREC Workshop on Temporal Annotation Standards*, 2002.
- [20] P. Tapanainen and T. Jaervinen. A non-projective dependency parser. In *Proceedings of the 5th Conference of Applied Natural Language Processing, ACL*, 1997.
- [21] A. Vasilakopoulos and W. J. Black. Temporally Ordering Event Instances in Natural Language Texts. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*, 2005.
- [22] B. Webber. Tense as Discourse Anaphor. *Computational Linguistics*, 14(2):61 – 73, 1988.

# Sequencing of verbs – a study on tense and aspect using unsupervised learning

Catherine Recanati, Nicoleta Rogovschi, Younès Bennani

LIPN - UMR 7030

CNRS - Université Paris 13

F-93430 Villetaneuse, France

Firstname.Secondname@lipn.univ-paris13.fr

## Abstract

We report here the results of an attempt at using data mining tools for inspecting sequences of verbs from French accounts of road accidents. This analysis comes from an original approach of unsupervised learning allowing the discovery of the structure of sequential data. The entries of the analyzer were only made for the verbs appearing in the sentences. It provided a classification of the linking between two successive verbs into four distinct clusters, allowing thus text segmentation. We give here an interpretation of these clusters by applying statistical analysis to independent semantic annotations.

## Keywords

Time, tense, aspect, semantics, discourse, unsupervised learning, data mining.

## 1. Introduction

Many studies underline the importance of time in the narrative structure of a text (see [13] for French narratives). Much has been written about the opposition between the *passé simple* and the *imparfait*. It has been shown that one could not carry out the analysis of the successions of events using only tenses without referring to aspect (see Kamp, Vet or Vlach in [6], or [4], [12], [2], etc.). There are also several links between aspect and others semantic phenomena, such as intentionality and causality. Since there are links between time and aspect, the idea of coupling tenses and aspectual categories seemed natural and promising to us. In this work, we shall attempt to detect regularities in sequences of verbs (within sentences) by focusing on their tense and aspectual categories, and on the assumption that this is possible, within a restricted framework, we shall assign them a “meaning.”

The texts we analyze are short accounts in French of road accidents intended for the insurers. Their main purpose is to describe the accident, and its causes, and to identify those who are responsible. The verbs in the reports were encoded as pairs (*cat*, *tense*), where *cat* is one of the four aspectual categories of a verb, and *tense* its grammatical tense. We sought here to isolate typical sequences of such “verbs” - on the hypothesis that, if such classes exist, they might

have an overall meaning, at the very least for the type of account considered. We are fully aware of the difficulty concerning the value of this work, this last being based on the postulate that such sequences (relatively poor from a syntactic-semantic point of view) can be meaningful, and that the classification we obtained is not contingent. However, paucity of resources is a great advantage for automatic applications, and the experiment thus deserved to be undertaken. Let us add that the mathematical tools used here make it possible to check the statistical validity of the categories obtained, and that our semantic validation has been carried out with annotations unused by the training process.

## 1.1 Interests of our formal approach

One of the interests of unsupervised learning is to allow the discovery of initially unknown categories. In this framework, the connectionist Self Organizing Maps [5] provide an efficient categorization with simultaneous visualization of the results. This visualization is given by the topological map of the data (two similar data are close on the map) providing at the same time an “intelligent” coding of the data in the form of prototypes. Since these prototypes are of same nature as the data, they are interpretable, and the map thus provides a summary of the data. From this coding, we took the Hidden Markov Models (HMM) to model the dynamics of the sequences of data (here, verbs of the sentences). The HMM are the best approach [7] to treat sequences of variable length and to capture their dynamics. This is the reason why these models have been widely used in the field of voice recognition and are particularly well adapted to our objective. To validate our hybrid approach, we used biological gene sequences. For technical details see [9].

## 1.2 Data encoding

We encoded a hundred or so texts containing 700 occurrences of verbs. In these texts, we considered all the sequences of at least two verbs delimited by the end of the sentences. The descriptions of the accidents mostly use the *imparfait* (24%) and the *passé composé* (34%), with a few sentences in the present tense. In addition, there are also



some (rare) occurrences of *passé simple* and of *plus-que-parfait*. There are however a significant number of present participles (11%), and infinitives (20%). We thus decided to retain all the tenses and carried out the training by using nine codes<sup>1</sup> for the tenses of verbs. For coding, the four aspectual categories of verbs originally introduced by Vendler [11] and Kenny were combined with the tense of the verb. Our indexing is based on our lexical conception of these categories [8]. Table 1 briefly summarizes the differences between these four semantic categories.

**Table 1. The four aspectual categories of verbs**

<p><b>Verbs of STATE</b> homogeneous, durative, habitual, dispositional <i>be / expect / know</i></p>	<p><b>Verbs of ACTIVITY</b> homogeneous process, unbounded <i>drive / run / zigzag</i></p>
<p><b>Verbs of ACCOMPLISHMENT</b> process bounded by an end <i>go through / reverse into</i></p>	<p><b>Verbs of ACHIEVEMENT</b> quasi punctual event <i>cross / hit</i></p>

**Example** « Le véhicule B circulait sur la voie de gauche des véhicules allant à gauche. Il a accroché mon pare-choc et m'a entraîné vers le mur du pont de Gennevilliers que j'ai percuté violemment ». This description will be first reduced to the sequences of verbs: (circulait, allant) / (a accroché, a entraîné, ai percuté) – which are then numerically encoded as (category, tense) pairs: (act., IM) (acc., pp) / (ach., PC) (acc., PC) (ach., PC).

## 2. First confirming of our intuition

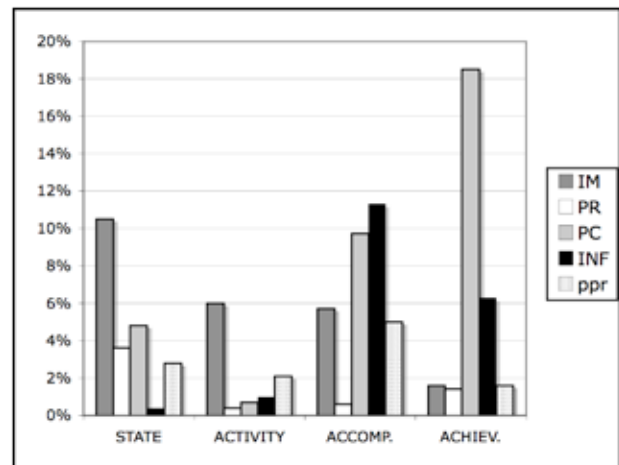
The first results are the percentages of tenses and aspectual categories. The verbs of state account for 24% of the corpus, of activity only 10%, of accomplishment 34% and of achievement 32%. The percentages of tenses by category given graphically in Figure 1 confirm the importance of our pairing (*cat, tense*). The nature of the aspectual categories, the aspectual specialization of grammatical tenses, and the typical structure of these accounts explain these percentages rather naturally.

### 2.1 Aspectual categories

#### 2.1.1 Verbs of state (24%)

More than 70% are in the *imparfait*, the present tense, and the *participe présent*. This is not surprising since states are homogeneous, often durative, or characterize an aptitude (habitual, generic). The not insignificant proportion of

*passé composé* is explained by the frequency of verbs like “want” or “can” which are classified as verbs of states.



**Figure 1: Distribution of main tenses by category**

The small proportion of present tense arises from the fact that the account is in the past tense and the *passé historic* (preterit or *passé simple*) is a literary style not appropriate for this kind of account.

#### 2.1.2 Verbs of activity (10%)

Similarly, because the activities indicate homogeneous and not limited situations, verbs of activities are distributed quite naturally with more than 79% in the *imparfait* tense and in the form of present participles. That 10% of the verbs are in the infinitive can be easily explained by the fact that activities are processes which have a beginning and which can thus be the complement of verbs like “start”, “want”, or simply can be introduced to mention a goal (in French) with the preposition “pour”.

#### 2.1.3 Accomplishments and achievements (34-32%)

Contrary to the two preceding categories, the telic character of these verbs explains their frequency in the *passé composé*. Achievements are mostly in the *passé composé* because being punctual (or of short time length), they indicate mainly a change of state. In contrast, accomplishments often occur in the *imparfait* and as present participles, because they have an intrinsic time length, and stress rather the process than its end. The global importance of these two categories is due to the fact that the report of an accident implies a description of the sequence of the successive events that caused it.

## 2.2 Aspectual specialization of tenses

There are three different points of view in the aspectual system of French [10]. A perfective point of view is expressed by the *passé composé* and the *passé simple*. It describes a situation as being closed, which includes states (the final point is then a change of state). An imperfective or neutral points of view present on the contrary open situations. A neutral point of view is expressed by the

<sup>1</sup> IM = *imparfait*, PR = *présent*, PC = *passé composé*, PS = *passé simple*, PQP = *plus-que-parfait*, inf = *infinitif*, ppr = *participe présent*, pp = *participe passé* and pps = *participe passé surcomposé*.

present tense. An imperfective point of view is expressed by the *imparfait* (or by the locution *en train de*). The opposition perfective/imperfective is realized in these texts by the opposition *imparfait/passé composé*. We borrow from Paul J. Hopper Table 2, which give an excellent description of the opposition of these two modes with respect to the narrative structure, focus, and aspect [3].

**Table 2. The Perfective/Imperfective distinction**

PERFECTIVE	IMPERFECTIVE
Strict chronological sequencing	Simultaneity or chronological overlapping
View of event as a whole, whose completion is a necessary prerequisite to a subsequent event	View of a situation or happening whose completion is not a prerequisite to a subsequent happening
Identity of subject within each discrete episode	Frequent changes of subject
Human topics	Variety of topics, including natural phenomena
Unmarked distribution of focus in clause, with presupposition of subject and assertion in verb	Marked distribution of focus (subject, instrument or sentence adverbial)
Dynamic, kinetic events	Static, descriptive situations
Foregrounding. Event indispensable to narrative	Backgrounding. State or situation necessary for understanding motives, attitudes, etc.

### 2.3 Typical structure

A description of an accident generally starts with sentences describing the circumstances before the accident. This first part of the description is thus in the *imparfait*, and also contains many present participles and infinitives. Here the account is in background. This first part is mainly circumstantial and contains a majority of verbs of states, but also some of activities and of accomplishments. The next part of the text contains a description of the accident that mentions the succession of events leading to the accident and finishes with the crucial moment of the impact. This part of the description of the accident uses mostly verbs of accomplishment and achievement, generally in the *passé composé*. It is characterized by a perfective mode, but since the goal is to indicate the responsibilities of the various actors, one still finds here many present participles and infinitive constructions connecting several verbs (“J’ai voulu freiner pour l’éviter” : “I wanted to slow down to avoid it”). At the end of the report, one occasionally finds a section consisting of comments on the accident that often contain an inventory of the damage. This third part is often short and less easy to characterize.

## 3. Classification of verbal sequences

Our unsupervised approach provided a classification of the pairs of two successive verbs (within the same sentence) in four groups (or clusters). The profiles of the transitions obtained are represented on maps incorporating a notion of proximity. The matrix of the distances between the profiles of transitions provides a distribution of these transitions. That the profiles of the transitions fall into four distinct clusters accords with the Davies and Bouldin quality standard of unsupervised classification [1]. This is to a certain extent an additional confirmation of the fruitfulness of our pairing tense/aspect, since this number of clusters (only four) is a small number.

### 3.1 Semantic interpretation

To provide the interpretation of the clusters obtained, we carried out a certain number of semantic annotations. Thus, to account for the usual structure of these texts, we indexed all the verbs with a number indicating the thematic part of the text in which they occurred (1-*circumstance*, 2-*accident* or 3-*comment*). We have also marked some verbs with the attributes *foreground* or *background* to indicate that this part of the account was in foreground or background. To detect possible causal chains of verbs leading to the accident, we marked the verbs with the attributes *causal* or *impact* when the verb was describing a direct cause of the accident, or indicating the impact itself.

We also marked the verbs of action according to the agent responsible for the action (*A* for the author of the text and the driver of the insured automobile about which the accident report is being made, *B* for a driver of the other vehicle, and *C* for a third person who might be involved in the accident). We also noted the presence of negation, and the description of objectives or possible worlds that did not occur (attributes *negation* and *inertia*). The marking of negation was not very discriminating, and that of agent not very helpful. Table 3 summarizes the main results that we obtained with a statistical analysis by making comparisons of our semantic markers within these four clusters.

**Table 3. Summary of statistical interpretations**

Cluster (IC) Impact and Comments	Cluster (AA) Actions leading to the Accident
Very strong causality, foreground, frequent impact, goals and alternatives	strong causality, neutral foregrounding, few impacts, many goals and alternatives
Cluster (CA) Circumstances or Appearance of incident	Cluster (C) Circumstances
Little causality, background, little impact, no goal or alternative	No causality, background, no impact, many goals and alternatives

## 3.2 Description of the four clusters

### 3.2.1 Typical pairs

Information concerning the Markov chains, transferred back to the topological map, enabled us to reduce the number of typical pairs (verb1, verb2) by pruning. Table 4 is a synthesis of the remaining typical pairs.

Table 4: Typical pairs

Types	verb 1	verb 2
C 1	state or act. IM	state or act., ppr
2	state IM (or PR)	acc., INF
3	act. or ach. IM	acc. (or ach.) INF
CA 4	state or act. IM	state (or ach.), IM
5	state or act. IM	state (or ach.), PC
AA 6	acc.(or ach.) INF	acc.( or ach.) INF (or ppr)
7	ach.(or acc.) PC	acc.( or ach.) INF
8	state PC	ach. INF
IC 9	ach.(or acc.) INF	ach. PC
10	ach. or state, PC	ach.(or acc) PC

### 3.2.2 Cluster (C) of Circumstances

In this cluster, the first verb is 93% in the *imparfait*, but only 7% in the present tense, while the second is 63% in the infinitive and 30% in the present participle. From the point of view of aspectual categories, the first verb is a verb of state 56% of the time, and the second 63% of the time, a verb of accomplishment (look at Table 4 for a finer synthesis). One of the reasons why we labelled it "Circumstances" is that it contains a strong majority of verbs belonging to the first part of the account (63%). The cluster (C) is the one where the *inertia* attribute (indicating a goal or a nearby possible world) is the most frequent. This is explained by the many accomplishments introduced by the French preposition "pour" or by auxiliary verbs in the *imparfait* indicating the intentions of the driver.

Cluster (C): Le véhicule de Mme X était à très peu de distance de mon véhicule, le passage étant impossible / Je descendais l'avenue du Général De Gaulle, roulant à 45 km/h / J'estimais avoir le temps / Je venais de doubler un véhicule / Je m'apprêtais à tourner à gauche / Je reculais pour repartir.

### 3.2.3 Cluster (CA) of Circumstances or of the Appearance of an incident

One notes here a great number of verbs of states (37, 5%) and activities (17%), even more than in the preceding cluster. There are, on the other hand, an average number of achievements (29%), absent from the first verb, but quite often present on the second. This distinguishes this cluster from the preceding one, where accomplishments played this role. Here, on the contrary, accomplishments are excluded from the second place, and are clearly under-represented (16, 5%). In this group, 36% of verbs come from the first part of the account, but there are also sequences finishing by an achievement in the *passé composé* coming from the second part. This cluster contains

also 25% of verbal sequences located between the two parts. This is why we called it "cluster of circumstances or of appearance of an incident".

Cluster (CA): Au moment où je démarrais, j'ai entendu le choc arrière. Je ne m'attendais pas à ce qu'un usager désire [me dépasser] car il n'y avait pas deux voies matérialisées sur la portion de route où je me trouvais / Je circulais à environ 45 km/h dans une petite rue à sens unique où stationnaient des voitures / Je roulais rue Pasteur quand une voiture surgit de ma droite. Pour l'éviter, je me rabattais à gauche et freinais.

### 3.2.4 Cluster (AA) of Actions leading to the Accident

The third cluster clearly marks the report of the accident itself. Fifty-six percent of the pairs come from the second part of the account. It is characterized by the abundance of achievements, to the detriment of states and activities, and it includes many infinitives. Nevertheless, the present participles and infinitives allowing, as in the cluster (C), the expression of goals and possible worlds, 26% of the pairs come in fact from the first part of the account. We noted also little foregrounding - the verbs being unmarked. Many verbs are found taking part in descriptions of the causal chain of the accident, but relatively few mention the impact.

Cluster (AA): J'ai voulu m'engager sur la deuxième file, lui laissant libre la première / Voulant dépasser un semi-remorque clignotant à droite, ce dernier tourna à gauche m'obligeant à braquer à gauche pour l'éviter / J'ai immédiatement commencé à freiner / Afin d'éviter le choc, j'ai braqué sur la gauche, pensant que / Le véhicule A a pris son tournant à vive allure, sans s'assurer de ma présence sur sa droite. / Je n'ai pu apercevoir Mr X.

### 3.2.5 Cluster (IC) of the Impact and of the Comments

The verbs of achievements (45%) appear here in a larger number than elsewhere, to the detriment of activities and states (only 14, 5%). This explains why this group is used to describe the accident (57%). One observes also an increase in infinitives and participles on the first verb, and a large increase in *passé composé* on the second verb to the detriment of all tenses - except the present (8%, slightly higher than the average). Perhaps this appearance of the present tense explains the strong proportion of comments (29% instead of 18%).

Cluster (IC): Je suis tombé de l'engin qui a fini sa course sur la voie de gauche. Le véhicule A circulant sur cette voie n'a pu stopper et a percuté mon véhicule / La voiture a dérapé sur la chaussée mouillée et a percuté un trottoir puis un mur de clôture. Le conducteur du camion avait bien mis son clignotant à gauche, mais sa remorque inversait le signal sur la droite. Ne m'ayant pas touché, le conducteur s'est déclaré hors de cause et n'a pas voulu établir de constat / Le conducteur du véhicule B me doublant par la droite a accroché mon pare-choc et m'a entraîné vers le mur amovible du pont de Gennevilliers que j'ai percuté violemment.

## 3.3 Comments

### 3.3.1 Aspectual categories

This categorization distinguishes quite well states and activities from events. In a more interesting way,

accomplishments are also distinguished from achievements, justifying the distinction in opposition to the more general notion of events. We also found that the expression of goals or alternatives often accompanies the use of verbs in the present participle or infinitive - which explains the ratio obtained by clusters (C) and (AA). However, the aspectual category used influences also this expression, because the second verb in these two clusters is generally an accomplishment. Moreover, the clusters (C) and (CA) (unmarked for this purpose) differ precisely in the type of event appearing in second place. In the same way, elements differentiating the clusters (AA) and (CC), which contain a majority of events, show that the cluster (AA), which favors accomplishments, although conveying a perfective mode, is little marked for narrative foreground. This cluster is also less concerned with the causes of the accident than the cluster (CC), and it makes little reference to the impact. Goals and intentions would thus be expressed more easily by accomplishments than by achievements - which would carry more causality. It should also be noticed that this classification underlines the importance of infinitive turns and present participles, and the subtlety of their linking.

Although having well detected the opposition perfective/imperfective, surprisingly this categorization puts in the same cluster (CA) imperfective sequences and *imparfait/passé simple* breaking points. One of the explanations is that our training algorithm did not take into account the linking of sentences (nor the notion of text), so that the succession of several sentences in the *imparfait*, and the typical structure of these accounts could not be detected. For this reason, an important part of the goal of our study could not be achieved. However, the results obtained are still interesting, since the three thematic parts that we have distinguished, do not fall into the four clusters in a uniform way. It should also be noticed that this classification underlines the importance of infinitive turns and present participles, and the subtlety of their linking.

### 3.3.2 Technical improvements

We built the HMM by moving a window of size 2: a verb is analyzed by taking into consideration the two verbs which precede it and follow it, but not the N-preceding or the N-following with  $N > 1$ . This is not important here, since our analysis is at the sentence level, (and in these accounts, a sentence contains rarely more than three verbs), but for an analysis of the entire structure of a text, we will need to add this. In addition, we would have liked to produce typical sequences of variable length instead of simple pairs. This result could be obtained automatically but we have not had enough time to implement this.

## 4. Conclusion

Our general project is to apply techniques of data mining to explore textual structures. We tried here to analyze sequences of verbs in sentences from a corpus of accounts

of road accidents. We obtained a classification of pairs of two successive verbs in four groups. We succeeded in satisfactorily validating these groups, by basing our judgment on the application of statistical analyses of semantic independent annotations. This validates the power of our coupling of the grammatical tense with the aspectual category of a verb. However, this work is still at its early stages, and many points remain to be elucidated. We regret not having been able to compare our statistical analysis on cross tense/category uses with those of other types of texts (and in particular with that of simple accounts of incidents). It indeed remains to determine what is the “typological” part of the isolated sequences. Finally, we warmly thank S. Davis and A. Nazarenko for their judicious comments.

## 5. References

- [1] D.L. Davies and D.W. Bouldin. A Cluster Separation Measure. In *IEEE Transactions on Pattern Analysis and Machine Learning*, 1(2), 1979.
- [2] L. Gosselin. *Sémantique de la temporalité en français*. Duculot, Louvain-la-Neuve, 1996..
- [3] J. Hopper. Some observations on the typology of focus and aspect in narrative language. In *Studies in Language* 3.1, pp. 37-64. J. Benjamins, Amsterdam, 1979.
- [4] H. Kamp and C. Rohrer C. Tense in Texts. In Bauerle R., Schwarze C. et von Stechow A. (eds), *Meaning, Use and Interpretation of Language*, pp. 250--269. De Gruyter, Berlin, 1983.
- [5] T. Kohonen. *Self-Organizing Map*. Springer, 1995.
- [6] R. Martin et F. Nef (eds). *Le temps grammatical*. In *Langage* n°64, H. Kamp (pp. 39-64), F. Vlach (pp. 65-79), C. Vet (pp. 109-124). Larousse, Paris, 1981.
- [7] L.R. Rabiner and B.H. Juang. An Introduction to Hidden Markov models. In *IEEE ASSP Magazine*, jan. 86, pp. 4-16, 1986.
- [8] C. Recanati et F. Recanati. La classification de Vendler revue et corrigée. *Cahiers Chronos* 4, La modalité sous tous ses aspects, pp. 167-184. Amsterdam/Atlanta, GA, 1999.
- [9] N. Rogovschi, Y. Bennani et C. Recanati. Apprentissage neuro-markovien pour la classification non supervisée de données structurées en séquences. In *Actes des 7<sup>èmes</sup> journées francophones Extraction et Gestion des Connaissances*. Namur, Belgique, 2007.
- [10] C.S. Smith. The parameter of aspect. *Studies in Linguistics and Philosophy*. Kluwer Academic publishers, 1991.
- [11] Z. Vendler. Verbs and Times. In *Linguistics in Philosophy*, pp. 9-121. Cornell University Press, Ithaca, New-York, 1967.
- [12] C. Vet. Relations temporelles et progression thématique. In *Études Cognitives* 1, *Sémantique des Catégories de l'aspect et du Temps*, pp. 131-149. Académie des Sciences de Pologne, Warszawa, 1994.
- [13] M. Guillaume.. *Grammaire temporelle des récits*. Minuit, Paris, 1990.

# Where Anaphora and Coreference Meet. Annotation in the Spanish CESS-ECE Corpus

Marta Recasens, M. Antònia Martí, Mariona Taulé  
CLiC, Centre de Llenguatge i Computació  
University of Barcelona  
Gran Via de les Corts Catalanes, 585  
Barcelona 08007, Spain  
{mrecasens,amarti,mtaule}@ub.edu

## Abstract

This paper describes the guidelines of the annotation scheme designed to enrich the Spanish CESS-ECE corpus with coreference information, which is a significant step towards the definition of an exhaustive typology of pronominal and full NP coreferential expressions and their relations for Spanish. The goal is twofold. From a computational perspective, this work establishes the formal foundations for the construction of the largest corpus of Spanish texts annotated from the morphological to the pragmatic level. This corpus, which will be publicly released, will be used to construct an automatic corpus-based coreference resolution system. From a linguistic point of view, hypotheses on coreferential expressions will be tested and validated on this framework.

## Keywords

Coreference resolution, anaphora resolution, corpus linguistics, annotation scheme.

## 1 Introduction

Natural Language Processing (NLP) applications such as information extraction, text summarization and question answering need to identify all the information that is said about one same entity throughout a text. Consequently, systems capable of resolving coreference –and, by extension, anaphora– are essential. There are basically two approaches: knowledge-based and corpus-based. However, as pointed out by Mitkov [11], “corpora annotated with anaphoric or coreferential links are still a rare commodity, and those that do exist are not of a large size.” Specifically, in Spanish, the field of computational coreference resolution is still highly knowledge-based.

With a view to building a corpus-based coreference resolution system for Spanish, our project is to extend the morphologically, syntactically and semantically annotated CESS-ECE corpus (500,000 words) with pronominal and full NP coreference information. We believe that the more consistent the linguistic basis underlying the annotation scheme is, the easier it is to build a state-of-the-art coreference resolution system. On the other hand, coreferential –anaphoric in particular– relations are very much specific to each language. Unlike English, for instance, Spanish has zero and clitic pronouns. Therefore, it is fundamental to define the typology of expressions (pronouns, full NPs and proper nouns) that can enter in coreferential relations in Spanish as well as the types of

relations.<sup>1</sup> This typology forms the basis for a flexible markup scheme, rich enough to cover the cases of coreference in Spanish.

Apart from being a useful resource for training and evaluating coreference resolution systems for Spanish; from a linguistic point of view, the annotated corpus will serve as a workbench to test for Spanish the hypotheses suggested by Ariel [1] and Gundel et al. [6] about the cognitive factors governing the use of referring expressions. The only way theoretical claims coming from a single person’s intuitions can be proved is on the basis of empirical data that have been annotated in a reliable way.

This paper lays the foundations for our ongoing project. Taking the CESS-ECE corpus as the starting point, we describe the adaptation of the MATE meta-scheme for anaphora annotation [15] by considering both the information already codified in the CESS-ECE corpus and the way new information should be annotated. Sound linguistic criteria guide the decisions made throughout the process.

The rest of the paper proceeds as follows: Section 2 delimits the frame of the coreferential and anaphoric phenomena that we deal with. A brief state of the art of anaphora resolution systems existing for Spanish is provided in Section 3. The guidelines of our annotation scheme and methodology are given in Section 4. Finally, Section 5 presents our conclusions and further work.

## 2 Coreference along a continuum

Anaphora is the linguistic phenomenon by which a word is interpreted with the help of some previous item (the antecedent) in the discourse. The anaphor and the antecedent may be coreferential or “colexical” –coinage of our own–, that is, they may have the same discourse referent (1a) or just share the semantic type (1b).

- (1) a. Llegaron con buenos resultados hasta los torneos de la final, pero en ellos perdieron.
- b. Los mejores equipos de la NBA son mejores que los nuestros.
- c. La capital de Francia...en París...<sup>2</sup>

---

<sup>1</sup> Anaphora in Spanish has been mainly studied from a descriptive grammar point of view [3]. From a pragmatic perspective, the recent study by Blackwell [2] tests the neo-Gricean maxims on the basis of oral data.

Coreference and anaphora are thus closely interrelated, although not all anaphoric relations are coreferential (1b), nor are all coreferential relations anaphoric (1c). Our main concern is coreference; as regards colexicality then, we limit ourselves to solving those colexical anaphors occurring in headless definite NPs in order to recover the semantic type of the head, as they may be part of a coreferential chain.

When speakers solve a coreferential link, they rely –to a greater or lesser extent– on linguistic or/and world knowledge. The more anaphoric a coreferential expression is, the more linguistic knowledge is required; the less anaphoric, the more world knowledge. The expression of coreference is best seen as a continuum, ranging from zero and anaphoric pronouns to self-sufficient definite descriptions (DD, see Section 4.2.1) and proper nouns (Figure 1). Fully aware that it is a too coarse simplification, we propose this gradation just as a starting point. One of the goals of our project from a linguistic point of view is to achieve a better understanding of the grades of this continuum.

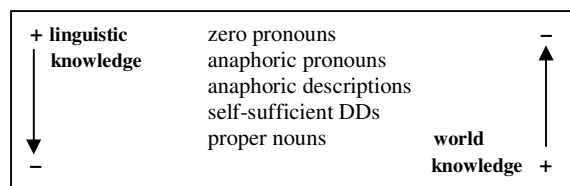


Figure 1: Range of expressions creating coreference

The piece of news in (2) illustrates different expressions referring to one same entity: *the Canary Islands*.

- (2) El número de parados registrado en Canarias en mayo subió y la cifra de desempleados en las islas se sitúa hoy en 89.764. Aunque es todavía pronto para sacar conclusiones, los políticos de la comunidad canaria ya han apuntado posibles causas y no descartan giros inesperados en la economía de la zona. De hecho, Ø es un territorio del que los periódicos suelen hablar, pero no precisamente de su tasa de paro.<sup>3</sup>

The entity is first evoked in the discourse by means of a proper noun (*Canarias*) and it is next expressed via an anaphoric description (*las islas*) –it is the previous proper noun that completes its referential meaning. Later it takes the form of a self-sufficient DD (*la comunidad canaria*), then again an anaphoric description (*la zona*). The last

<sup>2</sup> All translations throughout the paper are literal so as to make the Spanish wording as transparent as possible.

- (1) a. They got with good results to the final competitions, but they lost in them.  
 b. The best teams of the NBA are better than ours.  
 c. The capital of France...in Paris...

<sup>3</sup> (2) The number of unemployed people recorded in the Canary Islands in May increased and the number of unemployed in the islands is today 89,764. Although it is still early to draw conclusions, the politicians of the Canarian Community have already suggested possible causes and do not discard unexpected turns in the economy of the area. In fact, (it) is a region about which newspapers usually talk, but not precisely about its unemployment rate.

three elements of the coreference chain are a zero subject pronoun, a relative pronoun (*que*) and a possessive (*su*).

The whole of this continuum is the basis for the typology of coreferential expressions that our project focuses on.

### 3 Coreference resolution in Spanish

The computational coreference resolution in Spanish has been restricted to the resolution of third person anaphoric and zero pronouns [14] and to the resolution of descriptions introduced by the definite article or a demonstrative that corefer with another NP [13] by applying heuristics on shallowly parsed texts. Evaluated on a corpus containing 1,217 descriptions, Muñoz's [13] algorithm achieved 79.5% precision. Saiz-Noeda's [18] ERA system is an extension of the algorithm of [14], which, in turn, is an adaptation to Spanish of the set of constraints and preferences used by Lappin & Leass [9] in their system for English. Palomar et al.'s [14] algorithm makes use of lexical, morphological and syntactic information (partial parsing) and it obtained an accuracy of 76.8% when evaluated on two subsets (1,677 pronouns) from a corpus of a telecommunications handbook and the Lexesp corpus (made up of newspaper articles and narratives). Saiz-Noeda [18] improves the system by incorporating syntactic functions –which allows a revision and optimization of the constraints and preferences of the original algorithm– as well as semantic information –WordNet synsets measure the degree of semantic compatibility between the antecedent and the verb. In its best performance on a 3,000-word fragment (two opinion articles and a narrative text) from the Lexesp corpus, Saiz-Noeda [18] reports an accuracy of 94.49%. His evaluation was on a small corpus from a closed domain. It is not clear if the system generalizes to open-domain corpora and to the many types of coreferential relations we propose in this paper.

The knowledge-based approach is the one that has dominated anaphora resolution for a long time. However, since the 90s, in order to cater for the processing of unrestricted corpora –essential in the Internet field–, there has been a growing need for wide coverage systems. In this context, the machine-learning-based approach may be better suited than rule-based coreference anaphora resolvers. Some systems have tried using non-annotated corpora, but some linguistic issues –such as anaphora resolution– require annotated data, as little can be learnt from raw texts. We aim at testing the success of a learning-based coreference system trained on an annotated 500,000-word corpus.

With respect to corpus-based techniques, as far as we know, for Spanish there is no substantial corpus available in which coreferential or anaphoric relations are encoded. Besides, all the research on anaphora resolution carried out up to now has focused either on pronouns or on DDs, but no project has dealt with pronouns, full NPs and proper nouns all together as we do. In order to enrich the Spanish CESS-ECE corpus with coreference information, we draw on projects developed for English, considering the markup schemes, tools and strategies that have been

suggested, and making the adaptations, changes and extensions that we feel necessary given the conditions of the corpus and our purposes.

## 4 Coreference annotation scheme

The CESS-ECE corpus is a multilingual corpus that consists of a Spanish (CESS-ESP) and a Catalan corpus (CESS-CAT), 500,000 words each mostly coming from newspaper articles. The CESS-ECE corpus has already been annotated with morphological information (PoS), syntactic constituents and functions, argument structures and thematic roles, tagged with strong and weak named entities (NE), and the 150 most frequent nouns have their WordNet synset [10]. It is the largest annotated corpus of Spanish. The information already annotated is taken into account when planning the enrichment of the corpus with coreference links.

The annotation methodology of the CESS-ECE corpus is divided into two steps: a first automatic stage, and a second manual one. The former takes advantage of the annotation already contained in the corpus; while the latter enriches manually the automatic annotation and incorporates the anaphoric and coreferential links.

With regard to the annotation scheme, after considering different ones [5, 7, 8, 12, 15, 19], we opted to implement the scheme proposed in MATE/GNOME [16] for coreference annotation, because its great flexibility and modularity make it able to meet our needs. It is open to linguistic phenomena of languages other than English and, although designed for dialogues, it can be easily adapted to other textual genres. Besides, it keeps distinct the annotation of discourse entities from the annotation of links.

A number of coreference annotation projects have drawn on the MUC-6 and MUC-7 schemes [7], in which two NPs are considered to be coreferential if they refer to the same entity in the world. However, van Deemter & Kibble [20] have criticized the MUC Task Definition for violating the relation of coreference proper and mixing it with anaphora. The MATE scheme differs from that of MUC in that it is based on the discourse rather than the world. Following the discourse model [21], coreference and anaphora occur between discourse entities (DE), which may or may not refer to specific objects in the world. So the first stage in the development of an annotation scheme is the delimitation of which text constituents realize DEs that may enter in coreferential relations and are identified as markables with a <de> element.

### 4.1 First step: Automatic annotation

The starting point is the rich hierarchical syntactic annotation contained in the CESS-ECE corpus (see Figure 2). The general tag *sn* codes all NPs, while more specific tags are used to mark coordinated NPs (*sn.co*), adjunct NPs (*sn.j*), NPs containing a coordinated nominal group (*sn.x*), and elliptical subjects (*sn.e*). All these tags are also able to contemplate cases of discontinuities (splitting into a *sn.1n* and a *sn.1c* labels) and they contain a further specification –an additional letter at the end of the tag– if they are NEs: *o* for organizations, *l* for locations, *p* for

persons, *d* for dates, *n* for numbers (including percentages and money), and *a* for the rest.

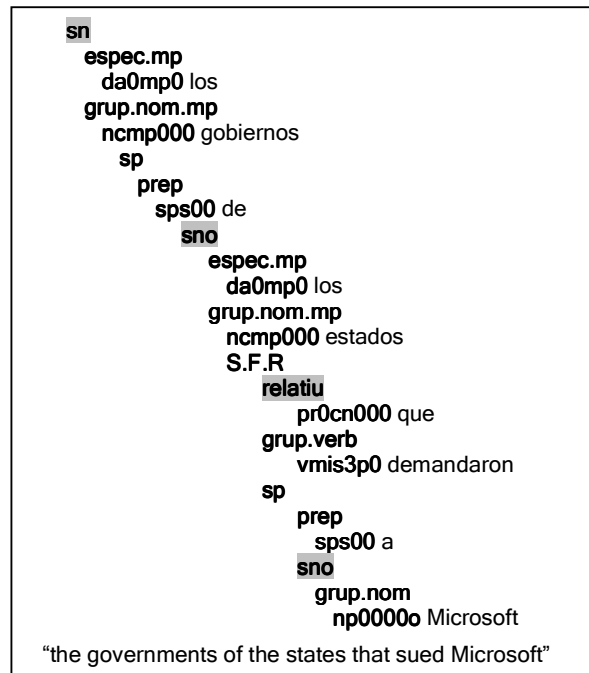


Figure 2: Fragment of a morphosyntactic tree from the CESS-ECE corpus

#### 4.1.1 Markables: <de> elements

As specified in Section 2, we aim at coding coreferential relations involving at least one NP –whose antecedent may be another NP, a VP, a clause or a sequence of clauses. Therefore, these are the text constituents that should be marked up. All NPs are automatically marked up, whether definite, indefinite, pronominal, bare nouns or proper names. Antecedents expressed by phrases other than nominal are later marked manually when necessary. The full syntactic annotation of the CESS-ECE corpus enables us to follow the MATE’s guidelines and do most of the markable identification task automatically, by instructing the computer to mark all *sn* (NPs) as <de> with an ID number, as shown by the highlighted nodes in Figure 2. Although not all NPs should be treated as markables, in this first automatic step no distinction is made. *Sn* syntactic tags treat relative clauses and appositional phrases as modifiers of the head noun, so both are included within the <de> tag. Although the Spanish reflexive pronoun *se* can also function as a verbal morpheme or as a mark for passive and impersonal constructions, the syntactic annotation already contains this information, so that the automatic annotation only identifies as <de> the uses that are really reflexive and so coreferential.

The fact that coordinated NPs, apart from their own tag, are syntactically marked with a tag for the larger NP means that three or more <de> are generated: one for each constituent NP and one for the larger NP. Subsequent references either to parts or to the whole coordination imply then no additional difficulty. Since relative

pronouns are tagged as *relatiu*, the computer is also instructed to mark them as `<de>`.

Unlike English, subject pronouns are usually omitted in Spanish –as they can be easily recovered from the verbal morphology. Otherwise a contrastive value is implied. So it is a great advantage that zero subject pronouns are syntactically already shown as \*0\*. It means that they can be automatically marked up as `<de>` and no other special tag is required.

On the other hand, in order to specify the antecedent of what we call “contextual descriptions”, we adapt the MATE’s possibility of annotating references to visible objects. Each piece of news is introduced by a `<universe>` element containing two universe entities (`<ue>`): the location and the time in which the piece of news was written (3). Both elements are automatically filled with the information heading each file.

(3) `<universe> <ue type=“location” id=“ue_1”> Toledo </ue> <ue type=“date” id=“ue_2”> 23.07.97 </ue> </universe>`

#### 4.1.2 Markables: the TYPE attributes

Apart from its ID, each `<de>` element has one or two TYPE attributes: the first specifies the type of NP –its degree of determination–, whereas the second appears only if the `<de>` is a NE or a self-sufficient DD. TYPE1 can be filled automatically by profiting from the morphological annotation of the corpus, thus copying the information contained in the specifier section (*espec.*), if there is any. In Figure 2, for instance, TYPE1 for the first `<de>` is filled with “*da0mp0*” (namely, determiner, article, masculine, plural). If the NP has no specifier, the information for TYPE1 is provided in the nominal group (*grup.nom*) section or in the node tag itself: “*rel*” for relative pronouns, “*co*” for NPs containing coordination, “*e*” for elliptical subjects (in this case, the verbal morphology is included as well), etc.

The TYPE2 attribute is automatically filled for NEs, whereas the identification of self-sufficient DDs is done in the manual annotation stage. After running the automatic annotation, the `<de>` elements obtained from the tree in Figure 2 are shown in (4).

(4) `<de id=“de_0” type1=“da0mp0”> los gobiernos de <de id=“de_1” type1=“da0mp0” type2=“NE-org”> los estados <de id=“de_2” type1=“rel”> que </de> demandaron a <de id=“de_3” type1=“np00000” type2=“NE-org”> Microsoft </de> </de> </de>`

## 4.2 Second step: Manual annotation

At this point two tasks need to be carried out. On the one hand, the automatic identification of markables is completed by adding unidentified ones. On the other hand, annotators have to annotate manually coreferential relations by incorporating the `<link>` element wherever necessary.

### 4.2.1 Adding markables

First, since incorporated clitics are not syntactically annotated in the corpus, annotators have to mark the verbal complex as a `<clit>` element (5), including as many `<clit>` as clitics there are.

(5) `<clit id=“clit_12” type1=“pp3cn000”> <clit id=“clit_13” type1=“pp3cna00”> dársele </clit> </clit>`<sup>4</sup>

Second, antecedents corresponding to a VP, a clause or a sequence of clauses are marked as `<seg>` elements.

Third, the TYPE2 attribute needs to be filled with the value “SD” for DDs which are considered to be self-sufficient, that is, NPs with the definite article that depend on no antecedent, but on world knowledge. Their autonomy can result from their generic reference (6a), their containing an explanatory modifier (6b, 6c) or their general uniqueness (6d).

- (6) a. los alemanes  
 b. las reservas de oro y divisas del Banco Central  
 c. los estados que demandaron a Microsoft  
 d. la policía<sup>5</sup>

Marking these DDs as “SD” can prove successful for a resolution system when learning to recognize definite NPs that, like proper nouns, can potentially be the first elements of a coreference chain.

On the other hand, annotators are advised to omit `<de>` elements which participate very rarely in coreferential relations, such as pronouns referring to an adjective, bound anaphors (within the scope of a quantifier), bare NPs with an attributive value, idiomatic expressions, and pronouns within fixed connectors.

### 4.2.2 Annotating coreference: <link>

The `<link>` elements serve to show coreferential relations holding between two discourse entities. This marking is especially useful for question answering, information extraction as well as text summarization. The ANCHOR attribute points to the ID of the antecedent. For the sake of simplicity, we do not distinguish between anaphora and cataphora, so that it is possible that the ANCHOR entity appears not before but after its related `<de>`. We agree to mark the closest antecedent, whether pronominal or not, as the ANCHOR.

The TYPE attribute of the `<link>` specifies the kind of coreferential relation and can take seven different values (the last three ones unique to our scheme):

- (i) **type=“ident”** (identity)

The two `<de>` share the same discourse referent. It may involve a full NP and a pronoun (7a), a proper noun and a pronoun, a proper noun and a full NP (7b), two proper nouns, or two full NPs, which may share the same head (7c) or stand in a synonymy, hypernymy or hyponymy relationship (7d). We also treat as identity relations the resolution of first and second person pronouns in quoted speech, as once within a written discourse, deictic pronouns are interpreted in an anaphoric way (7e).

- (7) a. El presidente boliviano y el jefe del partido de la oposición...ambos.  
 b. Microsoft...la firma.

<sup>4</sup> (5) ‘give-him/her/them-it’

<sup>5</sup> (6) a. the Germans  
 b. the gold and currency reserves of the Central Bank  
 c. the states that sued Microsoft  
 d. the police



- c. la falta de mano de obra en Cataluña...esta falta de mano de obra.
- d. un grupo de adolescentes...el equipo.
- e. “yo sigo” – dijo el director general de Seat.<sup>6</sup>

(ii) **type=“dx”** (discourse deixis)

The antecedent of the NP is a VP, a clause (8a), or a sequence of clauses (8b, 8c) –be it an event, fact, or proposition. The difficulty of deciding the exact textual part that serves as antecedent –which can be considerably long– has been pointed out by van Deemter & Kibble [20] and Poesio [15]. Given the relevance of events in NLP tasks, it is important for discourse deixis to have specific guidelines about how the ANCHOR should be marked. These guidelines will appear in [17]. The resolution of discourse deixis helps answer fusion in question answering and template merging in information extraction.

- (8) a. Si no cambia la situación meteorológica, cosa que el INM no prevé a corto plazo,...
- b. Pujol cree necesario que el Gobierno agilice los permisos de residencia a los inmigrantes para... Esta opinión...
- c. [...] Con esto no quiero decir que nosotros...<sup>7</sup>

(iii) **type=“poss”** (possessor)

The possessor link concerns possessive pronouns, NPs introduced by a possessive determiner, and possessive relatives. The coreference relation shows that the <de> antecedent is the possessor of the second <de>, which may express an object properly possessed as well as a part or an attribute of the possessor (9). Unlike cases of part-of bridging, possessor relations are straightforward, as the possessive makes them explicit.

- (9) El primer ministro mostró su preocupación.<sup>8</sup>

(iv) **type=“bridg”** (bridging)

This is a very broad class that encompasses all kinds of metonymic relations –to a greater or lesser extent– holding between two NPs (subset, member, etc.) (10), or between a NP and a VP, implicitly related. Bridging is treated within coreference in the sense that the link between the two discourse entities is established on the basis of the same reference point. A detailed specification of bridging subtypes is addressed in [17].

- (10) a. el cambio de 17 acciones de Alcan...los accionistas
- b. la tropa...uno de los soldados.<sup>9</sup>

<sup>6</sup> (7) a. The Bolivian president and the head of the opposition party...both.  
b. Microsoft...the firm.  
c. the lack of labour in Catalonia...this lack of labour.  
d. a group of adolescents...the team.  
e. “I go on” – said the general director of Seat.

<sup>7</sup> (8) a. If the meteorological situation does not change, something that the INM does not foresee in the short term...  
b. Pujol believes it necessary that the government speeds up the residence permits for immigrants to...This opinion...  
c. [...] With this I do not want to say that we...

<sup>8</sup> (9) The Prime Minister showed his concern.

(v) **type=“pred”** (predicative)

Following van Deemter & Kibble [20], we do not treat nominal predicates (11a) and appositional phrases (11b) as identity coreference. However, given that NPs identifying a discourse entity by its properties can be very relevant for some NLP tasks –such as Entity Detection and Recognition from ACE, and definitional question answering–, we have created the special “predicative link” type for these cases.

- (11) a. Villatoro es el director del diario Avui.
- b. Barnasants, el ciclo de canción de autor...<sup>10</sup>

(vi) **type=“rank”** (ranking)

The ranking link applies to NPs that refer to the numerical order of the elements of a given list. The ANCHOR is either a coordinated or a complex NP of the enumerative kind (12). This link helps “list” questions in question answering, e.g. “Name all the participants in the event.”

- (12) Por este orden, participaron en el acto Javier Krahe, Javier Ruibal y Loquillo.<sup>11</sup>

(vii) **type=“context”** (contextual)

Contextual descriptions are interpreted with respect to the spatial or temporal coordinates (13). Therefore, their ANCHOR is not a <de>, but one of the two universe entities from the <universe> element.

- (13) Este año las cifras están por debajo de la media.<sup>12</sup>

When considering the taxonomy of coreferential link types, we decided to include a second kind of <link> element –different from the coreferential one– so as to fill the semantic type of headless NPs. The <sem type:link> element (with an ANCHOR attribute) is limited to some NPs with adjectives (14a), PPs (14b) or relative clauses as heads.

- (14) a. Tres tipos de vestidos: los blancos, los...
- b. Hubo poca participación, pero la de los españoles...<sup>13</sup>

## 5 Conclusions and further work

In this paper we have presented a foundational step for the annotation of the CESS-ECE corpus with coreference information: the design of the guidelines and general criteria to carry out our project. This annotation scheme allows us to annotate a corpus sample and identify problems and unexpected cases that lead us to extend and refine the markup scheme. Therefore, the scheme here presented is open to new attributes and values. The outcome of this long process is the definition of an

<sup>9</sup> (10) a. the change of 17 shares of Alcan...the shareholders  
b. the troop...one of the soldiers.

<sup>10</sup> (11) a. Villatoro is the director of the Avui newspaper.  
b. Barnasants, the singer-writer song cycle,...

<sup>11</sup> (12) In this order, took part in the event Javier Krahe, Javier Ruibal and Loquillo.

<sup>12</sup> (13) This year the figures are below the mean.

<sup>13</sup> (14) a. Three types of dresses: the white ones, the...  
b. There was little participation, but that of the Spanish...

exhaustive typology of coreferential expressions in Spanish<sup>14</sup> and coreference relations.

In contrast to existing anaphora resolution systems for Spanish, our project covers the whole range of coreferential expressions, thus dealing with proper nouns, full NPs and pronouns all together. The existence of a rich syntactic annotation in the CESS-ECE corpus offers the possibility of doing most of the markable identification automatically, thus reducing the extent of manual annotation, which is well known to be a labour-intensive and time-consuming task.

The choice of the annotation tool is another key point, and we are considering the way in which the existing ones can be adapted to meet our needs. Regarding the annotation strategy, annotators meet periodically to discuss the doubtful cases and thus achieve a level of inter-annotator agreement as high as possible.

This work is the first step in the creation of the largest corpus with complex semantic annotation for Spanish. Once the CESS-ECE corpus is annotated following a scheme linguistically well founded, the goal of our project is twofold. From a computational perspective, the development of an automatic coreference resolution system by applying machine-learning techniques. Besides, the annotated corpus can be used by researchers to train and test automatic coreference resolution methods. From a linguistic point of view, we shall test the hypotheses suggested by Ariel [1] and Gundel et al. [6] on the basis of the annotated data in the CESS-ECE corpus. The linguistic study may lead us to infer generalisations about the expression of coreference in Spanish that can be used as heuristics. According to Botley & McEnery [4], the existing variety of approaches and methodologies to anaphora resolution calls for a synthesis. The combination of the machine-learning algorithms with the heuristics obtained from the linguistic analysis will be a fruitful synthesis, resulting in a hybrid coreference resolution system.

## Acknowledgements

We would like to thank Mihai Surdeanu for his helpful advice and suggestions.

This paper has been supported by the FPU grant (AP2006-00994) from the Spanish Ministry of Education and Science. It is based on work supported by the CESS-ECE (HUM2004-21127), Lang2World (TIN2006-15265-C06-06), and Praxem (HUM2006-27378-E) projects.

## References

- [1] M. Ariel. Referring and accessibility. *Journal of Linguistics*, 24(1):65-87, 1988.
- [2] S. Blackwell. *Implicatures in Discourse: The Case of Spanish NP Anaphora*. John Benjamins, Amsterdam, 2003.
- [3] I. Bosque and V. Demonte (editors), *Gramática descriptiva de la lengua española*, Real Academia Española / Espasa Calpe, Madrid, 1999.

- [4] S. Botley and A. McEnery (editors). *Corpus-based and computational approaches to discourse anaphora*. John Benjamins, Amsterdam, 2000.
- [5] C. Gardent and H. Manuélian. Création d'un corpus annoté pour le traitement des descriptions définies. *Traitement Automatique des Langues (TAL)*, 46(1):115-140, 2005.
- [6] J. K. Gundel, N. Hedberg, and R. Zacharski. Cognitive status and the form of referring expressions in discourse. *Language*, 69(2):274-307, 1993.
- [7] L. Hirschman and N. Chinchor. MUC-7 coreference task definition. In *MUC-7 Proceedings*. Science Applications International Corporation, 1997.
- [8] V. Hoste and W. Daelemans. Learning Dutch coreference resolution. In *Proceedings of the 15<sup>th</sup> Computational Linguistics in the Netherlands Meeting (CLIN 2004)*, 2005.
- [9] S. Lappin and H. J. Leass. An algorithm for pronominal anaphora resolution, *Computational Linguistics*, 20(4):535-561, 1994.
- [10] M. A. Martí, M. Taulé, L. Màrquez, and M. Bertran. CESS-ECE: a multilingual and multilevel annotated corpus, to appear. <http://www.lsi.upc.edu/~mbertran/cess-ece/publications>.
- [11] R. Mitkov. *Anaphora Resolution*. Longman, London, 2002.
- [12] R. Mitkov, R. Evans, C. Orasan, C. Barbu, L. Jones, and V. Sotirova. Coreference and anaphora: developing annotating tools, annotated resources and annotation strategies. In *Proceedings of the 3<sup>rd</sup> Discourse Anaphora and Anaphor Resolution Colloquium (DAARC)*, Lancaster, 49-58, 2000.
- [13] R. Muñoz. *Tratamiento y resolución de las descripciones definidas y su aplicación en sistemas de extracción de información*. Ph.D. Thesis, University of Alicante, Spain, 2001.
- [14] M. Palomar, A. Ferrández, L. Moreno, P. Martínez-Barco, J. Peral, M. Saiz-Noeda, and R. Muñoz. An algorithm for anaphora resolution in Spanish texts. *Computational Linguistics*, 27(4):545-567, 2001.
- [15] M. Poesio. MATE Dialogue Annotation Guidelines – Coreference. Deliverable D2.1, 2000. <http://www.ims.uni-stuttgart.de/projekte/mate/mdag>
- [16] M. Poesio. The MATE/GNOME proposals for anaphoric annotation, revisited. In *Proceedings of the 5<sup>th</sup> SIGdial Workshop on Discourse and Dialogue*, Boston, 154-162, 2004.
- [17] M. Recasens, M. A. Martí, and M. Taulé. Text as scene: discourse deixis and bridging relations. In *Proceedings of the Congreso Anual de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN2007)*, Sevilla, 2007.
- [18] M. Saiz-Noeda. *Influencia y aplicación de papeles sintácticos e información semántica en la resolución de la anáfora pronominal en español*. Ph.D. Thesis, University of Alicante, Spain, 2002.
- [19] A. Tutin, F. Trouilleux, C. Clouzot, E. Gaussier, A. Zaenen, S. Rayot, and G. Antoniadis. Annotating a large corpus with anaphoric links. In *Proceedings of the 3<sup>rd</sup> Discourse Anaphora and Anaphor Resolution Colloquium (DAARC)*, Lancaster, 2000.
- [20] K. van Deemter and R. Kibble. On coreferring: Coreference in MUC and related annotation schemes. *Computational Linguistics*, 26(4):629-637, 2000.
- [21] B. Webber. *A Formal Approach to Discourse Anaphora*. Garland Press, New York, 1979.

<sup>14</sup> This work extends easily to other Iberian Romance languages such as Catalan and Galician.

# An OWL- and XQuery-Based Mechanism for the Retrieval of Linguistic Patterns from XML-Corpora

Georg Rehm

SFB 441 Linguistic Data Structures  
Tübingen University  
Nauklerstrasse 35  
72074 Tübingen, Germany  
*georg.rehm@uni-tuebingen.de*

Richard Eckart

Dept. of English Linguistics  
TU Darmstadt  
Hochschulstrasse 1  
64289 Darmstadt, Germany  
*eckart@linglit.tu-darmstadt.de*

Christian Chiarcos

SFB 632 Information Structure  
Potsdam University  
Karl-Liebknecht-Strasse 24–25  
14476 Potsdam, Germany  
*chiarcos@uni-potsdam.de*

## Abstract

We present an approach for querying collections of heterogeneous linguistic corpora that are annotated on multiple layers using arbitrary XML-based markup languages. An OWL ontology is used to homogenise the conceptually different markup languages so that a common querying framework can be established.

our generic data model. Section 3 sketches the general approach, our system architecture, and the process flows. The main part of this paper, section 4, discusses the web-platform's query interface: first, we illustrate the technical aspects of querying multi-rooted trees. We subsequently introduce an ontology-based approach for homogenising the heterogeneous markup languages. Finally, we sketch the graphical interface and the output and visualisation modules.

## Keywords

Corpora, Corpus analysis, XML, querying, XQuery, OWL, ontologies, multi-rooted trees, annotation, multi-level annotation

## 1 Introduction

Annotated linguistic corpora can be used in several different scenarios: they can be employed in machine learning contexts to serve as training data, they can be used to build language models based on statistical properties, or corpora can serve as a resource in computer-assisted language learning software. In fact, there are so many possible ways in which corpora can be used effectively that their initial purpose has become overshadowed rather quickly. Traditionally, linguists compiled corpora in order to find answers for research questions on the basis of empirical evidence. After a corpus had been compiled using a number of criteria, it could be analysed using statistical methods.

We are concerned with devising a web-based corpus platform for a large collection of more than 60 heterogeneous linguistic corpora. One of the obstacles we are confronted with deals with exploring ways of providing homogeneous means of accessing this very large collection of diverse and complex linguistic resources. The user interface does not only have to generalise over several heterogeneous annotation formats, it has to be intuitively usable for linguists without expertise in XML, querying standards such as XQuery (see, e.g., [15]), or even the original markup languages. In other words, we want to lay a technical foundation for the interoperability and reusability of annotated linguistic corpora. We would like to enable academics who are not interested in the corpus annotation specifics to log onto the platform and to explore as well as to query the available corpora in an efficient and simple way.

Section 2 briefly highlights the most important properties of data formats for linguistic corpora and

## 2 A Homogeneous Data Model

Since the late 1990s, practically all corpus annotation formats have been realised as XML markup languages [11, 13, 20]. They come in two different flavours: traditionally, most corpus markup languages form hierarchies that are expressed by nested XML element trees (e.g., for the representation of syntactic constituents or document structures). In stark contrast to hierarchical data formats are markup languages that anchor a data set to a timeline (primarily used for the transcription of spoken language), see [2]. In timeline-based formats such as Exmaralda [18], the annotator can draw an arc from one anchor to another point on the timeline. However, these structures are not represented by nested XML element-trees, but with the help of attribute-value pairs. At the same time, both approaches usually encode several annotation layers concurrently, for example, information on morphological, syntactic, semantic, and pragmatic structures.

In our project we have to deal with both hierarchical and timeline-based corpora and we have to provide the means for enabling users to query both types of resources in a uniform way. In fact, the original annotation format will be irrelevant to the user, as the user interface and the underlying technology will abstract from any idiosyncrasies and peculiarities of the original data formats. We use an approach that is able to cope with the abovementioned difficulties [8, 19, 22] and that can be compared to the NITE Object Model [4]. We developed a tool that semiautomatically splits hierarchically annotated corpora that typically consist of a single XML document instance, into individual XML files, so that each file represents all the information related to a single annotation layer [21]; this approach guarantees that overlapping structures can be represented straightforwardly. Timeline-based cor-

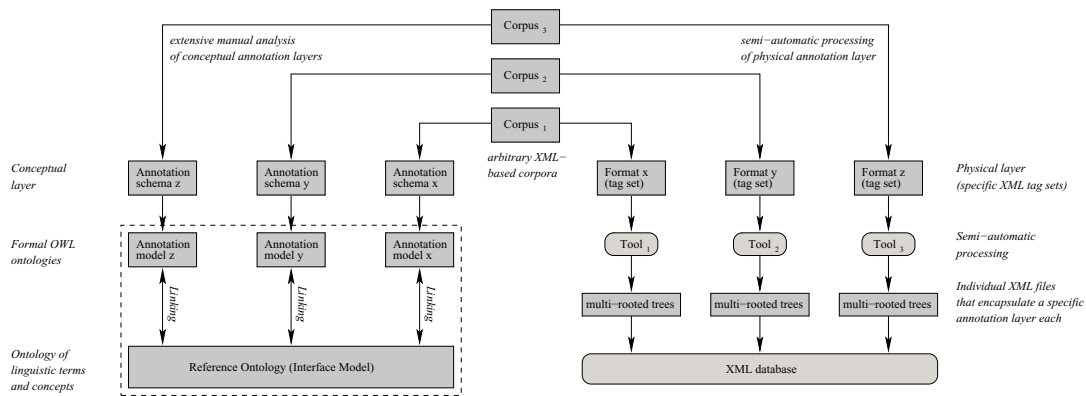


Fig. 1: The two main corpus processing workflows

pora are processed using another tool in order to separate the graph annotations that are also stored in individual XML files [21]. Our approach enables us to represent arbitrary types of XML-annotated corpora as individual files, i.e., individual XML element trees. These multi-rooted trees are represented as regular XML document instances, but, as a single corpus comprises *multiple* files, there is a need to go beyond the functionality offered by typical XML tools in order to enable us to process multiple files, as regular tools work with single files only.

### 3 System Architecture

First, a corpus to be imported into our corpus platform has to be analysed manually (figure 1). Depending on its corresponding markup language, the XML document instance is transformed into multi-rooted trees.

Some corpora can be transformed using simple XSLT stylesheets, while other corpora have to be processed using a custom set of tools: corpora annotated based on the hierarchical model are analysed by a tool that enables us to map XML elements, attributes and textual content onto one or more annotation layers. As soon as this mapping exists, the annotation layers can be exported as XML documents. A second tool can be used to split timeline-based corpora into a set of multi-rooted trees. Finally, these XML files are imported into an XML database (e.g., eXist). A third tool anchors all files to a set of primary data in order to allow query-time coordination between the individual files that represent a single-rooted tree each.

At the same time, the elements and attributes used in the markup languages are analysed and incorporated into an ontology that encapsulates knowledge about linguistic terms and concepts. The ontology is used to generalise over the specific and, at times, idiosyncratic names and labels used in the corpus markup languages and to provide a coherent, unified, and homogeneous perspective on the large set of heterogeneous corpora.

### 4 The Query Interface

There are several constraints for the web-based query interface we are currently developing. For this paper

the two most important issues are the implementation of a mechanism that enables XQuery queries that work on multi-rooted trees (section 4.1) and the integration of the ontology of linguistic annotations into the process of building an XQuery statement (section 4.2). In addition, we want to provide a graphical interface that can be intuitively used by linguists and other interested parties who know neither XML, XQuery, nor the XML-based markup languages used in the original corpora (section 4.3). Figure 2 shows the architecture of the query interface. We modified the XML database eXist so that it is able to cope with directing XQuery queries over multi-rooted trees.

#### 4.1 Querying Multi-Rooted Trees

As each annotation layer is contained in one XML document, a corpus represents a special form of a multi-rooted tree, i.e., a collection of trees that do not share nodes except the leaves containing annotated data. AnnoLab [9] is an XML/XQuery-based corpus query and management framework designed to deal with multi-rooted trees. An abstract data-model for corpus annotation was synthesized from various approaches (e.g., [4], [12], [14]) and consists of four tiers: (i) signal tier (annotated data), (ii) structure tier (annotation structure), (iii) feature tier (annotation features), (iv) location tier (a mapping between signal and structure tiers). XML's data-model itself, however, supports only three of the four tiers: signal (text-nodes), structure (element hierarchy), and feature tier (attributes). Furthermore, it combines the tiers into an ordered tree with non-overlapping leaves, leading to problems regarding projectiveness and overlapping segments. By introducing the location tier as a buffer between signal and structure, these problems can be resolved. In addition, the text-nodes from the XML data-model are replaced by *segments* that serve as placeholders for the signal, thus functioning as stand-off *anchors*. A segment addresses a signal using start and end offsets as well as a *signal identifier*. The rest of the XML data-model remains untouched, so that standard XQuery statements can be used. Assuming that an XML annotation contains the annotated text in document order in its text nodes, the conversion to the AnnoLab format (and back) can be done fully automatically.

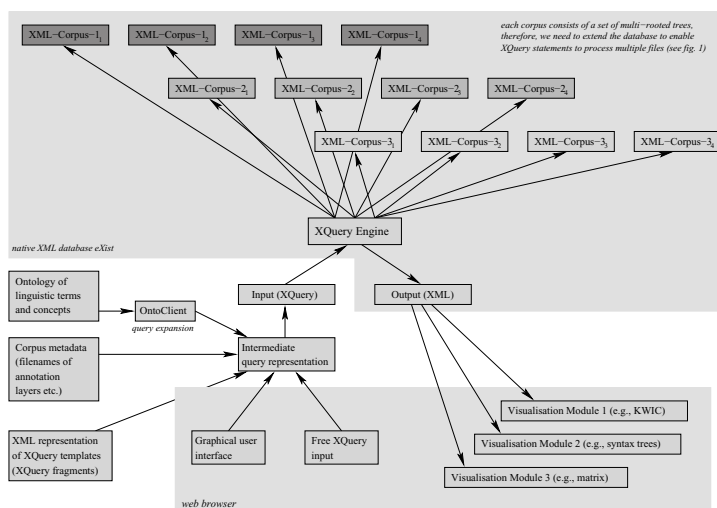


Fig. 2: Architecture of the web-based query interface

#### 4.1.1 XQuery Extensions

To access signals and to perform queries across multiple layers, AnnoLab provides a library of XQuery functions that are loaded into eXist as extensions. These extensions fall into two categories: (i) accessing the signal, (ii) coordinating queries across layers.

**Signal access** – To this category belong functions such as `get-text(N)` and `find-text(N, p)`. The first function takes as an argument a set of elements *N*. It collects all *segments* located under *N* and returns the text they address. The second function takes a set of elements *N* and a pattern *p*. It returns those segments under *N* that address text matching *p*.

**Layer coordination** – The functions in this category perform comparisons and calculations on *segments*. The function `overlapping(X, Y)` illustrates the general principle: it takes two sets of elements *X* and *Y*. These sets are expanded into two *segment* lists *A* = `seg(X)`, *B* = `seg(Y)` that contain all segments under *X* and *Y*. It returns all *a* in *A* that overlap with some *b* in *B*. Analogous functions exist for all 13 temporal relations formalized by Allen [1]. The functions can be used to specify the desired relations between *segments* originating from different annotation layers and, thus, to coordinate different layers.

All extension functions could be implemented in pure XQuery, however, for performance reasons and limitations in eXist, they were implemented in Java.

#### 4.1.2 Query Example

For the following example [9] assume an alignment layer `en_de.align` (see figure 3); its segments refer to two signals `de` (*Deutsch*, German) and `en` (English). Another layer `en.pos` contains `token` elements that have a `pos` feature (part-of-speech data for `en`).

The query (figure 4) yields all verb forms in the English text that are one or two tokens to the left of a determiner along with their translations into German. The query selects all tokens (`$eng`) from the POS layer and all alignments (`$aln`) from the alignment layer. The result set contains those combinations of segments and alignments that fulfill the specified conditions:

- line 4: the English part of the alignment layer has to overlap with a token from the part-of-speech layer,
- line 5: the token from the part-of-speech layer has to be a verb form (`pos` feature starting with V),
- lines 3+6: the first or second following token (`$next`) from the part-of-speech layer has to be a determiner (`pos` feature starting with DT).

This example demonstrates that using AnnoLab's XQuery extensions results in rather complex query statements that require a certain amount of XQuery knowledge. Each query depends on a consistent set of annotation elements, feature names, and feature values.

```

1 <signal id="de">Er schloss das Tor ab</signal>
2 <signal id="en">He locked the gate</signal>
3 <layer id="en_de.align">
4 <alignment>
5   [...]
6 <align>
7   <i role="de">
8     <seg start="3" end="9" sig="de">schloss</seg>
9     <seg start="19" end="20" sig="de">ab</seg>
10  </i>
11 <i role="en">
12   <seg start="3" end="8" sig="en">locked</seg>
13 </i>
14 </align>
15 [...]
16 </layer>

```

Fig. 3: Abbreviated alignment layer and signals

```

1 for $eng in ds:layer("en.pos")//token,
2   $aln in ds:layer("en_de.align")//align
3 let $next := $eng/following::token[position()<2]
4 where seq:overlapping($eng, $aln//i[@role="en"])
5 and starts-with($eng/@pos, "V")
6 and starts-with($next/@pos, "DT")
7 return
8 <t>
9   <eng>{txt:get-text($eng)}</eng>
10  <ger>{txt:get-text($aln//i[@role="de"])}</ger>
11 </t>
12
13 <t>
14   <eng>locked</eng>
15   <ger>schloss ab</ger>
16 </t>

```

Fig. 4: Query aligned signals using pos constraints

## 4.2 Creating XQuery Constraints

In order to provide a consistent approach for documentation and to enable a uniform query interface that applies to different annotation formats, we built an ontology that serves as a terminological reference, represented in OWL DL (see [6, 10] for similar approaches). This *reference model* is based on the EAGLES recommendations for morphosyntax, the general ontology for linguistic description [10], and the SFB632 annotation standard [7]. Currently it includes reference specifications for word classes, morphosyntax [5], and will be extended to other linguistic phenomena.

The *reference model* consists of three parts: a taxonomy of linguistic categories (modelled as OWL classes, e.g., NOUN, COMMONNOUN), a taxonomy of grammatical features (OWL classes, e.g., ACCUSATIVE), and relations (OWL properties, e.g., HASCASE). An *annotation model* is an ontology that represents one specific annotation scheme. We built, among others, formalised annotation models for the SFB632 annotation format [7], TIGER/STTS [17, 3], SUSANNE [16], and for the Uppsala corpus tagset. Annotation models include word classes, grammatical features, and relations. However, this structure is independent from the reference model as it relies on the original annotation documentation only. It can be seen as a formal interpretation of the annotation scheme (see figure 1).

In contrast to the reference model, annotation models include instances. Every instance corresponds to a tag or an annotation value in the original annotation scheme. It is augmented with the properties `HAS TAG` and `HAS TIER`, which provide the exact surface form of the corresponding annotation (e.g., `hasTag(VVZv)`) and the conceptual layer (e.g., `hasTier(pos)`). Instances are characterized by the word class they are assigned to (e.g., `susa:LEXICALVERB` and `susa:FINITEVERB`) and grammatical properties (e.g., `susa:HASPERSON(susa:THIRD)`, and `susa:HASNUMBER(susa:SINGULAR)`).

Annotation models and the reference model are linked by RDF descriptions (`rdfs:subClassOf`, `rdfs:subPropertyOf`): an annotation model acts as one specific instantiation of the reference model. This linking mechanism can also be applied to use definitions from external reference ontologies such as GOLD [10] as an optional *upper model* or *external reference model*. The internal reference model's purpose is to mediate between resource or language-specific annotation models and an external upper model. For the specification of queries, definitions provided by an external reference model may decrease the initial reluctance a user might have to work with the ontology.

### Ontology-Based Corpus Querying

According to the structure of the ontologies, any tag used in an annotation scheme corresponds to an indirect instance of a class in the reference model, which might be subject to further specification by (sub)properties of the reference model. Accordingly, any tag from an annotation model can be retrieved by a description in terms of OWL classes and properties from the reference model. If multiple annotation models are considered, such a description may be expanded

into a disjunction of tags from different tag sets or conceptual layers. OntoClient, a highly configurable query preprocessor implemented in Java, retrieves all individuals which correspond to an ontology-based description and translates them into a disjunction of tags. OntoQueries can be embedded in arbitrary code which remains untouched during query expansion. OntoClient's input as well as the output are specified by formal grammars. In the input, ontology-sensitive sub-queries are marked by curly braces, with the opening parenthesis followed by the CUE, e.g., a variable that describes the element whose attributes and attribute values are defined by the ontological description, the key word `in`, and an expression composed of ontological classes and properties.

```
RESULT := (CUE/@TIER="TAG" (or CUE/@TIER="TAG")*)
```

For every individual retrieved from the expansion of the OntoQuery expression, `TIER` and `TAG` are the values of the corresponding `HAS TIER` and `HAS TAG` properties. CUE is identical to a CUE element in the OntoQuery, thus, it has to be specified by the user.

```
for $eng in ds:layer("en.pos")//token,
2 $aln in ds:layer("en.de.align")//align
let $next := $eng/following::token[position()<2]
4 where seq:overlapping($eng, $aln//i[@role="en"])
and {$eng in Verb}
6 and {$next in Determiner}
return [...]
```

Fig. 5: Incorporating ontology-driven constraints into a query (modified version of the query shown in fig. 4)

Figure 5 shows a modified version of the sample query: for `{ $eng in Verb }` (line 5), OntoClient searches for the concept `VERB` in the reference model. Considering the `SUSANNE` annotation model, `susa:VERB` is retrieved as a subclass of the reference model concept, with sub-classes `susa:LEXICALVERB`, `susa:MODALVERB`, etc., which expand to a total of 44 instances (e.g., `susa:VVZv` is retrieved as an instance of `LEXICALVERB`). The value of `HAS TAG()` specifies the surface form of its tag, i.e., `VVZv`, the value of `HAS TIER()` specifies the conceptual layer, i.e., `pos`. OntoClient produces the following constraints:

```
($eng/@pos = "VVZv" or $eng/@pos = "VV0"
or $eng/@pos = "VV0i" [...])
```

Here, `$eng` is the cue from the original query, `pos` is the value of the property `hasTier`, and `VVZv` is the value of `hasTag`. In the additional annotation models, corresponding tags are listed as well, including multiple conceptual layers and a greater variety of tags.

OntoClient was originally developed as a preprocessor for corpus querying languages such as `CQP`, `TIGERSearch`, and `ANNIS-QL`, which are tailored to the needs of corpus linguists. However, OntoClient can be applied as a more general query preprocessor in order to produce XQuery constraints.

## 4.3 The Graphical Interface

We cannot expect our primary user group (i.e., linguists) to be proficient in XML-related querying languages such as XQuery. Instead, we want to provide an intuitive user interface that generalises as much as possible from the underlying data structures and querying



methods actually used. Our system will make heavy use of Ajax technologies (Asynchronous JavaScript and XML) so that a dynamic, interactive, drag-and-drop-enabled query interface can be provided. As the ontology of linguistic annotations (section 4.2) is a resource for homogenising heterogeneous markup languages, we will be able to provide abstract graphical representations of linguistic concepts (e.g., “noun”, “verb”, “preposition” etc.) that may have a specific set of features; furthermore, we will provide operands so that the linguistic concepts can be glued together by dragging and dropping these graphical representations onto a specific area of the screen, building a query step by step. In addition, users will be able to enter all kinds of annotated linguistic information, e.g., specific text, feature values, syntactic relations etc. (where possible, the information to be presented to the user will be constructed from the ontology). The abovementioned linguistic concepts as well as the operands are associated with XPath and XQuery fragments so that, after a query has been specified using this graphical interface, the individual fragments can be assembled into the final XQuery statement.

We want to provide several output and visualisation modules for query results, e.g., we will visualise queried corpus subsets that contain syntactic trees as trees, realised as SVG graphics, and we plan to represent data that is modelled using a timeline-based approach in a tabular fashion that highlights overlapping structures. One conceptual obstacle concerns the fact that, just like SQL, XQuery queries specify the output part of a query. We plan to introduce a processing layer that represents complex search result datatypes: as soon as each query template is associated with one such search result datatype (e.g., “syntax tree”, “matrix”, “kwic” etc.), we are able to map a specific query template onto a specific output or visualisation module so that the search result datatype specifies which output modules can be used.

For the representation of the query templates we will use an XML-based format in order to store all necessary data in one place: (a) the query template itself (i.e., an XQuery fragment, its associated linguistic concepts, and “free” corresponding variables); (b) the search result datatype; (c) function of the query (its linguistic scope); (d) source of the query; (e) the annotation layer the query refers to (e.g., syntax, information structure etc.); (f) instructions or a general description of the query; (g) one or more sample queries built upon the query template (i.e., example variable assignments that can be modified by the user).

## 5 Concluding Remarks

We presented an approach to querying XML-annotated corpora using standard techniques such as XPath and XQuery. As modern corpora are annotated on several layers, we extended a native XML database so that multi-rooted trees, representing one such annotation layer each, can be queried. One of our goals is to provide an intuitive, modern, flexible, and powerful search interface. As our web-platform has to cope with arbitrary annotation formats, we built an OWL ontology that encapsulates knowledge about the tag

sets used in these annotation schemes. The ontology can be used for query expansion, so that knowledge of the underlying data formats is not required.

## Acknowledgments

Part of the research on AnnoLab was supported by a grant from *Deutsche Forschungsgemeinschaft* (DFG) within the project *Linguistische Profile interdisziplinärer Register* (Darmstadt University of Technology). Part of the research presented was supported by a grant from *Deutsche Forschungsgemeinschaft* (DFG) within the project *Nachhaltigkeit linguistischer Daten*.

## References

- [1] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.
- [2] S. Bird and M. Liberman. A Formal Framework for Linguistic Annotation. *Speech Communication*, 33(1/2):23–60, 2001.
- [3] S. Brants, S. Dipper, P. Eisenberg, S. Hansen-Schirra, E. König, W. Lezius, C. Rohrer, G. Smith, and H. Uszkoreit. TIGER: Linguistic Interpretation of a German Corpus. *Journal of Language and Computation*, 2003.
- [4] J. Carletta, J. Kilgour, T. J. O’Donnell, S. Evert, and H. Voormann. The NITE Object Model Library for Handling Structured Linguistic Annotation on Multimodal Data Sets. In *Proc. of t. EAACL Workshop on Lang. Techn. a. t. Sem. Web*, 2003.
- [5] C. Chiarcos. An Ontology of Linguistic Annotation: Word Classes and Morphology. In *Proc. of DIALOG 2007*, 2007.
- [6] G. A. de Cea, G.-P. Asunción, I. Álvarez de Mon, and A. Pareja-Lora. Ontotag’s linguistic ontologies: Improving semantic web annotations for a better language understanding in machines. In *Proc. of ITCC’04*, pages 124–128, 2004.
- [7] S. Dipper, M. Götze, and S. Skopeteas, editors. *Information Structure in Cross-Linguistic Corpora: Annotation Guidelines for Phonology, Morphology, Syntax, Semantics, and Information Structure*, volume 7 of *ISIS*. 2007.
- [8] S. Dipper, E. Hinrichs, T. Schmidt, A. Wagner, and A. Witt. Sustainability of Linguistic Resources. In E. Hinrichs, N. Ide, M. Palmer, and J. Pustejovsky, editors, *Proc. of LREC 2006 Satellite Workshop Merging and Layering Linguistic Information*, pages 48–54, Genoa, Italy, May 2006.
- [9] R. Eckart and E. Teich. An XML-Based Data Model for Flexible Representation and Query of Linguistically Interpreted Corpora. In G. Rehm, A. Witt, and L. Lemnitzer, editors, *Data Structures for Linguistic Resources and Applications*. Gunter Narr, Tübingen, Germany, 2007.
- [10] S. Farrar and D. T. Langendoen. A Linguistic Ontology for the Semantic Web. *GLOT International*, 3:97–100, 2003.
- [11] N. Ide, P. Bonhomme, and L. Romary. XCES: An XML-based Standard for Linguistic Corpora. In *Proc. of LREC 2000*, pages 825–830, Athens, 2000.
- [12] C. Laprun and J. Fiscus. Recent improvements to the atlas architecture. In *Proc. of HLT 2002*, San Diego, 2002.
- [13] T. Lehmborg and K. Wörner. Annotation Standards. In A. Lüdeling and M. Kytö, editors, *Corpus Linguistics, Handbücher zur Sprach- und Kommunikationswissenschaft (HSK)*. de Gruyter, Berlin, New York, 2007. In press.
- [14] W. Lezius. *Ein Suchwerkzeug für syntaktisch annotierte Textkorpora*. Ph.D. thesis, University of Stuttgart, 2002.
- [15] J. Melton and S. Buxton. *Querying XML*. Morgan Kaufmann, Amsterdam etc., 2006.
- [16] G. Sampson. *English for the Computer. The SUSANNE Corpus and Analytic Scheme*. Clarendon, Oxford, 1995.
- [17] A. Schiller, S. Teufel, and C. Stockert. Guidelines für das Tagging deutscher Textkorpora mit STTS. Technical report, University of Stuttgart, University of Tübingen, 1999.
- [18] T. Schmidt. Time Based Data Models and the Text Encoding Initiative’s Guidelines for Transcription of Speech. *Working Papers in Multilingualism, Series B*, 62, 2005.
- [19] T. Schmidt, C. Chiarcos, T. Lehmborg, G. Rehm, A. Witt, and E. Hinrichs. Avoiding Data Graveyards: From Heterogeneous Data Collected in Multiple Research Projects to Sustainable Linguistic Resources. In *Proc. of E-MELD Workshop on Dig. Lang. Doc.*, East Lansing, Michigan, June 2006.
- [20] C. M. Sperberg-McQueen and L. Burnard, editors. *TEI P4: Guidelines for Electronic Text Encoding and Interchange*. Text Encoding Initiative Consortium, 2002.
- [21] A. Witt, O. Schonefeld, G. Rehm, J. Khoo, and K. Evang. On the Lossless Transformation of Single-File, Multi-Layer Annotations into Multi-Rooted Trees. In B. T. Usdin, editor, *Proc. of Extreme Markup Languages*, Montréal, Canada, 2007.
- [22] K. Wörner, A. Witt, G. Rehm, and S. Dipper. Modelling Linguistic Data Structures. In B. T. Usdin, editor, *Proc. of Extreme Markup Languages*, Montréal, Canada, 2006.

# Identifying relations between scientific objects within predicate structures

Jean Royauté  
LIF-CNRS, Univ. de la Méditerranée  
UMR 6166  
F-13 288 Marseille  
royaute@lif.univ-mrs.fr

Elisabeth Godbert  
LIF-CNRS, Univ. de la Méditerranée  
UMR 6166  
F-13 288 Marseille  
godbert@lif.univ-mrs.fr

Mohamed Mahdi Malik  
LIF-CNRS, Univ. de la Méditerranée  
UMR 6166  
F-13 288 Marseille  
mmahdi@lif.univ-mrs

## Abstract

Predicate structures and their identification present interesting properties for Information Extraction. Few researchers are interested in the relation that links a predicative noun (derived from verbs, from adjectives or not derived) and its arguments in a noun phrase with a predicative head. We show the complexity of these structures as well as paraphrase relations which link them to verbal constructions. We then describe the Link Grammar which we have developed to parse predicate noun phrases in the biology field. This grammar is integrated in the existing Link Parser grammar. Lastly, we show how we have enhanced the heuristics for the selection of the best parsing.

## Keywords

Information Extraction, predicate structure, predicate noun phrase, dependency grammar, link grammar.

## 1. Introduction

Most research in Information Extraction on proteomics focuses on relations between biological entities, for example interactions between genes or proteins, to create databases or knowledge bases. This type of relation is essentially based on verbal predicates and their core arguments, i.e. subject and complement. However, verbs are not the only predicate elements in a sentence. A great number of nouns also play this role. We illustrate this fact of language with four examples using the verb *activate* and its nominalization (*activation*): (a) *the DeltaNp73 promoter is not activated by E2F1...*; (b) *DeltaNp73 activates the hTERT promoter...*; (c) *activation of the hTERT promoter by DeltaNp73alpha...*; (d) *hTERT activation was also observed for ...*

These examples highlight the difficulty in connecting the various forms of the predicate *activate* in its verbal (for example, a-b) or its nominal forms (for example, c-d). These nominal occurrences are very numerous in scientific texts. Our objective is to define a method as well as tools for a robust analysis in order to identify, within the structure of NPs, the prepositional phrases (PPs) which are arguments of the predicate to identify relevant relations between biological entities more extensively.

In this article, we give a definition of predicate structures and highlight the complexity of these structures

by showing their various forms of surface. We then show how the predicate noun phrases (PNP) are integrated in the predicate structures as well as the various properties which make it possible to classify them. Lastly, we present the PNP grammar which we have created and integrated in the standard grammar of the Link Parser ([www.link.cs.cmu.edu/link/](http://www.link.cs.cmu.edu/link/)) [12], as well as heuristics allowing to select the best parsing.

## 2. Verbal or nominal predicate structures

To extract patterns of proteomic interactions, most research is based on verbal structures. These papers concern processing with either a complete parsing [6]; or a partial parsing of shallow-parsing type [1,5] or of pattern-matching type [4]. Alphonse et al. (2004) [1] are interested in nominal-verbal predicate structures, but only from a general point of view, without describing the different nominal patterns which show the complexity of the problem. A specific work on PP attachments on nominalizations [11] in proteomic texts achieves good results with linguistic heuristics, but the system does not produce information on the PP roles (subject, object or adjunct). Concerning nominalizations in other texts than biology, the NOMBANK project [7] automatically, semi-automatically and manually annotates, in corpus (with the Wall Street Journal Corpus of the Penn Treebank), predicate nouns (verbal, adjectival and other) with their argument relations and creates a lexical base of predicate nouns: NOMLEX-PLUS. We have oriented our work in this direction but only with an automatic and weekly-manual acquisition of nominal argument structures from verb structures of a biological and general lexicon: "Specialist Lexicon" [2].

We define as predicate, a word to which one can attach arguments. They are verbs, adjectives and predicate nouns. In a general way, a noun is known as predicative when it presents the same argument relations as a verb [8]. Each argument plays a precise conceptual role: subject, complement or adjunct. In the following, we name '*arguments*' all the core arguments of the predicate and '*adjunct*', other arguments. For example the PNP *milk concentration by ultrafiltration*, connected to the sentence, *ultrafiltration concentrates milk*, is formed with



a predicate head, *concentration*, followed or preceded by its arguments, *ultrafiltration* and *milk*, preceded or not by a preposition. We notice that between the two structures there is conservation of the arguments and that one could add an adjunct (*in the manufacture of cheese*).

We name predicate structure, a structured class of nominal, adjectival and verbal predicates in which the information data to be extracted are aggregated in a <predicate, argument(s)> form.

**Syntactic patterns:** In scientific sublanguages [3], syntactic patterns of a predicate structure generally correspond to various surface forms which convey the same information. More precisely, we call syntactic pattern a grammatical skeleton describing the various core arguments of the predicate in its saturated form and its eventual adjuncts. For a predicate noun, we define tuples of prepositions/conjunctions whose function is to mark, in a stable way, arguments of saturated predicative nominal phrases. For example, in the PNP *activation of the hTERT promoter by DeltaNp73alpha*, we show, with the patterns described in section 3, why the prepositions *by* and *of* respectively mark a subject and a complement noun. This marker capability is reduced in case of deletion of one or more arguments, particularly when these arguments are preceded by the preposition *of*.

### 3. NPs with predicate head

We are interested in nominalization of verbs in PNPs. Each of these PNPs can appear in various surface forms. We distinguish, on one hand, arguments of the associated verbal form with its subject and its essential complements, and, on the other hand, its adjuncts. We do not work for the moment on verbs and nominalizations with that-clauses or infinitive clauses. We show that the structure of the PNP is narrowly correlated with the nature of the verb which corresponds to the nominal predicate head. We use in our work a study of properties of French PNPs [8] and observations in corpus (Web, scientific articles, etc). We also use a syntactic lexicon: "The Specialist Lexicon" [2], ([www.nlm.nih.gov/pubs/factsheets/umlslex.html](http://www.nlm.nih.gov/pubs/factsheets/umlslex.html)) which describes various uses of verbs and their nominalizations. Leroy et al. (2002) [5] worked on such structures by using local parsing and Specialist Lexicon data. However the patterns they have used are based on transitive verbs and are linked to nominal constructions.

We present here seven classes of NPs, among the most significant, which can be linked to verbal constructions. These predicates represent a subset of predicate nouns of English identified in the Specialist Lexicon. In the examples, we use the notation  $N_0 V W$ , where  $N_0$  is the subject of the verb,  $V$  the verb and  $W$  a series, possibly empty, of complements ( $N_1 \dots N_n$ ) linked to this verb. As we will show, it is possible to put semantically in relation

an NP built with a predicate noun and a core sentence, in a unified way.

We have adopted, as first criteria of classification, the role of the preposition *of*: this preposition can mark a direct complement noun phrase (case of predicates linked to verbal constructions with direct complements) or it can mark a subject noun phrase (case of predicates linked to verbal constructions without complement or with prepositional complement). The second criteria is the ability to accept permutable arguments (subject and/or complements) which we note  $N_a$  and  $N_b$ . In Table 1 below, we show the seven classes of these nominalizations, with verbal and nominal constructions, and with examples.

### 4. A grammar of predicate noun phrases

We have written a complete grammar of NPs whose heads are verb nominalizations. We show from which principles and with which data we have completed this work.

#### 4.1 Data

"Specialist Lexicon" [2] is a lexicon which gives interesting syntactic and morphological information on verbs, adjectives and their nominalizations. "Specialist Lexicon" unfortunately does not give any information that can be directly processed on argument structures of nominalizations. However, from this information on verb complements, it is possible to deduce, for each of the verb nominalizations, which tuples of markers (prepositions, conjunctions) can mark an argument (subject or complement) in the PNP. If we examine the verb *to concentrate*, we note that this verb is nominalized with the noun *concentration* and has four verbal uses: (i) *intran*: shows that this verb has an intransitive use, i.e. without complement; (ii) *tran=np*: shows that this verb has a transitive use and that the direct complement is an NP; (iii) *tran=pphr(in,np)*: shows that this verb has a transitive use and that the preposition *in* marks the complement; (iv) *tran=pphr(upon,np)*: shows that this verb has a transitive use and that the preposition *upon* marks the complement.

We have integrated nominalizations of "Specialist Lexicon" in the PNP grammar which we have written and integrated in the standard grammar of the Link Parser. Starting from this lexicon, we highlighted 31 types of verbal uses. For our example with *concentrate*, type 27 can abstract two saturated verbal patterns:  $N_0 V N_1$  and  $N_0 V in/upon N_1$ . The first pattern corresponds to class 1 and parse structures such as  $N^{pred} of N_1 by N_0$ . The second pattern corresponds to class 4 and allows to parse the NP  $N^{pred} of N_0 in/upon N_1$ . Each class, which we have defined in section 3, makes it possible to define a saturated pattern. We have to accept as approximation the fact that certain complements could be optional, which is not always the case for all the verbs.

Table 1. Verbal patterns and nominal patterns associated

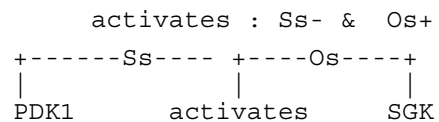
	Verbal patterns	Nominal patterns	Nb
Patterns with <i>of</i> as object marker	-1- N <sub>0</sub> V N <sub>1</sub> <i>IFN-gamma <u>activates</u> protein kinase C delta</i>	N <sup>pred</sup> of N <sub>1</sub> by N <sub>0</sub> <i><u>activation</u> of protein kinase C delta by IFN-gamma</i>	2,351
	-2- N <sub>0</sub> V N <sub>1</sub> Prep N <sub>2</sub> <i>N<sub>0</sub> <u>attributes</u> a protein fragment to a sequence</i>	N <sup>pred</sup> of N <sub>1</sub> Prep N <sub>2</sub> by N <sub>0</sub> <i><u>attribution</u> of a protein fragment to a sequence (by N<sub>0</sub>)</i>	720
Patterns with <i>of</i> as subject marker	-3- N <sub>0</sub> V <i>the femoral head <u>necroses</u></i>	N <sup>pred</sup> of N <sub>0</sub> <i><u>necrosis</u> of the femoral head</i>	200
	-4- N <sub>0</sub> V Prep N <sub>1</sub> <i>tryptophans <u>fluctuate</u> in gramicidin</i>	N <sup>pred</sup> of N <sub>0</sub> Prep N <sub>1</sub> <i><u>fluctuation</u> of tryptophans in gramicidin</i>	348
	-5- N <sub>0</sub> V Prep N <sub>1</sub> Prep N <sub>2</sub> <i>temperature <u>decreases</u> from 200 K to 70 K</i>	N <sup>pred</sup> of N <sub>0</sub> Prep N <sub>1</sub> Prep N <sub>2</sub> <i><u>decrease</u> of temperature from 200 K to 70 K</i>	10
Patterns with permutable arguments	-6- N <sub>a</sub> V with N <sub>b</sub> <i>genes <u>interact</u> with proteins</i> N <sub>a</sub> and N <sub>b</sub> V <i>genes <u>and</u> proteins <u>interact</u></i> N <sub>plur</sub> V <i>the two genes <u>interact</u></i>	N <sup>pred</sup> of N <sub>a</sub> with N <sub>b</sub> <i><u>interaction</u> of genes with proteins</i> N <sup>pred</sup> of/between N <sub>a</sub> and N <sub>b</sub> <i><u>interaction</u> of / between genes and proteins</i> N <sup>pred</sup> of /between N <sub>plur</sub> <i><u>interaction</u> between two genes</i>	64
	-7- N <sub>0</sub> V N <sub>a</sub> Prep N <sub>b</sub> <i>N<sub>0</sub> <u>connects</u> a new sequence with/to a cluster</i> N <sub>0</sub> V N <sub>a</sub> and N <sub>b</sub> <i>N<sub>0</sub> <u>connects</u> a new sequence and a cluster</i> N <sub>0</sub> V N <sub>plur</sub> <i>N<sub>0</sub> <u>connects</u> nodes</i>	N <sup>pred</sup> of N <sub>a</sub> with/to N <sub>b</sub> by N <sub>0</sub> <i><u>connection</u> of a new sequence with/to a cluster (by N<sub>0</sub>)</i> N <sup>pred</sup> of /between N <sub>a</sub> and N <sub>b</sub> by N <sub>0</sub> <i><u>connection</u> of/between a new sequence and a cluster (by N<sub>0</sub>)</i> N <sup>pred</sup> of /between N <sub>plur</sub> by N <sub>0</sub> <i><u>connection</u> of/between nodes (by N<sub>0</sub>)</i>	54

However, this optional status does not pose a problem for predicate noun phrases, insofar as all the arguments (except in some very rare cases) can be deleted. These regroupings have made it possible to define a specific grammar of predicate noun phrases for the Link Parser. This grammar integrates 3,747 nominalizations of verbs, which accounts for 95% of nominalizations of “Specialist Lexicon”.

#### 4.2 PNP grammar and parsing

Link grammars are a variant of dependency grammars. The result of the sentence parsing is a graph in which words are linked two by two with edges labeled by grammatical functions. With Link Parser (LP) [12] which allows this type of parsing, words are linked by a junction between a link X+ (towards the right) and a link X- (towards the left), where X is a tag. We can see below with the verb *activates* that the junction of the Ss+ and Ss- establishes the Ss link between the subject and the verb, and the Os link between the verb and its complement. The parsing of the sentence *PDK1 activates SGK* is represented by the graph below:

Grammar : PDK1 SGK : Ss+ or Os-



A Link Grammar rule is formed with a list of words associated with a more or less complex expression representing all links which belong to these words. The standard Link Parser grammar allows the attachment of a noun to any preposition which introduces an NP. The link which is used is always Mp for the prepositional modifier of a noun. Yet, in PNPs, tuples of prepositions (that can be used with conjunctions) precede and mark the arguments. We have therefore defined new links which we name “argument links”. These links identify the different arguments of a PNP during the sentence parsing.

To integrate these argument links in the grammar, we have created a subclass per syntactic pattern. The grammar consists of 57 subclasses. They are divided according to different uses of verbs and their nominalizations such as described in section 3.

Fig. 1. Sub-class nt1 : parsing of  $N^{pred}$  of  $N_1$  by  $N_0$  and  $N_1 N^{pred}$  by  $N_0$  ; Sub-class ni5 : parsing of  $N^{pred}$  of  $N_0$  in  $N_1$

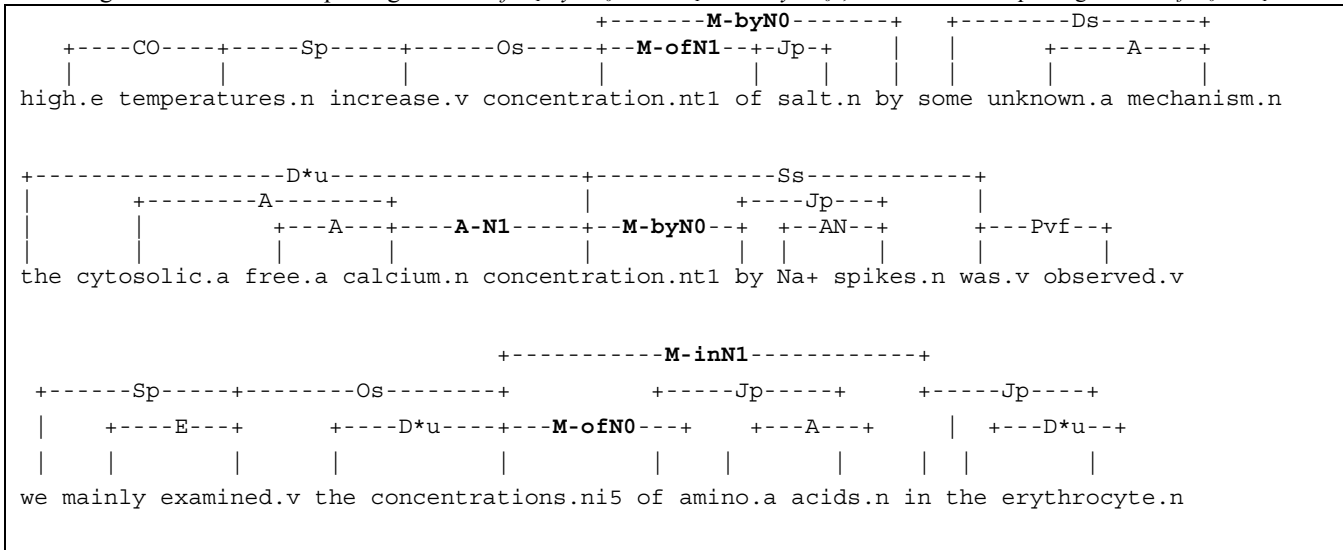
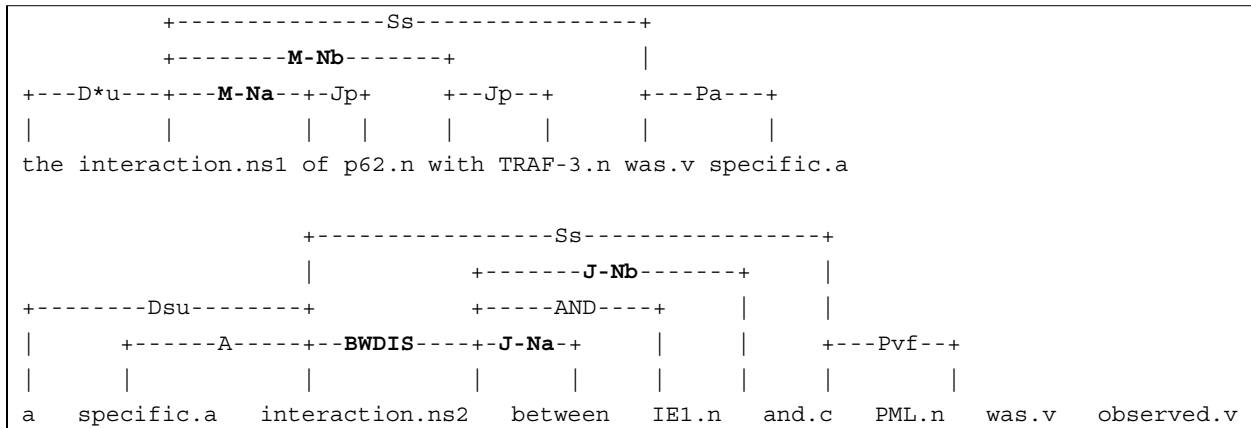


Fig. 2. Sub-class ns1 et ns2 : parsing of  $N^{pred}$  of  $N_a$  with  $N_b$  and  $N^{pred}$  of/between  $N_a$  and  $N_b$



If a word in the Link Parser grammar accepts several syntactic descriptions, it is necessary for it to appear each time in the grammar with a different extension. We have therefore added an extension (see Fig. 1-2) to the nominalizations which characterizes the subclass of every predicative noun and allows to accept the same predicative noun in different constructions.

In the previous section, we saw that *concentration* accepts several nominal constructions. We are going to itemize some significant constructions of this nominalization. The two constructions of Fig.1. concern class 1. The first one is in the form  $N^{pred}$  of  $N_1$  by  $N_0$ . and the second in the form  $N_1 N^{pred}$  by  $N_0$ . In the produced parsing, the M-byN0 link identifies the subject, while the M-ofN1 link marks the direct complement. In the second example, we point out that a particular link (A-N1) was created to process the case in which the object argument introduced by the preposition *of* is in a position of pre-

modifier. Let us point out that there is an ambiguity at this point, because, exceptionally, this position can be occupied by an adjunct. These two examples concern the subclass with the nt1 extension and allows to process the saturated form of this PNP. The last example of this figure shows the prepositional use of *concentration* corresponding to class 4 ( $N^{pred}$  of  $N_0$  Prep  $N_1$ ). As we can see, extensions ni5 characterize the class of verbal use with two arguments whose complement is marked by the preposition *in*. The M-ofN0 link identifies the subject introduced by *of*, while the M-inN1 link marks the complement introduced by the preposition *in*.

Another feature of the grammar is to be able to process very numerous complex forms in the genomics corresponding to class 6, where, as we have seen, subject and complement are permutable. We give above (Fig. 2.) two examples of parsing with *interaction*. In the first use, M-Na and M-Nb links respectively identify co-agents of

class  $N_a V$  with  $N_b$ . The subclass of this first use is marked by the extension ns1. For the second use, corresponding to the subclass ns2, we have created a specific link tied to *between* (BWDIS), allowing to distribute co-agents with two specific links (J-Na and J-Nb) around the conjunction *and*.

### 4.3 Filtering parse results

For each sentence it parses, the Link Parser outputs a set of linkages, corresponding to the set of all possible analyses in accordance with the grammar and ordered with a heuristic of relevance. The longer the sentence, the higher is the number of parsing. For best results [9], it is necessary to modify the standard heuristics of the LP. These results must then be processed by a post-processing, aiming at the extraction of each NP argument. This processing provides a grading of linkages, according to their relevance. Preference is given to linkages satisfying the following criteria: (i) each PNP, whenever it is compatible with its structure, has at least one argument; (ii) the number of argumental links attached with PNPs is maximum; (iii) in the case of a sequence on NPs, the first one is saturated. This heuristics of choice of best parses gives good results. We have conducted pre-evaluation on the first version of the grammar and tests on the last version. For the identification of relevant arguments of NPs, an accuracy of 88.5% has been obtained on a randomized sample of 60 sentences with nominalizations (with one or more arguments), from a corpus of 335 Medline abstracts [10]. Other tests and evaluations are carried out with the last version of the grammar.

## 5. Conclusion

In this paper, the complexity of nominal-verbal predicate structures has been described. A typology has been defined for a significant subset of PNPs with their different patterns and argumental structures. To test the validity of this work, the LP grammar has been modified: specific argumental links have been defined to identify the role of each argument. Using "Specialist Lexicon" data, specific entries have been integrated in the grammar. They modelize the different uses of each nominal structure and the possible ambiguities that can occur when the structure is not saturated. Tests and pre-evaluation show that PNP grammar and the heuristics of choice of best parses give good results.

## 6. Acknowledgements

We are very grateful to Christine Brun and Bernard Jacq of LGPD (Laboratoire de Génétique et Physiologie du Développement, Marseille), for having supplied us with their corpus of Medline abstracts tagged with gene nouns.

## 7. References

- [1] E. Alphonse, S. Aubin, P. Bessières, G. Bisson, T. Hamon, S. Lagarigue, A. Nazarenko, A-P. Manine, C. Nedellec, M. O. A. Vetah, T. Poibeau and D. Weissenbacher. *Event-based information extraction for the biomedical domain: the Caderige project*. International Workshop on Natural language, Processing in Biomedicine and its Applications (JNLPBA), 43-49. 2004.
- [2] A. C. Browne, A. T. McCray and S. Srinivasan. 2000. *The SPECIALIST lexicon technical report*, Lister Hill National Center for Biomedical Communications, National Library of Medicine, USA.
- [3] Z. Harris, M. Gottfried, T. Ryckman, P. Mattick, A. Daladier, T. N. Harris and S. Harris. *The form of Information of Science: Analysis of an immunology sublanguage*. Dordrecht & Boston: Kluwer Academic Publisher. 1989.
- [4] M. Huang, X. Zhu, Y. Hao, D. G. Payan, K. Qu, and M. Li. *Discovering patterns to extract protein-protein interactions from full texts*. Bioinformatics, 20(18):3604-3612. 2004.
- [5] G. Leroy, H. Chen, and J. D. Martinez. *A shallow parser based on closed-class words to capture relations in biomedical text*. Journal of Biomedical Informatics, 36:145-58. 2003.
- [6] D. M. McDonald, H. Chen, H. Su and B. B. Marshall. *Extracting gene pathway relations using a hybrid grammar: the arizonarelation parser*. Bioinformatics, 20(18):3370-3378. 2004.
- [7] A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young and R. Grishman. *The NomBank Project: An Interim Report*, In proceedings of HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation, ACL, 24-31, Boston. 2004.
- [8] R. Pasero, J. Royauté and P. Sabatier. *Sur la syntaxe et la sémantique des groupes nominaux à tête prédicative*. Linguisticae Investigationes, 27(1):83-124. 2004.
- [9] S. Pyysalo, F. Ginter, T. Pahikkala, J. Koivula, J. Boberg, J. Jrvinen, and Tapio Salakoski. *Analysis of Link Grammar on Biomedical Dependency Corpus Targeted at Protein-Protein Interactions*. In Proceedings of the International Workshop on Natural Language Processing in Biomedicine and its Applications, 2004.
- [10] J. Royauté, E. Godbert et M. M. Malik. *Groupes nominaux prédicatifs : Utilisation d'une grammaire de liens pour l'extraction d'information*, In proceedings of TALN-2006, Cahiers du Cental 2.2, 276-286, Vol 1, 2006.
- [11] J. Schuman and S. Bergler. *Postnominal Prepositional Phrase Attachment in Proteomics*, Proceedings of the BioNLP Workshop on Linking Natural Language Processing and Biology at HLT-NAACL 06, ACL 82-89, New York, 2006.
- [12] D. Temperley and D. Sleator. *Parsing English with a Link Grammar*. Carnegie Mellon University Computer Science technical report, CMU-CS-91-196, Carnegie Mellon University, USA. 1991.

# Global Learning of Context Weights from GermaNet

Michael Schiehlen  
Institute for Natural Language Processing  
University of Stuttgart  
Azenbergstr. 12  
D-70174 Stuttgart  
*Michael.Schiehlen@ims.uni-stuttgart.de*

## Abstract

The paper presents a distributional method for automatic acquisition of ranked word similarity lists from text. As usual, the similarity between two words is computed from their common contexts. The relative importance of each context is captured in a weight, and these weights are globally optimized so as to approximate similarity rankings extracted from the German WordNet. We show that this global learning approach performs significantly better than several state-of-the-art approaches that do not rely on learning. The paper also argues that context weights are an important ingredient in taxonomy construction, and that the importance of contexts for ontological categorization can be determined absolutely, i.e. independently of the filler items.

## Keywords

Lexical Acquisition, Word Similarity, WordNet, Machine Learning

## 1 Introduction

Reliable information on lexical semantics is crucial for many NLP applications. To cater for this need, large collections of semantic facts have been created by hand, often organized in the form of a taxonomy<sup>1</sup>. An example is the German WordNet, GermaNet [6], which covers 50,000 common nouns in its current version 5.0. And yet this is not nearly large<sup>2</sup> enough for many NLP applications. So automatic acquisition of at least the most basic semantic relations like hyponymy from large corpora is an absolute necessity. From a corpus perspective, information that is readily available about the relation between two words  $w_1$  and  $w_2$  is the strength of their syntagmatic and paradigmatic association [19]. In the literature, automatic acquisition methods for hyponymy have been based on each of these types of association. The syntagmatic approach, initiated by Hearst [7], infers hyponymy from co-occurrence in specific patterns like *and other* (1).

<sup>1</sup> Our usage of the term “taxonomy” is not intended to restrict the kind of relation encoded in the hierarchy in any way. Rather, it serves to distinguish hierarchies (“taxonomies”) from knowledge bases without a rigid order (“ontologies”).

<sup>2</sup> In the Huge German Corpus, a collection of 200 mio tokens of German newspaper text, only 4.3% of all common nouns (types) are in GermaNet 5.0. If only simplex nouns are counted, coverage rises to 32%.

### (1) whales and other mammals

The paradigmatic approach [8, 18, 14] is based on the hypothesis that semantic similarity between words (and ultimately hyponymy) is correlated with the syntactic contexts they share (distributional similarity), cf. example (2). Section 2 discusses the paradigmatic approach in greater depth.

### (2) Animals: live, breathe, eat, sleep.

Birds: live, breathe, eat, sleep, fly, have wings.

⇒ Birds are animals.

All contexts in (2) are relevant for deriving hyponymy, but other imaginable contexts have no or an adverse effect on classification (e.g. color, form, traits, length of life, or the countability property of animals/birds). Only a small subset of possible contexts is important for taxonomy construction. To formalize the distributional hypothesis, we first make the simplifying assumption that the corpus is comprehensive, i.e. contains all knowledge about the world. On this assumption, a taxonomy is informationally equivalent to the relevance criteria applied in its construction. Formally, a taxonomy  $T$  is a set of words  $W$  (rather: word senses) partially ordered by hyponymy  $>_T$ ; the construction criteria are encoded as a set of admissible contexts  $R_T$ ; and the corpus provides a function  $C$  mapping a word to the set of contexts in which it occurs in the corpus.

$$\forall w_1, w_2 \in W : w_1 >_T w_2 \leftrightarrow C(w_1) \cap R_T \subseteq C(w_2) \quad (3)$$

In reality, corpora are never comprehensive, data are noisy, and predications are used existentially (*some animals do fly*); all these aspects make eq. 1 invalid. Nevertheless, the equivalence in (1) can arguably be maintained if  $R_T$  is interpreted probabilistically, i.e. as a *weighting function* for context features. According to eq. 1, weights should be chosen so as to reproduce the taxonomy  $T$  given the particular contextual features extracted from the corpus. Afterwards, the  $\leftarrow$  direction of eq. 1 permits extending the taxonomy with words that only occur in the corpus. The work presented here focuses on a task that is more modest than full-scale taxonomy extension and also better suited to automatic evaluation: finding for each word the  $M$  words most similar according to the lexical resource, and ranking them as they are implicitly ranked by the resource.

The paper is organized as follows. Section 2 describes the paradigmatic approach to taxonomy con-

struction in more detail, and thus provides the background for the experiment described in Section 3. Section 3 presents the learning technique used in the experiment and compares it empirically with state-of-the-art approaches adapted from the literature. Section 4 discusses related work, while Section 5 concludes.

## 2 Distributional Similarity

Distributional similarity can be defined as the extent to which two words share contexts, but several terms used in that definition require clarification: What exactly is a context? How is the “extent” computed? Are all contexts equally important? Finally, how is distributional similarity used to arrive at a taxonomy? All these questions will be dealt with in this section.

### 2.1 Type of Context

The first question concerns the type of context relevant for lexical semantics. Typically, a context of a word  $w$  is another word, which either stands in a grammatical relation  $r$  to  $w$  [8, 14], or occurs in the same document [20] or the same  $n$ -word window. The first alternative groups together words on the basis of selectional restrictions, the second alternative clusters words that are topically related (assuming that each document is about a particular topic). For “tighter” thesauri like GermaNet, the first option is more appropriate [11], so we adopt this option here. Figure 1 shows the 10 words most similar to “Aal” (*eel*), as extracted with syntactic relations (on top), or with a context window of size 3 (in the middle). The contexts responsible for some of the quirky similarities reveal a range of “non-classificatory” properties of *eels*: They are slippery (like earthworms and mud), they swim (like algae), they bite (like boars), they are green (like spinach), they are cooked (like buckwheat) and killed (like buffaloes), they stand in some relationship to smoking and eating (like termites), they may be electric (like slides) and have something to do with faking (like black marketers).

### 2.2 Measures of Similarity

In contrast to semantic similarity, distributional similarity is directly computable. Generally, two words  $w_1$  and  $w_2$  are the more similar the more contexts they share. Let  $C_i$  be a vector where the  $k$ -th position is set to 1 iff word  $w_i$  occurs in context  $c_k$ . Table 1 lists some common similarity measures; PRF<sub>WW</sub> parametrically combines values modelled after precision (P), recall (R) and F-score. In the description of the measures,  $p$ -norms are used (e.g.  $\|\cdot\|_2$  for Euclidean distance,  $\|\cdot\|_1$  for Manhattan distance). The maxpos function yields the maximum only if its operands are positive numbers, and 0 otherwise.

### 2.3 Context Weights

When context vectors are extended from binary to real vectors, each context–word combination is assigned a weight that captures the importance of the context in

cosine

$$\frac{C_i * C_j}{\|C_i\|_2 \|C_j\|_2}$$

Dice<sub>Lin</sub> [14]

$$\frac{\|\min(C_i, C_j)\|_1 + \|\maxpos(C_i, C_j)\|_1}{\|C_i\|_1 + \|C_j\|_1}$$

Dice<sub>CM</sub> [5]

$$\frac{\|\min(C_i, C_j)\|_1 + \|\min(C_i, C_j)\|_1}{\|C_i\|_1 + \|C_j\|_1}$$

Jaccard<sub>CM</sub> [5]

$$\frac{\|\min(C_i, C_j)\|_1}{\|\max(C_i, C_j)\|_1}$$

PRF<sub>WW</sub> [24]

$$\gamma \frac{2P_{ij}R_{ij}}{P_{ij} + R_{ij}} + (1 - \gamma)(\beta P_{ij} + (1 - \beta)R_{ij})$$

where

$$P_{ij} = \frac{C_i \cdot \|C_j\|_0}{\|C_i\|_1} \text{ and } R_{ij} = P_{ji}$$

**Table 1:** *Similarity Measures*

classifying the word. Among the weighting schemes that have been proposed in the literature, pointwise mutual information [8] and a measure based on the t-test [5] have been shown to yield superior performance [24].

(4) MI

$$C_{ik} = \max(0, \log \frac{P(c_k, w_i)}{P(c_k)P(w_i)})$$

T-Test

$$C_{ik} = \max(0, \frac{P(c_k, w_i) - P(c_k)P(w_i)}{\sqrt{P(c_k)P(w_i)}})$$

## 2.4 Clustering Algorithms

With similarities at hand, hierarchical clustering algorithms can be used to automatically generate taxonomies. In the literature all types of clustering approaches have been explored: agglomerative [1] and divisive approaches [18], yielding hard membership [1] or probabilistic membership information [18].

## 2.5 Evaluation

Evaluation of automatically generated taxonomies is a delicate issue. The literature proposes three approaches, discussed below.

### 2.5.1 Synonymy

One task is to extract synonyms (or near-synonyms), as they occur in a thesaurus [8, 5].

Syn	Regenwurm, Karpfen, Forelle, Fisch, Bachforelle, Hecht, Lachs, Wildschwein, Braunalge, Hering <i>earthworm, carp, trout, fish, brook trout, pike, salmon, wild boar, brown algae, herring</i>
W=3	Termite, Forelle, Flunder, Spinat, Tintenfisch, Buchweizen, Schieber, Matsch, Fisch, Büffel <i>termite, trout, flounder, spinach, octopus, buckwheat, slide/black marketeer, mud, fish, buffalo</i>
WNet	Hering, Lachs, Salm, Karpfen, Forelle, Weller, Wels, Muräne, Hecht, Seenadel <i>herring, salmon, salmon, carp, trout, wels, wels, moray, pike, pipefish</i>

**Fig. 1:** 10 most similar words to Aal (eel) according to syntactic context (above), a window of size 3 (middle), GermaNet 5.0 (below)

### 2.5.2 Semantic Relatedness

The semantic relatedness information automatically extracted can also be compared with semantic relatedness implicit in a taxonomic resource like WordNet [14, 24]. Making WordNet similarity explicit requires some measure. Similarity measures that also take into account the frequency of word senses, e.g. [9], are reported to perform best. For German, however, there is no corpus annotated with WordNet senses like SemCor [17]. So we make use of the next-best option, Leacock and Chodorow’s measure [12], which is the normalized length of the paths to the lowest common concept node ( $D$  is the maximal path length).

$$S_{ij} = -\log \frac{\min_k |path_{ik}| + |path_{jk}|}{2D}$$

Next, the similarities in the resource (the *reference* solution) and those generated automatically (the *system* solution) need to be compared. Here averaged cosine is a widely used measure [14, 24].

$$(5) \quad \frac{1}{N} \sum_{i=1}^N \frac{S_i^{sys} \cdot S_i^{ref}}{\|S_i^{sys}\| \|S_i^{ref}\|}$$

Weeds and Weir [24] use ranks  $R$  in formula (5) instead of similarity scores  $S$ ; comparing ranks is better than directly comparing similarity values if it cannot be guaranteed that the two similarity measures  $S^{sys}$  and  $S^{ref}$  use the same scale. The following definition ensures that the values of ranks range from  $M$  (most similar) to 0 (least similar).

$$R_{ij} = \max(0, M - |\{k : S_{ik} > S_{ij}\}|)$$

### 2.5.3 Hyponymy

Finally, some approaches evaluate directly against the hyponymy relation, either by asking native speakers [1] or by comparison with WordNet [2]. Cimiano and Staab [3] make use of a more elaborate measure, viz. taxonomic overlap [15]. The idea behind this measure is that the similarity of two taxonomies  $T_1$  and  $T_2$  can be quantized with the number of hypernym/hyponym relations they have in common. The definition rests on the notion of semantic cotopy of word senses  $w$  (i.e. the set of all hyponyms and hypernyms of  $w$ ):

$$SC(w, T) := \{w' \in W_T : w' \leq w \vee w' \geq w\}$$

Semantic cotopies enter into the computation of taxonomic overlap as shown in (6), the average percentage of common hyponyms/hypernyms for each word.

(6) taxonomic overlap  $TO(T_1, T_2)$

$$\frac{1}{|W_1|} \sum_{w \in W_1} \frac{|SC(w, T_1) \cap SC(w', T_2)|}{|SC(w, T_1) \cup SC(w', T_2)|}$$

where

$$w' = \begin{cases} w & \text{if } w \text{ in } T_2 \\ \arg \max_{w''} \frac{|SC(w, T_1) \cap SC(w'', T_2)|}{|SC(w, T_1) \cup SC(w'', T_2)|} & \end{cases}$$

### 2.5.4 An Amalgamation of Evaluation Measures

All of the evaluation measures listed so far have their pros and cons. Thus, we amalgamated them into a new measure. This measure is based on Leacock and Chodorow’s [12] measure in so far as ranks are computed from the lengths of shortest possible paths in the resource. It uses the synonymy approach by ensuring that synonyms are assigned optimal ranks (they are linked by the shortest path). It integrates semantic cotopy by stipulating that non-synonymous hypernyms and hyponyms are assigned the second-best rank. We used a maximum rank value ( $M$ ) of 50. If several equally good lemmas occur on the rank list,  $M$  may be higher.

## 3 Experiment

### 3.1 Setup

In our experiments, we use a broad-coverage and high-speed parser [21, 22] that is able to extract dependency triples, i.e. pairs of words with a label for the grammatical relation holding between them. The parser is somewhat comparable to Minipar, but uses finite-state technology, which makes it an order of magnitude faster. It also produces more fine-grained results. The text basis in the experiments is a corpus of German newswire (HGC, Huge German Corpus), consisting of 200 million tokens. For easy access, parse results are coded with CQP, a tool of the IMS corpus workbench [10]. From this resource, we extracted all dependency tuples with a minimum frequency of 5.

The parse results were then linked to GermaNet 5.0. For efficiency, we only inspected the subsection “natürliches Objekt” (natural object), which still covers 18,883 senses distributed over 14,000 lemmas. First of all, GermaNet nouns were mapped to lemmas of the parse output, which reduced the numbers to 15,900

	t-test	mutual info
cosine	21.33%	21.88%
Dice <sub>Lin</sub>	10.30%	9.27%
Jaccard <sub>CM</sub>	21.35%	18.89%
OPAL	36.23%	

**Table 2: Results**

word senses and 11,647 lemmas. We also only considered lemmas that occurred in at least 3 different contexts (7,953 lemmas) and contexts that were common to at least 5 different lemmas (92,768 contexts).

We only evaluated for semantic relatedness (see Section 2.5.4 for details), using roughly 1/20th of all lemmas (the first 350 lemmas) as unseen test set. All results reported were obtained from this test set. Apart from the global learning approach (see Section 3.2), we tested the approaches that performed best in earlier evaluations [5, 24], viz. Jaccard with t-test and Jaccard with mutual information, along with some other measures. Table 2 states all results obtained on the test set. Generally, t-test performed better than mutual information (cf. Table 2).

### 3.2 Approach

In addition to state-of-the-art approaches, we explored a learning algorithm, more specifically an online passive-aggressive algorithm [4]. Online passive-aggressive algorithms have achieved good results in other areas of Natural Language Processing [16]. In online learning, the algorithm iterates over the training set, adjusting the weights  $\mathbf{w}$  of the features after inspecting each training instance.

The evaluation measure (5) makes it clear, that the basic blocks for learning should be individual lemmas  $i$ . For each lemma, our evaluation measure returns a ranking for all other lemmas  $j$  in the taxonomy. So a training instance  $\mathbf{x}$  is the pair of  $i$  and the set  $J$  of  $j$ 's. problem. We aim to learn a ranking via the context features that relate lemma  $i$  with each of the other lemmas  $j$ . Each such context feature  $c_k$  is associated with two weights: one weight  $w_k^c$  for  $c_k$ 's occurrence as link feature (i.e. common to  $i$  and  $j$ ) and one weight  $w_k^d$  for  $c_k$ 's occurrence as discriminating feature (i.e. occurring with  $i$  or  $j$  but not both). With the help of these weights, the similarity between  $i$  and all  $j$ 's is computed as follows:

$$S_{ij}^{sys} = \sum_{c_k \in C_i \cap C_j} w_k^c + \sum_{c_k \in C_i \setminus C_j \cup C_j \setminus C_i} w_k^d$$

The similarity function is then converted into a ranking:

$$R_{ij}^{sys} = \max(0, M - |\{k : S_{ik}^{sys} > S_{ij}^{sys}\}|)$$

One problem in comparing system ranking with reference ranking is that the reference ranking defines only a partial order. We complete this order so as to *optimize* agreement between reference and system ranking. Given a complete ranking  $R^{ref}$ , the loss incurred

at each training instance is defined as the divergence between the two rankings, so that it is computed via the cosine as follows.

$$L(R^{sys}, R^{ref}) = 1 - \frac{R_i^{sys} \cdot R_i^{ref}}{\|R_i^{sys}\| \|R_i^{ref}\|}$$

The online learning algorithm is based on the assumption that feature weights  $\mathbf{w}$  should be changed as little as possible (*passive*) provided that the system solution is incorrect by a margin at least as large as the loss  $L$  incurred by that solution (*aggressive*). In the following formula  $\Phi$  stands for feature vectors,  $\mathbf{w}$  for the weight vector,  $\mathbf{x}$  for the training instance,  $R^{ref}$  for the reference solution, and  $R^{sys}$  for the system solution, i.e. the single best solution derived with current weights.  $\mathbf{w} \cdot \Phi(\mathbf{x}, R^{ref})$  is the score assigned to the reference solution, and  $\mathbf{w} \cdot \Phi(\mathbf{x}, R^{sys})$  the overall best score, which leads to the system solution.

$$\mathbf{w}^{(t+1)} = \arg \min_{\mathbf{w} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{w} - \mathbf{w}^{(t)}\|^2$$

under the constraint that

$$\mathbf{w} \cdot \Phi(\mathbf{x}, R^{ref}) - \mathbf{w} \cdot \Phi(\mathbf{x}, R^{sys}) \geq \sqrt{L(R^{sys}, R^{ref})}$$

Having a closed form solution [4], the problem can be solved efficiently.

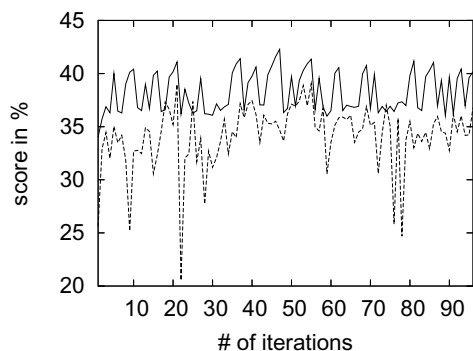
### 3.3 Results

Table 2 lists the results that the different approaches achieved on the test set under the evaluation measure presented in section 2.5.4. Online passive-aggressive learning (OPAL) clearly outperforms Jaccard with t-test and Jaccard with mutual information, and indeed all other state-of-the-art approaches discussed. We take these results to indicate the importance of context weights for the performance of the overall system. In contrast to the other approaches, OPAL uses context weights that are independent of the word types compared. Thus, its good performance corroborates the hypothesis propounded in the introduction that some contexts are inherently more important for ontological categorization than others. A cursory inspection of contexts with high linking weight shows many appositive relations but also modifiers like *uniformiert* (*uniformed*) or *pflanzlich* (*herbal*). Very frequent contexts tend to have low linking weights. When we look at the convergence properties of the algorithm, i.e. the relation between iterations over the training set and results obtained (cf. Table 3), we register a smooth gradual increase. Training performance (the higher curve) and test performance (the lower curve) are well aligned.

## 4 Related Work

Pereira et al. [18] present a framework of distributional clustering, that is comparable in some respects to the one presented here. They use conditional distributions  $P(c|w)$  for context vectors, and also associate





**Table 3:** Results for Iterations of OPAL

such distributions with nodes  $n$  internal to the hierarchy. Internal distributions are computed from leaf distributions via probabilistic class membership.

$$P(c|n) = \sum_w P(c|w)P(w|n)$$

There are, however, also major differences between their approach and ours. Their approach is unsupervised, while we make use of GermaNet. They use generative models, while we take a discriminative approach. Discriminative clustering, albeit on a different task, is also discussed by Li and Roth [13].

## 5 Conclusion

In this paper, we presented an approach to learn context weights from WordNet that brings substantial improvements over older unsupervised approaches. The approach not only learns context weights from the corpus and WordNet, but also globally optimizes for a rank-based evaluation measure. Future work will look at extending the approach to full-scale taxonomy construction. A probabilistic approach for this task has already been proposed by Snow et al. [23]. Their approach does not make use of iterative learning, however, and deals with taxonomy extension rather than taxonomy construction. It will be interesting to see in what way the evaluation measures described in section 2.5.3 can be applied in global learning.

Thanks are due to Kristina Spranger for discussion and proof-reading. I would also like to thank the audience at the GermaNet workshop 2007 in Tübingen for helpful hints and discussion.

## References

- [1] S. A. Caraballo. Automatic Construction of a Hypernym-Labeled Noun Hierarchy from Text. In *ACL'99*, pages 120–126, 1999.
- [2] P. Cimiano, A. Pivk, L. Schmidt-Thieme, and S. Staab. Learning Taxonomic Relations from Heterogeneous Sources of Evidence. In P. Buitelaar, P. Cimiano, and B. Magnini, editors, *Ontology Learning from Text: Methods, Evaluation and Applications*, volume 123 of *Frontiers in Artificial Intelligence*, pages 59–73. IOS Press, jul 2005.
- [3] P. Cimiano and S. Staab. Learning Concept Hierarchies from Text with a Guided Agglomerative Clustering Algorithm. In C. Biemann and G. Paas, editors, *Proceedings of the ICML 2005 Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods*, Bonn, Germany, Aug. 2005.
- [4] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online Passive-Aggressive Algorithms. *Journal of Machine Learning*, 7:551–585, 2006.
- [5] J. R. Curran and M. Moens. Improvements in Automatic Thesaurus Extraction. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition*, pages 59–66, Philadelphia, PA, 2002.
- [6] B. Hamp and H. Feldweg. GermaNet - a Lexical-Semantic Net for German. In *Proceedings of the ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, Madrid, 1997.
- [7] M. A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. In *COLING '92*, pages 539–545, 1992.
- [8] D. Hindle. Noun Classification From Predicate-Argument Structures. In *COLING '90*, pages 268–275, 1990.
- [9] J. J. Jiang and D. W. Conrath. Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, Taiwan, 1997.
- [10] H. Kermes and S. Evert. YAC – A Recursive Chunker for Unrestricted German Text. In *LREC '02*, pages 1805–1812, 2002.
- [11] A. Kilgarriff and C. Yallop. What's in a thesaurus. In *LREC '00*, pages 1371–1379, 2000.
- [12] C. Leacock and M. Chodorow. Combining local context and WordNet similarity for word sense identification. In C. Fellbaum, editor, *WordNet: An electronic lexical database*, pages 251–263. MIT Press, 1998.
- [13] X. Li and D. Roth. Discriminative Training of Clustering Functions: Theory and Experiments with Entity Identification. In *CONLL '05*, pages 64–71, 2005.
- [14] D. Lin. Automatic Retrieval and Clustering of Similar Words. In *COLING-ACL '98*, pages 768–773, Montreal, Canada, 1998.
- [15] A. Maedche and S. Staab. Measuring Similarity between Ontologies. In *Proceedings of the European Conference on Knowledge Acquisition and Management (EKAW)*, pages 251–263. Springer, 2002.
- [16] R. McDonald, K. Crammer, and F. Pereira. Online Large-Margin Training of Dependency Parsers. In *ACL'05*, 2005.
- [17] G. A. Miller, M. Chodorow, S. Landes, C. Leacock, and R. G. Thomas. Using a Semantic Concordance for Sense Identification. In *Proceedings of the ARPA Human Language Technology Workshop*, Plainsboro, NJ, 1994.
- [18] F. Pereira, N. Tishby, and L. Lee. Distributional Clustering of English Words. In *ACL'93*, pages 183–190, 1993.
- [19] R. Rapp. The Computation of Word Associations: Comparing Syntagmatic and Paradigmatic Approaches. In *COLING '02*, Taipei, Taiwan, 2002.
- [20] M. Sanderson and B. Croft. Deriving Concept Hierarchies from Text. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 206–213, 1999.
- [21] M. Schiehlen. A Cascaded Finite-State Parser for German. In *Proceedings of the Research Note Sessions of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, pages 163–166, Budapest, Hungary, Aug. 2003.
- [22] M. Schiehlen. Combining Deep and Shallow Approaches in Parsing German. In *ACL'03*, pages 112–119, Sapporo, Japan, 2003.
- [23] R. Snow, D. Jurafsky, and A. Y. Ng. Semantic Taxonomy Induction from Heterogenous Evidence. In *COLING '06*, Sydney, Australia, 2006.
- [24] J. Weeds and D. J. Weir. Co-occurrence Retrieval: A Flexible Framework for Lexical Distributional Similarity. *Computational Linguistics*, 31(4), 2005.

# Transferring Egyptian Colloquial Dialect into Modern Standard Arabic

Khaled Shaalan  
The Institute of Informatics  
The British University in Dubai,  
PO Box 502216, Dubai, UAE  
khaled.shaalan@buid.ac.ae

Hitham M. Abo Bakr  
Computer & System Dept  
Zagazig University  
hithamab@yahoo.com

Ibrahim Ziedan  
Computer & System Dept  
Zagazig University  
i.ziedan@yahoo.com

## Abstract

Arabic is rooted in the Classical or Qur'anical Arabic, but over the centuries, the language has developed to what is now accepted as Modern Standard Arabic (MSA). Arab colloquial dialects are generally only spoken languages, but recently the rate of colloquial written text increases dramatically as a medium of expressing ideas especially across the WWW, usually in the form of blogs and partially colloquial articles. Most of these written colloquial has been in the Egyptian colloquial dialect, which is considered the most widely dialect understood and used throughout the Arab world. We are able to reuse MSA processing tools with colloquial Arabic by transferring colloquial Arabic words into their corresponding MSA words. The advantages of this lexical transfer are to facilitate the communication with colloquial Arabic speakers and restoring it to the standard language in use nowadays. This paper addresses the transfer techniques between colloquial Arabic and MSA, which have not yet been closely studied before. In particular, we present a rule-based lexical transfer approach for converting Egyptian colloquial words into their corresponding MSA words. This process involves morphological analysis and lexical acquisition of colloquial words.

## Keywords

Colloquial Arabic dialects processing, and transferring Egyptian Arabic into Modern Standard Arabic.

## 1. Introduction

Colloquial Arabic is a collective term for the spoken languages or dialects of people throughout the Arab world. Although it is descended from Arabic, it is considered a separate language. Speakers of some of these dialects are unable to understand speakers of other Arabic dialects. Recently, the rate of colloquial written text increases dramatically. Modern Standard Arabic (MSA) is the official Arabic language taught and understood all over the Arabic world. MSA has many challenges concerning the development of morphological and syntactic processing tools.

These significant tools will become more complicated if they include in parallel the handling of Colloquial Arabic problems.

Today Egyptian Arabic, also known as *Masri*, is the dialect spoken in Egypt by more than 70 million people. It is understood across the Middle East due to the predominance of Egyptian media, making it one of the most widely spoken and most widely studied varieties of Arabic. For this reason we selected Egyptian Arabic to prove the capability of our approach in transferring a Colloquial Arabic dialect into MSA.

In literature, there are few researches that relate colloquial Arabic to MSA [6, 7]. These researches have focused on the spoken colloquial features of Arabic while our research focuses on written colloquial Arabic. Our approach is to develop transfer techniques that are able to perform the lexical mapping between written colloquial Arabic and MSA. The resultant front-end module will make it easy to incorporate colloquial Arabic into existing MSA tools. This will widen the coverage of current Arabic natural language processing applications to include colloquial languages or dialects of Arabic. Our proposed research builds the linguistic transformation resources between colloquial Arabic and MSA using the rule-based method. The data collection process will gather colloquial words from Arabic websites across the Web.

The paper is structured as follows. Section 2, discusses the challenges in handling written colloquial Arabic. In Section 3, we propose solutions for these problems. Section 4 gives background information. Section 5 concentrates on handling the deviation of Egyptian Arabic from MSA. Section 6 gives some concluding remarks.

## 2. Challenges in Handling Written Colloquial Arabic with Regard to MSA

Language processing of colloquial Arabic is a difficult task. The reasons of this difficulty come from several sources:

1- *Arabic Script*. There two ways that colloquial Arabic speaker use in their writing of colloquial words. One way is to Romanize the colloquial word (written using the Latin alphabet) and hence has to be transliterated from Arabic to English. Informal chatting across chat rooms or exchanged SMS messages in the Arab community usually done using Romanized letters. The other way is to write Arabic words using lexographic Arabic letters. Colloquial normal Arabic letters.

2- *Deviation from MSA*. There are five main deviations from MSA:

- Distortion of verbs (e.g.  
بليته من بلته - صرَّيَّبه من صرَّيَّه - حاكتب من ساكتب -  
(ماتأعد من أما تقعد).
- Distortion of nouns. (e.g.  
الخير من الخير - ده من هذا - جمهور خايف من خائف -  
من جمهور - مين من من - فين من أين).
- Distortion of Pronouns and letters meanings.  
(e.g.  
(عصايتي من عصاي - احنا من نحن - هو من هو).
- Distortion of the structure of the word form  
(e.g.  
اتوب من تتأب - اتاوى من اوى - بغبعان من بيغاء -  
(تلات شهور من ثلاثة شهور).
- Replace the characters and movements.  
(e.g.  
تعبان من تعبان - نوم من نوم - سقب من نقب - شبط من  
(شبت "اي تعلق").

3- *Lack of syntactic rules*. There are no identified grammar rules for colloquial dialects.

4- *Lexical expansion rate*. As colloquial Arabic is more popular than MSA, it is very often to observe much more newly added expressions/words as apposed to MSA.

## 3. The Proposed Approach

For the problems introduced in the previous section, we give suggestions for each of which.

To solve problem of writing colloquial Arabic in Latin alphabet, we propose the following process:

- Detect Romanized words in the input and transliterate these words into Arabic lexographic letters,

- Normalize the words such as removing repeated characters that is usually used to informally indicate emotions, and
- Lookup the Colloquial-to-MSA lexicon for the closest colloquial word match and return the corresponding colloquial entry.

As an example, the phrase "Meeesh 3aweez 7agh" will be converted to "ميش عاوز حاجة" (I do not need anything).

To solve the problem of the deviation of Egyptian Arabic from MSA, the major contribution of this research, we used an existing mature MSA lexicon (Buckwalter lexicon version2, [3]<sup>1</sup>) to build the Colloquial-to-MSA lexicon such that both their entries coexist in one lexicon. We followed the same morphological analysis approach of this tool in analyzing the colloquial Arabic word. A rule-based lexical transfer approach is use to transform the analyzed colloquial Arabic word into MSA word(s).

To solve the problem of the lack of identified colloquial syntactic rules, we suggest solving this problem with empirical corpus-based techniques from Example Based Machine Translation (EBMT) [8, 9]. This has incurred building a parallel corpus of both the colloquial and MSA text. The development of such corpus is relatively new and will be published elsewhere.

To solve the problem of acquiring new colloquial words/expressions, we propose a process based on EBMT techniques that maintains the lexicon and keeps it up-to-date. This sophisticated process will gather Arabic text from the Web. The text is analyzed in order to recognize the unknown lexical items. An Arabic specialist has to take a decision of whether or not to add the unknown lexical item to the lexicon.

## 4. The Buckwalter Morphological Analyzer

We build our system on top of Buckwalter Arabic Morphological Analyzer Version 2.0 [3]. His morphological analysis depends on a dictionary of prefixes, a dictionary of suffixes, a stem dictionary, and three checking tables for testing the validity of a word analysis. The

<sup>1</sup> See the description of the Buckwalter's Arabic morphological analyzer  
<http://www.qamus.org/morphology.htm>

morphological analyzer tries to breakdown the input Arabic word into three elements: prefix, stem, and suffix. If all the three word elements are found in their respective lexicons, then their respective morphological categories are used to determine whether they are compatible. If all the morphological category pairs are compatible, then the morphological analysis is valid.

Each entry in the three lexicon files consists of four tab-delimited fields:

1. the entry (prefix, stem, or suffix) without short vowels and diacritics,
2. the entry (prefix, stem, or suffix) with short vowels and diacritics,
3. its morphological category (used for the compatibility between prefixes, stems, and suffixes), and
4. its English gloss(es), including selective POS data within XML tags `<pos>...</pos>`

Only fields 1 and 3 are required for morphological analysis. Fields 2 and 4 provide additional information once the morphology analysis is succeeded in producing the analyzed word(s). Arabic script data in the lexicons is provided in the Buckwalter transliteration scheme.

The following is a description of the three lexicon files:

- *dictPrefixes* contains all Arabic prefixes and their concatenations. Sample entry:  
w wa Pref-Wa `<pos>wa/CONJ</pos>`
- *dictSuffixes* contains all Arabic suffixes and their concatenations. Sample entry:  
p ap NSuff-ap [fem.sg]  
`<pos>ap/NSUFF_FEM_SG</pos>`
- *dictStems* contains all Arabic stems. Sample entries:  
ktb katab PV write  
ktb kotub IV write

There are three compatibility tables; each of the three compatibility tables lists pairs of compatible morphological categories:

- Compatibility table *tableAB* lists compatible Prefix and Stem morphological categories, such as:  
NPref-Al N  
NPref-Al N-ap
- Compatibility table *tableAC* lists compatible Prefix and Suffix morphological categories, such as:  
NPref-Al Suff-0  
NPref-Al NSuff-u

- Compatibility table *tableBC* lists compatible Stem and Suffix morphological categories, such as:  
PV PVSuff-a

## 5. The Proposed Solution of Transferring Colloquial Arabic Dialect to MSA

Our proposed transfer techniques are based on previous studies of the transformations between the MSA and colloquial Arabic [1, 2, 4, 5]. We used the indicated variations to acquire the lexical transfer rules that can be used to derive the MSA word from a corresponding colloquial Arabic word. Additional rules will be acquired and judged by an Arabic specialist during the lexical acquisition process. These rules are used to analyze the input colloquial word and produce the target MSA word(s).

### 5.1 Examples of Egyptian Colloquial Word to MSA Transformations

The colloquial Arabic word is normally derived from a well-formed MSA word. This process can be traced back to the distortion (transformation) made to the MSA word that has changed it to a colloquial Arabic word form. The analysis of the relationship between well-formed MSA Arabic words and colloquial words has been discussed by many linguists [1, 2, 4, 5]. Table 1 shows distortion examples and how to transfer them into MSA words.

The transfer between Egyptian Arabic dialect and MSA is one-to-many transformation. This means some Egyptian Arabic words can be transferred in one or more steps through lexicon lookup as the mapping involves more than one morpheme. For example, the Egyptian word *ازيك* "How are you?" is transformed to two MSA words "كيف حالك?". Other examples are:

- ماورد (Ma2 ward) : ماء ورد
- كلشينكان (Koleshenkan) : كل شيء كان
- أجرنك (2agranak) : لا جرم انك وتقال في العامية
- أجرنك شاطر أي لا جرم انك شاطر
- أشمعنا (2eshMe3na) : ايش المعني
- إكمه (2kmeno) : كما انه
- بسملة (Besmellah) : بسم الله

In colloquial language processing, a word might be added to the lexicon which does not have a

corresponding word in the formal language. This is also the case in Egyptian colloquial (e.g. the word "بقى" can be used to indicate either an exclamation or an interrogation such that both the symbols "?!") appear together at the end of the sentence. This is best explained by the following examples:

- "بقى أنت تعمل كده؟" which is transferred to MSA as "أنت تفعل هذا؟!" (Do you do this?), and
- "ازيك ببقى؟" which is transferred to MSA as "كيف حالك؟!" (How are you?).

**Table 1.** Examples that illustrate the relation between MSA words and Egyptian Arabic words

MSA	EGW	Distortion Type	Handling method
يد	ايد	Replace of vowels "فتتح الاول والعامية تكسره"	Add new stem and assign the same rules as (colloquial Arabic word (CAW)
وأنا نحن	ونا احنا	Distortion in Pronouns and letters meaning "التحريف في الضمانر"	Add new stem and assign the same rules as CAW
البارحة أمس	امبارح	Distortion in Pronouns and letters meaning "التحريف في حروف المعاني ال - ام"	Add new stem and assign the same rules as CAW "البارحة"
قال ياليت متاع تمطى سلخفاء	أال ياريت بتاع تمطع زحلفة	Replace the characters and vowels. "البريد الحروف"	Add new stem and assign the same rules as CAW
ابتل ارتمى ارتوى اششوى افترض	اتبل اترمى اتروى اتشوى اتفرض	Distortions in the structure of the word "تقدم التاء علي فاء الفعل في صيغة أفعل"	Add new stem and assign the same rules as CAW

## 5.2 Lexicon Structure

We enhanced the Buckwalter's lexicon tables with new extra fields:

- *ID*: An identifier to distinguish each word segment. This field is used for indexing purposes,
- *SegmentType*: it can be either MSA (Ar-Ar), Egyptian dialect (Ar-Eg) or other dialects such as Jordanian dialect (Ar-Jr) for future extension of the lexicon.

- *NewSegmentPosition*: this is the new position of the word segment, which indicates its proper order, within the target MSA word or sentence. This field takes one of the following values:
  - same position (SP),
  - start of word (SoW),
  - end of word (EoW),
  - start of sentence (SoS),
  - end of sentence (EoS), and the like.

For example, the Egyptian colloquial sentence "جيت امتي؟" (you came when?) is literally transformed to the MSA sentence "جئت متى؟" (you came when?). Given that the word "امتي" takes the value "SoS" for the *NewSegmentPosition* field, the transformation moves this word to the beginning of the sentence in order to get the target MSA sentence "متي جئت؟" (When did you come?).

## 5.3 Mapping Rules

A new database file, called *Mapping Table (MT)*, is introduced to encode the mapping rules between Egyptian Arabic to MSA. This table uses the value of the lexicon's ID field to cross reference the lexical entries inside the rules. The mapping is either one-to-one or one-to-many. An entry of this table has three fields: source colloquial word, target colloquial word, and the mapping mode. The mapping mode takes either of two values: 0 indicates one-to-one and 1 indicates one-to-many. In the following we will present examples of mapping rules along with their related lexicon entries.

*Example 1: mapping the colloquial interrogative "امتي" (when) to the MSA word "متي".*

This rule will be represented in the MT by an entry with the values: source colloquial interrogative=ID 79831, target MSA interrogative ID=64063, and mapping mode=0, where the source and target words entries in the lexicon are:

```
64063 mtY mataY FW-Wa when متى
      ٲٲٲٲ mataY/INTERROG_PART
      ٲٲٲٲ /أداة استفهام mataY_2
      ٲٲٲٲ mtY(1) 1) مي SP Ar-Ar

79831 >mtY >mtY FW-Wa when أمٲٲٲٲ
      ٲٲٲٲ >mtY/INTERROG_PART
      ٲٲٲٲ أمٲٲٲٲ /أداة استفهام >mataY
      ٲٲٲٲ >mtY أمٲٲٲٲ SoS Ar-Eg
```

*Example 2: mapping the colloquial prefix “عال” (on-the) to the MSA words “على” (on) and the prefix article “ال” (the).*

These rules will be represented in the MT by two entries (one-to-many): 1) source colloquial prefix=ID 79835, target MSA preposition=46196, and mapping mode=0, and 2) source colloquial prefix=ID 79835, target MSA article=15, and mapping mode=1.

In addition to adding colloquial prefixes, stems and suffixes to the corresponding lexicon database file, the compatibility database files should also be modified to include entries that will verify the recognized prefix, stem, and suffix of the input Egyptian Arabic word. Consequently, the colloquial prefix “عال” (EAl) will have also entries in the respective compatibility tables: *tableAB*, and *tableAC*. As a matter of fact, these entries will be treated in a similar way to the MSA prefix “بال” (BiAl). In order to distinguish between MSA and colloquial entries, we used the prefix “C\_” as an indicator of a colloquial entry, e.g. the morphological category of “عال” (EAl) is “C\_NPref-EAl” while the MSA for “بال” (BiAl) is “NPref-BiAl”

## 6. Conclusion

We have investigated the variations between Egyptian Arabic and MSA, and introduced lexical transfer techniques between these languages. These techniques reuse existing Arabic morphological analysis resources and enhance these resources with meta data of Egyptian Arabic. Our approach is able to transfer written Egyptian colloquial dialect into its corresponding MSA forms in order to cope with the dramatic increase of written colloquial dialects. This step showed that it is easy to incorporate colloquial Arabic dialects into existing MSA tools. We hope these techniques to be applied to other colloquial Arabic dialects such as Moroccan, Levantine and Gulf Arabic. Moreover, using MSA Arabic as a hub language, into and out of which all transfer is done, will make the transfer among these Arabic colloquial dialects straight way such that speakers of one dialect is able to read and understand written material of other Arabic dialects.

## References

1. Shawki Deef, Tahrifat Al Amiah Lil Fousah Fi El Kawaad wa Al Bonian we Al Horouf wa Al Harakat , تحريفات العامية , للفصحى في القواعد والبيئات والحروف والحركات , Dar El Maaref, Egypt, 1994.
2. Scocrates Spiro ,”An Arabic – English Dictionary of the Colloquial Arabic of Egypt”, Lebanon Bookshop Publisher, Lebanon, 1973
3. Tim Buckwalter, Buckwalter Arabic Morphological Analyzer Version 2.0 LDC Linguistic Data Consortium, University of Pennsylvania, 2004. Available at <http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2004L02>
4. Ahmed Taymour, "Moagam Taymour Al Kbir: volume 1, 2 & 3", "معجم تيمور", "مجلد 1 ؛ 2 ؛ 3 :الكبير", Dar El Afak el Arabia, Egypt, 2003.
5. Ibn El hanbaly, "Bahr ul-awwam fi ma asaba fihil a'wam, " بحر العوام فيما أصاب فيه " , Ibn Zietoun, Syria,1937
6. Owen Rambow, David Chiang, Mona Diab and Nizar Habash, The final report: Parsing Arabic Dialects (version I), CSLP, JHU, Baltimore, USA, 2006.
7. Nizar Habash and Owen Rambow, MAGEAD:A Morphological Analyzer and Generator for the Arabic Dialects, In the Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, Sydney, PP 681–688, July 2006.
8. Ralf D Brown, Example Based Machine Translation in the Pangloss System, In the proceedings of The 16th International Conference on Computational Linguistics, Copenhagen (COLING-96), pp 169-174, 1996.
9. Ralf D Brown and Robert Frederking Applying Statistical English Language Modeling to Symbolic Machine Translation, In the Proceedings of the 6th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-95), Leuven, Belgium, pp 221-239, 1995.

# WordSets: Finding Lexically Similar Words for Second Language Acquisition

Vera Sheinman  
Department of Computer Science  
Tokyo Institute of Technology, Japan  
vera46@cl.cs.titech.ac.jp

Takenobu Tokunaga  
Department of Computer Science  
Tokyo Institute of Technology, Japan  
take@cl.cs.titech.ac.jp

## Abstract

We introduce a method of expanding a multiple-words input by a short list of similar words in a manner suitable for Second Language Acquisition (SLA). Similarity for that purpose is determined based on two aspects, semantic relations and typicality. Finding words with similar typicality is particularly important for SLA tasks. The study incorporates, and shows the advantage of a recently introduced distance measure that uses the Web as its corpus. The value of the proposed method is demonstrated by empirical experiments on word lists provided by teachers.

## Keywords

Second language acquisition, lexical acquisition, similar words, typicality, familiarity, similarity.

## 1. Introduction

Computational modeling of Second Language Acquisition (SLA) may be a great step toward a deeper understanding of how humans acquire new languages. Rappoport and Sheinman in [14] proposed a preliminary computational model of SLA. One of the components of their model is the prior conceptual knowledge of the learner. Existence of such knowledge is one of the major differences between SLA and First Language Acquisition (FLA). Hence, it requires special attention in SLA studies. In their study that component was constructed manually and was tailored to a specific corpus. A construction of an extensive model of learners' conceptual system is important. Ontology is one of the ways to do so, reflecting the recent beliefs about the structure of conceptual knowledge in psycholinguistic research. WordNet [12] may be viewed as one of the most extensive ontologies of that kind available.

This study introduces a method to compute conceptual categories, based on several examples. Proposed method will allow for (semi)automatic construction of an adult learner's conceptual system model. Additionally, this method may be applied as a tool for language courseware authoring, as well as a helpful tool for language learners, or even native speakers that are missing a word. For instance, if there is a difficulty retrieving the word for 'kiwi', entering examples of similar fruits such as 'apple' and 'lemon' might be a way to retrieve the missing word.

The type of learning that we analyze for the purpose of this study is generalization from examples, similar to [14]. After the learner hears enough examples in the second language, he is ready to generalize into a construction and he is able to generate new phrases. Learners are unlikely to generalize after a single example. In our study we require an input of at least two words to trigger recognition of a conceptual category and automatic extension of it. The scope of the current study is English nouns.

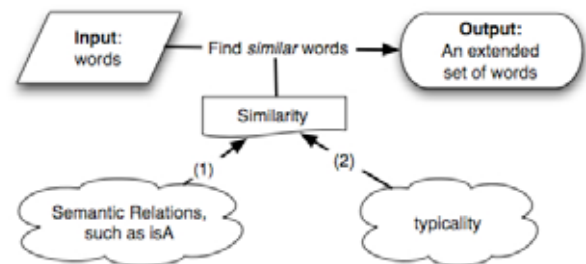


Figure 1: Diagram of Problem Definition

## 2. Problem Definition

A sketch of the problem that we suggest to solve automatically in the current study is shown in Figure 1. The 'WordSets' method, and an application implementing it, are the key products of this study. As part of the solution to this problem, we define 'similarity suitable for SLA tasks'. We focus on two aspects of similarity, described in the subsections below.

### 2.1 Semantic Relations

Words or concepts may be represented in an extensive network, such as WordNet, with many types of links connecting them. For instance, one such link is the 'isA' relation, or in terms of WordNet, the hyponym-hypernym relation. Focusing on two concepts out of the whole network reduces the numerous possibilities to consider to only the links that connect them. Choosing more than two concepts reduces the links even more, and provides further information about the similarity of these concepts.

The given input words share some semantic relations. We detect two such relations by looking for the least common subsumer of the given concepts, traversing the appropriate relation links in WordNet network.

## 2.2 Typicality

Some concepts are more common than others, while some are rare or even obscure. More common concepts are usually more likely to be encountered, and it is more important to learn the words representing them in the early stages of SLA. Typicality of the given words provides further information about the words and about the desired extension, and it should not be underestimated. If the given words share similar typicality, their most suitable extensions should share that typicality as well.

Consider the following example, in the context of a learner searching for an extension for a set of words he provides:

```
Input: olive, navy, maroon
Output: red, blue, yellow
```

The words provided in the input are not obvious choices for colors. Extending the set by the most basic colors will not provide the information that is probably being sought. In the context of a learner, if he knows such words as 'maroon', it is improbable that he does not know 'blue'. The information in the output will be redundant for him.

In the opposite case, for very typical members of a category provided as the input, presenting complex words as the most similar extensions will overwhelm the learner. Moreover, it will not be useful for courseware authoring that seeks simple category members, easily recognized by students. Additionally, it will not be useful for modeling the core conceptual system of the most useful concepts based on typical examples.

## 3. Related Work

There is a large body of research and products that deal with finding similar words for a single entry. Additionally, there is an extensive body of work for measuring semantic similarity between two given words. Some of these studies base their similarity measures on WordNet [3]. Others exploit various computational techniques to measure such similarity in a corpus [5], explore psycholinguistic data, etc. One of the major directions is distributional similarity. An influential work by Lin [10] in this field analyzes syntactic features from a corpus, and comes up with rather broad clusters of similar words, synonyms and hyponyms mixed. Weeds and Weir [15] provide an excellent survey on distributional similarity techniques. It is still difficult to distinguish among the various semantic relations such as hyponyms or holonyms by these techniques, a knowledge that we need to protect the learners from unnecessary information.

Most previous studies refer to WordNet as the major available lexicon. Some previous studies on lexical similarity [6], [15], [16] use WordNet as the golden standard for evaluation purposes, especially for nouns. In this study, we focus mainly on ordering similar nouns by

typicality, using well-defined semantic relations, and hence we extract words similar to the input words directly from the ready constructed WordNet, using WordNet-based similarity measures. In this sense other studies on similarity are complementary to this study.

We work with an input of at least two entries, similarly to learners that generalize based on at least two examples. This task is essentially different from the task of finding similar items based on a single example, that most of the lexical acquisition works tackle.

The problem of providing similar items based on entry of several words may be viewed as **Ontology Learning** - provided existing entries in an existing category, this category is extended.

Although in reality some examples that learners encounter may be erroneous, they will still be able to create correct generalizations eventually. However, for the purpose of this study we compute the set of items that are equally similar to each one of the input entries, leaving possible inaccuracies or inconsistencies in the input set out of its scope.

Nation [13] recommends that language teachers avoid introducing words from lexical sets simultaneously. Some textbooks [18] follow this recommendation, and extend the lexical sets gradually. The research in this field is complementary to our work. Automatic construction of semantically related concepts might help teachers and textbook authors to be aware of such limitations.

### 3.1 GoogleSets

GoogleSets [7] is one of the projects in Google labs that provides a friendly tool to extend sets of words. Similarly to the proposed method, it receives multiple words as its input and provides an output of words similar to the input. GoogleSets is an efficient, dynamic, and generic application. It works for any kind of inputs (simple words, movie names, numbers, etc.), using the Web as its corpus.

Table 1. GoogleSets Results Example

GoogleSets Output (first 8 words) for <b>Input:</b> Doctor, Engineer
<i>Bureaucrat, Fixer, Enforcer, Trader, Adventurer, Soldier, Scientist</i>

However, lacking any specific linguistic objectives or any linguistic knowledge augmentation, it may not provide for building ontologies of conceptual systems of humans, or serving as a tool for learners. Table 1 shows an example of this idea. 'Doctor' and 'Engineer' are both very typical professions, and it is likely to assume that such similarly typical items as 'Nurse' or 'Teacher' are anticipated as the output. Instead, 'Bureaucrat', whose semantic similarity to the input set is questionable is the first word returned. Also, 'Enforcer', which is much less typical than the provided examples is one of the top results.



Although 'Soldier' and 'Scientist' come last on the extension list, they seem to be the best extensions.

Our method may be viewed as an adaptation of the GoogleSets results to make it suitable for SLA purposes.

### 3.2 Normalized Google Distance (NGD)

Cilibrasi and Vitanyi [5] introduce a distance measure between concepts, intended for large corpora such as the Web. Using the whole Web as the corpus, with the computational ease of acquiring page counts is a good method to obtain averaged information about what is typical and what is not. NGD is incorporated in our method for measuring typicality of words.

## 4. The Proposed Method

Given a set of words  $W = \{w_1, \dots, w_n \mid n \geq 2\}$  (1) as the input, our method comprises 4 stages leading to output of a set of similar words. These 4 stages are described in the subsections below.

### 4.1 Disambiguation

In this stage, we perform word sense disambiguation (WSD) to determine the semantics of the words in (1). We assume that the words in (1) are similar enough, and consequently they can serve as the context for each other of the words in the set. The procedure is as follows.

**Step 1:** For each word  $w_i$  in  $W$  (1), acquire its noun senses  $\{n_{i1}, n_{i2}, \dots\}$  from WordNet 2.1,

$$S = \{\{n_{11}, \dots, n_{1m}\}, \dots, \{n_{n1}, \dots, n_{nk}\}\}. \quad (2)$$

**Step 2:** For each combination of senses in (2), compute the sum of Lesk similarity measures [1] between its members pairwise.

**Step 3:** Determine the combination with the highest sum of similarities  $SD = \{n_{1x}, \dots, n_{ny}\}$ . (3)

There are several approaches for WSD task. In this study we search for semantic relation information, and it makes sense to use WordNet-based similarity measures to perform disambiguation.

Budanitsky and Hirst [3] in their thorough evaluative survey suggest that the measure by Jiang-Conrath [8] is superior to other WordNet-based measures. However, this measure does not provide any results for many entries. Additionally, although this measure is very effective in measuring similarity between entries that share the same hypernym in WordNet hierarchy, it is not as effective for entries that are similar by other relations, such as meronymy. As opposed to Jiang-Conrath, Lesk measure that is based on gloss overlaps in WordNet reflects similarity between words with meronymy relation equally well. Recent studies [11] report on Lesk outperformance of Jiang-Conrath for the purposes of WSD.

The meronymy relation is important for our task where the input words often tend to be parts (meronyms)

of some concept. For instance, the words 'bumper' and 'window' that are both meronyms of 'car' cannot be disambiguated by Jiang-Conrath. However, Lesk provides a correct disambiguation for them.

### 4.2 Detection of Semantic Relations

We assume that the word senses in (3) share some semantic relations. Two shared relations may be detected automatically using WordNet relations:

$$\begin{aligned} Z_1 &= \text{least\_common\_holonym\_in}(SD), \\ Z_2 &= \text{least\_common\_hypernym\_in}(SD), \quad (4) \\ R &= \{\text{meronyms}(Z_1), \text{hyponyms}(Z_2)\}. \end{aligned}$$

$Z_1$  in (4) may be non-existent, due to the structure of WordNet. For instance,  $\text{apple}\#n\#1$ <sup>1</sup> and  $\text{pear}\#n\#1$  do not share a holonym. In such case the relation  $\text{meronyms}(Z_1) = \phi$ .  $Z_2$ , however, always exists.

### 4.3 Extension

In this stage the set of word senses  $SD$  (3) is extended by adding the word senses that are acquired by recursive WordNet traversal for each of the relations in  $R$  (4),

$$\begin{aligned} E_1 &= \{n_{1x}, \dots, n_{ny}, e_{11}, \dots, e_{1m}\}, \\ E_2 &= \{n_{1x}, \dots, n_{ny}, e_{21}, \dots, e_{2h}\}. \end{aligned} \quad (5)$$

The items that are deeper by more than one level than the deepest item in the input in the WN hierarchy are not added to the input. This is done, in order to prevent overly specific items, or instances appearing in the same lexical set with other items. For example, consider 'airport' and 'bank' provided as an input. In the context of extraction of words from examples, the user might expect to see 'hospital', or 'gas station' as other examples of institutions, rather than 'Kennedy airport' or 'Mutual Savings Bank' that are of greater specificity than the items in the input.

For the simplicity of calculation, we remove relations that have a very general hypernym, such as 'object' or 'substance'. We determine the intended extension as too general when  $Z_2$  is closer to the WordNet root than to the items in the input, so that  $\min_{SD} (\text{depth}(n_{ij}) - \text{depth}(Z_2)) > 2/3 (\text{depth}(Z_2))$ .

The pruning techniques mentioned above will malfunction in certain cases, due to the unbalanced state of WordNet hierarchy. Better methods will be considered in the future studies.

### 4.4 Ranking Procedure

The suggested ranking procedure is the key part of our study. It is counter-productive to overwhelm learners with information. Ranking the results will allow us to

<sup>1</sup> The notation  $\text{apple}\#n\#1$  stands for the first noun sense for the word 'apple' in WordNet. It refers to the fruit 'apple'.

differentiate between the more useful and less useful extensions of the given set.

Given the extended sets of word senses (5), the elements of each set will be ranked by their typicality (section 2.2). The items with typicality level closest to the input words will be ranked the highest. The Web is a huge corpus, with plethora of domains evening out the typical usages. We use frequencies in the Web as the markers for typicality.

In order to calculate typicality we use the distance measure of NGD (section 3.2). NGD requires  $M$  (the total number of pages indexed by a search engine). Most of the large search engines do not declare this number. We estimate  $M$  by retrieving the number of webpages that include the word ‘*the*’, and restrict the search to English pages. An interesting study [2], suggests an improvement for this kind of estimation. We plan to experiment with the suggested measure in the future.

An interesting feature of NGD is that it tends to cluster items not only by their similarity, but also by their frequency. For instance, the colors ‘*red*’, and ‘*blue*’ are clustered together, apart from ‘*pink*’, and ‘*wine*’, which seem more similar to ‘*red*’ than ‘*blue*’ [4].

NGD measures the distance between two items -  $x$ ,  $y$ . We measure the distance between a set of items to one item -  $X$ ,  $y$ . For the purpose of this study we used the

$$distance(X, y) = \sum (NGD(x, y) \mid x \in X). \quad (6)$$

The smaller the distance of an item from the input set, the higher its ranking.

When submitting queries to a search engine, we once again use words, rather than WordNet senses. Hence, we need further disambiguation, in order to prevent many results such as “*Apple computer*” biasing our calculation when dealing with an input of ‘*apple*’ and ‘*pear*’. This is achieved by incorporation of NGD. Similarity is measured between each input word and the word in question. We implement the distance measure using estimated counts by Yahoo.

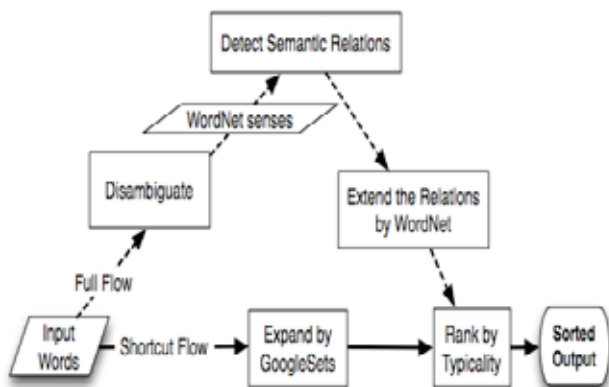


Figure 2: Two possible flows for ‘WordSets’

## 5. Shortcut Flow

The main focus of this study is on the ranking of words by their similarity to the words provided in the input. In order to evaluate only this stage, and also in order to provide solution for the cases when WordNet does not include the input words, we introduce an alternative shortcut flow. The two possible flows in general are overviewed in Figure 2. The steps of the shortcut flow are presented below.

**Step 1:** Expand the input words by the larger set in GoogleSets.

**Step 2:** Standardize the results, due to inconsistency of GoogleSets results in terms of capital letters and such. This step is performed using the validity check provided in WordNet. All the nouns are stored in their singular form in low-case letters for consistency.

**Step 3:** Rank the results by the same ranking procedure as described in section 4.4.

**Step 4:** Output the results sorted by their ranking.

## 6. Evaluation

In order to test our method, we have performed several evaluation procedures as described in the subsections below.

Table 2. The evaluation of the full flow using WordNet

Word lists	Precision% full / reduced	Recall% full / reduced
Family	8 / 49	76 / 49
Colors	9 / 78	83 / 78
Vegetables	11 / 33	81 / 33
Buildings	0 / 0	0 / 0
Fruits	3 / 27	30 / 27
Clothes	5 / 21	47 / 21
House	4 / 7	19 / 7
Tools	3 / 50	88 / 50
Body	4 / 18	34 / 18
Animals	2 / 6	12 / 6
<b>Macro average</b>	<b>5 / 29</b>	<b>47 / 29</b>
<b>Micro average</b>	<b>6 / 36</b>	<b>54 / 36</b>

### 6.1 Lexical Sets from Word Lists

Ten lexical sets were retrieved from word lists provided by English teachers for beginners [9] from a site for English learners in Japan. For each one of the lexical sets two of its members were randomly chosen as the input words. The rest of the words served as test set. Both, the full procedure using WordNet (section 4), and the shortcut procedure (section 5) were performed for at least two different input sets for each word list. In total 32 different input sets were tested, and 32 hyponyms and 5 meronyms relations were detected. In cases when the size of the acquired set was big enough the set was reduced to

the same size as the appropriate word list size after sorting it by ranks. We compared the precision rates for the full set (before ranking) vs. the reduced set (after ranking).

**Table 3. Shortcut flow evaluation**

Comparison of our method (WS) with GoogleSets (GS)				
Word lists	Precision % full / reduced		Recall % full / reduced	
	GS	WS	GS	WS
Family	41 <sup>2</sup> / 61	42 / 65	82 / 56	82 / 59
Colors	24 / 89	24 / 69	100 / 89	100 / 69
Vegetables	29 / 47	36 / 56	67 / 47	79 / 56
Buildings	8 / 8	9 / 11	9 / 8	11 / 9
Fruits	44 / 56	48 / 53	100 / 56	100 / 53
Clothes	28 / 37	23 / 33	34 / 30	26 / 26
House	3 / 4	4 / 5	2 / 2	3 / 3
Tools	<b>8 / 25</b>	<b>8 / 38</b>	44 / 25	44 / 38
Body	43 / 52	33 / 38	66 / 52	46 / 38
Animals	56 / 66	62 / 69	51 / 31	53 / 30
<b>Macro avg.</b>	<b>28 / 44</b>	<b>29 / 44</b>	<b>59 / 39</b>	<b>57 / 38</b>
<b>Micro avg.</b>	<b>29 / 47</b>	<b>30 / 47</b>	<b>63 / 43</b>	<b>64 / 43</b>

To illustrate the evaluation process consider the word list for 'tools' that contains 10 words: *drill, hammer, knife, plane, pliers, saw, scissors, screwdriver, vise, and wrench*. Two input word pairs were randomly chosen 'drill, pliers', and 'hammer, vise'. For the first input set, 228 words were extracted from WordNet, and 43 words were extracted from GoogleSets. Precision and recall values were first calculated for these lists comparing them to the original word list of tools. As the next step we sorted both of the lists by our ranking procedure and reduced each of the sets to the first 10 words. Then, we recalculated precision and recall for the shorter lists to evaluate our ranking procedure's contribution. For comparison of the sorting we also reduced the list by GoogleSets in the same manner, without ranking it. The same procedure was performed for the second input set.

Our main purpose in the analysis is to show improvement of precision for the reduced ranked lists. Perfect precision values cannot be anticipated, because the chosen lists are a sample of word lists that typically appear in textbooks. They may omit some words, due to size limitations or other reasons. However, improvement of precision after ranking shows good tendency toward conformity with the teachers' opinions. Recall values are expected to decrease due to reduction of the acquired sets. The precision values for the full procedure that are shown in Table 2 clearly suggest that the ranking procedure

successfully cleans the word sets from redundant items, increasing the precision by 6 times on average for each list. The best ranking was achieved for *colors* with inputs 'orange, white', 'black yellow', and 'green, purple'.<sup>3</sup>

The precision results for the ranking procedure in comparison with GoogleSets show similar values on average (see Table 3). Precision in this experiment is higher than in the full flow (see Table 2), due to better order by similarity and typicality of items in GoogleSets, compared to non-existent order in WordNet synsets. Note the better precision and recall for the ranked *tools* set with inputs 'drill, pliers' and 'hammer, vise'. Ranked lists show better results for 6 word lists, and worse precision for *colors, fruits, clothes* and *body parts*.

## 6.2 Familiarity Rating

Familiarity values used for this experiment were extracted from the MRC Psycholinguistic Database [17]. The total number of rated words extracted was 4896, from the lowest rating of 101, to the highest of 657.

All the words (total of 19) in the category of 'vegetables' that appear both in WordNet and in familiarity rating were extracted. One copy of the list, noted by *F*, was sorted according to its familiarity rates, another copy *X* was ranked using the ranking procedure as described in section 4.4 using the top two familiar items from *F* as the input. The order of the two lists was compared summing the absolute error as following.

$$rank_L(x) = \text{the position of item } x \text{ in list } L$$

$$error(X) = \sum |rank_F(x) - rank_X(x)|$$

The error for the ranked set is 48 and the mean error (calculated combinatorically) is 96. The order of the ranked set is two times more similar to the list *F* than the average. Discrepancies in the order of the sets are anticipated. One of the contributions to the inconsistency may be relatively old dating of the familiarity rating experiments. The typicality ratings are based on a more recent language that appears in the web.

## 7. Discussion

We have pointed out the needs of SLA in the field of computerized lexical acquisition. Motivated by them, we have divided the former known notion of similarity into two aspects of semantic similarity and typicality level similarity, and we have presented a method for semisupervised lexical acquisition from multiple words input based on this new notion. Our method is web-based, hence, providing dynamic results that reflect the changes that happen in the language use from day to day.

<sup>2</sup> The precision values for GS and WS before reduction, sometimes differ due to the standartization procedures applied on GoogleSets result before ranking it (step 2 in section 5)

<sup>3</sup> In some cases, the results acquired from WordNet were too general, or there were errors in the disambiguation. In such cases, we reran the tests with additional input words.

We implemented the suggested method using the distance measure of NGD, and compared it to the existing application of GoogleSets. NGD is a universal measure that measures distance over all the implicit similarity aspects between two items. It does not require an annotated or parsed corpus. We have shown its applicability to the similarity by typicality level. We plan to compare its usefulness with additional approaches and similarity measures in the future.

Integration of the presented method into computational modeling of SLA seems to be a much needed direction. Additionally to the theoretical value, being able to extend several example words by words of similar typicality and semantic category may be applicable in several ways.

One way is automatic acquisition of lexical sets for textbooks authoring. Currently, textbook authors construct lexical sets, and word lists by manual work, relying on their memory and expertise. Language changes dynamically, textbooks have to be reissued and lexical sets needed for them have to be reinvented. Instead, a dynamic method that reflects the modern language use, because it is Web-based, and that takes the typicality of words into consideration will reduce the costs, and will provide richer resources for the text authors' consideration.

Another useful application of the proposed method would be as an extension for a dictionary. It will provide for cases that a certain word belongs to the passive vocabulary, but cannot be retrieved directly. Furthermore, it will be helpful in cases when the word in the target language does not have an equivalent in learner's first language<sup>4</sup> of the learner, and bilingual dictionary cannot be used for that purpose. For instance, the Russian word for 'light blue' ('голубой' – *goluboy*) is a very basic color name, of similar typicality to such basic colors as 'red' or 'blue'. A possible English equivalent 'azure' exists, but it is much less typical in English. The learner that wants to learn, or reinforce his knowledge about basic colors in Russian will easily retrieve the ubiquitous word for 'light blue' by providing the Russian equivalents for 'blue' and 'red' to WordSets. If the word is already in his passive vocabulary he will recognize it. Otherwise, he will look it up in the bilingual dictionary that will be complementary to WordSets in such case.

Word lists by language teachers provide a good combination of similarity by semantics and by typicality in a way useful for learners, hence being important resources for evaluation. The empirical evaluation provided in this study shows a clear improvement of precision by ranking a set of similar words. It also demonstrates comparability of the established method to GoogleSets and a general conformity with the familiarity

ratings. However, a limited choice of manually constructed word lists as the evaluation data cannot fully reflect its advantages and deficiencies. We plan an extensive evaluation procedure with human subjects that are language learners in the near future.

The scope of the current study is English. However, we believe, that the suggested method may be applied for other languages in a similar manner, given large corpora and a WordNet in another language.

## 8. References

- [1] S. Banerjee and Ted Pedersen. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. CICLing-02, Mexico, 2002.
- [2] I.A. Bolshakov and S.N. Galicia Haro. How many pages in a given language does Internet have? (In Russian). Computational Linguistics and Intellectual Technologies. Dialogue-2003, pp. 76-82, Nauka, Moscow, Russia, 2003.
- [3] A. Budanitsky and G. Hirst. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. Computational Linguistics, 32(1):13-47, 2006.
- [4] R. Cilibrasi and P. Vitanyi. The ComLearn Toolkit, <http://clo.complearn.org/clo/showexpnum/1166446698/experiment.s.html>, 2003.
- [5] R. Cilibrasi and P. Vitanyi. The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering* 19(3): 370-383, 2007.
- [6] D. Davidov and Ari Rappoport. Efficient Unsupervised Discovery of Word Categories Using Symmetric Patterns and High Frequency Words. ACL, Sydney, 2006.
- [7] Google™ Sets. <http://labs.google.com/sets>, 2002.
- [8] Jay Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *COLING*, Taiwan, 1997.
- [9] C. Kelly and L. Kelly. <http://www.manythings.org/vocabulary>. 2005-2006.
- [10] D. Lin, Automatic Retrieval and Clustering of Similar Words. COLING-ACL, Montreal, 1998.
- [11] D. McCarthy, Rob Koeling, et al. Predominant Word Senses in Untagged Text. ACL. Barcelona, Spain, 2004.
- [12] G.A. Miller et al, WordNet. A Lexical Database for the English Language. Cognitive Science Lab, Princeton University. <http://www.cogsci.princeton.edu/~wn>, 2006
- [13] Paul Nation. Learning Vocabulary in Lexical Sets: Dangers and Guidelines. *TESOL Journal*, v. 9, n. 2, pp. 6-10, 2000.
- [14] Ari Rappoport and V. Sheinman. A Second Language Acquisition Model Using Example Generalization and Concept Categories. Workshop on Psychocomputational Models of Human Language Acquisition, ACL, Ann Arbor, 2005
- [15] J. Weeds and D. Weir. Co-occurrence Retrieval: A Flexible Framework for Lexical Distributional Similarity. *Computational Linguistics*, V. 31, Issue 4, 2005.
- [16] D. Widdows and B. Dorow, A Graph Model for Unsupervised Lexical Acquisition. *COLING*, Taiwan, 2002.
- [17] Wilson, M.D. The MRC Psycholinguistic Database: Machine Readable Dictionary. *Behavioural Research Methods, Instruments and Computers*, 20(1), 6-11, 1988.
- [18] Y. Yamazaki and D. Mitsuru. Hakase: Basic Japanese for Students. 3A Corporation. Tokyo, 2006.

<sup>4</sup> By first language we refer to any language that the learner knows, not necessarily one, for this matter

# Disambiguating Tense, Aspect and Modality Markers for Correcting Machine Translation Errors

Anil Kumar Singh, Samar Husain, Harshit Surana, Jagadeesh Gorla, Dipti Misra Sharma\* and Chinnappa Guggilla†

## Abstract

All languages mark tense, aspect and modality (TAM) in some way, but the markers don't have a one-to-one mapping across languages. Many errors in machine translation (MT) are due to wrong translation of TAM markers. Reducing them can improve the performance of an MT system. We used about 9000 sentence pairs from an English-Hindi parallel corpus. These were manually annotated with TAM markers and their mappings. Based on this corpus, we identify the factors responsible for ambiguity in translation. We present the results for learning TAM marker translation using CRF. We achieved an improvement of 17.88% over the baseline.

## Keywords

Machine Translation, Tense, Aspect, Modality, TAM Markers

## 1 Introduction

Tense, aspect and modality are important elements of natural languages. They are needed for specifying the information about the world which is temporal in nature, or tell us something about the status of an action, or about the ability to perform an action. In some languages, they also govern the realization of a particular case marker. Different languages have different systems for marking such temporal (including aspectual and modal) information. In other words, TAM markers used by different languages don't have a one-to-one correspondence. TAM markers are not the only device used for expressing temporal information, but they can be very useful for NLP. They are a bit like function words. Like prepositions, even if not to the same extent, they can help in arriving at the correct syntactic and semantic analysis of a sentence. At the same time, they have an inherent meaning (even if ambiguous). This means that they are a bit like content words too. In this paper, we argue that these markers are under-utilized sources of linguistic information.

We first explain how we are defining TAM markers. Then we show that a significant percentage of errors in machine translation are due to wrong translation of TAM markers. We prepared an annotated parallel corpus to study the possibility of correcting these errors. The aim was to improve the performance of an

MT system. We annotated around 9000 sentence pairs from a sentence aligned English-Hindi parallel corpus with TAM markers and their mappings. Based on this corpus, we present the lists of most frequent markers and their translations. We also present the results of our experiments on learning translations of TAM markers using Condition Random Fields or CRF [4] and also show that we can improve the accuracy of an MT system by using this method. For our experiments, we have used the 0.73 version of the Shakti MT system [6].

## 1.1 What Exactly are TAM Markers

TAM markers are the combination of inflections (*en, ing, nA<sup>1</sup>, tA*) and auxiliary verbs (*is, been, HE, thA*) or modals (*can, should, sakanA, paDZA*) or words indicating negativity (*not, naHIM*). These combinations together provide the information about tense, aspect and modality.

We can explain this by a hypothetical example from an English to Hindi machine translation system:

**SL:** So what happens now?

**TL:** to aba kyA HogA?

'So now what will-happen?'

**TL (Default):** \* to aba kyA HonA HE?

In the example above, SL is the source language (English) sentence, TL is a correct translation in the target language (Hindi) followed by the literal English version of the correct Hindi translation. Finally, TL (Default) is the translation provided by the MT system, assuming that everything is correct except the TAM marker, because it was taken from a TAM dictionary with a one-to-one mapping.

In our terminology, we would say that in the SL sentence, *happens* has *PRES* (simple present) TAM marker, while the marker in TL for *HogA* is *gA* (future or hypothetical).

## 1.2 Empirical Evidence of the Problem

To get empirical evidence for our contention that wrong translation of TAM markers is a notable source of errors in MT, we extracted 250 random English sentences from the corpus. These sentences were run through the MT system. We manually checked these

\*Language Technology Research Centre, IIIT, Hyderabad, India, {aiklavya,samarhusain,surana.h}@gmail.com, jagadeesh.gorla@gmail.com and dipti@iiit.ac.in

†Applied Research Group, Satyam Computer Services Ltd, IISC campus, Bangalore, chinnappa.guggilla@satyam.com

<sup>1</sup> To represent text in Indian languages, we have used the RR notation. In this notation, capitalization roughly means longer length for vowels, and a small *h* after a consonant means aspiration.

<b>English</b>	PRES, PAST, to_0, ing, is_en, will_0, en, can_0, are_en, was_en, may_0, has_en, have_en, should_0, were_en, should_be_en, would_0, must_0, can_be_en, had_en, do_0, is_ing, to_be_en, by_ing,
<b>Hindi</b>	tA_HE, HE, nA, yA, thA, 0_sakatA_HE, gA, yA_jAtA_HE, yA_HE, nA_cAHIye, 0_raHA_HE, 0_kara, tA, yA_gayA_HE, yA_gayA, ye, tA_thA, yA_thA, 0_jAtA_HE, 0_gayA, HogA, yA_jAnA_cAHIye, yA_jA_sakatA_HE,

**Table 1:** *Most frequent TAMs*

sentences and marked the errors due to wrong translation of TAM markers. We found that 152 sentences had such errors. The number of wrongly translated TAM markers was 163 out of a total of 296 markers. This shows that there is empirical evidence of TAM markers being the cause of a significant number of errors in machine translation. If we can reduce these errors by using a better technique for TAM marker translation or for correcting such errors in the MT system output, we can improve the performance of the MT system.

## 2 Previous Work

Tense, aspect and modality have been studied extensively by linguists, both separately and as part of the study of temporal information encoded in natural languages. One of the most well known works in the first category is by Bybee et al. [2]. Their book discusses how tense, aspect and modality have evolved in different languages.

Vendler’s work [10] on verb classification with respect to time (or tense) is the basis of a lot of work on tense. In this work, he claimed that almost all the verbs can be classified into a few classes. Richenbach, in the classic work called ‘The Tenses of Verbs’ [8], suggests that the times of events can be located with respect to a deictic centre, which makes them similar to pronouns (the anaphoric view of tenses).

A lot of work has been done on temporal information from a computational point of view too. Dorr and Olsen [3] use a Lexical Conceptual Structure (LCS) based representation of Levin’s classes [5]. The aspectual classes are defined in terms of three features (telicity, dynamicity and durativity) and can be used to help in machine translation and generation.

Tense, aspect and modality in Indian languages have also been studied from a linguistic point of view. The book ‘Tense and Aspect in Indian Languages’ edited by Lakshmi Bai and Mukherji [1] contains a collection of a few such papers.

This paper has a different focus because we are concentrating on machine translation, whereas the focus of works mentioned above was language understanding or information extraction. As has been observed by many, some elements of machine translation may not require deep analysis of meaning.

In our opinion, TAM markers as a separate class of entities have not been given as much importance as they deserve, though they have been considered indirectly in the form of verb inflections and auxiliary verbs etc.

## 3 Problem Formulation

The problem we are addressing in this paper can be formulated as a disambiguation problem. In that sense it is similar to both preposition disambiguation and word sense disambiguation because TAM markers are like function words as well as content words, as mentioned earlier.

At a higher level of abstraction, the problem can also be formulated as a classification task. This formulation is more suitable than that of word sense disambiguation for our purposes because TAM markers form a closed class and the number of classes, though more than for prepositions, is small enough (50-200 for English and many Indian languages) to allow machine learning techniques such as CRF to be used.

If we classify TAM markers by considering contextual similarity of TAM markers, the problem becomes similar to POS tagging by using CRF:

$$t_i = f(s_i, c_i); \quad (1)$$

where  $s_i$  is the  $i^{th}$  TAM marker in the SL sentence and  $t_i$  is the translation of  $s_i$ ,  $f$  represents a classification algorithm based on CRF, and  $c_i$  is the context for  $s_i$ . The CRF implementation that we have used was CRF++ [9]. The features we experimented with are described later in the Section-6.3.

## 4 Markers, Annotation and Dictionary

In this section we will first discuss the development of a set of TAM markers for a particular language, i.e., deciding on the set of TAM marker classes. We then discuss how a better TAM marker dictionary can be built. Finally, we describe how the parallel corpus was annotated with TAM markers and their mappings.

### 4.1 Developing a TAMMSet

How many TAM markers does a language have? Are they naturally and unambiguously very well defined? Not quite. We have to *design* a set of TAM markers for a particular language. This requires linguists, or at least well informed native speakers to sit down and list all possible TAM markers. The task of building this TAMMSet somewhat resembles the task of building a part of speech (POS) tagset for a particular language. In other words, even though they are linguistically significant, there may not be a universally acceptable set of markers for a language. Similarly, the set designed for one language may not be applicable for another.

English	Frequent Hindi Senses	English	Frequent Hindi Senses
PRES	HE, tA_HE, nA, yA_HE, gA, tA, 0_jAtA_HE, yA, ye, yA_jAtA_HE, 0_sakatA_HE, 0_kara, 0_raHA_HE,	PAST	yA, thA, tA_thA, HE, yA_thA, 0_gayA, nA, tA_HE, gA, yA_HE, 0_kara, yA_gayA, tA, tA_raHA
to_0	nA, ne.ke.liye, tA_HE, yA 0_sakatA_HE, HE, ye, 0_kara, gA, nA_HE,	ing	nA, tA_HE, 0_kara, HE, tA_HuA, 0_raHA_HE, tA, yA, ne.ke.liye,
is_en	yA_jAtA_HE, tA_HE HE, yA_gayA_HE, 0_kara	will_0	gA, HogA, tA_HE 0_sakatA_HE, HE

**Table 2:** Correspondence of some TAM categories (English-Hindi)

## 4.2 TAM Marker Dictionary

We had started with a basic TAM dictionary that was being used for machine translation. It was basic in the sense that it had only one-to-one mappings of TAM markers and the list of markers was shorter than the one we have after the new markers have been added during the annotation of the parallel corpus. Moreover, it was not compiled from the study of a corpus. Our aim was to build a proper TAM dictionary which can have one-to-many mappings corresponding to the possible senses of a TAM marker, just like an ordinary dictionary of words. The new dictionary was also to contain at least one example sentence in both SL and TL for every sense of a TAM marker.

## 4.3 TAM Marker Annotation

For annotating TAM markers, we selected at random more than 4000 short (up to 15 words) sentences and 5000 long sentences from a sentence aligned parallel English-Hindi corpus. These sentences were marked up using an interface by five different annotators. Some sets of sentences were then validated by a person different from the one who originally did the annotation. Annotators were given an initial list of SL and TL TAM markers, but they were asked to add a new marker if they thought it was required. An annotated sentence would look like this:

**SL:** So what [happens]<sub>PRES</sub> now?

**TL:** to aba kyA [HogA]<sub>gA</sub>?

**Mapping:**  $PRES_1 \rightarrow gA_1$  (future)

## 4.4 Marker Lists from the Parallel Corpus

A list of most frequent markers (ranked according to frequency in the corpus) is given in Table-1. Table-2 gives a list of most frequent TAM marker mappings for English-Hindi. It is clear that the problem is not trivial and is a bit like word sense disambiguation.

## 5 Why TAM Markers: Another Example

There might be other ways of achieving the same kind of improvement in machine translation. Why use TAM markers? We will try to explain by an example how they can be useful. Consider the following text:

**SL:** We [don't like]<sub>PRES\_not</sub> that horse [flying]<sub>ing</sub> in the sky. [Shoot it down]<sub>IMPER</sub>.

**TL:** AsamAna meM [uDZane vAlA]<sub>ne.vAlA</sub> vo ghoDZA HameM acchA [naHIM laga raHA]<sub>nahIM+0.raHA</sub>. use [mAra girAO]<sub>0.O</sub>.

'sky in that-flies that horse we not-like. it shoot down.'

**Mapping:**  $PRES\_not_1 \rightarrow nahIM + 0\_rahA_2$ ,  $ing_2 \rightarrow ne.vAlA_1$ ,  $IMPER_3 \rightarrow 0\_O_3$

Now consider another variation of the same sentence pairs, *superficially* only slightly different:

**SL:** We [don't like]<sub>PRES\_not</sub> horses [flying]<sub>ing</sub> in the sky. We [shoot them down]<sub>PRES</sub>.

**TL:** AsamAna meM [uDZane vAle]<sub>ne.vAlA</sub> ghoDZe HameM acche [naHIM lagate]<sub>nahIM+tA</sub>. Hama unHeM [mAra girAte HeM]<sub>0.tA\_HE</sub>.

'sky in that-fly horses we not-like. we them shoot down.'

**Mapping:**  $PRES\_not_1 \rightarrow nahIM + tA_2$ ,  $ing_2 \rightarrow ne.vAlA_1$ ,  $PRES_3 \rightarrow 0.tA\_HE_3$

Note that in  $0\_O$ , the first  $0$  is zero and is a place holder or wild card for verbs, while the second one is capital  $o$ , representing the inflection used for imperatives (*IMPER* for English).

What this example shows can be summarized as:

- The same TAM information can be expressed differently in different languages. TAM markers (at least partially) capture this difference. In the first set above,  $PRES\_not$  (present with negation) gets translated as  $nahIM+0.raHA$ , while in the second as  $nahIM+tA$ . The only change in the SL sentence was that 'that horse' was substituted by 'horses'. We can perform deep semantic analysis to get a correct translation in such a case, but it might not be possible in the near future for most (if not all) language pairs for obvious reasons. Or we could translate TAM marker separately.
- On the surface, only a slight change in the second case ('we shoot' instead of 'shoot') leaves the sentence no longer imperative, which changes the translation from ( $0\_O$  to  $0.tA\_HE$ ). This will again be difficult to handle by semantic analysis, but is made easier if we use TAM markers.

## 6 Automatic Translation of TAM Markers

As indicated earlier, TAM markers can be translated by rule based, statistical or hybrid techniques. Theoretically, all these techniques can give good results. We have used a statistical or machine learning based technique.

### 6.1 Identifying TAM Markers

Though identifying TAM markers in the SL sentences is not the focus of the current work, one obvious method (which is already being used for machine translation) is through simple linguistic rules. In most cases it seems to work, provided that resources like dictionaries and morphological analyzer are available. However, for our experiments, we had the manually annotated markers in the corpus. We inserted them in the correct place in the MT system, so that we could see the results only for TAM marker translation, avoiding the errors in TAM marker identification.

### 6.2 Factors in TAM Marker Translation

The correct translation of an SL TAM marker depends on several factors. The preceding sections have already indicated them. In this section, we will take up all these factors and relate them to some possible solutions. The first item of information needed is, of course, the SL TAM marker. We assume that it is known since we are using TAM markup from the annotated corpus, rather than relying on the TAM marker computation module of the MT system, which can make mistakes (this is, of course, just for evaluation of TAM marker translation alone). A solution based only on the one-to-one TAM dictionary uses only this information. Three more factors are the distributions of SL and TL markers and their mappings in the corpus (or, ideally, in the language). A distributional similarity based solution, e.g. the IBM models [7] could take these factors into account. The most important factors for our proposed solution are the contexts in the SL sentence and in the TL output by the MT system (which we have to correct).

We have also tried to find the specific factors (or specific parts of the context) which determine this choice. One interesting example is given below:

**SL:** Who *wants*<sub>PRES</sub> to lose their jobs?.

**TL:** apanI nOkarI kOna khonA cAHegA<sub>gA</sub>?  
'one's job who lose will-want?'

**Mapping:**  $PRES_1 \rightarrow gA_1$

In this case (which is frequent in the corpus), the fact that the sentence is a question seems to determine that *PRES* will be translated as *gA* (future). Some other factors that seem to determine the choice of 'TAM sense' in the target language are the properties of the main verb, certain words (other than the verb), the type of the clause, etc. In fact, infinitives seem to have their own way of getting translated (see *to\_0* in Table-2).

The proposed CRF based solution tries to take into account all the factors mentioned in this section. However, so far we have evaluated only with the context in the SL sentence, not with context in TL output by the MT system. This point is elaborated more in the next section.

### 6.3 Features for CRF

For now, we are only using the context from the SL sentence for learning and evaluation. We experimented on four sets of features for CRF. These were:

- **F1:** SL TAM marker, *verb\_lex* (verb lexical item), *verb\_cat* (verb category), *verb\_lex-2* (word at a distance -2 from the current verb), *verb\_lex-1*, *verb\_cat-2*, *verb\_cat-1*
- **F2:** SL TAM marker, *verb\_lex*, *verb\_cat*, *verb\_lex-2/verb\_lex-1* (combination of *verb\_lex-2* and *verb\_lex-1*), *verb\_cat-2/verb\_cat-1*
- **F3:** SL TAM marker, *verb\_lex*, *verb\_cat*, *verb\_lex-2/verb\_lex-1*, *verb\_cat-2/verb\_cat-1*, *head\_lex-2/head\_lex-1* (combination of lexical items for the head of the previous two chunks), *head\_cat-2/head\_cat-1*
- **F4:** SL TAM marker, *verb\_lex*, *verb\_cat*, 0 or 1 (1 if there is a conjunct except 'and' in the sentence, otherwise 0), *verb\_lex-2/verb\_lex-1*, *verb\_cat-2/verb\_cat-1*, *head\_lex-2/head\_lex-1*, *head\_cat-2/head\_cat-1*

## 7 Evaluation

In this section, we first describe the experimental setup and the evaluation method used by us. Then we present the results obtained.

### 7.1 Evaluation Method

For evaluation, we first conducted experiments on the four feature sets mentioned earlier to select the one which is likely to give to the best performance with CRF. We calculated the precision of TAM marker translation in three cases with the best feature set (F3):

- **A:** TAM dictionary with one-to-one mappings (baseline)
- **B:** MT system with output corrected using CRF (first evaluator)
- **C:** MT system with output corrected using CRF (second evaluator)

To prepare our training set we take the intermediate output from the MT system (after the POS tagger and the chunker) to get the context features. To evaluate which feature set was best (to be used for final evaluation), we divided the subset of the corpus into two parts (3470 sentences with 3908 markers, 530 sentences with 616 markers) by randomly selecting sentences. By training on the bigger set and testing on



Feature Set	Precision
<b>F1</b>	50.75%
<b>F2</b>	48.87%
<b>F3</b>	51.70%
<b>F4</b>	51.51%

**Table 3:** Results for four feature sets used for classification by CRF

	Precision
<b>A. TAM Dictionary</b>	46.05%
<b>B. CRF-1</b>	63.63%
<b>C. CRF-2</b>	64.22%

**Table 4:** Improvement in precision for TAM marker translation

the smaller one, we selected the most promising feature set. For this step, we used the markers in the parallel corpus as reference for evaluation. The feature set F3 gave the best performance (51.70% precision) and we used it for evaluation on the MT system.

For final evaluation, we took a subset of short sentences (6-15 words) from the annotated corpus and run them through the MT system and we correct the TAM marker in the MT system output based on learning by CRF. We tested on 439 sentences. The lower limit on the size of the sentences was to avoid fragments which were not really complete sentences. The higher limit was fixed because the MT system was not always able to process long sentences. Note that the output of the MT system was required to extract the features for learning. This is why we could not use the longer sentences.

## 7.2 Results

The evaluation was performed by two different evaluators. One was a professional translator while other was from computational linguistics background. Precision was calculated for default translation using the TAM dictionary (the baseline) and on the MT system with CRF corrected output.

The evaluators checked the TL markers in the context of their being meaningful keeping both the SL sentence and the intermediated MT system output in mind. The precision for the baseline was 46.05%. Learning by CRF gave a precision of 63.93%, which was significantly better than the baseline.

## 8 Observations Based on the Results

Based on the results obtained by correcting the MT system output using the CRF based marker classification, we present some observations about the errors and some suggestions for improving the results further:

1. Many of the errors were in translating *PRES* (simple present). On examining the training and testing data we found that this was because its distribution in the training and testing data was highly

imbalanced, i.e., our testing data was very unfair for evaluating the translation of this marker.

2. Another reason for errors in translating *PRES* is that it is more ambiguous. There are more ways in which it can be translated and many of them are quite frequent.
3. Some of the errors in translating *PRES* can be taken care of by simple rules. For example, in reported speech, the correct translation is usually *tA.HE*.

## 9 Conclusions and Future Work

We described the parallel corpus annotated with TAM markers and their mappings and listed the most frequent of them. We discussed why TAM markers are important for MT and gave linguistic and empirical evidence for this. The problem was formulated as a classification task. The technique we used for machine learning was CRF. We tested for four sets of features and selected the best one. Using this best feature set, we experimented on improving TAM translation. We were able to get a precision of 63.93%, which was significantly better than the baseline, which used a TAM dictionary with a one-to-one mapping. Based on our observation of the output, we suggested some ways to further improve the results. Another task for the future to use the context from the TL output given by the MT system, because TAM marker translation depends on the structure selected by the MT system for the translated sentence.

## References

- [1] B. L. Bai and A. Mukherji, editors. *Tense and Aspect in Indian Languages*. Centre for Advanced Study in Linguistics, Osmania University, Hyderabad., 1993.
- [2] J. Bybee, R. Perkins, and W. Pagliuca. *The Evolution of Grammar: Tense, Aspect, and Modality in the Languages of the World*. University of Chicago Press., 1994.
- [3] B. J. Dorr and M. B. Olsen. Deriving verbal and compositional lexical aspect for nlp applications. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics., 1997.
- [4] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.
- [5] B. Levin. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago., 1993.
- [6] LTRC. A brief outline of shakti machine translation system, 2004. Language Technologies Research Centre, International Institute of Information Technology, Hyderabad, India, <http://ltrc.iit.ac.in/showfile.php?filename=projects/shakti.php>.
- [7] V. J. D. P. Peter F. Brown, Stephen A. Della Pietra and R. L. Mercer. Mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics*, pages 19(2):263–311, 1993.
- [8] H. Reichenbach. The tenses of verbs. In *Elements of Symbolic Logic*. The Macmillan Company, New York., 1947.
- [9] S. Saravagi. Crf project page, 2005. A java implementation of Conditional Random Fields for sequential labeling. <http://crf.sourceforge.net/>.
- [10] Z. Vendler. Verbs and times. In *Linguistics in Philosophy*. Cornell University Press., 1967.

# Evaluating Algorithms for the Generation of Referring Expressions: Going Beyond Toy Domains

Ielka van der Sluis and Albert Gatt and Kees van Deemter  
Department of Computing Science  
University of Aberdeen  
{ivdsluis, agatt, kvdeemte}@csd.abdn.ac.uk

## Abstract

We describe a corpus-based evaluation methodology, applied to a number of classic algorithms in the generation of referring expressions. Following up on earlier work involving very simple domains, this paper deals with the issues associated with domains that contain ‘real-life’ objects of some complexity. Results indicate that state of the art algorithms perform very differently when applied to a complex domain. Moreover, if a version of the Incremental Algorithm is used then it becomes of huge importance to select a good preference order, because some preference orders are prone to generating very unnatural output. Finding good preference orders, however, can be difficult, as we show. These results should contribute to a growing debate on the evaluation of NLG systems, arguing in favour of carefully constructed *balanced* and *semantically transparent* corpora.

## Keywords

generation of referring expressions, corpus-based evaluation of algorithms, Natural Language Generation

## 1 Introduction

This paper evaluates some classic algorithms for the Generation of Referring Expressions (GRE), which focus on the question of Content Determination. We ask how well these algorithms model the semantic content of expressions produced by people. It replicates the methodology used in [8], which carried out an evaluation using relatively simple domains of objects with well-defined properties. In addition to presenting new evaluation results on a novel, more complex domain, this paper poses a number of questions regarding the adequacy of existing GRE algorithms when they are deployed in scenarios involving complex objects. In contrast to ‘toy’ domains, such objects afford human authors with a much larger variety of referential possibilities, with potentially more inter-author variation. This has some consequences for existing GRE algorithms that rely on predefined general or domain-specific ‘preferences’ for content determination, whereby some properties of objects are prioritised over others. While psycholinguistic research has indeed shown that such preferences exist, the results have tended to rely on precisely the kinds of simple objects that characterise most proposals in the

GRE literature. Our aims in this paper are to (a) examine the feasibility of constructing a semantically-annotated corpus for GRE evaluation in complex scenarios; (b) evaluate the performance of current content determination heuristics for GRE on such scenarios; (c) compare this performance to our earlier results on more limited domains.

GRE is a semantically intensive sub-task of microplanning in NLG. GRE algorithms take as input a Knowledge Base (KB), which lists domain entities and their properties (attribute-value pairs), together with a set of intended referents,  $R$ . The output is a distinguishing description of  $R$ , that is, a logical form which distinguishes this set from its distractors. Most work in the area focuses on very simple objects, with attributes such as *colour* and *shape*. With complex real-world objects, the relevant properties are not always easy to ascertain. For instance, in describing a *person*, attributes such as *shape* become problematic, whereas *hair-colour*, *beard-colour*, *has-glasses* and *height* are not only more relevant, but also more numerous. Some properties (e.g. a person’s freckles) may only be used occasionally, or not at all. Because of more variation, GRE algorithms might be expected to perform worse on complex domains, compared to those where objects are simple and stylised. For an evaluation which compares output to the human gold standard represented by a corpus, another problem is the potential for lack of agreement between corpus annotators. This is especially non-trivial since *semantic and pragmatic transparency* are prerequisites for corpora in GRE as we have argued in [23]. Semantic transparency means that all the relevant knowledge available to the human authors of the corpus is known. Similarly, pragmatic transparency ensures that the authors’ communicative intentions are known. Ideally, the corpus should be balanced in both respects so that, for example, different kinds of referents occur an equal number of times.

This paper describes the construction of a corpus of this kind, involving a moderately complex domain whose inhabitants are (black & white photographs of) *people*. The resulting corpus is then used to compare some classic GRE algorithms with human descriptions. Wherever appropriate, we shall highlight the ways in which our experiences and findings differed from the ones involving a simpler domain [8].

## 2 Related work

The current state of the art in GRE is dominated the Incremental Algorithm (IA) of Dale and Reiter [5], which has served as a starting point for later models which sought to extend the expressiveness and coverage of GRE [10, 14, 15, 17, 22]. The IA was proposed as a better match to human referential behaviour relative to some predecessors, notably Dale’s [4] Full Brevity (FB) and Greedy (GR) heuristics, which emphasise *brevity* as the main determinant of adequacy. In contrast, the IA performs hillclimbing along a predetermined list of domain attributes. This *preference order* reflects general or domain-specific preferences, which is the main reason for the IA’s predicted superiority. However, the preference order strongly impacts the IA’s performance, since in a domain with  $n$  attributes, there are in principle  $n!$  different incremental algorithms.

Few empirical evaluations have been conducted in this area, and those that were done were limited to descriptions of objects that can be identified with only a few clearly distinguishable attributes like *colour* or *type*. [13] and [9] compared the IA to some alternative models, using the COCONUT dialogue corpus, where pieces of furniture are described with four attributes at most. [24] used a small corpus of descriptions of drawers, using *colour* and *location* attributes only. Apart from using simple domains, these studies meet the transparency requirements mentioned above to a very limited degree. Though COCONUT dialogues were elicited against a well-defined domain, [12] has emphasised that reference, in COCONUT, was often intended to satisfy intentions over and above identification. Thus, evaluating the IA against this data may not have done justice to a content determination strategy designed solely to achieve this aim. Furthermore, Gupta and Stent used an evaluation metric that included aspects of the syntactic structure of descriptions (specifically, modifier placement), thus arguably obscuring the role of content determination.

### 2.1 Computing Similarity

One question that the studies mentioned above raise relates to how human-authored and automatically generated descriptions should be compared. A measure of recall (as used in the Jordan/Walker and Viethen/Dale studies) indicates coverage, but does not measure the *degree* of similarity between a description generated by an algorithm and a description in the corpus, punishing all mismatches with equal severity. To obtain a more fine-grained measure, we use the Dice coefficient of similarity shown in (1). Let  $D_1$  and  $D_2$  be two descriptions, and let  $att(D)$  be the attributes in any description  $D$ . The coefficient takes into account the number of attributes that an algorithm omits in relation to the human gold standard, and those it includes, making it more optimally informative. Because descriptions could contain more than one instance of an attribute (e.g. ‘the young man with the glasses and the old man who also wears glasses’), the sets of attributes for this comparison were represented as multisets.

TYPE	HASBEARD	HASGLASSES	AGE
man	1	0	old
man	1	1	young
man	0	1	old
man	0	0	young

**Table 1:** Attributes and example targets as defined in the corpus domains

$$dice(D_1, D_2) = \frac{2 \times |att(D_1) \cap att(D_2)|}{|att(D_1)| + |att(D_2)|} \quad (1)$$

## 3 A transparent corpus of references

We constructed and annotated a balanced corpus that pairs each description in the corpus with a logical form that is cast in terms of the domain with respect to which the description was produced. Our corpus contains ca. 1800 descriptions, collected through a controlled experiment run over the web. Participants in the experiment were asked to identify one or two objects from a set of distractors shown on their computer screen, by typing distinguishing descriptions as though they were interacting remotely with another person. One within-subjects variable was the use of different domains: (1) artificially constructed pictures of household items and (2) real photographs of people, yielding two sub-corpora. In this paper, we discuss how the latter sub-corpus is gathered, annotated and used to evaluate various GRE algorithms. Throughout the paper, we compare with our findings on the furniture corpus [8].

### 3.1 Materials and design

The people sub-corpus consists of 810 descriptions from 45 native or fluent speakers of English. Participants described photographs of men in 18 trials, each corresponding to a domain where there were one or two clearly marked target referents and six distractors (also men), placed in a 3 (row)  $\times$  5 (column) grid. The use of these pictures was based on previous experimental work using the same set [19].

In addition to their location (on which more below), all targets could be distinguished via the three attributes shown in Table 1. Thus, the targets differed from their distractors in whether they had a beard (HASBEARD), wore glasses (HASGLASSES) and/or were young or old (AGE). The corpus is semantically balanced, in that for each possible combination of the attributes, there was an equal number of domains in which an identifying description of the target(s) required the use of those attributes (modulo other possibilities). We refer to this as the *minimal description* (MD) of the target set. However, results of earlier studies with the same set of photographed persons indicated that speakers use other attributes to identify the photographed people as well (e.g. whether the person wears a tie, a suit or has a certain hairstyle or colour). These too were included in the corpus annotation, for a total of 9 attributes per photograph.

By contrast, objects in the furniture sub-corpus were invariably described using at most four attributes.

The present study focusses on the subset of the corpus descriptions which do *not* contain locative expressions ( $N = 342$  from 19 authors)<sup>1</sup>. For comparison, we use the subset of the household/furniture sub-corpus which also does not contain locatives ( $N = 444$  descriptions from 27 authors). Comparing the furniture and people descriptions, the variation amongst the people descriptions is expected to be higher and the annotation of the people descriptions is expected to be more difficult.

The experiment manipulated another within-subjects variable in addition to the domain, namely Cardinality/Similarity (3 levels):

1. **Singular** (SG): 6 domains contained a single target referent.
2. **Plural/Similar** (PS): 6 domains had two referents, which had identical values on the MD attributes. For example, both targets might be wearing glasses in a domain where HASGLASSES='1' sufficed for a distinguishing description.
3. **Plural/Dissimilar** (PD): 6 Plural trials, in which the targets had different values of the minimally distinguishing attributes.

Plural referents were taken into account because plurality is pervasive in NL discourse. The literature suggests that they can be treated adequately by minor variations of the classic GRE algorithms ([7, 11]), as long as the descriptions in question refer distributively [20]. This is something we considered worth testing.

### 3.2 Corpus annotation

To make the corpus semantically transparent, we designed a XML annotation scheme [18] that pairs each corpus description with a representation of the domain in which the description was produced (see Figure 1(a)). In order to match the descriptions produced by the participants in the study with the domain representations, the entities in the people domain are represented with 9 attribute tags in total. Six of them, HASGLASSES, HASBEARD, HASHAIR, HASSHIRT, HASTIE, HASUIT have a boolean value. The other four attributes have nominal values: the attribute TYPE has values **person** or **other**, the attribute AGE has value **old** or **young**, HAIRCOLOUR has values **dark**, **light** or **other**, and finally ORIENTATION, which captures the gaze direction of a photographed man, has three possible values **frontward**, **leftward** or **rightward**. If a part of a description could not be resolved against the domain representation, it was enclosed in an OTHER attribute tag with the value **other** for **name**. This was necessary in 62 descriptions (18.2%), a figure which is much larger than that obtained in the simpler furniture domain, in which only 3.3% of descriptions contain OTHER tags.

Figure 1(b) shows the annotation of a plural description in the people domain. ATTRIBUTE tags enclose segments of a description corresponding to

properties, with **name** and **value** attributes which constitute a semantic representation compatible with the domain, abstracting away from lexical variation. For example, in Figure 1(b), the expression *with black facial hair* is tagged as HASBEARD, with the value 1. Note that HASBEARD encloses the HAIRCOLOUR tag used for *black*. The DESCRIPTION tag in Figure 1(b), permits the automatic compilation of a logical form from a human-authored description. Figure 1(b) is a plural description enclosing two singular ones. Correspondingly, the logical form of each embedded description is a conjunction of attributes, while the two sibling descriptions are disjoined, as shown in (2).<sup>2</sup>

$$\begin{aligned}
 & ([Age: old] \wedge [type: person] \wedge \\
 & [Orientation: frontward]) \vee ([hasBeard: 1] \wedge \\
 & [hairColour: dark] \wedge [type: person])
 \end{aligned} \tag{2}$$

### 3.3 Annotator reliability

The reliability of the corpus annotation scheme was evaluated in a study involving two independent annotators (hereafter A and B), both postgraduate students with an interest in NLG, who used the same annotation manual [18]. They were given a stratified random sample of 540 target descriptions consisting of 270 descriptions from each domain. For both the furniture and the people domain they were given 2 descriptions from each Cardinality/Similarity condition, from each author in the corpus. To estimate inter-annotator agreement, we compared annotations of A and B against those by the present authors, using the Dice coefficient described above. We believe that Dice is more appropriate than agreement measures (such as the  $\kappa$  statistic) which rely on predefined categories in which discrete events can be classified. The 'events' in the corpus are NL expressions, each of which is 'classified' in several ways (depending on how many attributes a description expresses), and it was up to an annotator's judgment, given the instructions, to select those segments and mark them up.

Inter-annotator agreement was high in both sub-corpora, as indicated by the mean and modal (most frequent) scores. In the furniture domain, both A and B achieved similar agreement scores with the present authors (A: mean = .93, mode = 1 (74.4%); B: mean = .92; mode = 1 (73%)). They also evinced substantial agreement among themselves (mean = .89, mode = 1 (71.1%)). In the people domain A's annotations were in slightly better agreement with our annotations than B's (A: mean = .84, mode = 1 (41.1%); B: mean = .78; mode = 1 (36.3%)). The annotators had a somewhat higher agreement among themselves than with the annotations of the present authors in the people domain (mean = .89, mode = 1 (70%)).

Overall, these results suggest that the annotation scheme used is replicable to a high degree. As expected however, these results also indicate that annotating complex object descriptions is more difficult than ones elicited in simple domains.

<sup>1</sup> Location was manipulated as a between-subjects factor. Participants were randomly placed in groups which varied in whether they could use location or not.

<sup>2</sup> In the phrases of interest, disjunction or set union is the semantic correlate of the use of *and* in a plural description.

```

<ENTITY type='target'>
  <ATTRIBUTE name='type' value='person' />
  <ATTRIBUTE name='age' value='old' />
  <ATTRIBUTE name='hasBeard' value='0' />
  <ATTRIBUTE name='hasGlasses' value='0' />
  <ATTRIBUTE name='orientation'
value='frontward' />
  ...
</ENTITY>
<ENTITY type='target'>
  <ATTRIBUTE name='type' value='person' />
  <ATTRIBUTE name='age' value='young' />
  <ATTRIBUTE name='orientation'
value='frontward' />
  <ATTRIBUTE name='hasBeard' value='1' />
  <ATTRIBUTE name='hasGlasses' value='0' />
  <ATTRIBUTE name='orientation'
value='frontward' />
  ...
</ENTITY>

```

(a) Fragment of a domain

```

<DESCRIPTION num='plural'>
  <DESCRIPTION num='singular'>
    <ATTRIBUTE name='Age' value='old'>elderly
    </ATTRIBUTE>
  <ATTRIBUTE name='type' value='person'>man
  </ATTRIBUTE>
  <ATTRIBUTE name='orientation' value=
'frontward'>who is facing the front
  </ATTRIBUTE>
</DESCRIPTION>
and
<DESCRIPTION num='singular'>
  <ATTRIBUTE name='type' value='person'>man
  </ATTRIBUTE>
  <ATTRIBUTE name='hasBeard' value='1'>with
  <ATTRIBUTE name='hairColour' value='dark'>
    black</ATTRIBUTE>
  facial hair</ATTRIBUTE>
</DESCRIPTION>
</DESCRIPTION>

```

(b) Example Description

Fig. 1: Annotation example: 'elderly man who is facing the front and man with black facial hair'

## 4 Evaluating the algorithms

We evaluated the three algorithms introduced earlier, all of which can be characterised as search problems [3]:

1. **Full Brevity (FB)**: Finds the smallest distinguishing combination of properties.
2. **Greedy (GR)**: Adds properties to a description, always selecting the property with the greatest discriminatory power.
3. **Incremental (IA)**: Performs gradient descent along a predefined list of properties. Like GR, IA incrementally adds properties to a description until it is distinguishing.

The performance of these algorithms was tested with respect to 342 descriptions in the 'people' corpus. Among other things, they were compared to a baseline (RAND), which randomly added properties true of the referent(s) to the description until it was distinguishing. Because the IA always adds TYPE [5], the same trick was applied to all algorithms, to level the playing field.<sup>3</sup>

In addition, we extended the algorithms to cover the plural descriptions in the people corpus, using the algorithm of [21]. This algorithm first searches for a distinguishing description through literals in the KB, failing which, it searches through disjunctions of increasing length until a distinguishing description is found. This approach was applied to FB and GR as well as the different versions of IA.

We had several general expectations regarding this evaluation. In particular, we expected all algorithms to perform worse with respect to the people descriptions than with respect to the furniture descriptions,

simply because the larger number of attributes means that there is more room for error. Before we can delve more deeply into these matters, we need to ask what we mean when we speak about the IA, given that this search method gives rise to different algorithms depending on the way in which attributes are ordered.

### 4.1 Preference orders for the IA

In simple situations, such as the furniture domain in this corpus, which contained 3 attributes (apart from LOCATION), the number of ways in which attributes can be grouped into a preference order for the IA was limited [8]. Psycholinguistic evidence also facilitates the task. For instance, it is known that attributes such as COLOUR tend to be included in descriptions even when they are not required [16, 6, 1], while relative attributes requiring comparison to other objects (such as SIZE), are cognitively more costly and more likely to be omitted [2]. In a more complex domain, such as the people domain in this corpus, the larger number of attributes increases the possible number of preference orders, and testing them all is unfeasible. Moreover, many of these attributes will not have been studied in the psycholinguistics literature. Let us see how these issues pan out in the (only moderately complex!) people domain.

Although the experimental trials on which the people corpus is based were composed in such a way that the targets could be distinguished with a combination of the attributes HASBEARD, HASGLASSES and AGE, the descriptions contain many other attributes. With the 9 attributes (excluding TYPE) that were needed to annotate the bulk of descriptions in the people corpus, there are as many as  $9! = 362880$  possible preference orders.<sup>4</sup> How might one narrow down 362880 prefer-

<sup>3</sup> In the corpus 91% of descriptions in the people domain and 93.5% in the furniture domain contain a TYPE.

<sup>4</sup> For the algorithm evaluation we exclude the OTHER tag, because it represents a variety of unregistered properties.

	mean (SD)	sum
<b>type</b>	1.39 (.64)	475
<b>hasGlasses</b>	.68 (.78)	231
<b>hasBeard</b>	.66 (.56)	226
<b>hairColour</b>	.61 (.54)	210
<b>hasHair</b>	.46 (.62)	158
<b>orientation</b>	.21 (.48)	73
<b>age</b>	.10 (.36)	34
<b>hasTie</b>	.04 (.18)	12
<b>hasSuit</b>	.01 (.11)	4
<b>hasShirt</b>	.01 (.09)	3

**Table 2:** Means, Standard Deviations (SD), and Sum frequencies of attribute usage in the people domain.

ence orders to a manageable number? This is evidently an art rather than a science, but it will be instructive to see how one might reason, and how successful or unsuccessful this type of reasoning can be.

Having built the corpus, the natural approach is perhaps to count the frequencies of occurrence of each of the attributes. Table 2 shows that HASGLASSES (G), HASBEARD (B), HAIRCOLOUR (C), and HASHAIR (H), are relatively likely to be included in a description. Arguably, a person’s age is an attribute that needs comparison (e.g. with the ages of the distractors), so one might assume that AGE (A) is less preferred than HASGLASSES and HASBEARD.

In the corpus annotation, HAIRCOLOUR can only appear when the HASHAIR or the HASBEARD tag is included in the description (see Section 3). Accordingly, one can reasonably restrict the number of possible preference orders by the constraint that HAIRCOLOUR, can only be positioned in the preference order when preceded by HASHAIR and HASBEARD. The following 8 IAs are tested: IA-GBHC, IA-GHBC, IA-HBGC, IA-HBCG, IA-HGBC, IA-BHGC and IA-BGHC and the 3 algorithms that perform best are presented in the next section. In addition the IA with the worst of all preference orders, IA-WORST, was tested as a baseline case. The preference order used by this algorithm lists the attributes in reverse-frequency order (e.g. HASSHIRT > HASSUIT > HASTIE > AGE > ORIENTATION > HASHAIR > HASBEARD > HAIRCOLOUR > HASGLASSES).

## 4.2 Differences between algorithms

As indicators of the performance of algorithms, we use mean and modal (most frequent) scores, as well as the *perfect recall percentage* (PRP: the proportion of Dice scores of 1). Pairwise t-tests are used to compare the average Dice scores of each algorithm to RAND and to GR. These comparisons are reported using subjects ( $t_S$ ) and items ( $t_I$ ) as sources of variance.

Table 3 displays scores averaged over all three Cardinality/Similarity conditions; we return to the differences between these below. It shows the results of the three best IAs (IA-GBHC, IA-GHBC and IA-BGHC) from the eight IAs that were tested on the people descriptions. Also shown are the results of the IA with the worst preference order IA-WORST as well as the performance of the FB, the GR on the same descriptions. To enable a comparison with the evaluation of algorithms tested on the furniture descriptions the results for GR and for the version of the IA that performed best in

this domain are included in the table as well. The latter algorithm, IA-COS, is a version using a preference order consisting of three attributes (COLOUR > ORIENTATION > SIZE).

**Results.** All eight IA variations that were evaluated with the people corpus perform significantly better than RAND. This baseline achieved a mean Dice score of .47 (SD= .24; PRP=2.6%; Mode= .33). Of the three best IAs shown in Table 3, the algorithm with the highest mean, IA-GBHC, has a modal score of 1 in 21.3% of the cases. (This is also achieved by IA-GHBC.) A pairwise t-test tells us that the IA-GBHC algorithm performs significantly better than IA-GHBC, though its average dice score is only better by subjects ( $t_S = 10.720$ ,  $p = .01$ ;  $t_I = 1.678$ , *ns*). These figures suggest that even when only the first four attributes in the preference order are varied, differences in performance are already noteworthy. The IA with the worst preference order performs very badly, and much worse than any other algorithm that was considered. Its mean Dice score is .33 and the best match it receives with the descriptions in the corpus is .75, which happened for only one description (thus, its PRP was 0).

The by-subject and by-item analysis for the GR algorithm presented in Table 4 shows that FB performed slightly worse than GR, but only by subjects ( $t_S = -4.147$ ,  $p = .01$ ). Interestingly, GR also matches the people descriptions better than some of the IAs that were tested. This is most obviously true for IA-WORST, but also for IA-BHCG and IA-HGCB (two of the eight IA algorithms that were tested, but whose values are not shown in Table 3). For instance, IA-BHCG (mean= .60; SD= .21) was significantly worse by subjects than GR ( $t_S = 3.187$ ,  $p = .01$ ;  $t_I = 1.159$ , *ns*). On the other hand, the average dice scores of GR are significantly lower than the IA that performed best in our analysis, IA-GBHC ( $t_S = -3.332$ ,  $p = .01$ ;  $t_I = -3.236$ ,  $p = .01$ ). These results indicate a very substantial impact of preference orders, which offers a note of caution: in practice, identifying a preference order is not always trivial, and minor variations in attribute orderings can have a significant impact.

Although in the people domain there exists a particular IA algorithm that performs better than the GR algorithm, our findings suggest strongly that only a few of the 362880 IA algorithms render better results than GR. So even though the relative discriminatory power of a property (as used by GR) or the overall brevity of a description (as used by FB) may not exactly reflect human tendencies, these factors are certainly worth considering when one has difficulties in determining a preference order in complex domains like this one. When confronted with a new and ‘complex’ domain, in which attribute preferences are unknown, a properly modified GR algorithm is a better choice than an arbitrary IA.

Turning to a comparison of furniture and people domains, focusing on the best IAs, their mean scores seem to differ substantially, with IA-COS obtaining .83 on furniture descriptions, compared to .69 obtained by IA-GBHC on the people corpus. Nevertheless, the PRP scores tell a different story: 24.1% on 444 furniture descriptions against 21.3% on 342 people descriptions

	PEOPLE						FURNITURE	
	IA-GBHC	IA-GHBC	IA-BGHC	IA-WORST	FB	GR	IA-COS	GR
<b>Mean (SD)</b>	.69 (.23)	.66 (.25)	.68 (.22)	.33 (.13)	.60 (.27)	.64 (.24)	.83 (.13)	.79 (.16)
<b>Mode</b>	1.00	1.00	.67	.29	1.00	1.00 & .67	1.00	.8
<b>PRP</b>	21.3	21.3	17.3	0.0	19.6	19.3	24.1	18.7
compared to RAND $t_S$	12.080	12.747	14.737	-12.967	8.397	9.724	7.002	3.333
compared to RAND $t_I$	8.794	5.642	7.026	-12.254	3.371	5.227	4.632	1.169
compared to GR $t_S$	-3.332	-3.332	-4.385	15.034	-4.147	-	2.972	-
compared to GR $t_I$	-1.310	1.310*	1.582*	10.007	-1.678*	-	2.117*	-

**Table 3:** Scores for the three best IAs, IA-WORST, FB and GR in the people domain. Related figures for IA and GR in the furniture domain are included for comparison. Values of  $t$ -tests by subjects ( $t_S$ ) and items ( $t_I$ ) compare each to the Random Baseline RAND and to GR (\* $p$  = not significant, otherwise  $p \leq .01$ ).

	SINGULARS		SIMILAR PLURALS		DISSIMILAR PLURALS	
	IA-GBHC	IA-COS	IA-GBHC	IA-COS	IA-GBHC	IA-COS
<b>Mean (SD)</b>	.78 (.19)	.92 (.12)	.77 (.22)	.80 (.11)	.51 (.15)	.79 (.13)
<b>Mode</b>	1.00	1	1.00	.8	1.00	.8
<b>PRP</b>	21.3	60.8	21.3	7	21.3	.8

**Table 4:** Scores the algorithms as a function of Cardinality/Similarity.

seem fairly comparable.

One explanation of the overall worse performance of the algorithms on the people domain, which was hypothesised in Section 1, is that there is greater scope for inter-author variation in more complex domains, and perhaps also greater scope for variation within the descriptions produced by the same author. As an approximate indicator of this, we computed the average number of attributes that descriptions in the two domains had. This was clearly higher in the people domain (3.64) than in the furniture domain (2.02). More important than a measure of central tendency however, is the variance. At 2.24, variance in the number of attributes across descriptions of people was substantial, compared to a mere .66 in furniture. This largely confirms our expectations, as well as offering an explanation for some of the different results obtained in the two sub-corpora.

The final part of our analysis concerns the relative performance of the algorithms on singular and plural descriptions. Table 4 displays scores for the best-performing IAs in the furniture and in the people domain as a function of the Cardinality/Similarity variable. Results in the people domain suggest that the algorithm performs approximately equally well in the ‘singular’ and ‘plural similar’ conditions. Pairwise comparisons showed no significant difference between these two conditions. The difference between ‘singulars’ and ‘dissimilar plurals’ was substantial ( $t_S = -14.784$ ,  $p = .01$ ;  $t_I = -8.250$ ,  $p = .01$ ). The same was true of ‘similar’ and ‘dissimilar’ plurals ( $t_S = -10.773$ ,  $p = .01$ ;  $t_I = -8.701$ ,  $p = .01$ ). One reason for the worse performance on the ‘dissimilar’ condition is that here, algorithms needed to use disjunction. Under the generalisation of the IA by [21], this involves searching through disjointed combinations of increasing length, a process which obscures the notion of preference incorporated in the preference order.

A similar analysis by [8] on the different Cardinality/Similarity conditions in the furniture corpus showed a somewhat different picture. All algorithms tested in that paper performed better on singular descriptions, but the difference between ‘similar’ and

‘dissimilar’ plurals was not as dramatic. One of the reasons for this has to do with TYPE. In the people domain, all entities had the same value of this attribute (*man*). This means that authors avoided coordination (semantic disjunction) in the ‘plural similar’ domains, producing descriptions such as *the men with the beard*. In the furniture domains, referents in ‘plural similar’ domains had different basic-level values of TYPE, and authors were more likely to use disjunction, with descriptions such as *the red table and the red chair*. This interpretation suggests that the basic problem encountered by all algorithms in both domains was with disjunction (which had to be used in the similar cases for furniture descriptions, because of the different values of TYPE).

## 5 Conclusions

Our study of the *people* domain has significantly reinforced a number of conclusions that we were only able to formulate tentatively when studying the simpler *furniture* domain. In particular:

- As in the furniture domain, the ‘best’ IA outperformed all other algorithms, but unlike the furniture domain, the ‘worst’ IA was significantly worse than FB and GR.
- The best IA in the furniture domain performed much better than the best IA in the people domain, although the PRP scores of these algorithms were similar.
- The total number of preference orders for IA was much larger in the people domain than in the furniture domain, and it proved difficult to find efficient ways of zooming in on preference orders that perform well.
- The complexity of the people domain makes itself felt with particular force in the algorithmic performance on dissimilar plurals.

Reflecting on these results, one might argue that the Incremental Algorithm (IA) is not suitably named. IA

is not really an *algorithm* but a strategy that can be used by a variety of algorithms and only becomes concrete when a preference order is selected. We showed that, in complex domains, different IAs can perform very differently, so that it is important to distinguish between them and ask which one suits a particular domain and genre best.

What are the practical implications of these results for designing NLG systems to be deployed in novel scenarios? The results of [8] suggested that selecting a preference order matters considerably, even in simple domains. The present work shows that these differences become huge when descriptions of more complex objects are considered. Moreover, psycholinguistic principles are of limited help in selecting a manageable subset of ‘promising’ preference orders. On the positive side, our results indicate that information about the frequency of occurrence of each attribute in a corpus *can* help. One might, of course, ask how useful this finding is for someone who has not studied the domain/genre before. Such a person, after all, does not possess the corpus to compute the frequencies of attributes. One might hope, however, that a quicker, less controlled experiment would give frequency information that could be used to similar effect, but this is a question for future research.

We have sometimes described the ‘people’ domain that was studied in this paper as if it were complex. But even though the objects in the domain are messier and more complex than the ones that have figured in most previous studies, calling this domain complex is arguably an overstatement. For example, the domain contains only a limited number of people, and nothing else than people, and that *relations* between people were not even taken into account. One wonders how reasonable preference orders might be chosen in any truly complex domain, how a controlled experiment could be set up in such a domain, or how a workable annotation scheme could be devised for gaining information about speakers’ behaviour in such situations. It seems likely to assume that the problems revealed by our study will be even greater in such domains.

## References

- [1] A. Arts. *Overspecification in Instructive Texts*. PhD thesis, University of Tilburg, 2004.
- [2] E. Belke and A. Meyer. Tracking the time course of multidimensional stimulus discrimination. *European Journal of Cognitive Psychology*, 14(2):237–266, 2002.
- [3] B. Bohnet and R. Dale. Viewing referring expression generation as search. In *Proc. IJCAI-05*, 2005.
- [4] R. Dale. Cooking up referring expressions. In *Proc. ACL-89*, 1989.
- [5] R. Dale and E. Reiter. Computational interpretation of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(8):233–263, 1995.
- [6] H. J. Eikmeyer and E. Ahlsèn. The cognitive process of referring to an object: A comparative study of german and swedish. In *Proc. 16th Scandinavian Conference on Linguistics*, 1996.
- [7] C. Gardent. Generating minimal definite descriptions. In *Proc. ACL-02*, 2002.
- [8] A. Gatt, I. van der Sluis, and K. van Deemter. Evaluating algorithms for the generation of referring expressions using a balanced corpus. In *Proc. ENLG-2007*, 2007.
- [9] S. Gupta and A. J. Stent. Automatic evaluation of referring expression generation using corpora. In *Proc. 1st Workshop on Using Corpora in NLG*, 2005.
- [10] H. Horacek. An algorithm for generating referential descriptions with flexible interfaces. In *Proc. ACL-97*, 1997.
- [11] H. Horacek. On referring to sets of objects naturally. In *Proc. INLG-04*, 2004.
- [12] P. W. Jordan. Influences on attribute selection in redescription: A corpus study. In *Proc. CogSci-00*, 2000.
- [13] P. W. Jordan and M. Walker. Learning content selection rules for generating object descriptions in dialogue. *Journal of Artificial Intelligence Research*, 24:157–194, 2005.
- [14] J. D. Kelleher and G.-J. Kruijff. Incremental generation of spatial referring expressions in situated dialog. In *Proc. ACL-COLING-06*, 2006.
- [15] E. Krahrmer and M. Theune. Efficient context-sensitive generation of referring expressions. In K. van Deemter and R. Kibble, editors, *Information Sharing*. Stanford: CSLI, 2002.
- [16] T. Pechmann. Incremental speech production and referential overspecification. *Linguistics*, 27:89–110, 1989.
- [17] A. Siddharthan and A. Copestake. Generating referring expressions in open domains. In *Proc. ACL-04*, 2004.
- [18] I. van der Sluis, A. Gatt, and K. van Deemter. Manual for the TUNA corpus: Referring expressions in two domains. Technical report, University of Aberdeen, 2006.
- [19] I. van der Sluis and E. Krahrmer. The influence of target size and distance on the production of speech and gesture in multimodal referring expressions. In *Proc. ICSLP’04*, Jeju, Korea, 2004.
- [20] M. Stone. On identifying sets. In *Proc. INLG-00*, 2000.
- [21] K. van Deemter. Generating referring expressions: Boolean extensions of the incremental algorithm. *Computational Linguistics*, 28(1):37–52, 2002.
- [22] K. van Deemter. Generating referring expressions that contain gradable properties. *Computational Linguistics*, 2006. to appear.
- [23] K. van Deemter, I. van der Sluis, and A. Gatt. Building a semantically transparent corpus for the generation of referring expressions. In *Proc. INLG-06 (Special Session on Data Sharing and Evaluation)*, 2006.
- [24] J. Viethen and R. Dale. Algorithms for generating referring expressions: Do they do what people do? In *Proc. INLG-06*, 2006.



# The complexity of linguistically motivated extensions of tree-adjoining grammar

Anders Søgaard  
University of Copenhagen  
*anders@cst.dk*

Timm Lichte  
University of Tübingen  
*timm.lichte@uni-tuebingen.de*

Wolfgang Maier  
University of Tübingen  
*wo.maier@uni-tuebingen.de*

## Abstract

This paper proves the NP-hardness of three extensions of tree-adjoining grammar (TAG): FO-TAG [2], RSN-MCTAG [6] and TT-MCTAG [7]. The complexities of these extensions have all been presented as open problems in the literature. The extensions have been proposed to model scrambling and free word order phenomena in languages such as German, Korean and Japanese. It is shown that one of them also generates the MIX language. Finally, some polynomial time fragments are defined.

## 1 Introduction

Natural language has been shown to contain constructions which can not be adequately represented using context-free grammar (CFG), such as cross-serial dependencies. While first shown to exist in Swiss German [12], they can also be found in Tagalog [8]. Several formalisms have been introduced that provide more expressive power than CFG while staying computationally tractable, i.e. while retaining polynomial recognition. One of these formalisms is tree-adjoining grammar (TAG). Some knowledge of TAG is assumed in this paper. Consult [5] otherwise for a nice introduction. Informally, a TAG consists of finite sets of terminals and nonterminals and finite sets of initial and auxiliary trees. Larger trees are derived by substituting  $\downarrow$ -marked frontier nodes with trees or by adjoining trees (that have a root node and a  $*$ -marked frontier node with the same nonterminal) to interior nodes. The language of a TAG is the set of strings that are in the yield of trees that can be derived from an initial tree.

[2] showed that TAG does not have the expressive power necessary to capture scrambling and free word order phenomena in languages such as German, Korean and Japanese. Here's an example from Korean:

- (1) *jatongcha-lul keu-ka surihakess-tako*  
car.DEF.ACC PRO.3SG.NOM repair.INF  
*yakosokhaessta*  
promise.FIN  
'He promises to repair the car.'

In Korean, adjuncts and arguments scramble. In this case, an argument of the lower clause even appears in the upper clause. The structure is thus discontinuous. Scrambling over clause boundaries is sometimes referred to as long-distance scrambling. Such long-distance scrambling is beyond the expressive power of TAG, but even scrambling phenomena within the clause receive rather unelegant analyses in TAG.

ID/LP grammar was proposed by [11] for scrambling and free word order within the clause. In ID/LP grammar, the productions of CFG are split into immediate dominance (ID) and linear precedence (LP). For instance, the production  $S \rightarrow \alpha\beta$  is split into the (un-ordered) ID  $S \rightarrow \alpha\beta$  and the LP  $\alpha \prec \beta$ . The LPs can be relaxed or removed. In other words, free word order and scrambling do not complicate grammars, but simplify them. ID/LP grammar still only generates context-free languages, but in fact the universal recognition problem becomes NP-hard. This was proven for the fragment of ID/LP grammar with no LPs (UCFG) in [1] by an interesting application of the vertex cover problem; this problem is also used here to establish the NP-hardness of FO-TAG, RSN-MCTAG and TT-MCTAG.

The vertex cover problem involves finding the smallest set  $V'$  of vertices in a graph  $D = \langle V, E \rangle$  such that every edge has at least one endpoint in the set. Formally,  $V' \subseteq V : \forall \{a, b\} \in E : a \in V' \vee b \in V'$ . The problem is thus an optimization problem, formulated as a decision problem:

- INSTANCE: A graph  $D = \langle V, E \rangle$  and a positive integer  $k$ .  
QUESTION: Is there a vertex cover of size  $k$  or less for  $G$ ?

Say  $k = 2, V = \{a, b, c, d\}, E = \{\langle a, c \rangle, \langle b, c \rangle, \langle b, d \rangle, \langle c, d \rangle\}$ ; for instance. One way to obtain a vertex cover is to go through the edges and underline one endpoint of each edge. If you can do that and only underline two vertex symbols, a vertex cover has been found. Since  $|V| = 4$ , this is equivalent to leaving two vertex symbols untouched. Consequently, the vertex cover problem for this specific instance is encoded by this UCFG, where  $\delta$  is a bookkeeping dummy symbol:

$S$	$\rightarrow$	$\rho_1\rho_2\rho_3\rho_4UU\delta\delta\delta\delta$
$\rho_1$	$\rightarrow$	$a c$
$\rho_2$	$\rightarrow$	$b c$
$\rho_3$	$\rightarrow$	$b d$
$\rho_4$	$\rightarrow$	$c d$
$U$	$\rightarrow$	$aaaa bbbb cccc dddd$
$\delta$	$\rightarrow$	$a b c d$

$\rho_i$  captures the  $i$ th edge in  $E$ . The input string  $\omega = aaaabbbbccccdddd$ . One derivation tree in our example will have the form:

$$[[aaaa]_U [bbbb]_U [c]_{(b,c)} [c]_{(a,c)} [c]_{(c,d)} [c]_\delta [d]_{(b,d)} [d]_\delta [d]_\delta [d]_\delta]_S.$$

Generally, the first production has as many  $\rho_i$ 's as there are edges in the graph,  $|V| - k$  many  $U$ 's and  $|E| \times |V| - |E| - |E| \times (|V| - k)$  many  $\delta$ 's, i.e. the length of the string minus the number of edges and the extension of  $|V| - k$  many  $U$ 's. The  $\rho_i$  productions are simple,  $U$  extends into  $|E|$  many  $a$ 's or  $b$ 's and so on, and  $\delta$  extends into all possible vertices. Since the grammar and input string can be constructed in polynomial time from an underlying vertex cover problem  $\langle k, V, E \rangle$ , universal recognition of UCFG must be at least as hard as solving the vertex cover problem. Since the vertex cover problem is NP-hard [4], the universal recognition problem for totally unordered type 2 grammars is therefore NP-hard. It is easy to see that it is also in NP. Simply guess a derivation, linear in the size of the string, and evaluate it in polynomial time.

Several extensions, as already mentioned, have been proposed for long-distance scrambling. See [6] for a partial survey. Most proposals in one way or another relax the notion of immediate dominance between mothers and daughters in trees. Note that immediate dominance is already relaxed in TAG, since new trees can be adjoined to daughter nodes.

FO-TAG is probably the simplest proposal, as it is very similar to ID/LP grammar. In variants of multicomponent TAG (MCTAG), the relaxation of immediate dominance is obtained by splitting auxiliary trees into smaller trees and dominance (not immediate dominance) links. The grammar then consists of sets of trees rather than just elementary trees, except for the initial trees. It is usually a restriction that every auxiliary tree in a set must be applied to the derived tree, in a single derivation step. In the absence of any other restrictions, the fixed recognition problem of (even lexicalized) MCTAG can be shown to be NP-hard [10, 3]. [3] proves the NP-hardness of the fixed recognition problem of a particular grammar that solves all instances of the three-partition problem by accepting only some input.

Several variants of MCTAG have appeared in the last years, and their complexities have been presented as open problems. In this paper, our concern is with FO-TAG [2], RSN-MCTAG [6] and TT-MCTAG [7]. It seems that none of these extensions of TAG are able to easily reconstruct the three-partition problem, but they all have the power to solve the vertex cover problem. Or more accurately, for every instance of the vertex cover problem, there is a polynomial (and linear) translation into a grammar of one of these kinds and a string such that the string is only recognized

by the grammar iff the source problem instance has a solution. Since the vertex cover problem is known to be NP-complete, the universal recognition problems of the three extensions are thus NP-hard. It is trivial to show that the extensions are also NP-complete.

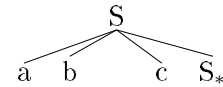
## 2 FO-TAG

Free order TAG (FO-TAG) was introduced in [2]. The definition is presented in Definition 2.1.

**Definition 2.1.**  $G$  is a free order tree-adjoining grammar iff  $G = \langle N, T, I, A, S \rangle$  such that  $G' = \langle N, T, I, A, S \rangle$  is a tree-adjoining grammar [5], except that initial and auxiliary trees are now tuples of unordered trees and LPs of the form  $\alpha < \beta$  where  $\alpha, \beta \in N$ .

The language of a FO-TAG is, as in the case of TAG, the set of strings that are in the yield of the trees that can be derived from an  $S$ -rooted initial tree by adjunction and substitution.

**Example 2.2.** FO-TAG does not seem to generate the MIX language, but the totally unordered extension of it does, i.e. if a language  $L$  is generated by a FO-TAG, the language generated by its totally unordered extension is the set of all permutations of strings in  $L$ . See [13] for the notion of total unordering. The MIX language is conjectured not to be mildly context sensitive. It consists of all strings in  $\{abc\}^*$ ; that is, every string that consists of the same number of  $a$ 's,  $b$ 's and  $c$ 's. To see this, consider the auxiliary tree:

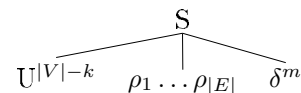


This generates the language whose permutations is the MIX language with an appropriate initial tree to begin the derivation and an appropriate auxiliary tree to end it.

For our NP-hardness proof, it is shown how to reconstruct the vertex cover problem in FO-TAG. The theorem is a trivial consequence of the result in [1], since every UCFG is also a FO-TAG.

**Theorem 2.3.** *The recognition problem of FO-TAG is NP-hard.*

*Proof sketch 2.4.* For each problem instance  $D, k$  we construct a FO-TAG  $G = \langle N, T, I, A, S \rangle$  and a string  $\sigma$  such that  $\sigma$  is in the language of  $G$  iff  $D, k$  has a solution. First  $\sigma$  is defined as the concatenation of  $|E|$  many  $v_i$ 's for each  $v_i \in V$ . So for the instance  $k = 2, V = \{a, b, c, d\}, E = \{\langle a, c \rangle, \langle b, c \rangle, \langle b, d \rangle, \langle c, d \rangle\}$ , for example, a possible string is  $aaaabbbbccccdddd$ . It is then defined that the tuple of the elementary tree



with  $m = |E| \times |V| - |E| - |E| \times (|V| - k)$ . The  $S$ -initial tree has  $|V| - k$  many  $U$ 's as daughters. Consequently, there are only  $k$  vertices left to cover the graph. It should not be difficult to see how the proof proceeds; it is, in all respects, analogous to the proof for UCFG.

### 3 RSN-MCTAG

Restricted multicomponent TAG with shared nodes (RSN-MCTAG) was introduced in [6]. Its formal definition is presented in Definition 3.1.

**Definition 3.1.**  $G$  is a restricted multicomponent tree-adjoining grammar with shared nodes iff  $G = \langle N, T, I, A, \mathcal{A}, S \rangle$  such that  $G' = \langle N, T, I, A, S \rangle$  is a tree-adjoining grammar [5], and where  $\mathcal{A} \subseteq 2^{I \cup A}$ .

The next step is to define a relation on the derivation tree  $R_s$  for node-sharing. A derivation tree is a tuple  $D = \langle \text{Trees}, \text{Drvs} \rangle$ , where  $\text{Trees} \subseteq (I \cup A)$ , and  $\text{Drvs} \subseteq \text{Trees} \times \text{Trees} \times \text{GornAddrs}$ , where  $\text{GornAddrs}$  is the set of Gorn addresses.  $R_s$  is defined:

$$R_s = \{ \langle n_1, n_2 \rangle \mid n_1, n_2 \in \text{Trees}, n_2 \text{ is immediately dominated by } n_1 \text{ or there are } t_1, \dots, t_k \in \text{Trees} \text{ such that } t_1 \text{ is immediately dominated by } n_1, n_2 = t_k \text{ and for all } i, 1 \leq i \leq (k-1) : \langle t_i, t_{i+1}, p' \rangle \text{ with } t_i \text{ being an auxiliary tree with root note address } p' \}$$

The language of a RSN-MCTAG is the set of strings that are in the yield of the trees that can be derived from an  $S$ -rooted initial tree by simultaneous adjunction and substitution of all elements of sets, with derivation tree  $D = \langle \text{Trees}, \text{Drvs} \rangle$  such that for every  $\{\beta_1, \dots, \beta_n\} \in \mathcal{A}$ ,  $\beta_i \in \text{Trees}$ , and there is a  $\gamma$  such that  $\beta_i$  is immediately dominated by (is the daughter of)  $\gamma$  in the derived tree or linked to  $\gamma$  by a chain of root adjunctions. In other words,  $R_s(\gamma, \beta_i)$ . In addition, at least one  $\beta_j$  must be immediately dominated by  $\gamma$  in the derivation tree. Due to the simultaneity constraint, no two  $\beta_i, \beta_j$  can dominate each other.

**Example 3.2.** Neither RSN-MCTAG or SN-MCTAG [6], that is, RSN-MCTAG without the immediate dominance restriction on set application, generate the MIX language. This is possible, however, if we give up the constraint that no two  $\beta_i, \beta_j$  can dominate each other. To see this, consider the set:

$$\left\{ \begin{array}{c} S \\ / \quad \backslash \\ a \quad S^* \end{array} \quad , \quad \begin{array}{c} S \\ / \quad \backslash \\ b \quad S^* \end{array} \quad , \quad \begin{array}{c} S \\ / \quad \backslash \\ c \quad S^* \end{array} \right\}$$

and the initial tree:

$$\begin{array}{c} S \\ | \\ \epsilon \end{array}$$

It should be relatively easy to see that this generates the MIX language if the auxiliary trees in a set can dominate each other.

For our NP-hardness proof, it is shown how to reconstruct the vertex cover problem in RSN-MCTAG:

**Theorem 3.3.** *The recognition problem of RSN-MCTAG is NP-hard.*

*Proof sketch 3.4.* For each problem instance  $D, k$  we construct a RSN-MCTAG  $G = \langle N, T, I, A, \mathcal{A}, S \rangle$  and a string  $\sigma$  such that  $\sigma$  is in the language of  $G$  iff  $D, k$  has a solution. First  $\sigma$  is defined as the concatenation of  $|E|$  many  $v_i$ 's for each  $v_i \in V$ . So for the instance  $k = 2, V = \{a, b, c, d\}, E = \{\langle a, c \rangle, \langle b, c \rangle, \langle b, d \rangle, \langle c, d \rangle\}$ , for example, a possible string is *aaaabbbbccccdddd*. It is then defined that  $N = \{D, U, S, e_1, \dots, e_{|E|}, \delta\}$ . For each  $e_m = \langle n_i, n_j \rangle \in E$ , singleton sets are introduced:

$$\left\{ \begin{array}{c} e_m \\ | \\ n_i \end{array} \right\} \text{ and } \left\{ \begin{array}{c} e_m \\ | \\ n_j \end{array} \right\}$$

For each  $v_i \in V$ , singleton sets are introduced:

$$\left\{ \begin{array}{c} U \\ | \\ v_i^{|E|} \end{array} \right\}$$

The set in Figure 1 is then introduced. Finally, the  $S$ -initial tree is added:

$$\begin{array}{c} S \\ | \\ S \downarrow^{|E| \times |V|} \end{array}$$

The set in Figure 1 needs to saturate  $|V| - k$  many  $U$ 's. Consequently, there are only  $k$  vertices left to cover the graph. Consequently, only the trees that relate edge nonterminal symbols  $e_i$  with the terminals that are not used to build  $U$ 's can be build. In our example, this will be those with terminals  $c, d$ . The  $\delta$ -trees are just there for technical reasons.

The reconstruction shows that the recognition problem of RSN-MCTAG is NP-hard.

The proof also applies to SN-MCTAG [6], of course. Moreover the proof is independent on constraints such as tree-locality and set-locality, since the elementary trees of each set all apply, simultaneously, to the same tree and, therefore, the same set.

### 4 TT-MCTAG

Multicomponent TAG with tree tuples (TT-MCTAG) is introduced in [7]. Its formal definition is presented in Definition 4.1.

**Definition 4.1.**  $G$  is a multicomponent tree-adjoining grammar with tree tuples iff  $G = \langle N, T, I, A, \mathcal{T}, S \rangle$  such that  $G' = \langle N, T, I, A, S \rangle$  is a tree-adjoining grammar [5], and where  $\mathcal{T} \subseteq (I \cup A) \times 2^A$  such that for each  $\langle \gamma, \{\beta_1, \dots, \beta_n\} \rangle \in \mathcal{T}$  the frontier nodes of the destination tree  $\gamma$  include at least one terminal symbol.

The next step is to define a relation on the derivation tree  $R_s$  for node-sharing. This is just as in RSN-MCTAG. A derivation tree is a tuple  $D = \langle \text{Trees}, \text{Drvs} \rangle$ , where  $\text{Trees} \subseteq (I \cup A)$ , and  $\text{Drvs} \subseteq \text{Trees} \times \text{Trees} \times \text{GornAddrs}$ , where  $\text{GornAddrs}$  is the set of Gorn addresses.

The language of a TT-MCTAG is the set of strings that are in the yield of the trees that can be derived

$$\left\{ \begin{array}{c} S \\ | \\ \rho_1 \end{array} \dots \begin{array}{c} S \\ | \\ \rho_{|E|} \end{array}, \left( \begin{array}{c} S \\ | \\ U \end{array} \right)^{|V|-k}, \left( \begin{array}{c} S \\ | \\ \delta \end{array} \right)^m \right\}$$

**Fig. 1:** A set in the RSN-MCTAG reconstruction of the vertex cover problem.  $m = |E| \times |V| - |E| - |E| \times (|V| - k)$ .

from an  $S$ -rooted initial tree by adjunction and substitution with derivation tree  $D = \langle \text{Trees}, \text{Drvs} \rangle$  such that for every  $\langle \gamma, \{\beta_1, \dots, \beta_n\} \rangle \in \mathcal{T}$ ,  $\beta_i \in \text{Trees}$ , and either  $\beta_i$  is substituted for a frontier node in  $\gamma$ , or adjoined to an interior node of  $\gamma$  or  $R_s(\gamma, \beta_i)$ .

**Example 4.2.** TT-MCTAG also generates the MIX language. To see this, consider the tree tuples:

$$\left\langle \begin{array}{c} S \\ / \backslash \\ a \quad S^* \end{array}, \left\{ \begin{array}{c} S \\ / \backslash \\ b \quad S^* \end{array}, \begin{array}{c} S \\ / \backslash \\ c \quad S^* \end{array} \right\} \right\rangle$$

$$\left\langle \begin{array}{c} S \\ / \backslash \\ b \quad S^* \end{array}, \left\{ \begin{array}{c} S \\ / \backslash \\ a \quad S^* \end{array}, \begin{array}{c} S \\ / \backslash \\ c \quad S^* \end{array} \right\} \right\rangle$$

$$\left\langle \begin{array}{c} S \\ / \backslash \\ c \quad S^* \end{array}, \left\{ \begin{array}{c} S \\ / \backslash \\ a \quad S^* \end{array}, \begin{array}{c} S \\ / \backslash \\ b \quad S^* \end{array} \right\} \right\rangle$$

and the tree tuple:

$$\left\langle \begin{array}{c} S \\ | \\ \epsilon \end{array}, \emptyset \right\rangle$$

It should be relatively easy to see that this generates the MIX language. The saturation requirement in TT-MCTAG ensures that you use up all the trees in the tuples, whenever destination trees are introduced.

For our NP-hardness proof, it is shown how to reconstruct the vertex cover problem in TT-MCTAG:

**Theorem 4.3.** *The recognition problem of TT-MCTAG is NP-hard.*

*Proof sketch 4.4.* For each problem instance  $D, k$  we construct a TT-MCTAG  $G = \langle N, T, I, A, \mathcal{T}, S \rangle$  and a string  $\sigma$  such that  $\sigma$  is in the language of  $G$  iff  $D, k$  has a solution. First  $\sigma$  is defined as the concatenation of  $|E|$  many  $v_i$ 's for each  $v_i \in V$ , prefixed by the symbol  $\dagger$ . So for the instance  $k = 2, V = \{a, b, c, d\}, E = \{\langle a, c \rangle, \langle b, c \rangle, \langle b, d \rangle, \langle c, d \rangle\}$ , for example, a possible string is  $\dagger a a a b b b b c c c c d d d d$ . It is then defined that  $N = \{U, S, e_1, \dots, e_{|E|}, \delta\}$ . For each  $e_m = \langle n_i, n_j \rangle \in E$ , tree tuples are introduced:

$$\left\langle \begin{array}{c} e_m \\ | \\ n_i \end{array}, \emptyset \right\rangle \text{ and } \left\langle \begin{array}{c} e_m \\ | \\ n_j \end{array}, \emptyset \right\rangle$$

For each  $v_i \in V$ , tree tuples are introduced:

$$\left\langle \begin{array}{c} U \\ | \\ v_i \end{array}, \left\{ \left( \begin{array}{c} U \\ / \backslash \\ v_i \quad U^* \end{array} \right)^{|E|-1} \right\} \right\rangle$$

Finally, the tree tuple in Figure 2 is introduced.

The  $S$ -initial tree needs to saturate  $|V| - k$  many  $U$ 's. Consequently, there are only  $k$  vertices left to cover the graph. Consequently, only the edge-tuples – that is, the ones with destination trees with interior nodes  $\rho_i$  – with the terminals that are not used to build  $U$ 's can be build. In our example, this will be those with terminals  $c, d$ . The  $\delta$ -trees are just there for technical reasons.

The reconstruction shows that the recognition problem of TT-MCTAG is NP-hard. It is, as already said, trivial to show NP-inclusion, since derived trees are clearly polynomial in the length of the input. Consequently, it is possible to guess a model and evaluate it in polynomial time.

## 5 Polynomial time fragments

This section defines some fragments of the above extensions whose recognition problems can be solved in polynomial time. Our first fragment is FO-TAG( $k$ ). An FO-TAG is a FO-TAG( $k$ ) iff a discontinuous tuple, that is, a tuple in which the linearization of the tree is not fully specified by the LPs, has a yield of at most  $k$  terminals.

**Theorem 5.1.** *The recognition problem of FO-TAG( $k$ ) is in P.*

*Proof sketch 5.2.* Consider the simpler case of UCFG( $k$ ), defined in an analogous fashion. Our first step is to define a chart. See [13] for a similar construction. If you have a UCFG  $G = \langle N, T, P, S \rangle$  and some string  $\omega_1 \dots \omega_n$ . Construct  $G_\omega = \langle N_\omega, T_\omega, P_\omega, \{ {}_1 S_n \} \rangle$  such that

$$T_\omega = \{ \omega_1, \dots, \omega_n \}$$

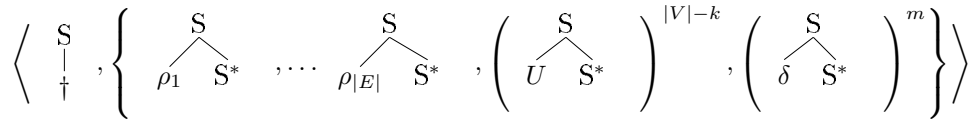
and, recursively

- (a)  $(\omega_i \in T_\omega \text{ and } A \rightarrow \omega_i \in P) \implies ({}_i A_i \in N_\omega \text{ and } {}_i A_i \rightarrow \omega_i \in P_\omega)$
- (b)  $({}_i B_{j,j+1} C_k, \dots, {}_{m-1} X_m \in N_\omega \text{ and } A \rightarrow \{B, C, \dots, X\}) \implies ({}_i A_m \in N_\omega \wedge {}_i A_m \rightarrow \{ {}_i B_{j,j+1} C_k, \dots, {}_{m-1} X_m \in P_\omega)$

The upper bound on  $|P_\omega|$  is roughly:

$$\sum_{i < n} (|N| \times (n - i) \times \sum_{j < (n-i)} (|N|^{n-j} \times (n - i) \times (n - j)))$$

To see this, note that there are  $\frac{n^2+n}{2}$  spans to assign one of  $|N|$  nonterminals, and that each nonterminal in the chart with span of length  $n - i$  may correspond to, roughly,



**Fig. 2:** The  $S$ -initial tree in the TT-MCTAG reconstruction of the vertex cover problem.  $m = |E| \times |V| - |E| - |E| \times (|V| - k)$ .

$$\sum_{j < (n-i)}^{0 \leq j} (|N|^{n-(i+j)} \times (n-i) \times (n-(i+j)))$$

productions, since you can partition the span in  $(n-i) \times (n-(i+j))$  and assign  $N^{(n-(i+j))}$  many combinations of nonterminals for each partitioning.

In UCFG( $k$ ), this number is much lower, namely

$$\sum_{i < k}^{0 \leq i} (|N| \times (n-i) \times \sum_{j < (k-i)}^{0 \leq j} (|N|^{k-(i+j)} \times (k-i) \times (k-(i+j)))) + \sum_{i < n}^{k < i} (|N|^3 \times (n-i))$$

if it is assumed that any rule that spans more than  $k$  positions is binary. The bound implies parsing in  $\mathcal{O}(n^3)$ . This reflects that a  $k$  bound on yield means a  $k'$  bound on arity, and if there is such a bound, the underlying CFG can be constructed in  $\mathcal{O}(k'!)$  time. In FO-TAG( $k$ ), the same effect is seen on charts. See [14] for a chart-based parsing algorithm for ordinary TAG. It is easy to see that since charts are polynomial in the length of the string, so is the time complexity of the recognition problem.

Similarly, a  $k$  bound on the number of nodes in unordered elementary trees will allow you to generate the underlying TAG in  $\mathcal{O}((k-1)!)$ . [6] defines a polynomial fragment of RSN-MCTAG (RSN-MCTAG( $k$ )) by adding a restriction that roughly means you can only have  $k$  elementary trees between elements of tree sets.

A similar constraint can be imposed on TT-MCTAG. Or we can impose a  $k$ -gap degree constraint, as in non-projective dependency parsing [9]. In fact, the two constraints are intimately related. If there is a  $k$  bound on the elementary trees that can occur between elements of tree sets, there is also a  $k'$  bound, linear in  $k$ , on the gap degree.

FO-TAG( $k$ ) is weakly equivalent to TAG and FO-TAG, but the  $k$ -constraint is not harmless, from a linguistic point of view, since intra-clausal unordering is relevant for arbitrary yields. The second proposal, to restrict the number of nodes in unordered elementary trees, seems more realistic; most grammars have reasonable bounds on the number of nodes. The same applies to RSN-MCTAG( $k$ ). [9] shows that a low gap degree is realistic for a number of languages.

## 6 Conclusion

It was shown that FO-TAG [2], RSN-MCTAG [6] and TT-MCTAG [7], three extensions of tree-adjoining grammar that are suited for analyzing scrambling and free word order phenomena in languages such as German, Korean and Japanese, have NP-hard universal recognition problems. All three extensions are also

NP-complete, but only one of them, TT-MCTAG, generates the MIX language. Some polynomial time fragments were defined. The NP-hardness proofs imply that tree-local MCTAG is NP-hard, while weakly equivalent to TAG. Consequently, any translation from tree-local MCTAG into TAG is exponential. This mirrors the relation between ID/LP grammars and CFG.

## References

- [1] E. Barton. The computational difficulty of ID/LP parsing. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, pages 76–81, Chicago, Illinois, 1985.
- [2] T. Becker, A. K. Joshi, and O. Rambow. Long-distance scrambling and tree adjoining grammars. In *Proceedings of the 5th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–26, Berlin, Germany, 1991.
- [3] L. Champollion. Lexicalized non-local MCTAG with dominance links is NP-complete. In *Proceedings of Mathematics of Language 10*, Los Angeles, California, 2007. To appear.
- [4] M. Garey and D. Johnson. *Computers and intractability*. W. H. Freeman & Co., New York, New York, 1979.
- [5] A. K. Joshi and Y. Schabes. Tree-adjoining grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 69–124. Springer, Berlin, Germany, 1997.
- [6] L. Kallmeyer. Tree-local multicomponent tree-adjoining grammars with shared nodes. *Computational Linguistics*, 31(2):187–225, 2005.
- [7] T. Lichte. An MCTAG with tuples for coherent constructions in German. In *Proceedings of the 12th Conference on Formal Grammar*, Dublin, Ireland, 2007. To appear.
- [8] A. Maclachlan and O. Rambow. Cross-serial dependencies in tagalog. In *Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+6)*, pages 100–104, Venice, Italy, 2002.
- [9] J. Nivre. Constraints on non-projective dependency parsing. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 73–80, Trento, Italy, 2006.
- [10] O. Rambow and G. Satta. Formal properties of non-locality. In *Proceedings of the Second International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+2)*, Philadelphia, Pennsylvania, 1992.
- [11] S. Shieber. Direct parsing of ID/LP grammars. *Linguistics and Philosophy*, 7:135–154, 1984.
- [12] S. Shieber. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343, 1985.
- [13] A. Sogaard. Polynomial charts for totally unordered languages. In *Proceedings of the 16th Nordic Conference of Computational Linguistics*, pages 183–190, Tartu, Estonia, 2007.
- [14] K. Vijay-Shanker and D. Weir. Parsing some constrained grammar formalisms. *Computational Linguistics*, 19(4):591–636, 1993.

# Combining Deterministic Processing with Ambiguity-Awareness

Kristina Spranger  
IMS, University of Stuttgart  
Azenbergstraße 12  
70 174 Stuttgart, Germany  
*Kristina.Spranger@ims.uni-stuttgart.de*

## Abstract

In order to analyze large amounts of unrestricted text, it is highly desirable to construct a system that skirts exponentiality at the parsing level. This is exactly what is attained by syntactic underspecification. We present an underspecification approach that basically consists of constraining the parsing process by incorporating restrictive rules in the grammar on which the parser relies. Starting from the output of a deterministic parser, an underspecified representation is generated. In order to prove that our underspecification approach works, we show how all syntactic readings can be reconstructed from the underspecified representation. As a linguistic working example we opted for quantifying noun groups in German since they are a detail of language where the interplay of syntax, semantics and the lexicon becomes particularly apparent and since, therefore, these constructions are manifold ambiguous.

## Keywords

Syntactic Underspecification, (Deterministic) Parsing, Deep Processing, Ambiguity-Awareness

## 1 Introduction

In parsing, the major problem that has to be dealt with is the problem of ambiguity and with it the exponential increase in the number of analyses. By definition, syntax does not provide enough constraints to distinguish a meaningful from a syntactically well-formed, but meaningless or incomplete utterance. So, many problems cannot be resolved by purely syntactic knowledge sources.

Hindle (cf. [4]) formulated a reasonable set of requirements for a parser of unrestricted text. According to Hindle, such a parser: should provide at least some syntactic analysis for any input - grammatical or not; it should give a partial analysis when a complete analysis is not achievable; it should provide one single analysis for each input text; and, it should process text in a reasonable amount of time. A deterministic parser directly satisfies several of these requirements.

## Combining Deterministic Parsing with Ambiguity-Handling – Syntactic Underspecification

An apparent drawback of deterministic parsers is the need for “forced guessing”: as deterministic parsers return only exactly one analysis they are forced to solve many locally unresolvable ambiguities (cf. [9], and [11]). Thus, deterministic parsers have to make decisions without access to the requisite knowledge (for a detailed discussion see [7]).

By using syntactic underspecification, parsing decisions that are known to be not resolvable by syntax can be handed over to subsequent modules that are equipped with the relevant (context) knowledge. So, syntactic underspecification allows to combine deterministic - and, therefore, efficient - processing with what we call “ambiguity-awareness” (cf. [13]): a well-defined set of cases that are known to be ambiguous may be represented in the parsing output and processed further on in an adequate way.

In cases that are known to be ambiguous only “minimal” chunks are annotated. “Minimal” chunks take part in each possibly intended syntactic reading; they are so to say the smallest common denominator of all syntactic readings. The minimal chunks are then combined into an underspecified representation of the ambiguous token sequence in question.

That means, syntactic underspecification skirts ambiguity at the parsing level: it tackles ambiguity successfully, and does away with exponentiality. Basically, in our underspecification approach the parsing process is constrained by incorporating restrictive rules in the grammar on which the parser is based in situations where ambiguity would otherwise occur. Starting from the parsing output, a representation can be generated that is “expandable” to different possible analyses. That is to say, analyses that are underspecified as to some aspect of the syntax are consistent with a set of different analyses.

We illustrate and explain our ambiguity-handling approach by means of a linguistically interesting working example: the quantifying noun group in German.

## 2 Quantifying Noun Groups

Collections of individuals and portions of masses and collectives can be referred to by means of quantifying noun groups in which one nominal quantifies over the kind of entity indicated by the other nominal (cf. [6],

[2]). Nominal quantifiers originate as concrete or abstract nouns, are used for creating a measuring or counting unit which may further be counted, and show a variety of types (cf. [14], [3], [1]).

**The Structure of the Quantifying Noun Group in German** The quantifying noun group consists of a numeral, a quantifying constituent, and a quantified constituent (cf. [8]). There is no obligatory component in this construction: each component can be deleted if it is contextually deducible. The only constraint is that at least two of the three components are explicitly realised. There are, of course, certain (contextually determined) combinatory restrictions to the combinability of different realisations of the components (cf. [10]).

In [13], a classification of quantifying noun groups is presented that is mainly based on the specific nature of the different types of quantity nouns.

**Automatic Analysis – Two Sources of Ambiguity** There are basically two sources of ambiguity that complicate the automatic analysis of quantifying noun groups - and that therefore can hamper a (linguistically intuitive) correct annotation (cf. [12])<sup>1</sup>:

**1. Quantifying Noun Groups versus (Accidental) Sequences of two Separate Noun Phrases**

There are no syntactic arguments that argue against identic syntactic analyses for the bold typed noun phrases in sentences (1) and (2).

- (1) Als Trostpflaster für die Verlierer gab es *eine Tafel Schokolade*.  
*a bar chocolate.*  
 ‘There was a bar of chocolate as consolation for the losers.’
- (2) Wo früher Kellner mit *einer Tafel Frauen* auf die Tanzfläche baten, fiept und piepst es heute neben den Kaffeetassen.  
*besides the coffee cups.*  
 ‘Where in former times waiters asked women to enter the dancefloor by blackboards, it cheeps and peeps besides the coffee cups nowadays.’

Examples (1) and (2) show that the linear surface order does not suffice in order to distinguish between a real quantifying noun group (cf. sentence (1), figure 1), and an accidental sequence of a spurious quantifying constituent and a spurious quantified constituent (cf. sentence (2), figure 2).

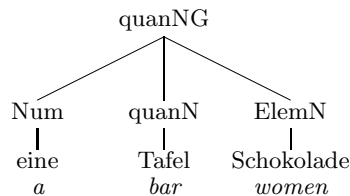


Fig. 1: Correct analysis for the bold typed noun phrase in sentence (1)

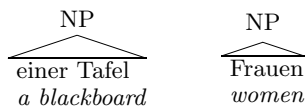


Fig. 2: Correct analysis for the bold typed noun phrase in sentence (2)

**2. Indication of Measurement versus Measure Argument of Adjective**

The linear surface order does not suffice in order to distinguish between the “indication-of-measurement-reading” (cf. sentence (3) where the indication of measurement, namely *2000 Hektar*, specifies the total quantity of what *alte Industrieflächen* refers to) and the “measure-argument-of-adjective-reading” (cf. sentence (4) where the indication of quantity, namely *50 bis 100 Hektar*, is the measure argument of the adjective *groß*):

- (3) **2000 Hektar alte Industrieflächen**  
*2000 hectare old industrial areas*  
 ‘2000 hectares of old industrial areas’
- (4) **50 bis 100 Hektar große Flächen**  
*50 to 100 hectare large areas*  
 ‘50 to 100 hectares large areas’

Even though the linear surface order is the same for both sentences the correct syntactic analyses differ from each other (cf. figures 3 and 4).

In both cases of ambiguity the parser should deliver a representation that comprises all syntactically possible readings. To this end, we extended the YAC-parser (cf. [5]) with the possibility of (syntactic) underspecification.

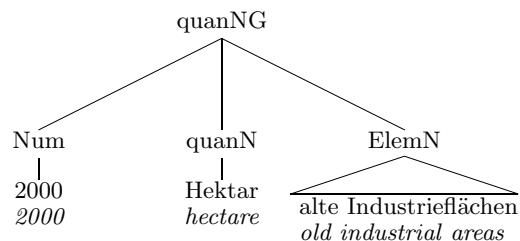
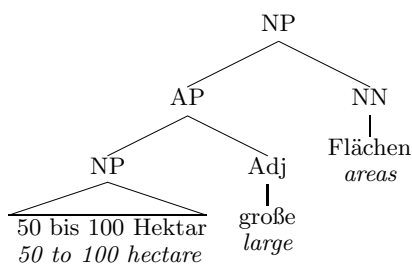


Fig. 3: Indication-of-measurement-reading (cf. sentence (3))

<sup>1</sup> The empirical base of the presented observations is the *HGC* (**H**uge **G**erman **C**orpus - approximately 200 million words German newspaper corpora).



**Fig. 4:** *Measure-argument-of-adjective-reading* (cf. sentence (4))

### 3 The Underspecification Approach

By a specification process we understand any linguistic process that transforms one representation of a linguistic object into anyone of several alternative more specific representations of the same linguistic object. Following this definition of a specification process, any process of disambiguation is actually a specification process. Abstractly, the use of underspecified representations and their stepwise transformation can be described as follows: there is a given formalism  $R$  in which both underspecified and fully specified representations are formulated. One and the same linguistic expression  $LE$  will in general have many different representations in  $R$ .

The set  $R(LE)$  of representations of  $LE$  is partially ordered in terms of specificity by the relation  $>_{LE}$ ; the fully specified representations of  $R(LE)$  are maximal with respect to  $>_{LE}$ . It is usually assumed that each  $R(LE)$  contains at least one fully specified representation and that for each  $r$  in  $R(LE)$  there is at least one fully specified  $r'$  in  $R(LE)$  such that  $r' \geq_{LE} r$ .

In general, an underspecified representation admits the transformation into more than one fully specified representation. The less underspecified a representation is, the smaller the set of fully specified representations into which it can be transformed.

**The Generation of the Underspecified Representation** As a working example we look at the three-fold ambiguous noun group given in sentence (5).

(5) drei Jahre alter Wein  
*three years old wine*

1. 'three years old wine' [drei Jahre<sub>NP</sub>][alter Wein<sub>NP</sub>]
2. 'three years of old wine' [drei Jahre alter Wein<sub>quan.NG</sub>]
3. 'wine that is three years old' [[[drei Jahre<sub>NP</sub> alter<sub>AP</sub>] Wein<sub>NP</sub>]

**The Parsing Output - Minimal Chunks** Supposed the example sentence (5) occupies the corpus positions 0 through 3:

0 1 2 3  
drei Jahre alter Wein

the minimal chunks outputted by the parser are the following<sup>2</sup>:

1.  $\langle NP, 0, 1, 1 \rangle$  (*drei Jahre* (0 ... 1), syntactic head: *Jahre*)
2.  $\langle NP, 2, 3, 3 \rangle$  (*alter Wein* (2 ... 3), syntactic head: *Wein*)
3.  $\langle NP, 3, 3, 3 \rangle$  (*Wein* (3 ... 3), syntactic head: *Wein*)
4.  $\langle AP, 2, 2, 2 \rangle$  (*alter* (2 ... 2), syntactic head: *alter*)

Starting from the parsing output a representation is generated that is "expandable" to different possible analyses.

**The Underspecified Representation** The underspecified representation is generated by assembling all possible starting positions, ending positions, and head positions in one representation per phrasal category. The complete underspecified representation is the set of underspecified representations that span the whole ambiguous token sequence.

The underspecified representation of our working example contains the following two 4-tuples:

1.  $\langle NP, \{0, 2, 3\}, \{1, 3\}, \{1,3\} \rangle$
2.  $\langle AP, \{0, 2\}, \{2\}, \{2\} \rangle$

### 4 The Reconstruction Process

The following six work steps are executed in order to reconstruct all readings assembled in the underspecified representation: (1) Derive all constituents comprised in the underspecified representation; (2) apply structural constraints; (3) compute the dominance and precedence relations between each pair of constituents; (4) compute "unordered trees"; (5) apply dominance and precedence relations on the unordered trees; and (6) check if the trees are "complete" syntactic descriptions.

**Deriving the Comprised Constituents** First, all constituents that are comprised in the underspecified representation are derived. The derivation simply consists in combining all possible starting, ending, and head positions:

- $$R_1^1: \langle NP, 0, 1, 1 \rangle, R_1^2: \langle NP, 0, 3, 1 \rangle$$
- $$R_1^3: \langle NP, 0, 1, 3 \rangle, R_1^4: \langle NP, 0, 3, 3 \rangle$$
- $$R_1^5: \langle NP, 2, 1, 1 \rangle, R_1^6: \langle NP, 2, 3, 1 \rangle$$
- $$R_1^7: \langle NP, 2, 1, 3 \rangle, R_1^8: \langle NP, 2, 3, 3 \rangle$$
- $$R_1^9: \langle NP, 3, 1, 1 \rangle, R_1^{10}: \langle NP, 3, 3, 1 \rangle$$
- $$R_1^{11}: \langle NP, 3, 1, 3 \rangle, R_1^{12}: \langle NP, 3, 3, 3 \rangle$$
- $$R_2^1: \langle AP, 0, 2, 2 \rangle, R_2^2: \langle AP, 2, 2, 2 \rangle$$

<sup>2</sup> The first position of the 4-tuple is occupied by the phrasal category, the second position specifies the starting position of the considered chunk, the third position the ending position, and the last position specifies the position of the syntactic head.



**Applying Structural Constraints** Then, structural constraints are applied to the derived constituents so that syntactically ill-formed constituents are ruled out from the beginning.

We apply the following two constraints:

1.  $start_i \leq end_i$ ; and
2.  $start_i \leq head_i \leq end_i$

and rule out six representations ( $R_1^3$ ,  $R_1^5$  through  $R_1^7$ , and  $R_1^9$  through  $R_1^{11}$ ).

**The Relations holding between each pair of Constituents** Next, the precedence and dominance relations that hold between each pair of constituents are computed and plotted in a matrix.

We define four possible relations:

1. dominance relation “ $\gg$ ”, “ $\ll$ ”:

$c_i \gg c_j$ , iff:

- $start_i < start_j$  &  $end_i \geq end_j$  & ( $head_j = head_i$  if  $start_j < head_i < end_j$ ); or
- $start_i \leq start_j$  &  $end_i > end_j$  & ( $head_j = head_i$  if  $start_j < head_i < end_j$ )<sup>3</sup>.

2. sibling relation “o”:

- $start_i > end_j$ ; or
- $end_i < start_j$ .

3. identity relation “ $\equiv$ ”:

- $cat_i = cat_j$ <sup>4</sup> &  $start_i = start_j$  &  $end_i = end_j$  &  $head_i = head_j$

4. exclusion relation “X”:

- no other relation holds between the constituents  $c_i$  and  $c_j$

	$R_1^1$	$R_1^2$	$R_1^4$	$R_1^8$	$R_1^{12}$	$R_2^1$	$R_2^2$
$R_1^1$	$\equiv$	$\ll$	X	o	o	$\ll$	o
$R_1^2$	$\gg$	$\equiv$	X	$\gg$	$\gg$	$\gg$	$\gg$
$R_1^4$	X	X	$\equiv$	$\gg$	$\gg$	$\gg$	$\gg$
$R_1^8$	o	$\ll$	$\ll$	$\equiv$	$\gg$	X	$\gg$
$R_1^{12}$	o	$\ll$	$\ll$	$\ll$	$\equiv$	o	o
$R_2^1$	$\gg$	$\ll$	$\ll$	X	o	$\equiv$	$\gg$
$R_2^2$	o	$\ll$	$\ll$	$\ll$	o	$\ll$	$\equiv$

**Table 1:** Relations holding between each pair of Constituents

In table 1, the constituent described by the representation given on the left hand side is related to the constituent described by the representation given on the top.

<sup>3</sup> Obviously, in case of a prepositional phrase introduced by a preposition,  $start_i$  must be smaller than  $start_j$ .

<sup>4</sup>  $cat_i = cat_j$  reads  $c_i$  and  $c_j$  have the same phrase category.

**Computing Unordered Trees** In the fourth step of the reconstruction process, all possible different sets are computed that contain only constituents that may occur together, i.e. that contain no constituents between which an exclusion relation holds. These sets of “compatible” representations are unordered trees, since, so far, the type of relation that holds between compatible representations is disregarded.

The reconstruction component starts with the first row index of table 1. It checks whether the represented constituent can occur together with the constituent represented by the second<sup>5</sup> column index. If these two constituents can occur together, a set  $S_1$  is started containing both representations.

Then, the algorithm checks whether the constituent represented by the next column index can occur together with both constituents whose representations are elements of  $S_1$ . If this is the case, the active representation - i.e. the representation that is now considered - is just added to the set; otherwise, a new set  $S_2$  is started that contains the active representation and those representations of  $S_1$  with which the active representation can cooccur.

In this way, the reconstruction module runs through the whole row: it tests for each representation  $R_i^j$  and for each set  $S_k$  with which representations of  $S_k$  the active representation  $R_i^j$  can cooccur. If the active representation can cooccur with all representations of a given set, it is just added to this set; otherwise, a new set  $S_{k+1}$  is started that contains only those representations of  $S_k$  with which the active representation can cooccur.

Following the described algorithm, four sets  $S_1$  through  $S_4$  are computed starting from table 1:

1.  $S_1: \{R_1^1, R_1^2, R_1^8, R_1^{12}, R_2^2\}$
2.  $S_2: \{R_1^4, R_1^8, R_1^{12}, R_2^2\}$
3.  $S_3: \{R_1^1, R_1^2, R_1^{12}, R_2^1, R_2^2\}$
4.  $S_4: \{R_1^4, R_1^{12}, R_2^1, R_2^2\}$

**The Hierarchical Representations of the “Unordered Trees”** In a fifth step, the information about the dominance and sibling relations - encoded in table 1 - is used in order to build up the hierarchical representations of the possible readings:

1.  $T_1$  (cf. figure 5) that is made up of the constituents contained in  $S_1$  describes two syntactic readings:
  - (a) one noun phrase, i.e. the quantifying noun group
  - (b) two separate noun phrases
2.  $T_2$  (cf. figure 6) that is made up of the constituents contained in  $S_2$  is no complete syntactic description (*drei Jahre* is missing).
3.  $T_3$  (cf. figure 7) that is made up of the constituents contained in  $S_3$  means “wine that is three years old”.

<sup>5</sup> Obviously, it does not look for the first representation given on the top, since this is identical to the first representation given on the left hand side.

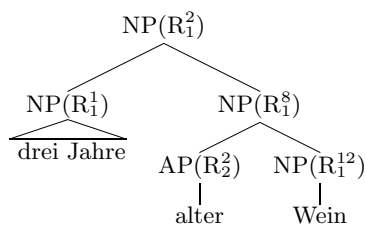


Fig. 5: *Syntax Tree T<sub>1</sub>*

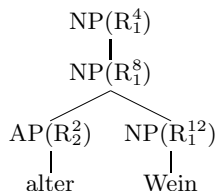


Fig. 6: *Syntax Tree T<sub>2</sub>*

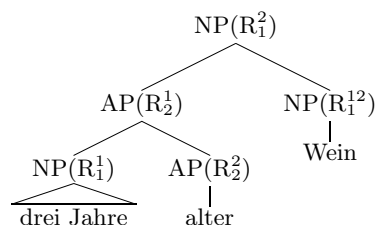


Fig. 7: *Syntax Tree T<sub>3</sub>*

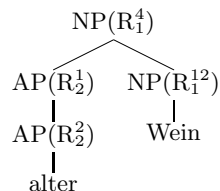


Fig. 8: *Syntax Tree T<sub>4</sub>*

4. T<sub>4</sub> (cf. figure 8) that is made up of the constituents contained in S<sub>4</sub> is no complete syntactic description (*drei Jahre* is missing).

The reconstruction process shows that the underspecified representation contains all syntactically valid readings. That is to say, the underspecified representation is the adequately specific representation.

## 5 Conclusion

The ambiguity handling strategy proposed in this article is best described by ambiguity-awareness:

- The parser is informed about ambiguous constructions and delivers minimal chunks only. So, it does not spoil the system's chance of getting the intended analysis from the beginning - although a deterministic parser is used.
- An underspecified representation that contains all locally valid readings is computed, and can be written back into the corpus. So, the reading intended by the author is available to applications using this output.

We call this behaviour ambiguity-aware since it is actually not a disambiguation strategy, but, in contrast, the output of a deterministic parser is completed by alternative analyses in a controlled way.

The strategy of syntactic underspecification is a linguistically sound way to tackle exponential ambiguity. A solution in which potentially problematic structures are dealt with by means of underspecified representations in order to resolve the ambiguity when enough information is available is a linguistically motivated solution for the disambiguation problem rather than a technical workaround to camouflage it. It is elegant as well, since it enables the linguist to determine very accurately which ambiguities are to be specified, and in which way.

## References

- [1] A. Akmajian and A. Lehrer. NP-like Quantifiers and the Problem of Determining the Head of an NP. *Linguistic Analysis*, 4(2):395–413, 1976.
- [2] C. Eschenbach. Maßangaben im Kontext - Variationen der quantitativen Spezifikation. In S. W. Felix, C. Habel, and G. Rickheit, editors, *Kognitive Linguistik*, pages 207–228. Westdeutscher Verlag, Opladen, 1994.
- [3] J. Higginbotham. Mass and Count Quantifiers. *Linguistics and Philosophy*, 17(5):447–480, 1994.
- [4] D. Hindle. A Parser for Text Corpora. In B. T. Atkins and A. Zampolli, editors, *Computational Approaches to the Lexicon*, chapter 5, pages 103–151. Oxford University Press, Oxford, 1994.
- [5] H. Kermes. *Off-line (and Online) Text Analysis for Computational Lexicography*. PhD thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, 2003.
- [6] W. G. Klooster. *The Structure Underlying Measure Phrase Sentences*, volume 17 of *Foundations of language / Supplementary series*. Reidel, Dordrecht, 1972.
- [7] H. Langer. *Parsing-Experimente: Praxisorientierte Untersuchungen zur automatischen Analyse des Deutschen*. Europäischer Verlag der Wissenschaften, 2001.
- [8] E. Löbel. *Apposition und Komposition in der Quantifizierung. Syntaktische, semantische und morphologische Aspekte quantifizierender Nomina im Deutschen*. Number 166 in *Linguistische Arbeiten*. Max Niemeyer Verlag, Tübingen, 1986.
- [9] G. Neumann, C. Braun, and J. Piskorski. A Divide-and-Conquer Strategy for Shallow Parsing of German Free Text. In *Proceedings of ANLP-2000*, pages 239–246, Seattle, Washington, 2000.
- [10] J. Oesterle. *Syntaktische und semantische Aspekte von Maßkonstruktionen im Deutschen*. PhD thesis, Centrum für Informations- und Sprachverarbeitung, LMU München, 1995.
- [11] M. Schiehlen. Experiments in German noun chunking. In *Proceedings of the 19th International Conference on Computational Linguistics*, 2002.
- [12] K. Spranger. Some Remarks on the Annotation of Quantifying Noun Groups in Treebanks. In *Proceedings of the 6th International Workshop on Linguistically Interpreted Corpora (LINC-2005)*, pages 81–90, 2005.
- [13] K. Spranger. *Combining Deterministic Processing with Ambiguity-Awareness - The Case of Quantifying Noun Groups in German*. PhD thesis, IMS, Universität Stuttgart, 2006.
- [14] H. Wiese and J. Maling. Beers, kaffi, and Schnaps - Different grammatical options for 'restaurant talk' coercions in three Germanic languages. *Journal of Germanic Linguistics*, 17(1):1–38, 2005.

# Learning Canonical Forms of Entailment Rules

Idan Szpektor  
Dept. of Computer Science  
Bar Ilan University  
Ramat Gan, 52900  
Israel  
szpekti@cs.biu.ac.il

Ido Dagan  
Dept. of Computer Science  
Bar Ilan University  
Ramat Gan, 52900  
Israel  
dagan@cs.biu.ac.il

## Abstract

We propose a modular approach to paraphrase and entailment-rule learning that addresses the morpho-syntactic variability of lexical-syntactic templates. Using an entailment module that captures generic morpho-syntactic regularities, we transform every identified template into a canonical form. This way, statistics from different template variations are accumulated for a single template form. Additionally, morpho-syntactic redundant rules are not acquired. This scheme also yields more informative evaluation for the acquisition quality, since the bias towards rules with many frequent variations is avoided.

## Keywords

Textual Entailment, Paraphrases, Knowledge Acquisition

## 1 Introduction

In many NLP applications such as Question Answering (QA) and Information Extraction (IE), it is crucial to recognize that a specific target meaning can be inferred from different text variants. For example, a QA system have to deduce that “*Mozart wrote the Jupiter symphony*” can be inferred from “*Mozart composed the Jupiter symphony*” in order to answer “*Who wrote the Jupiter symphony?*”. This type of reasoning has been identified as a core semantic inference paradigm by the generic *textual entailment* framework [5].

An important type of knowledge representation needed for such inference is *entailment rules*. An entailment rule, e.g. ‘ $X \text{ compose } Y \rightarrow X \text{ write } Y$ ’, is a directional relation between two *templates*. Templates represents text patterns with variables that typically corresponds to semantic predicates. In an entailment rule, the left hand side template is assumed to entail the right hand side template in certain appropriate contexts, under the same variable instantiation. Such rules capture rudimentary inferences and are used as building blocks for more complex inference. For example, given the above entailment rule, a QA system can identify “*Mozart*” as the answer for the above question. A major obstacle for further advances in semantic inference is the lack of broad-scale knowledge-bases for such rules [1]. This need sparked intensive research on automatic acquisition of entailment rules (and similarly paraphrases). These algorithms’ strength is in learning relations between lexical-syntactic templates, which capture lexical-based knowledge and world knowledge (see Section 2.1).

One noticeable phenomenon of lexical-syntactic templates is that they have many morpho-syntactic variations, which (largely) represent the same predicate and are semantically equivalent. For example, ‘ $X \text{ compose } Y$ ’ can be expressed also by ‘ $Y \text{ is composed by } X$ ’ or ‘ $X$ ’s composition of  $Y$ ’. Current learning algorithms ignore this morpho-syntactic variability. They treat these variations as semantically different, learning rules for each variation separately. This leads to several undesired consequences. First, statistics for a particular semantic predicate are scattered among different templates. This may result in insufficient statistics for learning a rule in any of its variations. Second, though rules may be learned in several variations (see Table 1), in most cases only a small part of the morpho-syntactic variations are learned. Thus, an inference system that uses only these learned rules would miss recognizing a substantial number of variations of the sought predicate.

It therefore makes more sense to design a modular architecture. In it, a separate entailment module recognizes entailing variations that are based on generic morphological and syntactic regularities (morpho-syntactic entailments). We propose to use such a module first at learning time, by learning only canonical forms of templates and rules. Then, applying the module also at inference time, in conjunction with the learned lexical-based canonical rules, guarantees the coverage of all morpho-syntactic variations of a given canonical rule.

Our proposed approach poses two advantages. First, the statistics from the different morpho-syntactic variations accumulate for one template form only. The improved statistics may result, for example, in learning more rules. Second, the learning output is without redundancies due to variations of the same predicate. Additionally, the evaluation of learning algorithms is more accurate when the bias towards templates with many frequent variations is avoided.

In this work we implemented a morpho-syntactic entailment module that utilizes syntactic rules for major syntactic phenomena (like passive and conjunctions) and morphological rules that address nominalizations. We then applied the module within two entailment rule acquisition algorithms. We measured redundancy removal of about 6% out of all rules learned. For one of the algorithms, we measured an increase of about 12% in the number of lexically different correct templates that were learned using our approach. Finally, we applied the morpho-syntactic entailment module also at inference time in a Relation Extraction setup for protein-interaction. In a preliminary experiment, we found that the rules learned using our new scheme yielded some improvement in recall.

Morpho-Syntactic Variations	
$X$ compose $Y \rightarrow X$ write $Y$	$X$ is composed by $Y \rightarrow X$ write $Y$
$X$ accuse $Y \leftrightarrow X$ blame $Y$	$X$ 's accusation of $Y \leftrightarrow X$ blame $Y$
$X$ acquire $Y \rightarrow X$ obtain $Y$	acquisition of $Y$ by $X \rightarrow Y$ is obtained by $X$

**Table 1:** Examples of learned rules that differ only in their morpho-syntactic structure.

Template	Single-feature Approach (DIRT)		Anchor-Set Approach Common Features
	X-vector Features	Y-vector Features	
$X$ compose $Y$	Bach, Beethoven Mozart, he	symphony, music sonata, opera	$\{X='Mozart'; Y='Jupiter symphony'\}$ , $\{X='Bach'; Y='Sonata Abassoonata'\}$
$X$ write $Y$	Tolstoy, Bach, author, Mozart, he	symphony, anthem, sonata, book, novel	

**Table 2:** Examples for features of the anchor set and single-feature approaches for two related templates.

## 2 Background

### 2.1 Entailment Rule Learning

Many algorithms for automatically learning entailment rules and paraphrases (which can be viewed as bidirectional entailment rules) were proposed in recent years. These methods recognize templates in texts and identify entailment relations between them based on shared features.

These algorithms may be divided into two types. The prominent approach identify an entailment relation between two templates by finding variable instantiation tuples, termed here *anchor-sets*, that are common to both templates [13, 18, 2, 12, 20, 16]. Anchor-sets are complex features, consisting of several terms, labelled by their corresponding variables. Table 2 (right column) presents common anchor-sets for the related templates ' $X$  compose  $Y$ ' and ' $X$  write  $Y$ '. Typically, only few common anchor-sets are identified for each entailment relation.

A different single-feature approach is proposed by the DIRT algorithm [10]. It uses simple, less informative but more frequent features. It constructs a feature vector for each variable of a given template, representing the context words that fill the variable in the different occurrences of the template in the corpus. Two templates are identified as semantically related if they have similar vectors. Table 2 shows examples for features of this type. DIRT parses a whole corpus and limits the allowed structures of templates only to paths in the parse graphs, connecting nouns at their ends.

In this paper we implemented the TEASE algorithm [20]. It is an unsupervised algorithm that acquires entailment relations from the Web for given input templates using the anchor-set approach (we required at least two common anchor-sets for learning a relation). We also implemented the DIRT algorithm over a local corpus, the first CD of Reuters RCV1<sup>1</sup>. Both algorithms process *lexical-syntactic templates*, which are represented by parse subtrees. All sentences are parsed using the Minipar dependency parser [9].

For a given input template  $I$ , these algorithms can be viewed as learning a list of output templates  $\{O_j\}_1^{n_I}$ , where  $n_I$  is the number of templates learned for  $I$ . Each out-

put template is suggested as holding an entailment relation with the input template, but current algorithms do not specify the entailment direction(s). Thus, each pair  $\{I, O_j\}$  induces two candidate directional entailment rules: ' $I \rightarrow O_j$ ' and ' $O_j \rightarrow I$ '.

As shown in previous evaluations the precision of DIRT and TEASE is limited [10, 2, 20, 19]. Currently, their application should typically involve manual filtering of the learned rules, and the algorithms' utility is reflected mainly by the amount of correct rules they learn. Specifically, DIRT learns a long tail of low quality rules with less significant statistics, which still yield a positive similarity value.

The learned entailment rules and paraphrases can be used at *inference time* in applications such as IE [18, 14, 17] and QA [10, 13, 8], where matched rules deduce new target predicate instances from texts (like the 'compose  $\rightarrow$  write' example in Section 1).

### 2.2 Morpho-Syntactic Template Variations

Lexical syntactic templates can take on many morpho-syntactic variations, which are usually semantically equivalent. This phenomenon is addressed at the inference phase by recognizing semantically equivalent syntactic variations, such as passive forms and conjunctions (e.g. [14]). Some work was done to systematically recognize morphological variations of predicates [11, 7], but it was not applied for entailment inference.

In contrast, current methods for learning lexical-syntactic rules do not address the morpho-syntactic variability at learning time at all. Thus, they learn rules separately for each variation. This results in either learning redundant rules (see Table 1) or missing some of the relevant rules that occur in a corpus. Moreover, some rules might not be learned in any variation. For example, if for each of the rules ' $X$  acquire  $Y \rightarrow X$  own  $Y$ ', ' $Y$  is acquired by  $X \rightarrow X$  own  $Y$ ' and ' $X$ 's acquisition of  $Y \rightarrow X$  own  $Y$ ' there are no sufficient statistics then none of them will be learned.

To sum up, though several problems rise from disregarding the morpho-syntactic variability, there is still no sound solution for addressing it at learning time.

<sup>1</sup> <http://about.reuters.com/researchandstandards/corpus/>

### 3 A Modular Approach for Entailment Rule Learning

A natural solution for addressing the morpho-syntactic variability in templates is a modular architecture, in which a separate entailment module recognizes entailing variations that are based on generic morphological and syntactic regularities.

In our scheme, we use this morpho-syntactic entailment module to transform lexical-syntactic template variations that occur in a text into their *canonical form*. This form, which we chose to be the active verb form with direct modifiers, is entailed by other template variations. We next describe our implementation of such a module and its application within entailment rule acquisition algorithms.

#### 3.1 Morpho-Syntactic Canonization Module

We implemented a morpho-syntactic module based on a set of *canonization rules*, highly accurate morpho-syntactic entailment rules. Each rule represents one morpho-syntactic regularity that is eliminated when the rule is applied to a given template (see examples in Table 3 and Figure 1).

Our current canonization rule collection consists of two types of rules: (a) syntactic-based rules; (b) morpho-syntactic nominalization rules. We next describe each rule type. As we use the Minipar parser, all rules are adapted to Minipar's output format.

**Syntactic-based Rules** These rules capture entailment patterns associated with common syntactic structures. Their function is to simplify and generalize the syntactic structure of a template.

In the current implementation we manually created the following simplification rules: (a) passive forms into active forms; (b) removal of conjunctions; (c) removal of appositions; (d) removal of abbreviations; (e) removal of set description by the 'such as' preposition. Table 3 presents some of the rules we created together with examples of their effect.

**Nominalization Rules** Entailment rules such as 'acquisition of  $Y$  by  $X \rightarrow X$  acquire  $Y$ ' and ' $Y$ 's acquisition by  $X \rightarrow X$  acquire  $Y$ ' capture the relations between verbs and their nominalizations. We automatically derived these rules from Nomlex, a hand-coded database of about 1000 English nominalizations [11], as described in [15]. These rules transform any nominal template in Nomlex into its related verbal form. These rules preserve the semantics of the original template predicate. We chose the verbal form as the canonical form since for every predicate with specific semantic modifiers there is only one verbal active form in Nomlex, but typically several equivalent nominal forms.

**Chaining of Canonization Rules** Each of the syntactic rules decreases the size of a template. In addition, nominalization rules can only be applied once for a given template, since no rule in our rule-set transforms a verbal template into one of its nominal forms. Thus, applying rules until no rule can apply is a finite process. In addition, each of our rules is independent of the others, operating on a different set of dependency relations. Consequently, applying

any sequence of rules until no other rule can apply will result in the same final canonical template form. Figure 1 illustrates an example for rule chaining.

#### 3.2 Applying the Canonization Module

When a morpho-syntactic entailment module is utilized at inference time (e.g. [14]), it recognizes a closure of morpho-syntactic variations for a lexical-syntactic template. Accordingly, acquisition algorithms may learn just a single morpho-syntactic variation of a template.

With this modular scheme in mind, we propose to solve the learning problems discussed in Section 2.2 by utilizing the morpho-syntactic entailment module at learning time as well. We incorporate the module in the learning algorithms (TEASE and DIRT in our experiment) by converting each template variation occurrence in the learning corpus into an occurrence of a canonical template. Thus, the learning algorithms operate only on canonical forms.

As discussed in Section 1, when canonization is used, no morpho-syntactically redundant rules are learned, with respect to the variations that are recognized by the module. This makes the output more compact, both for storage and for use. In addition, the statistical reliability of learned rules may be improved. For example, rules that could not be learned before in any variation may be learned now for the canonical form.

Methodologically, previous evaluations of learning algorithms reported accuracy relative to the redundant list of rules, which creates a bias for templates with many frequent variations. When this bias is removed and only truly different lexical-syntactic rules are assessed, evaluation is more efficient and accurate.

### 4 Evaluation

We conducted two experiments: (a) a manual evaluation of the contribution of the canonization module to TEASE and DIRT, based on human judgment of the learned rules; (b) a Relation Extraction evaluation setup for a protein interaction data-set.

#### 4.1 Human Judgement Evaluation

We have selected 20 different verbs and verbal phrases<sup>2</sup> as input templates for both TEASE and DIRT, and executed both the baseline versions (without canonization), marked as  $TEASE_b$  and  $DIRT_b$ , and the versions with the canonization module, marked as  $TEASE_c$  and  $DIRT_c$ . The results of the executions constitute our test-set rules.

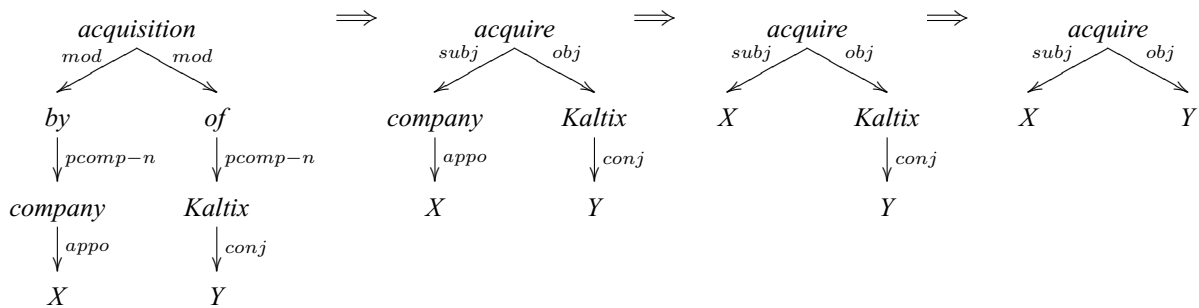
As discussed in Section 2.1, both TEASE and DIRT do not learn the direction(s) of an entailment relation between an input template  $I$  and a learned output template  $O$ . Thus, we evaluated both candidate directional rules, ' $I \rightarrow O$ ' and ' $O \rightarrow I$ '.

**Rule Evaluation** The prominent approach for evaluating rules is to present them to human judges, who assess whether each rule is correct or not. Generally, a rule is considered correct if the judge could think of reasonable

<sup>2</sup> The verbs are: accuse, approve, calculate, change, demand, establish, finish, hit, invent, kill, know, leave, merge with, name as, quote, recover, reflect, tell, worsen, write.

Rule	Description	Original Template	Simplified Template
passive to active	$X \xleftarrow{pcomp-n} \text{by} \xleftarrow{by-subj} V$ $\implies X \xleftarrow{subj} V$	$X \xleftarrow{pcomp-n} \text{by} \xleftarrow{by-subj} \text{find} \xrightarrow{obj} Y$	$X \xleftarrow{subj} \text{find} \xrightarrow{obj} Y$
conjunction	$Z \xrightarrow{conj} Y \implies Y$	$X \xleftarrow{subj} \text{find} \xrightarrow{obj} \text{gold} \xrightarrow{conj} Y$	$X \xleftarrow{subj} \text{find} \xrightarrow{obj} Y$
apposition	$Z \xrightarrow{appo} Y \implies Y$	$X \xleftarrow{subj} \text{find} \xrightarrow{obj} \text{protein} \xrightarrow{appo} Y$	$X \xleftarrow{subj} \text{find} \xrightarrow{obj} Y$
abbreviation	$Z \xrightarrow{spellout} Y \implies Y$	$X \xleftarrow{subj} \text{find} \xrightarrow{obj} \text{NDA} \xrightarrow{spellout} Y$	$X \xleftarrow{subj} \text{find} \xrightarrow{obj} Y$

**Table 3:** Some of the syntactic rules used in our implementation, together with usage examples (the application of the second rule and the third rule is demonstrated in Figure 1).



**Fig. 1:** Chaining of canonization rules that transforms the path template between the arguments  $\{X='Google'; Y='Sprinks'\}$ , which occurs in the sentence “We witnessed the acquisition of Kaltix and Sprinks by another growing company, Google”, into a canonized template form. The first rule applied is a nominalization rule, followed by removal of apposition and removal of conjunction (as described in Table 3). As can be seen, applying the rules in any order will result in the same final canonized form.

contexts under which it holds. However, it is difficult to explicitly define when a learned rule should be considered correct under this methodology.

Instead, we follow the evaluation methodology presented in [19], where each rule  $L \rightarrow R$  is evaluated by presenting the judges not only with the rule but rather with a sample of sentences that match its left hand side  $L$ . The judges then assess whether the rule holds under each specific example sentence. The precision of a rule is computed by the percentage of examples for which entailment holds out of all “relevant” examples in the judged sample. A rule is considered correct if its precision is higher than 0.8 (see [19] for details). This instance-based approach was shown to be more reliable than the rule-based approach.

## 4.2 TEASE Evaluation

We separated the templates that were learned by  $TEASE_c$  into two lists: (a) a *baseline-templates* list containing templates also learned by  $TEASE_b$ ; (b) a *new-templates* list containing templates that were not learned by  $TEASE_b$ , but learned by  $TEASE_c$  thanks to the improved statistics. In total, 3871 templates were learned: 3309 in the baseline-templates list and 562 in the new-templates list. Inherently, every output template learned by  $TEASE_b$  is also learned in its canonical form by  $TEASE_c$ , since its supporting statistics may only increase.

We randomly sampled 100 templates from each list and evaluated their correctness according to the methodology in Section 4.1. We retrieved 10 example sentences for each rule from the first CD of Reuters RCV1. Two judges, fluent English speakers, evaluated the examples. We randomly split the rules between the judges with 100 rules (942 examples) cross annotated for agreement measurement.

**Results** First, we measured the redundancy in the rules learned by  $TEASE_b$  to be 6.2% per input template on average. We considered only morpho-syntactic phenomena that are addressed in our implementation. This redundancy was eliminated using the canonization module.

Next, we evaluated the quality of each rule sampled using two scores: (1) micro average **Precision**, the percentage of correct templates out of all learned templates, and (2) average **Yield**, the average number of correct templates learned for each input template, as extrapolated for the sample. The results are presented in Table 5. The agreement between the judges was measured by the Kappa value [4], which is 0.67 on the relevant examples (corresponding to substantial agreement).

We expect  $TEASE_c$  to learn new rules using the canonization module. In our experiment, 5.8 more correct templates were learned on average per input template by  $TEASE_c$ . This corresponds to an increase of 11.6% in average Yield (see Table 5). Examples of new correctly

Rule	Sentence	Judgment
$X$ clarify $Y \rightarrow X$ prepare $Y$	He didn't clarify <b>his position on the subject</b> .	Left not entailed
$X$ hit $Y \rightarrow X$ approach $Y$	<b>Other earthquakes</b> have hit <b>Lebanon</b> since '82.	Irrelevant context
$X$ regulate $Y \rightarrow X$ reform $Y$	<b>The SRA</b> regulates <b>the sale of sugar</b> .	No entailment
$X$ stress $Y \rightarrow X$ state $Y$	<b>Ben Yahia</b> also stressed <b>the need for action</b> .	Entailment holds

**Table 4:** Example sentences for rules and their evaluation judgment.

learned templates are shown in Table 6.

There is a slight decrease in precision when using  $TEASE_c$ . One possible reason is that the new templates are usually learned from very few occurrences of different variations, accumulated for the canonical templates. Thus, they may have a somewhat lower precision in general. Overall, the significant increase in Yield is much more important, especially if the learned rules are later filtered manually (see Section 2.1).

Template List	Avg. Precision	Avg. Yield
$TEASE_b$	30.1%	49.8
$TEASE_c$	28.7%	55.6
$DIRT_b$	24.7%	46.9
$DIRT_c$	24.9%	47.5

**Table 5:** Average Precision and Yield of the output lists.

Input Template	Learned Template
$X$ accuse $Y$	$X$ blame $Y$
$X$ approve $Y$	$X$ take action on $Y$
$X$ demand $Y$	$X$ call for $Y$ , $X$ in demand for $Y$
$X$ establish $Y$	$X$ open $Y$
$X$ hit $Y$	$X$ slap $Y$
$X$ invent $Y$	grant $X$ patent on $Y$ , $X$ is co-inventor of $Y$
$X$ kill $Y$	$X$ hang $Y$ , charge $X$ in death of $Y$
$X$ named as $Y$	hire $X$ as $Y$ , select $X$ as $Y$
$X$ quote $Y$	$X$ cite $Y$
$X$ tell $Y$	$X$ persuade $Y$ , $X$ say to $Y$
$X$ worsen $Y$	$X$ impair $Y$

**Table 6:** Examples for correct templates that  $TEASE$  learned only after using canonization rules.

### 4.3 DIRT Evaluation

Unlike  $TEASE$ ,  $DIRT$  has a very long noisy tail of candidate templates (see Section 2.1). However,  $DIRT$  poses no hard threshold for filtering out this long tail. Instead, we follow [10], who evaluated only the top- $N$  templates learned for each input template. [10] set  $N$  to be 40, but this choice seems quite arbitrary. We set  $N$  to be 190 to assess an output list that is similar in size to  $TEASE$ 's output. Before selecting the top 190 templates, we removed redundant templates from  $DIRT_b$ , those that are just morpho-syntactic variations of a template with a higher score. We converted the remaining templates to their canonical forms.

We separated the templates learned for each input template into three lists: (a) a *common-templates* list containing templates that appear in both  $DIRT_b$  and  $DIRT_c$  top-190 lists; (b) a *new-templates* list containing templates that appear only in the  $DIRT_c$  list; (c) an *old-templates* list containing templates that appear only in the  $DIRT_b$  list. Out of the 3800 templates selected from each  $DIRT$  version output, 3353 were in the common-list and 447 were in each of the new/old lists.

We sampled 100 templates from each list and evaluated their correctness (10 sentences for each rule). One judge evaluated the sample. The evaluation results were affirmed by an additional evaluation by one of the authors.

**Results** We measured the redundancy in the rules learned by  $DIRT_b$  to be 5.6% per input template on average. This redundancy was removed using the canonization module. We found that only about 13% of the learned templates were learned by both  $TEASE$  and  $DIRT$ . This shows that the algorithms do not compete but rather largely complement each other in terms of Yield, since they learn from different resources.

13.3% of the top-190 templates learned by  $DIRT_b$  were replaced by other templates in  $DIRT_c$ , as the change in statistics results in different template ranking. We measured Precision and Yield as in Section 4.2. The results are presented in Table 5.

As can be seen, the performance of  $DIRT_c$  is basically comparable to that of  $DIRT_b$ . It seems that in typical paraphrase acquisition algorithms like  $TEASE$ , which use complex and more informative features that are infrequent, adding more statistics results in higher quality learning. On the other hand,  $DIRT$  is based on frequent simple features that are less informative. Under this approach, adding some more statistics does not seem to dramatically change the overall score of a rule. Perhaps a more substantial increase in the statistics, such as by adding more canonization rules, will result in a positive change.

Overall, it is useful to incorporate canonization also in  $DIRT$  in order to remove the redundancy within the learned rules but also to enable a uniform architecture for applying rules learned by different algorithms.

### 4.4 Relation Extraction Evaluation

To illustrate the potential contribution of the increased number of learned rules we conducted a small-scale experiment in a Relation Extraction (RE) setup over a data-set of protein interactions [3]. The task is to identify pairs of proteins that are described in a text as interacting.

We have set a simple partial replication of the RE configuration presented in [14]. We used ' $X$  interact with  $Y$ ' as the only input template for both  $TEASE_b$  and  $TEASE_c$ , which learned entailment rules containing this template

from the Web. We then extracted protein pairs using the rules learned. For canonization at inference time, we used only the rules described in Section 3.1 (a wider range of matching techniques should be used in order to reach higher recall).

Table 7 presents the results of our two TEASE versions for a test set of about 600 mentions of interacting pairs. There is a relative improvement of about 10% in recall, which reflects the yield increase in  $TEASE_c$ . These results are preliminary and of small scale, but they illustrate the potential benefit of learning with canonization.

We note that TEASE precision in this experiment, which was measured over actual applications of the learned rules in the test set, is much higher than that of Section 4.2, where the percentage of correctly learned rules was measured. This shows that many incorrectly learned rules are not applicable in typical contexts and thus rarely deteriorate overall performance.

Implementation	Recall	Precision
$TEASE_b$	9.4%	83%
$TEASE_c$	10.4%	87.5%

**Table 7:** Results for the protein interaction setup using TEASE with and without canonization.

## 4.5 Analysis

Parser errors are one of the main reasons that variations are sometimes not transformed into their canonical form. These errors result in different parse trees for the same syntactic constructs. Thus, several parser dependent rules may be needed to capture the same phenomenon. Moreover, it is difficult to design canonization rules for some parsing errors, since the resulting parse trees consist of structures that are common to other irrelevant templates. For example, when Minipar chooses the head of the conjunct ‘Y’ in “The interaction between X and Y will not hold for long” to be ‘interaction’ and not ‘X’, the appropriate nominalization rule cannot be applied. These errors affect both the learning phase, where statistics are not accumulated to the appropriate canonical form, and the inference phase, where a variations of a canonical rule are not recognized.

Finally, we note that the reported results correspond only to the phenomena captured by our currently implemented canonization rules. Adding more rules that cover more morpho-syntactic phenomena is expected to increase the performance obtained by our canonization scheme. For example, there are many nominalizations that are not specified in the current Nomlex version, but can be found in other resources, such as WordNet [6].

## 5 Conclusions

We proposed a modular approach for addressing morpho-syntactic variations of templates when learning entailment rules, based on rule canonization. We then used it for template canonization in two state-of-the-art acquisition algorithms. Our experiments showed that redundancy is removed while new correct rules are learned. We also showed initial improvement in a Relation Extraction setting when using the additional rules learned with the canonization

module. Finally, we suggest that the evaluation of rules in a canonical form is more accurate, since the bias for templates with many frequent variations learned is removed.

In future work we plan to investigate other types of entailment knowledge that can contribute to canonization, such as synonyms. We also plan to add additional syntactic and morpho-syntactic rules, which were not covered yet.

## Acknowledgements

The authors would like to thank Chen Erez for her help in the experiments. We also want to thank Efrat Brown, Ruthie Mandel and Malky Rabinowitz for their evaluation. This work was partially supported by the Israeli Ministry of Industry and Trade under NEGEV Consortium ([www.negev-initiative.org](http://www.negev-initiative.org)) and the IST Programme of the European Community under the PASCAL Network of Excellence IST-2002-506778.

## References

- [1] R. Bar-Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. The second pascal recognising textual entailment challenge. In *Second PASCAL Challenge Workshop for Recognizing Textual Entailment*, 2006.
- [2] R. Barzilay and L. Lee. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of NAACL-HLT*, 2003.
- [3] R. Bunescu, R. Ge, K. J. Rohit, E. M. Marcotte, R. J. Mooney, A. K. Ramani, and Y. W. Wong. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine (Special Issue on Summarization and Information Extraction from Medical Documents)*, 2004.
- [4] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960.
- [5] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. *Lecture Notes in Computer Science*, 3944:177–190, 2006.
- [6] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. Language, Speech and Communication. MIT Press, 1998.
- [7] O. Gurevich, R. S. Crouch, T. H. King, and V. de Paiva. Deverbal nouns in knowledge representation. In *Proceedings of FLAIRS*, 2006.
- [8] S. Harabagiu and A. Hickl. Methods for using textual entailment in open-domain question answering. In *Proceedings of ACL*, 2006.
- [9] D. Lin. Dependency-based evaluation of minipar. In *Proceedings of the Workshop on Evaluation of Parsing Systems at LREC*, 1998.
- [10] D. Lin and P. Pantel. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360, 2001.
- [11] C. Macleod, R. Grishman, A. Meyers, L. Barrett, and R. Reeves. Nomlex: A lexicon of nominalizations. *Proceedings of EURALEX*, 1998.
- [12] C. Quirk, C. Brockett, and W. Dolan. Monolingual machine translation for paraphrase generation. In *Proceedings of EMNLP*, 2004.
- [13] D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. In *Proceedings of ACL*, 2002.
- [14] L. Romano, M. Kouylekov, I. Szpektor, I. Dagan, and A. Lavelli. Investigating a generic paraphrase-based approach for relation extraction. In *Proceedings of EACL*, 2006.
- [15] T. Ron. Generating entailment rules using online lexical resources. *Masters thesis, Computer Science Department, Bar Ilan University*, 2006.
- [16] S. Sekine. Automatic paraphrase discovery based on context and keywords between ne pairs. In *Proceedings of IWP*, 2005.
- [17] S. Sekine. On-demand information extraction. In *Proceedings of the COLING/ACL Main Conference Poster Sessions*, 2006.
- [18] Y. Shinyama, S. Sekine, K. Sudo, and R. Grishman. Automatic paraphrase acquisition from news articles. In *Proceedings of HLT*, 2002.
- [19] I. Szpektor, E. Shnarch, and I. Dagan. Instance-based evaluation of entailment rule acquisition. In *Proceedings of ACL*, 2007.
- [20] I. Szpektor, H. Tanev, I. Dagan, and B. Coppola. Scaling web-based acquisition of entailment relations. In *Proceedings of EMNLP*, 2004.



# Sub-word Based Language Modeling for Amharic

Martha Yifiru Tachbelie, Wolfgang Menzel

Department of Informatik, University of Hamburg, Germany

{tachbeli, menzel}@informatik.uni-hamburg.de

## Abstract

This paper presents sub-word based language models for Amharic, a morphologically rich and under-resourced language. The language models have been developed (using an open source language modeling toolkit - SRILM) with different n-gram order (2 to 5) and smoothing techniques. Among the developed models, the best performing one is a 5gram model with modified Kneser-Ney smoothing and with interpolation of n-gram probability estimates.

## Keywords

Language modeling, sub-word based language modeling, morph-based language modeling, Amharic.

## 1. Introduction

### 1.1. Amharic Word Morphology

Amharic is a member of the Ethio-Semitic languages, which belong to the Semitic branch of the Afro-Asiatic super family [23]. It is related to Hebrew, Arabic, and Syrian. Amharic is a major language spoken mainly in Ethiopia. According to the 1998 census, it is spoken by over 17 million people as a first language and by over 5 million as second language throughout different regions of Ethiopia. Amharic is also spoken in other countries such as Egypt and Israel [4].

Like other Semitic languages such as Arabic, Amharic exhibits a root-pattern morphological phenomenon. A root is a set of consonants (called radicals) which has a basic 'lexical' meaning. A pattern consists of a set of vowels which are inserted (intercalated) among the consonants of the root to form a stem. The pattern is combined with a particular prefix or suffix to make a single grammatical form [3] or to form another stem [2]. For example, the Amharic root *sbr* means 'break', when we intercalate the pattern *ä-ä* and attach the suffix *ä* we get *säbbärä* 'he broke' which is the first form of a verb (3<sup>rd</sup> person masculine singular in past tense as in other semitic languages) [3]. In addition to this non-concatenative morphological feature, Amharic uses different affixes to form inflectional and derivational word forms.

Some adverbs can be derived from adjectives but, adverbs are not inflected. Nouns are derived from other basic nouns, adjectives, stems, roots, and the infinitive form of a verb by affixation and intercalation. For example, from

the noun *läḡ* 'child' another noun *läḡnät* 'childhood'; from the adjective *däg* 'generous' the noun *däḡnät* 'generosity'; from the stem *sənəf*, the noun *sənəfna* 'laziness'; from root *qld*, the noun *qäləd* 'joke'; from infinitive verb *mäsəbär* 'to break' the noun *mäsəbäriya* 'an instrument used for breaking' can be derived.

Case, number, definiteness, and gender marker affixes inflect nouns. Table 1 presents, as an example, the genitive case markers that inflect nouns.

Table 1. Genitive case markers (Adapted from [21])

Person	Singular		Plural
	Vowel ending	Consonant ending	
1 <sup>st</sup>	-ye	-e	-aččn
2 <sup>nd</sup> masculine	-h	-ih	-ačču
2 <sup>nd</sup> feminine	-š	-iš	
2 <sup>nd</sup> polite	-wo	-wo	-aččäw
3 <sup>rd</sup> masculine	-w	-u	
3 <sup>rd</sup> feminine	-wa	-wa	
3 <sup>rd</sup> polite	-aččäw	-aččäw	

Adjectives are derived from nouns, stems or verbal roots by adding a prefix or a suffix. For example, it is possible to derive *dənəgayama* 'rocky' from the noun *dənəgay* 'rock, stone'; *zənəgu* 'forgetful' from the stem *zənəg*; *sänäf* 'lazy' from the root *s\_n\_f* by suffixation and intercalation. Adjectives can also be formed through compounding. For instance, *hodäsäfi* 'tolerant, patient', is derived by compounding the noun *hod* 'stomach' and the adjective *säfi* 'wide'. Like nouns, adjectives are inflected for gender, number, and case [2].

Unlike the other word categories such as noun and adjectives, the derivation of verbs from other parts of speech is not common. The conversion of a root to a basic verb stem requires both intercalation and affixation. For instance, from the root *gdl* 'kill' we obtain the perfective verb stem *gäddäl-* by intercalating pattern *ä ä*. From this perfective stem, it is possible to derive passive stem (*tägäddäl-*) and causative stem (*asgäddäl-*) using prefixes *tä-* and *as-*,

<sup>1</sup> For transcription purpose, IPA representation is used with some modification.

respectively. Other verb forms are also derived from roots in a similar fashion.

Verbs are inflected for person, subject, object, gender, number, and tense [2]. Table 2 shows how a perfective Amharic verb inflects for person, subject, gender and number. Other elements like negative markers also inflect verbs in Amharic.

**Table 2. Inflection of a perfective verb**

Person	Singular	Plural
1 <sup>st</sup>	säbbärku/hu	säbbärn
2 <sup>nd</sup> masculine	säbbärh/k	
2 <sup>nd</sup> feminine	säbbärš	säbbäräčču
2 <sup>nd</sup> polite	säbbäru	
3 <sup>rd</sup> masculine	säbbärä	
3 <sup>rd</sup> feminine	säbbäräčč	säbbäru
3 <sup>rd</sup> polite	säbbäru	

From the above brief description of Amharic word morphology it can be seen that Amharic is a morphologically rich language. It is this feature that makes development of language models for Amharic challenging. The problems posed by Amharic morphology to language modeling were illustrated by [17] who, therefore, recommended the development of sub-word based language models for Amharic.

## 1.2. Language Modeling

In language modeling, the problem is to predict the next word given the previous words [13]. It is fundamental to many natural language applications such as automatic speech recognition (ASR) and statistical machine translation (SMT). LM has also been applied to question answering, text summarization, paraphrasing and information retrieval [5].

The most widely used language models are statistical language models. They provide an estimate of the probability of a word sequence  $W$  for a given task. The probability distribution depends on the available training data and how the context has been defined [10]. [25] indicated that large amounts of training data are required in statistical language modeling so as to ensure statistical significance.

Even if we have a large training corpus, there may be still many possible word sequences which will not be encountered at all, or which appear with a statistically non-significant frequency (data sparseness problem) [25]. In morphologically rich languages, there are even individual words that might not be encountered in the training data irrespective of its size (Out of Vocabulary words problem).

Morphologically rich languages have a high vocabulary growth rate which results in high perplexity and a large number of out of vocabulary words [22]. As a solution, sub-word units are used in language modeling to improve the quality of language models and consequently the performance of applications that use the language models ([6]; [24]; [9]; [12]; [8]).

We have developed sub-word (morpheme-based) language models for Amharic. As to our knowledge, this is the first attempt made for this language. Section 2 presents the development of the language models and the perplexity results obtained. But, before that we would like to discuss about the evaluation metrics used in language modeling.

## 1.3. Evaluation Metrics

The best way of evaluating language models is measuring its effect on the specific application for which it was designed [15]. However this is computationally expensive and hard to measure. An alternative is to evaluate a language model by the probability it assigns to some unseen text (test set), a text which is not used during model training. Better model will assign a higher probability to the test data [11]. Both cross entropy and perplexity are computed on the basis of this probability.

Cross-entropy of a language (sequence of words)  $W$  according to a model  $m = P(w_i/w_{i-N+1} \dots w_{i-1})$  can be calculated as:

$$H(W) = -\lim_{N \rightarrow \infty} \frac{1}{N} \log P(w_1 w_2 \dots w_N) \quad (1)$$

Where,  $N$  is the number of tokens in a test text. When  $N$  is sufficiently large, cross entropy can be calculated based only on our probability model as follows:

$$H(W) \approx -\frac{1}{N} \log P(w_1 w_2 \dots w_N) \quad (2)$$

This measures the average surprise of the model in seeing the test set and the aim is to minimize this number. Cross entropy is inversely related to the probability assigned to the words in the test data by the model. That means a high probability leads to a low cross entropy.

Perplexity is a related evaluation metric, which is used most commonly and computed as:

$$PP = 2^{H(W)} \quad (3)$$

$$= P(w_1 w_2 \dots w_N)^{-\frac{1}{N}} \quad (4)$$

$$= \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i/w_1 \dots w_{i-1})}} \quad (5)$$

Perplexity can be interpreted as the branching factor of a language model. Therefore, models with low perplexity values are better models. As it can be seen from equation 5, a higher conditional probability of the word sequence leads to lower perplexity. Thus, minimizing perplexity is equivalent to maximizing the test set probability [11]. Because perplexity is the most commonly used evaluation metric, we also evaluated our language models on the basis of perplexity values.

Since the calculation of both cross entropy and perplexity is based on the number of tokens in a test set, vocabularies must be the same when perplexities or cross entropies are compared. Otherwise, the measures are not comparable. When we have different token counts, models can only be compared on the basis of the probability they assign to the test sets.

## 2. Data Preparation

### 2.1. The Corpus

A text corpus consisting of 48,090 sentences and 1,542,697 tokens has been prepared. The electronic text is obtained from ethiozenna archive which contains written newscast. Since the target application domain is speech recognition, the text has been normalized accordingly.

After normalization, the text corpus has been merged with another one prepared by [17] from the same domain. The combined text corpus, used in the experiment, consists of 120,261 sentences or 2,348,151 tokens or 211,178 types. Table 3 presents the frequency distribution of words in the combined text corpus.

**Table 3. Word frequency distribution**

Frequency	Number of words
1	121329
2 - 10	69538
11 - 100	17358
101 - 1000	2655
1001 - 10000	293
10001 - 20000	3
above 20000	2

As it can be noted from Table 3, more than 50% (121,329) of the words occur only once (hapax legomena) in the corpus. This indicates the morphological richness of the Amharic language. Although much effort has been exerted to clean the data, there are still misspelled words and

correcting them is difficult, as there is no available spelling checker for the language. The existence of misspellings may also contribute to the large number of hapaxes. However, our corpus is not the only one to include large number of hapaxes. Zemánek (2005) indicated that CLARA (Corpus Linguae Arabicae), an Arabic corpus, consists of more than 50% hapax legomena. On the other hand, in our corpus only 5 words appear with a frequency of above 10000. These words are function words such as wəsəṭ 'in'.

### 2.2. Morphological Analysis

Developing a sub-word language model requires to have a word parser which splits word forms into its constituents. Different people ([1]; [20]; [16]) have attempted to develop morphological analyzer for Amharic using different methods. However, none of the systems can be directly used for this project. The systems developed by [1] and [20] suffer from lack of data. The morphological analyzer developed by [16] seems to exhibit a dearth of lexicon. It has been tested on 207 words and it analyzed less than 50% (75 words) of the words. Moreover, the output of the system is not directly useful for this project which needs the morphemes themselves instead of their morphological features. Since the source code of the analyzer is not yet made available, it is not possible to customize it.

An alternative approach is offered by unsupervised corpus-based methods which do not need annotated data. These methods are particularly interesting for resource scarce languages like Amharic.

Two freely available, language independent unsupervised morphology learning tools have been identified: Linguistica [7] and Morfessor [14]. Both tools have been tried on a subset of our corpus (9996 sentences). Unfortunately, it has been found out that Linguistica divides every word into exactly two constituents even if a word actually consists of more than two morphemes. Thus, Morfessor which tries to identify all the morphemes found in a word has been used for the subsequent experiments.

Morfessor requires a list of words as an input. The developers of Morfessor found out that Morfessor, evaluated on Finnish and English data sets, gives better morph segmentation when it is provided with a list of word types. To compare these findings with the situation in Amharic, two word lists have been prepared from the corpus: a list of tokens and a list of types.

Since Morfessor has been trained on two different word lists, there are two outputs (morph segmentation) and, therefore, two kinds of morph-segmented corpora: `token_based_corpus` and `type_based_corpus`. `Token_based_corpus` is a morph corpus where the morphs have been found by analyzing the list of tokens whereas in `type_based_corpus` the morphs have been found by analyzing the word type list.

### 3. Experiments

#### 3.1. Morpheme-based Language Models

The tool used for language modeling purpose is SRI Language Modeling toolkit (SRILM) [19]. SRILM is a freely available open source language modeling toolkit.

Each corpus is divided into three parts: training set, development and evaluation test sets with a proportion of 80:10:10.

Trigram models with Good-Turing smoothing and Katz-backoff have been developed for both corpora. A significant difference in perplexity (860.47 for the token\_based\_corpus and 117.43 for the type\_based\_corpus) has been observed. The reason for this difference might be due to the fact that the number of unsegmented words in token\_based\_corpus (45,767) is greater than that of the type\_based\_corpus (11,622). This conforms to the finding of [14] that segmentation is less common when word tokens are used as data. Accordingly, only the type\_based\_corpus has been used for subsequent experimentation.

N-gram models of order 2 to 5 have been tried. The effect of different smoothing techniques (Good-Turing, Absolute discounting, Witten-Bell, Natural discounting, modified and unmodified Kneser-Ney) on the quality of language models has been studied. The best results obtained for each smoothing technique are presented in Table 4.

Table 4. Perplexity results

N-gram	Smoothing technique	Perplexity
4gram	Good-Turing with Katz backoff	113.24
5gram	Absolute Discounting with 0.7 discounting factor	112.79
5gram	Witten-Bell	110.88
5gram	Natural Discounting	117.37
4gram	Modified Kneser-Ney	107.54
5gram	Unmodified Kneser-Ney	103.63

As it can be seen from Table 4, the best performing model is a 5gram model with unmodified Kneser-Ney smoothing. This result is in line with the finding of [18] that Kneser-Ney and its variation outperform other smoothing techniques.

Probability estimates of different n-gram order have been interpolated for Witten-Bell, Absolute discounting and modified Kneser-Ney smoothing techniques. Interpolation has been tried only for these three smoothing techniques because SRILM toolkit supports interpolation only for them. Table 5 shows the best results for each smoothing technique.

Table 5. Perplexity results with interpolation

N-gram	Smoothing Techniques	Perplexity
4gram	Witten-Bell	112.1
5gram	Modified Kneser-Ney	101.38
4gram	Absolute Discounting with 0.7 discounting factor	118.38

Interpolating n-gram probability estimates at the specified order n with lower order estimates sometimes yield better models [19]. Our experiment verified this fact. A 5gram model with Kneser-Ney smoothing and interpolation of n-gram probability estimates has a perplexity of 101.38. For the other smoothing techniques an increase in perplexity has been observed. The best performing model has a perplexity of 102.59 on the evaluation test set.

As indicated by [19], discarding unknown words or treating them as a special “unknown word” token affects the quality of language models. Thus, unknown words<sup>2</sup> have been mapped to a special “unknown word” token for the best model indicated in Table 5 and an increase in perplexity (to 102.26) has been observed. This might be due to the fact that there are only 76 out of vocabulary words.

#### 3.2. Word-based Language Models

To compare these results, we have also developed word-based language models. For this purpose, we used the corpus from which the morph-segmented corpus has been prepared. Table 6 shows the perplexity of word-based models. The 5gram model with unmodified Kneser-Ney is the best model compared with the other word-based language models.

Table 6. Perplexity of word-based models

N-gram	Smoothing technique	Perplexity
3gram	Good-Turing with Katz backoff	1151.29
5gram	Absolute Discounting with 0.7 discounting factor	1147.04
5gram	Witten-Bell	1236
5gram	Natural Discounting	1204.14
4gram	Modified Kneser-Ney	1107.32
5gram	Unmodified Kneser-Ney	1078.16

Interpolation of n-gram probability estimates has also been tried for the three smoothing techniques for which SRILM

<sup>2</sup>sub-word units are considered as words in sub-word based language models

supports interpolation. As it can be seen from Table 7, improvement with interpolation has been achieved for a 5gram model with modified Kneser-Ney. The other two smoothing techniques have lower perplexity values without interpolation.

**Table 7. Perplexity of word-based models with interpolation**

N-gram	Smoothing Techniques	Perplexity
5gram	Witten-Bell	1241.41
5gram	Modified Kneser-Ney	1059.38
3gram	Absolute Discounting with 0.7 discounting factor	1158.63

The optimal quality has been obtained with 5gram language model with modified Kneser-Ney, interpolation of n-gram probability estimates, and a mapping of unknown words to a special “unknown word” token. This model has a perplexity of 879.25 and 873.01 on the development and evaluation test sets, respectively.

The perplexities of our word-based language models are very high compared to what has been reported by [17], where the maximum perplexity of a bi-gram word-based language model was 167.889. To discover the reason behind the difference, we have developed word-based language models using our corpus in the same fashion as [17] did.

In [17] HLStats, HBuild and HSGen modules of the HTK toolkit [25] have been used since the version of the HTK toolkit used did not incorporate HLM language modeling toolkit. HLStats create a bigram probability, HBuild converts the bigram language model into lattice format and HSGen generates sentences from the lattice and calculates the perplexity.

Using this method it has been possible to develop a bi-gram word-based language model with a perplexity of 239.45. The perplexity is high compared to the one reported by [17], but this is not a surprise to us since the size of the training corpus used in our experiment is larger.

The problem with this method is that it calculates the perplexity from automatically generated sentences and there is no guarantee for the correctness of these sentences. In addition, when the same experiment is conducted repeatedly, the perplexity values also vary from experiment to experiment, as the sentences generated are different. Therefore, we can not directly compare the perplexity of the word-based language models of our experiment with the one reported by [17] because the test sentences used to calculate the perplexities are completely different.

### 3.3. Influence of Data Quality

Although we expect that the high perplexity of our word-based language models to be mainly due to the morphological richness of the language, spelling errors

might also contribute. To estimate the influence of spelling errors, we have conducted two experiments.

For these experiments, two data sets have been prepared: data\_set\_I and data\_set\_II. About 10,000 sentences of our corpus have been manually checked for spelling errors and merged with the data used in [17] for the speech recognition experiments. This forms data\_set\_I that consists of 21,922 sentences and 425,359 tokens. Data\_set\_II is prepared in the same way except that the spelling errors in the 10,000 sentences have not been corrected. It consists of 21,917 sentences and 429,795 tokens. These data have been divided into training set, development and evaluation test set with a proportion of 80:10:10 and word-based language models have been developed.

**Table 8. Word-based models with data\_set\_I**

N-gram	Smoothing technique	Perplexity
4gram	Absolute Discounting with 0.7 discounting factor	981.464
4gram	Witten-Bell	1091.03
5gram	Natural Discounting	1013.81
3gram	Modified Kneser-Ney	970.285
3gram	Unmodified Kneser-Ney	940.046

**Table 9. Word-based models with data\_set\_II**

N-gram	Smoothing technique	Perplexity
4gram	Absolute Discounting with 0.7 discounting factor	988.073
5gram	Witten-Bell	1096.71
4gram	Natural Discounting	1022.22
3gram	Modified Kneser-Ney	986.471
3gram	Unmodified Kneser-Ney	955.999

As it can be observed from Table 8 and 9, the best models are the tri-gram models with unmodified Kneser-Ney smoothing for both data sets. The perplexity values are 940.046 and 955.999 for data\_set\_I and data\_set\_II, respectively. When n-gram estimates are interpolated, the four-gram models with modified Kneser-Ney smoothing have the lowest perplexity for both data sets, as shown in Table 10 and 11.

**Table 10. Interpolated word-based models data\_set\_I**

N-gram	Smoothing technique	Perplexity
4gram	Absolute Discounting with 0.7 discounting factor	979.125
4gram	Witten-Bell	1084.92
4gram	Modified Kneser-Ney	936.898

**Table 11. Interpolated word-based models data\_set\_II**

N-gram	Smoothing technique	Perplexity
4gram	Absolute Discounting with 0.7 discounting factor	987.89
4gram	Witten-Bell	1092.23
4gram	Modified Kneser-Ney	953.953

Mapping the out of vocabulary words to a special “unknown word” token reduced the perplexity of the best performing model developed using data\_set\_I by 349.487 (from 936.898 to 587.411). This model has a perplexity of 613.983 on the evaluation test set. For data\_set\_II, a perplexity reduction of 372.632 (from 953.953 to 581.321) have been observed as a result of mapping unknown words to “unknown word” token. The latter model has a perplexity of 578.627 on evaluation test set.

There is still a very high perplexity for the best models developed using data\_set\_I, which is free from spelling errors. This enables us to conclude that correcting spelling errors did not reduce the high perplexity of word-based models and, therefore, the sole source for the high perplexity is the morphological feature of the language.

### 3.4. Comparison of Sub-word and Word-based models

The perplexity values of word-based and morph-based models are not comparable as the test sets used have quite different token counts. In this case, it is better to consider the probability assigned to the test sets by the models. A model that assigns high probability is considered as a better model. To avoid underflow, log probabilities are considered and, therefore, we actually compared the log probabilities.

The total log probability of the best performing morph-based model (A 5gram model with Kneser-Ney smoothing and interpolation of n-gram probability estimates, indicated in Table 4) is -834495. Whereas, the corresponding word-based model has a total log probability of -705218. Table 12 depicts the log probabilities of best morph-based model and the corresponding word based model which has a perplexity of 1059.38 (see Table 7).

**Table 12. Log probabilities I**

Models	Log Probabilities
Best performing morph-based model	-834495
Corresponding word-based model	-705218

The best performing word-based language model (5gram model with unmodified Kneser-Ney, interpolation of n-gram probabilities, and mapping of unknown words to “unknown word” token) has a total log probability of -726095, while

the total log probability of the corresponding morph-based model is -836215 although its perplexity is 102.26. Table 13 shows this fact. This tells us that word-based models have high log probability and, therefore, are the better models although their perplexity is higher.

**Table 13. Log probabilities II**

Models	Log Probabilities
Best performing word-based model	-726095
Corresponding morph-based model	-836215

On the other hand, sub-word based language models offer the benefit of reducing the out of vocabulary words rate from 13,500 to 76. This is a great achievement, as the out of vocabulary words problem is severe in morphologically rich languages in general, and Amharic in particular.

## 4. Conclusion

In this paper we described an attempt to develop sub-word based language models for Amharic. Since Amharic is one of the less resourced languages, we have used freely available softwares or toolkits (Morfessor for morphological parsing and SRILM for language modeling) in the course of our experiment.

Substantial reduction in the out of vocabulary rate, which is a severe problem in morphologically rich languages, has been observed as a result of using sub-words. In this regard, using sub-word units is preferable for the development of language models for Amharic. Low perplexity values have been obtained with morph-based language models. However, when comparing the quality based on the probability assigned to the test sets, word-based models seem better. Therefore, recognition experiments will be necessary to study the utility of the models in a particular application scenario.

We also observed that the output of the morphological analyzer consists of unsegmented words that should have been segmented. Efforts along this line might also improve the morph-based model.

No attempt has been made so far to deal with the non-concatenative root-pattern morphology of the language. A complete morphological decomposition of a semitic language will include affix segmentation as well as decomposition into root and pattern. Thus, a word in Amharic can be decomposed into root, pattern and one or more affix morphemes. Mere consideration of these morphemes as a language modeling unit might result in loss of word level dependencies since the root consonants of the words may stand too far apart. Therefore, new approaches, which capture word level dependencies, for modeling semitic languages in general, and Amharic in particular are

required. Building a separate model for root consonants and the other morphemes (patterns and affixes), and interpolating the models might help to capture word level dependencies. Currently, we are working in this direction.

## 5. Acknowledgment

We would like to thank University of Hamburg for financial support. Our thanks also goes to Solomon Teferra Abate who allowed us to use his text corpus.

## 6. References

- [1] Abiyot Bayou (2000) Developing Automatic Word Parser for Amharic Verbs and Their Derivation, M.Sc. thesis, Addis Ababa University, Addis Ababa.
- [2] Baye Yemam (1986 EC.) *yāamarāña sāwasāw*. Addis Ababa: EMPDE
- [3] Bender, M. L., J. D. Bowen, R. L. Cooper and C. A. Ferguson (1976) *Language in Ethiopia*. London: Oxford University Press.
- [4] Ethnologue (2004) Available at: [http://www.ethnologue.com/show\\_language.asp?code=AMH](http://www.ethnologue.com/show_language.asp?code=AMH)
- [5] Gao, Jianfeng and Lin Chin-Yew (2004) "Introduction to the Special Issue on Statistical Language Modeling" *ACM transactions On Asian Language Information Processing*. 3(2): 87-93
- [6] Geutner, P. (1995) Using morphology towards better large-vocabulary speech recognition systems. In *Proceedings of ICASSP*, 445-448.
- [7] Goldsmith, John. (2000) *Linguistica: An Automatic Morphological Analyzer*. The Proceedings from the Main Session of the Chicago Linguistic Society's Thirty-sixth Meeting. Arika Okrent and John Boyle (eds.) 36-1.
- [8] Hirsimäki, Teemu et. al. (2005) Morphologically Motivated Language Models in Speech Recognition. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR)*.
- [9] Ircing, P., P. Krebc, J. Hajic, S. Khudanpur, F. Jelinek, J. Psutka and W. Byrne (2001) On large vocabulary Continuous Speech Recognition of Highly Inflectional Language – Czech. In *Proceeding of the European Conference on Speech Communication and Technology*.
- [10] Juqua, Jean-Claude and Jean-Paul Haton (1996) *Robustness in Automatic Speech Recognition: Fundamentals and Applications*. London: Kluwer Academic Publishers.
- [11] Jurafsky, Daniel and James H. Martin (2006) *Speech and Language Processing: An Introduction to Speech Recognition, Computational Linguistics and Natural Language Processing*. Draft of 2<sup>nd</sup> edition. Available at: <http://www.cs.colorado.edu/~martin/slp2.html>
- [12] Kirchhoff, Katrin et al. (2002) *Novel Speech Recognition Models for Arabic*. Johns-Hopkins University Summer Research Workshop, Final Report. Available at: [http://ssli.ee.washington.edu/people/katrin/arabic\\_resources.html](http://ssli.ee.washington.edu/people/katrin/arabic_resources.html)
- [13] Manning, Christopher D. and Hinrich Schütze (1999) *Foundations of Statistical Natural Language Processing*. London: The MIT Press.
- [14] Mathias Creutz and Krista Lagus (2005) *Unsupervised Morpheme Segmentation and Morphology Induction from Text Corpora using Morfessor 1.1*. Publications in Computer and Information Science, Report A81, Helsinki University of Technology.
- [15] Rosenfeld, Ronald (1997) *Statistical Language Modeling and N-grams*. Available at: <http://www.cs.cmu.edu/afs/cs/academic/class/11761-s97/WWW/tex/Ngrams.ps>.
- [16] Saba Amsalu and Dafydd Gibbon (2005) *Finite State Morphology of Amharic*. In *Proceedings of RANLP, Bulgaria*, P. 47 – 51.
- [17] Solomon Teferra Abate (2006) *Automatic Speech Recognition for Amharic*. Ph.D. Thesis Available at: <http://www.sub.uni-hamburg.de/opus/volltexte/2006/2981/pdf/thesis.pdf>
- [18] Stanley F. Chen and Joshua Goodman (1998) *An Empirical Study of Smoothing Techniques for Language Modeling*. Available at: <http://people.csail.mit.edu/regina/6864/slides/goodman.pdf>
- [19] Stolcke, Andreas (2002) *SRILM - An Extensible Language Modeling Toolkit*. Available at: <http://www.speech.sri.com/projects/srilm/>
- [20] Tesfaye Bayu (2002) *Automatic Morphological Analyzer for Amharic: An Experiment Employing Unsupervised Learning and Autosegmental Analysis Approaches*. M.Sc. Thesis, Addis Ababa University, Addis Ababa.
- [21] Titov, E. G. (1976) *The Modern Amharic Language*. Moscow: Nauka Publishing House.
- [22] Vergyri, Dimitra, Katrin Kirchhoff, Kevin Duh and Andreas Stolcke (2004) *Morphology-Based Language Modeling for Arabic Speech Recognition*. Available at: <http://www.speech.sri.com/cgi-bin/run-distill?papers/icslp2004-arabic-lm.ps.gz>.
- [23] Voigt, R. M. (1987) "The classification of central Semitic" *Journal of Semitic Studies*, 32: 1 21.
- [24] Whittaker, E. W. D. and P. C. Woodland (2000) *Particle-based language modeling*. In *Proceeding of International Conference on Spoken Language Processing*, Beijing, China.
- [25] Young, Steve, Dan Kershaw, Julian Odell, Dave Ollason, Valtcho Valtchev and Phil Woodland (2000) *The HTK Book*. Available at: <http://nesl.ee.ucla.edu/projects/ibadge/docs/ASR/htk/htkbook.pdf>.

# Resolving coreference using an outranking approach

Olivier Tardif

Université de Provence

29, avenue Robert-Schuman

13621 Aix-en-Provence cedex 1

olivier.tardif@etu.univ-provence.fr

Grégory SMITS

France Télécom division R&D

2, avenue Pierre Marzin

Lannion 22307, France

gregory.smits@orange-ftgroup.com

## Abstract

Most machine learning methods used for coreference resolution do not allow expert users much control over many aspects where expert knowledge would be useful. Indeed all the training, decision and much of the optimisation processes occur in a "black box" system, which furthermore makes it difficult to get a precise interpretation of the results. We propose a machine learning method (MCDA) which makes it possible to inject *a priori* knowledge in the system, and lets the user see its precise effect in the overall resolution process.

## 1 Introduction

Coreference (or *anaphora*) resolution is arguably one of the "hard problems" of NLP : since (Hobbs 78) much work has been devoted to the question, and even with constant (but slow) improvement of the results obtained in the past thirty years, no single algorithm really stands out from the others, and no single method can overcome all the difficulties of the job. This state of affairs calls for exploratory work : here we present the results of a new approach for coreference resolution based on multi-criteria decision aid (MCDA), a technique which allows to mix expert knowledge and machine learning, and also gives rather precise feedback on the impact of each information used as an attribute in the classification process.

In section 2 we describe the corpus we used, the pair extraction algorithm and the attributes used to describe each pair. Next, we detail the MCDA method. Finally, we compare our approach to standard machine learning methods for coreference resolution.

## 2 Background details

In recent years, machine learning techniques, for example (Soon *et al.* 01), (Ng & Cardie 02b) or (Yang *et al.* 03) have proven most effective for resolving coreference. However most machine learning methods do not allow for a detailed interpretation of the results : they act as a black box, and

the expert hardly has any control over the classification process. Also these methods do not allow to specify explicitly the relative importance and potential relations between the attributes used to characterize each instance.

Our coreference resolution task is comparable to the one in MUC-7 (Hirschman & Chinchor 97) : mainly the objective is to find in a given text, for all proper names denoting people, places and organizations all nominal expressions coreferent with them. In what follows we will review all steps of the processing prior to the training and classification stages.

### 2.1 The corpus

To train and test the algorithm, 80 texts (from 500 to 1000 words long) were selected from a larger corpus<sup>1</sup> of the french newspaper *le Monde* from 1989 to 1990. They were annotated, first automatically for POS tagging, syntactic relations and other information like number, gender and semantic class by TiLT, an NLP toolbox developed at Orange R&D labs, and then manually to specify the markables and all coreference relations between them.

We consider that two expressions are coreferent if they denote the same object in the real world. We tried to mark as coreferent only the cases that were not problematic, typically, coreference between singular pronouns, common and proper nouns, and we did not consider the problem of temporality in predication. For example, in a sentence like "Henry Higgins, who was formerly sales director off Sudsy Soaps, became president of Dreamy Detergents (Kibble & vanDeemter 00).", even though *sales director* and *president* cannot really be said to be coreferent because they refer to two temporally distinct "states" of *Henry Higgins*, here we consider them coreferent.

In the end, our 80 texts contain 3504 expressions in a coreference relation, distributed in 683

<sup>1</sup>over 8000 texts.



chains.

## 2.2 Extraction algorithm

We approach coreference resolution like a classification task, where each pair of potential coreferent expressions is submitted to a classifier that must "decide" whether the pair's elements are (or not) coreferent. This decision is based on an attribute vector describing the pair, and on statistical data gathered during a training phase. We thus have to build a set of pairs from the annotated texts : starting at the beginning of each text, for each markable  $m_i$  (pronoun, common noun or proper noun) we extract all the markables P preceding it. We then create a pair with  $m_i$  for each markable in P. Since coreference is a rare relation, the distribution of positive versus negative instances is rather skewed, as was noted in (Ng & Cardie 02a); to minimize this problem, we keep only the pairs where the first element is a proper noun.

We divide these pairs into 3 types, depending on the category of the second element ( $m_i$ ). The reason for this is that we noticed (Tardif 06) that the attributes used to describe each pair are not of equal importance depending on the category of the coreferring expressions. For example, the average distance between a proper name and a coreferent pronoun is smaller than between two coreferent proper nouns; also the typographic similarity is less relevant between a proper noun and a common noun than it is between two proper nouns<sup>2</sup>. The three classes are then : proper noun and proper noun (NPR-NPR), proper noun and common noun (NPR-NCOM) and proper noun and pronoun (NPR-PRO).

## 2.3 Attributes

Each extracted pair is associated with an attribute vector that describes it. There are different types of attributes, namely typographic, categorial or syntactic, some describing the instance pair (e.g the distance, words in common or agreement features between both expressions), and other describing the context of the expressions it contains (e.g. *isParallel*, which specifies the fact that both expressions have the same grammatical function, or *isSubject*, which is true when the proper name in the pair is in subject position).

<sup>2</sup>But it is not completely unusable, e.g. the substring "president" in *President Johnson* versus *the president*.

## 3 A multicriteria decision problem

### 3.1 Taking expert knowledge into account in the sorting method

The problem we are faced with, consists on identifying valid antecedent/anaphora pairs from a set of extracted candidates. The decision of considering a candidate as a valid coreference case relies on the performances achieved on its associated attributes vector 2.3.

Several methods can be used to answer this problem, e.g. bayesian classifiers, decision trees, etc. However, these methods are all based on the interpretation of observed phenomena in a training corpus. The learning model acquired from a training phase is hardly interpretable for an expert, and intuitions and expert knowledge are difficult to integrate in such models.

So, in this paper, we propose to take advantage of MCDA methods and more precisely, an outranking approach called ELECTRE TRI (Roy 91). This method takes into account the intuitions and knowledge of a human expert as a preference model, which indicates the way attributes, considered as criteria, have to be computed to perform a sorting decision.

### 3.2 ELECTRE TRI

ELECTRE TRI is a MCDA method dedicated to sorting problems. This method assigns each candidate of a set  $A : \{a_1, a_2, \dots, a_n\}$  to one of the predefined classes  $C : \{c_1, c_2, \dots, c_m\}$ . A candidate  $a_i$  is assigned to a class  $c_k$  ( $a_i \in c_1$ ), if its performances achieved on the different criteria  $F : \{f_1, f_2, \dots, f_k\}$ ,  $f_j(a_i)$  being the performance of  $a_i$  on the  $j^{th}$  criterion, are acceptable with the limit profile  $L : \{l_1, l_2, \dots, l_k\}$  associated to  $c_k$ . In our case,  $C : c_0, c_1$  where  $c_1$  is the class of validated pairs. The way the candidates' performance vectors are compared to classes acceptability profiles is based on a preference model made from the intuitions and knowledge of a domain expert. This preference model is composed of :

- criteria weights  $W : \{w_1, w_2, \dots, w_k\}$ ;
- preference thresholds  $P : \{p_1, p_2, \dots, p_k\}$ , where  $p_j$  specifies the smallest difference between  $f_j(a_i)$  and  $l_j(c_1)$  compatible with the acceptance of  $a_i \in c_1$  on  $f_j$ ;
- indifference thresholds  $Q : \{q_1, q_2, \dots, q_k\}$ , where  $q_j$  specifies the largest difference between  $f_j(a_i)$  and  $l_j(c_1)$  that preserves indifference with the acceptance of  $a_i \in c_1$  on  $f_j$ ;

- veto thresholds  $V : \{v_1, v_2, \dots, v_k\}$  are used to filter candidates performing too weakly on a given criterion.
- a final cutting level  $lambda \in [0.5, 1]$ .

This preference model is used by ELECTRE TRI to establish outranking relations  $S$  between candidates  $a_i$  and defined classes  $c_j$ , where  $a_i S c_j$  means that  $a_i$  is considered an “acceptable” candidate for  $c_j$ . An outranking situation occurs if a sufficient majority of criteria (concordance value) validates the assertion of outranking and none of the criteria in the minority (discordance value) is opposed “too strongly” with this assertion. The concordance value  $c(a_i, c_1) = \frac{\sum_{j \in F} w_j \cdot c_j(a_i, c_1)}{\sum_{j \in F} w_j}$  is computed from partial concordance indices  $\forall j \in V$ :

$$c_j(a_i, c_1) = \begin{cases} 0, & \text{if } l_j(c_1) - f_j(a_i) \geq p_j \\ 1, & \text{if } l_j(c_1) - f_j(a_i) \leq q_j \\ ]0, 1[ & \text{otherwise} \end{cases}$$

Partial discordance indices are calculated  $\forall j \in F$ :

$$d_j(a_i, c_1) = \begin{cases} 1, & \text{if } l_j(c_1) - f_j(a_i) \geq v_j \\ ]0, 1[, & \text{if } p_j < l_j(c_1) - f_j(a_i) < v_j \\ 0, & \text{if } l_j(c_1) - f_j(a_i) \leq p_j \end{cases}$$

Finally, concordance and discordance values are merged in a credibility index  $\sigma(a_i, c_1) \in [0, 1]$ , which is then interpreted to establish an outranking relation:  $a_i S c_1$  if  $\sigma(a_i, c_1) \geq \lambda$ , where:

$$\sigma(a_i, c_1) = c(a_i, c_1) \cdot \prod_{j \in \bar{F}} \frac{1 - d_j(a_i, c_1)}{1 - c(a_i, c_1)}$$

$$\bar{F} = j \in F : d_j(a_i, c_1) > c(a_i, c_1)$$

Thus, in our application case  $a_i$  is considered as a coreference if  $\sigma(a_i, c_1) \geq \lambda$  then  $a_i S c_1$ .

### 3.3 Some heuristics to facilitate the determination of expert preferences

Despite this interesting property of being a fully parameterized method, it is sometimes hard for an expert to express his intuitions through numeric preferences. This is why, based on the manually annotated corpus (Sec. 2.1), we propose some heuristics to identify possible preferences for each considered criterion.

On the training part of the corpus (70%), three performance tables are built, one for each type of pair (NPR–NPR NPR–NCOM NPR–PRON). These tables bring together extracted candidate pairs, their manually assigned class (correct  $c_1$  or incorrect  $c_0$ ) and their associated computed

performance on each considered criterion.

We have used the features weighting algorithm RELIEF (Kononenko 94) to approximate iteratively the representativeness of each criterion for the class of the correct anaphora. At the end of the process, each criterion  $f_j$  is associated to a weight  $w_j \in [-1, 1]$ . A negative weight indicates that the criterion is not representative of the correct anaphora class. Positive weights are then reused for the concordance computation.

Moreover, from the performances tables, we establish on each criterion domain definition range the distribution curves of the number of correct candidates and incorrect candidates. We then interpret these curves to identify interesting phases corresponding to:

- correct anaphora acceptability threshold, where correct candidates begin to appear;
- a preference situation, where correct candidates emerge principally;
- an indifference space, where correct and incorrect candidates can not be clearly separated;
- a veto situation, where correct candidates are clearly separated from incorrect ones.

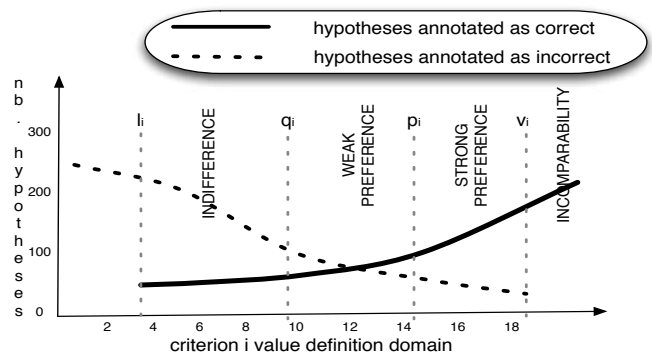


FIG. 1 – Distribution curves interpretation

Using such heuristics, we propose possible values for the construction of the preference model, which can then be refined and tuned by the expert. When a preference model is stabilized, we compute the credibility index of the relation  $a_i S c_1$  for each example of the performances table. We then search for the cutting level  $\lambda$  which separates, with the lowest error rate, the correct and incorrect candidates based on their computed credibility index.

## 4 An attempt at evaluation

To evaluate our approach, we have first established a baseline using a C4.5 decision tree<sup>3</sup>. 70% of each performance table (NPR-NPR, NPR-NCOM, NPR-PRON) has been used as training data to build the decision tree. On the testing corpus (30%), we have obtained the following results :

- NPR-NPR prec.=0.929; rec.=0.901; F-meas.=0.915
- NPR-NCOM prec.=0.742; rec.=0.2; F-meas.=0.323
- NPR-PRON prec.=0.375; rec.=0.207; F-meas.=0.267

Based on our own knowledge and intuitions about the anaphora resolution problem, we formalized the preferences about the way criteria had to be computed. Table 1 illustrates our own preferences about the 5 most important criteria we have identified (for display reasons, only weights are presented).

TAB. 1 – Preferences over criteria weights

NPR-NPR			
Expert model		inferred model	
Criteria	Weight	Criteria	Weight
wordsInCommon	0.25	isSimil	0.3636
sharedLexemes	0.25	diffWords	0.3150
distinctWords	0.15	subString	0.3054
subString	0.125	strSimil	0.2775
isAcronym	0.075	commWords	0.1885
strSimil	0.075	sharedLexemes	0.1859
NPR-NCOM			
Expert model		inferred model	
Criteria	Weight	Criteria	Weight
distExp	0.06	agrNum	0.1019
distTer	0.06	distExp	0.0658
agrGen	0.06	distTer	0.0651
appo	0.06	sameSentence	0.0582
sharedLexemes	0.06	isDefinite	0.0507
countOcc	0.03	agrGen	0.0361
NPR-PRON			
Expert model		inferred model	
Criteria	Weight	Criteria	Weight
agrNbrStrict	0.12	agrGen	0.0846
agrGenStrict	0.12	distExp	0.0821
agrNum	0.1	distTer	0.0771
agrGen	0.1	countOcc	0.0676
distExp	0.1	agrNum	0.0565
isSubj	0.1	distinctWords	0.0434
...	...	...	...

<sup>3</sup> implemented by the Weka java API (Witten & Eibe 05)

Using an expert preference model and an empirically defined cutting level of 0.75, the following results have been obtained :

- NPR-NPR prec.=0.865; rec.=0.952; F-meas.=0.907
- NPR-NCOM prec.=0.882; rec.=0.3051; F-meas.=0.405
- NPR-PRON prec.=0.94; rec.=0.405; F-meas.=0.566

We have then constituted a preference model with the values proposed by the heuristics presented in Sec. 3.3. Table 1 shows that using RELIEF our intuitions about the most important criteria have been validated, and also that "more accurate" weights are defined. We noticed that with the different heuristics, interesting cases of preference, indifference and veto thresholds naturally emerge, which were difficult to identify *a priori* by an expert.

Using this preference model and cutting levels : NPR-NPR  $\gamma = 0.55$ ; NPR-NCOM  $\gamma = 0.53$ ; NPR-PRON  $\gamma = 0.71$ , we have obtained the following results :

- NPR-NPR prec.=0.919; rec.=0.95; F-meas.=0.934
- NPR-NCOM prec.=0.443; rec.=0.544; F-meas.=0.488
- NPR-PRON prec.=0.852; rec.=0.448; F-meas.=0.587

Finally, the preference model obtained using heuristics 3.3 was refined in order to integrate complementary knowledge, like increasing the weight of criteria that have not been identified as relevant in the training corpus. For example, marginal phenomenon like acronyms or appositions which can be interesting features for respectively NPR-NPR and NPR-NCOM anaphora identification, are not frequent enough in the corpus to be considered by the RELIEF algorithm. Using such a mixed preference model and the cutting levels : NPR-NPR  $\gamma = 0.59$ ; NPR-NCOM  $\gamma = 0.51$ ; NPR-PRON  $\gamma = 0.69$ , we have obtained the following results :

- NPR-NPR prec.=0.924; rec.=0.945; F-meas.=0.944
- NPR-NCOM prec.=0.527; rec.=0.461; F-meas.=0.492
- NPR-PRON prec.=0.724; rec.=0.538; F-meas.=0.617

### 4.1 Some interpretations

For all three types of nominal pairs, we note a correspondence between expert intuitions and weights inferred from the data. In each case the same attributes are among the most significant for both weighting methods, e.g. `countOcc` (the number of times the first expression in a pair appears in the text) in NPR-PRON pairs, or the distance in words (`distTer`) and in nominal expressions (`distExp`) for NPR-NCOM pairs; etc.

There are however a few differences worth mentioning. First, the attributes `isAcronym` in NPR-NPR (the fact that one expression can be the acronym of the other) and `appo` in NPR-NCOM (both expressions in an apposition rela-

tion), highly weighted by the expert, do not get such a high score in the RELIEF method; conversely, `isDefinite` in NPR-NCOM is not given much importance by the expert's judgment, whereas the learning algorithm ranks it high.

Both cases illustrate advantages of a mixed approach like the one we propose. The former is an example of data sparseness, where the learning method ignores a relevant attribute when there are too few positive instances in the training corpus; indeed, `isAcronym` is true only in 13 cases on the 21017 NPR-NPR training instances<sup>4</sup>. This shows that a learning method can benefit from expert input for highly relevant but rare attributes. The second case shows that a corpus-based method can be used to optimize expert knowledge, here by increasing the weight of specific attributes.

Of course, learning methods rely heavily on the quality of the information present in the training corpus; this is not the case for expert knowledge. In our study a few attributes (e.g. `appo`; `isSubject`, when the pair's first element is in subject position; or `sameArgDomain`, when both expressions in the pair are arguments to the same predicate) depend on the results of a linguistic analysis. The more performant the analysis is, the less noise we will have, and, hopefully, the more these attributes will tend to be considered relevant by a method like RELIEF. This dependence on the preprocessing stage make the difference between both methods hard to quantify precisely.

From the point of view of coreference resolution, the results obtained clearly show the need for a distinct treatment of nominal pairs according to the category of their constituents. For example with NPR-NPR pairs the most relevant attributes are the ones pertaining to the general similarity of both expressions; these attributes are less significant for the other two types of pairs. Also, even though the difference between NPR-NCOM and NPR-PRON is not as marked as with NPR-NPR, we still can see that distance is a stronger attribute for NPR-PRON, as well as `countOcc`, the number of occurrences of the first expression in a pair.

<sup>4</sup>Note that `isAcronym` is always negative for NPR-NCOM and NPR-PRON pairs.

## 5 Perspectives and conclusion

In this paper we proposed an approach to coreference resolution that differs from previous work on two major points: the use of MCDA methods to identify valid coreference cases and the distinct treatment of candidate pairs according to their type. We showed that such method make possible the combination of human and statistical knowledge and that good results are obtained this way. Using the RELIEF method on a training corpus, we have noticed that the features associated to extracted candidate pairs have different weights functions of their types, which in our opinion justifies the use of three distinct processings rather than one for all candidate pairs, like previous approaches do.

## References

- [Hirschman & Chinchor 97] (Hirschman & Chinchor 97) Lynette Hirschman and Nancy Chinchor. Muc-7 coreference task definition. *Proceedings of MUC-7, Science Applications International Corporation*, 1997. 1
- [Hobbs 78] (Hobbs 78) Jerry R. Hobbs. Resolving Pronoun References. *Lingua*, 44 :311–338, 1978. 1
- [Kibble & vanDeemter 00] (Kibble & vanDeemter 00) Rodger Kibble and Kees van Deemter. On Coreferring. Coreference in MUC and Related Annotation Schemes. *Computational Linguistics*, 26 :4 :629–637, 2000. 1
- [Kononenko 94] (Kononenko 94) I. Kononenko. Estimating attributes: Analysis and extensions of relief. In *In proceedings of the European Conference on Machine Learning*, 1994. 3
- [Ng & Cardie 02a] (Ng & Cardie 02a) Vincent Ng and Claire Cardie. Combining Sample Selection and Error-Driven Pruning for Machine Learning of Coreference Rules. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002. 2
- [Ng & Cardie 02b] (Ng & Cardie 02b) Vincent Ng and Claire Cardie. Improving Machine Learning Approaches to Coreference Resolution. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002. 1
- [Roy 91] (Roy 91) B. Roy. The outranking approach and the foundations of electre methods. In *Theory and Decision*, pages 49–71, 1991. 2
- [Soon et al. 01] (Soon et al. 01) Wee Meng Soon, Tou Ng, Hwee, and Daniel Chung Yong Lim. A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, 27 :4, 2001. 1
- [Tardif 06] (Tardif 06) Olivier Tardif. Résoudre la coréférence à l'aide d'un classifieur bayésien naïf. *Actes de la 13e conférence sur le traitement automatique des langues naturelles (TALN2006)*, 2006. 2
- [Witten & Eibe 05] (Witten & Eibe 05) Ian H. Witten and Frank Eibe. *Data Mining: Practical machine learning tools and techniques*. 2005. 4
- [Yang et al. 03] (Yang et al. 03) Xiaofeng Yang, Guodong Zhou, Jian Su, and Chew Lim Tan. Coreference Resolution Using Competition Learning Approach. *Proceedings of ACL2003*, 2003. 1

# Comparing Document Segmentation Strategies for Passage Retrieval in Question Answering

Jörg Tiedemann  
University of Groningen  
Alfa Informatica,  
P.O. Box 716  
9700 AS Groningen, the Netherlands  
*j.tiedemann@rug.nl*

## Abstract

Information retrieval (IR) techniques are used in question answering (QA) to retrieve passages from large document collections which are relevant to answering given natural language questions. In this paper we investigate the impact of document segmentation approaches on the retrieval performance of the IR component in our Dutch QA system. In particular we compare segmentations into discourse-based passages and window-based passages with either fixed sizes or variable sizes. We also look at the effect of overlapping passages and sliding window approaches. Finally, we evaluate the different strategies by applying them to our question answering system in order to see the impact of passage retrieval on the overall QA accuracy.

## Keywords

passage retrieval, question answering, document segmentation, information retrieval

## 1 Introduction

Question answering (QA) systems commonly include a passage retrieval component to reduce the search space for information extraction modules when looking for appropriate answers of a given natural language question. Most systems rely on standard information retrieval (IR) techniques to retrieve relevant passages. In general, one prefers to work with textual units smaller than documents in QA systems. This is not only because of efficiency reasons but also because QA requires high recall in order to identify possible answers. Recall in general is improved by increasing the number of units retrieved but answer extraction methods are too expensive to work on a high number of large documents. Furthermore, answer extraction is less likely to make mistakes if the textual units are small and focused on relevant passages instead of documents that may contain a lot of extra information irrelevant to answering the question.

There are two general strategies to passage retrieval in QA [13]: (1) a two-step strategy of retrieving documents first and then selecting relevant passages within these documents (search-time passaging), and (2) a

one-step passage retrieval strategy (index-time passaging), see, for instance, [4]. Furthermore, in the first strategy we can distinguish between approaches that return only one passage per relevant document (for example the widely used Okapi model [14]; see [15] for a discussion on other algorithms) and the ones that allow multiple passages per relevant document to be returned (for instance [11]). In our QA system we adopt the second strategy (index-time passaging) using a standard IR engine to match keyword queries generated from a natural language question with passages in the index. Thus, we always allow multiple passages per document to be returned (which is also preferable according to [13]) and the IR engine decides for the overall ranking of all passages. The focus of this paper is to investigate the impact of different passaging approaches within the chosen setup.

Passage retrieval in QA is different from ordinary IR in at least two points: Firstly, queries are generated from user questions and not manually created as in standard IR. Secondly, the units to be retrieved are usually much smaller than documents in IR (as mentioned already). Here, the division of documents into passages is crucial. The textual units have to be big enough to ensure IR works properly and they have to be small enough to enable efficient and accurate QA. In this paper, we investigate the impact of changing the segmentation size on the retrieval performance and on the overall QA results. For this we look at fixed-sized and variable-sized segmentations using different degrees of redundancy. We compare our results with standard segmentations using the document structure.

The advantages of passage retrieval over full-text document retrieval has been investigated in various studies, see, e.g., [8, 1, 6, 7]. The main argument for passage retrieval is based on the normalization of textual units especially in cases where documents come from very diverse sources. In IR the task of comparing diverse documents with each other and with a given query is a serious problem and standard approaches have a lot of shortcomings when applying similarity measures to documents of various sizes and text types. The contents of the dataset we are working with is evidently very diverse. Most of the documents are very short but the longest one contains 625 sentences. The distribution of document sizes in our collection is plotted in figure 1.

Standard measures using, for instance, vector-space



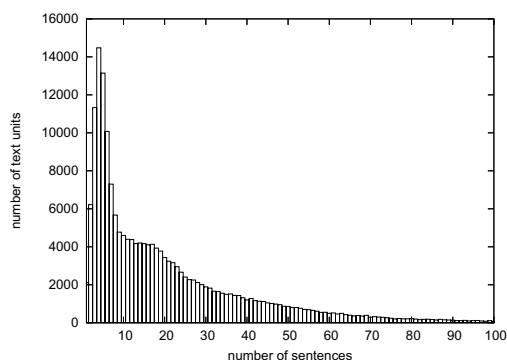


Fig. 1: Distribution of document sizes in terms of sentences they contain in the Dutch CLEF corpus

based models for ranking documents according to their relevance have often a strong bias for certain text types raising problems of discrimination between documents of different lengths and content densities. Passages on the other hand provide convenient units to be returned to the user avoiding such ranking difficulties [8]. We can distinguish two approaches to the incorporation of passages in information retrieval: (1) using passage-level evidence to improve document retrieval [1, 6] and (2) using passages directly as the unit to be retrieved [8, 7]. We are only interested in the second approach as we prefer small units to be returned in QA.

---

Minister Andriessen ( economische zaken ) wil dat ondernemers die door de watersnood in Limburg zijn gedupeerd , sneller en goedkoper krediet kunnen krijgen om hun bedrijf aan de gang te houden .

De rente van 7,5 procent op de middenstandskredieten moet worden gehalveerd en de provisie van 3 procent moet vervallen . Andriessen , gisteren voor de NOS-televisie : ” Het moet ook allemaal veel sneller dan gebruikelijk . De mensen moeten niet stikken in de papieren . ”

KNOV-voorzitter Kamminga antwoordde meteen dat niet voldoende te vinden : ” De klappen zijn daar z groot , dat voor velen een goedkope lening niet zal helpen . Er zal meer moeten gebeuren . ”

Pagina 3 :

VVD vraagt 100 miljoen voor rampgebied

Verscheidene ministers bespreken vanmiddag wat er voor het rampgebied moet gebeuren . Het ministerie van binnenlandse zaken kon gisteravond nog niet zeggen met welke voorstellen minister Dales naar het overleg komt . Het kabinet heeft 15 miljoen gulden toegezegd , maar enkele ministers hebben al laten weten dat dit bedrag moet worden verhoogd .

De Tweede-Kamerleden Van Rey en De Korte ( VVD ) hebben er bij minister Kok ( financin ) op aangedrongen voor de getroffen gebieden 100 miljoen gulden beschikbaar te stellen van de meevaller van ruim 4 miljard gulden op de rijksbegroting van 1993 .

---

Fig. 2: Discourse passages using paragraph markup.

Passages can be defined in various ways. An obvious way is to use logical divisions given in the documents such as sections and paragraphs. Existing markup or segmentation heuristics (such as empty lines) can be used to detect these units. Such segmentations based on document structure are known as **discourse passages** [8]. Problems with this approach often arise

with special structures such as headers, lists and tables which are easily mixed with other units such as proper paragraphs. Hence, discourse passages can vary substantially in terms of size and contents and similar problems as with standard IR may appear. An example segmentation of a document in our collection using existing paragraph markup is shown in figure 2.

Despite of the remaining diversity the variety in size is smaller at the paragraph level than at the document level and, therefore, the problems with a length bias in IR partly fades away.

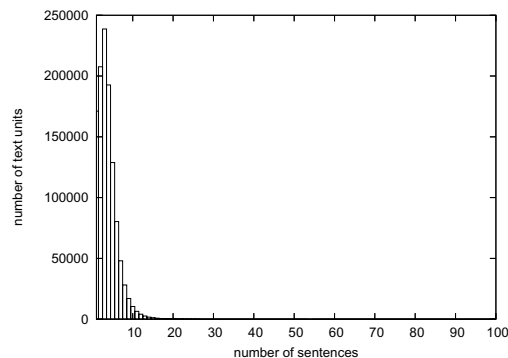


Fig. 3: Distribution of paragraph sizes in terms of sentences in the Dutch CLEF corpus

The distribution of paragraph sizes in our collection is plotted in figure 3. It shows that there is less divergence among paragraphs compared to the document size distribution. However, the longest paragraph still contains 156 sentences. The fact that we still have to deal with a large variety of paragraphs can be seen in figure 4 which plots the distribution of paragraph sizes in terms of characters.

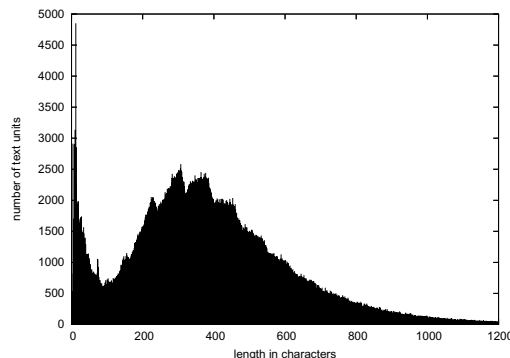


Fig. 4: Distribution of paragraph sizes in terms of characters in the Dutch CLEF corpus

Another type of passages are **semantic passages**. Here, the main idea is to split documents into semantically motivated units using some topical structure. TextTiling is an approach to such a segmentation using word frequencies to recognize topic shifts [5]. We do not include semantic passages in our experiments.

Finally, there are window-based passages that use fixed or variable-sized windows to segment documents into smaller units. Usually, windows are defined in terms of words or characters [8, 12]. However, sentences or paragraphs can also be used to define passage

windows [16, 10]. Commonly, window-based passages have a fixed length using non-overlapping parts of the document. However, dynamic definitions of windows have been proposed in order to create passages of variable lengths and starting positions, i.e., passages with overlapping parts [8, 12]. Arbitrary passages of fixed sizes can also be seen as sliding windows. More details are discussed later in the paper.

The remainder of the paper is organized as follows: Section 2 includes an overview of the retrieval component in our QA system and a detailed description of the various segmentations included in our experiments. In section 3 our results are shown and discussed and, finally, section 4 summarizes the paper with conclusions and some prospects for future work.

## 2 Passage retrieval in Joost

In our research, we are working with Dutch open-domain question answering. Our QA system, Joost, includes two strategies: (1) A table-lookup strategy using fact databases that have been created off-line, and, (2) an “on-line” answer extraction strategy with passage retrieval and subsequent answer identification and ranking modules. We will only look at the second strategy as we are interested in the passage retrieval component and its impact on QA performance. Let us first have a look at the retrieval module in our system.

### 2.1 Overview of the retrieval module

The passage retrieval component in our system implements an interface to several open-source IR engines. The query is generated from the given natural language question after question analysis. Keywords are sent to the IR engine and results (in form of sentence IDs) are returned to the QA system. The passage retrieval component generates the query in the required format used by the IR engine and translates retrieved units (for example paragraphs) into sequences of sentence IDs which are needed by the subsequent answer extraction modules. The retrieval component is transparent and we can easily switch between different IR engines and even combine them in various ways.

In the experiments described here, we apply Zettair [9], an open-source IR engine developed by the search engine group at the RMIT University in Melbourne, Australia. It implements a very efficient standard IR engine with high retrieval performance according to our experiments with various alternative systems. Zettair is developed for English and was mainly used for the TREC retrieval tasks. In our experiments we used version 0.6.1 and the Okapi BM-25 metric with standard parameters [14]. We applied the system to our Dutch data without any special adjustments and it seems to be very robust and comparable (in terms of retrieval performance) to other systems such as Lucene with integrated Dutch stemming and stop word filtering. Zettair is optimized for speed and is very efficient in both, indexing and retrieval. The outstanding speed in indexing is very fortunate for our experiments in which we had to create various indexes with different document segmentations which are discussed in the following section.

### 2.2 Document segmentation

We work with the Dutch data from the QA tasks at the cross-lingual evaluation forum (CLEF) [2]. The document collection used there is a collection of two daily newspapers from the years 1994 and 1995. It includes about 190,000 documents (newspaper articles) with altogether about 4 million sentences including approximately 80 million words. The documents include additional markup to segment them into paragraphs. Naturally, we apply this segmentation in one of our retrieval experiments. Note, that headers, signatures and other small units are treated as paragraphs on their own in the data. The average length of a paragraph is therefore rather small: around 4 sentences. Paragraph sizes may vary a lot depending on the document structure. This may influence the retrieval performance significantly which is our main motivation for the experiments with alternative segmentation approaches as described below.

We decided to define passages in terms of sequences of sentences as our QA system expects complete sentences for extracting answer candidates. Hence, passages of the same size (in terms of sentences) may have different lengths in terms of words and characters. We also define document boundaries as “hard” boundaries, i.e., passages may never come from more than one document in the collection.

Using this setup we apply the following segmentation strategies in our retrieval experiments:

**Window-based passages:** Documents are split into passages of fixed size (in terms of number of sentences). As mentioned earlier, we respect document boundaries and never cross them when creating a passage to be indexed. We use various sizes from 1 up to 10 sentences.

**Variable-sized arbitrary passages:** In this approach, passages may start at any sentence in each document and may have variable lengths. This is implemented by adding redundant information to our standard IR index: We create passages starting at every sentence in a document for each length defined (for instance lengths 1 up to 5 sentences). In this way we include many overlapping passages in our index that may be considered when querying the database. The IR ranking will decide which one to use when matching queries to documents. We define the following settings: arbitrary passages from 1 to 5 sentences, 5 to 10 sentences, and 1 to 10 sentences.

**Sliding window passages:** A sliding window approach also adds redundancy to the index by sliding over documents with a fixed-sized window (again in terms of number of sentences). All passages have the same size but may start at arbitrary positions in each document.<sup>1</sup> In our experiments, we apply sliding window passages for sizes from 2 to 10.

The segmentation strategies described above do not use any semantic or discourse information from the

<sup>1</sup> Note that we still do not cross document boundaries, i.e., passages may not start at any other sentence included in the last passage of a document except the first one.

documents (except for the document and sentence boundaries). We are interested in comparing such knowledge-poor approaches with the retrieval based on available paragraph markup. Especially, we would like to know if there is a length-based preference of the IR engine which would have a negative impact on variable-sized settings. Here, an advantage of the fixed-size segmentation should be observable. On the other hand, it is interesting to see whether it is preferable to include redundant information in the index. In these cases the system can directly compare various competing document segmentations returning the one with the best match. The variable-size approach has the additional advantage that the IR engine may even decide the passage size necessary to match the given query. However, a general length bias of the IR engine would again have a negative impact on the retrieval results when variable-sized passages are involved.

The following section describes the experiments carried out using the settings described above.

### 3 Experiments

#### 3.1 Setup

All experiments were carried out with the same data sets. The entire Dutch CLEF document collection is used to create the index files with the various segmentation approaches. For evaluation we applied questions from the previous Dutch QA tasks at CLEF. In particular we used all annotated questions (annotated with their answers) from the tracks in 2003, 2004, and 2005. Altogether, there are 777 questions, each question may have several answers. For each setting we retrieved 20 passages per question<sup>2</sup> using the same query generation strategy (basically using all words in the question). We used several measures to evaluate the retrieval performance:

**Mean reciprocal ranks:** The mean of the reciprocal rank of the first passage retrieved that contains a correct answer.

$$MRR_{IR} = \frac{1}{N} \sum_1^N \frac{1}{rank(\text{first\_relevant\_passage})}$$

**Coverage:** Percentage of questions for which at least one passage is retrieved that contains a correct answer [13].

**Redundancy:** The average number of passages retrieved per question that contain a correct answer [13].

We use simple string matching to decide whether a correct answer is included in a passage or not. We also count the number of sentences contained in all passages retrieved. The main purpose of passage retrieval is to reduce the search space for subsequent answer extraction modules which works on the sentence level. Hence, the number of sentences retrieved has a large impact on the QA system and its efficiency.

<sup>2</sup> In [3] the authors show that about 20 passages are optimal for the end-to-end performance of their QA system. We experienced similar results when experimenting with different numbers of retrieved passages.

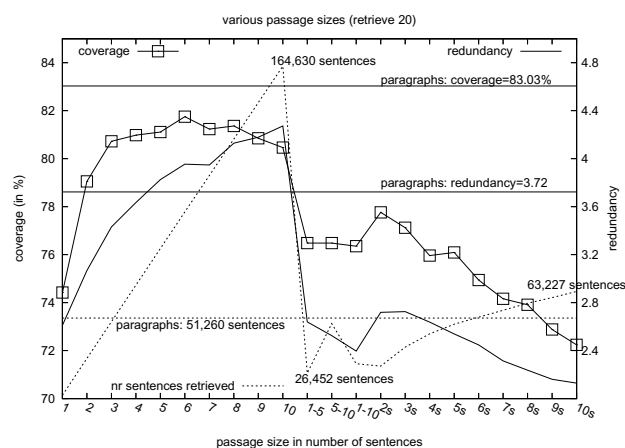


Fig. 5: Coverage and redundancy of passages retrieved for various segmentation strategies. 1-5, 5-10, 1-10 refer to the variable-size arbitrary passages and settings with the su x 's' refer to sliding window approaches. Note that there are 3 different scales on the Y-axis (one for coverage, one for redundancy and one for number of sentences). Hence, the curves should not be compared directly with each other.

#### 3.2 Coverage and redundancy

Intuitively, recall is more important in passage retrieval for QA than precision as mentioned already before. Passage retrieval is merely a filtering routine to make on-line QA feasible. It is a bottleneck because text segments that have been missed by the retrieval component are lost forever and cannot be found by any other means of the system. Therefore, we like to achieve the highest **coverage** possible to support question answering. Furthermore, we like to get as many relevant passages as possible to make it easier for the answer extraction modules to spot possible answers. This reduces the likelihood of selecting the wrong answer string by providing stronger evidence for the correct ones. Hence, high **redundancy** is desired as well. Figure 5 plots coverage and redundancy of the various approaches. Both measures are not directly comparable as they use different scales. Still, it is interesting to plot them on top of each other in order to illustrate dependencies between them.

As we can see in figure 5, coverage is best for the segmentation approach using pre-defined paragraphs. Redundancy on the other hand can be improved by considering larger units such as window-based segmentation techniques with 7 or more sentences. Using larger units increases the chance of including an answer in a selected passage. However, as illustrated in the figure the number of sentences to be searched is significantly increased for these segmentation approaches. The lowest scores are achieved for the sliding window approaches with large window sizes. This is somewhat surprising but the redundancy in the data seems to have a negative influence on the retrieval performance. We believe that the drop in coverage and redundancy is due to the overlap of passages. Many passages are included in the index with only small differences between them (one or a few sentences). These



passages are probably ranked similarly and, therefore, many overlapping passages are retrieved. In this way, the chance of finding an alternative relevant section is decreased. Hence, redundancy goes down and also coverage is decreased because of less variety in the retrieval results. However, we did not include a qualitative analysis of the results to support this hypothesis. If the hypothesis is true and many overlapping passages cause the drop in retrieval we could easily implement additional constraints to avoid such results. However, this has not been done in the present study.

### 3.3 Mean reciprocal ranks

In the case where passage retrieval is purely seen as a filtering step, ordering of the retrieved documents does not play a role. Ranking possible answers is then done entirely based on information extraction patterns according to the question independent of the ranking provided by the retrieval component. Therefore, coverage and redundancy should be sufficient to describe the quality of passage retrieval. However, the amount of data to be searched influences answer extraction modules not only in terms of efficiency but also in terms of error rates. Large amounts of data to be searched increase the likelihood of erroneous decisions made by answer extraction. Furthermore, the ranking of the passage retrieval component is usually an important clue to rank sentences with answer candidates. Hence, in our system, retrieval scores are incorporated in the final ranking equation. Therefore, we will now look at the mean reciprocal ranks of retrieved passages. These scores are compared with the performance of the overall QA system using the various passage retrieval strategies. For the latter we use mean reciprocal ranks again but this time in terms of answers found by the question answering system (using the first 5 answers only):

$$MRR_{QA} = \frac{1}{N} \sum_1^N \frac{1}{rank(\text{first\_correct\_answer})}$$

In figure 6 the mean reciprocal ranks for passage retrieval and for question answering are compared. Again, we also plot the number of sentences retrieved for each segmentation strategy.

Surprisingly, we can see a lot of differences in the plot of the passage retrieval MRR ( $MRR_{IR}$ ) and the MRR of the QA system ( $MRR_{QA}$ ). The  $MRR_{IR}$  scores are increased with larger passage sizes but the corresponding  $MRR_{QA}$  scores decline. On the other hand,  $MRR_{QA}$  scores are well above the other approaches (except paragraph segmentation) when using variable-sized paragraphs. This is also surprising when comparing the MRR scores to coverage and redundancy plotted in figure 5. The variable-sized segmentation approaches did not score well, neither on coverage nor on redundancy but they did very well on  $MRR_{QA}$ . Here, we can clearly see the effect of retrieval size in terms of number of sentences. Concluding from the experimental figures small units are preferred in the variable-sized segmentation approach. The number of sentences retrieved is comparable to the

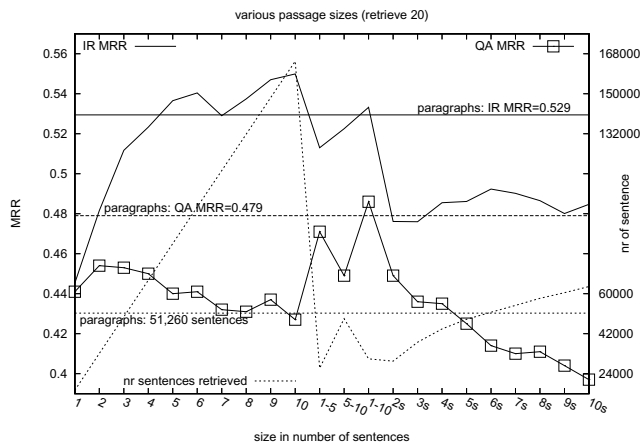


Fig. 6: Mean reciprocal ranks of passage retrieval (IR MRR) and question answering (QA MRR) for various segmentation strategies. There are two scales on the Y-axis in this plot (MRR and number of sentences).

retrieve 20	sent	par	doc
nr of sentences	16,545	51,260	323,582
coverage (%)	74.42	83.03	84.70
redundancy	2.61	3.72	5.33
$MRR_{IR}$	0.445	0.529	0.617
$MRR_{QA}$	0.441	0.479	0.432

Table 1: Discourse-based segmentation: 20 sentences/paragraphs/documents per question

fixed-size window approaches with sizes 2 and 3. For example, the average number of sentences retrieved for variable-sized passages of size 1 to 10 is about 40 per question which refers to an average passage size of 2 sentences (20 passages are retrieved per question). The  $MRR_{IR}$  scores, however, are much better for variable-sized passages than for fixed-sized passages. There are probably quite a few one-sentence passages in the one-to-x passage approaches and some larger passages where it is necessary to include larger context to match the query. To summarize the discussion, retrieving little amounts of precise data is apparently preferable for question answering compared to larger retrieval results even with better coverage. Hence, measuring retrieval performance in terms of coverage, redundancy and mean reciprocal ranks only is misleading according to our data. Most relevant is the relation between the measures just mentioned and the average size of the retrieved text units.

### 3.4 Discourse passages

In the comparison above, we could also see that the discourse-based segmentation performs best in terms of  $MRR_{QA}$  except for a slight improvement when using variable-sized arbitrary passages of size 1 to 10. However, this improvement is not significant and does not justify the extra redundancy in the retrieval database. In general, IR does not seem to be harmed by variable passage lengths.

We now also compare different discourse-based segmentations: sentence level, paragraph level and doc-

	75 sent	20 par	5 doc
nr of sentences	61,075	51,260	80,264
coverage (%)	83.03	83.03	75.45
redundancy	5.35	3.72	1.86
$MRR_{IR}$	0.451	0.529	0.607
$MRR_{QA}$	0.458	0.479	0.407

**Table 2:** *Discourse-based segmentation: 75 sentences, 20 paragraphs, 5 documents per question*

ument level segmentation. Table 1 summarizes the results.

The scores for the discourse-based segmentations follow the same tendencies as the other segmentation techniques. Larger units produce better performance in passage retrieval but cause a larger search space for sub-subsequent QA modules. As we can see at the  $MRR_{QA}$  scores, paragraph level segmentation performs best in our setup even though coverage and redundancy are below document retrieval results. Sentence retrieval on the other hand is not preferable due to low coverage and redundancy even though it produces the least amount of data to be searched.

Finally, we also compare the three discourse-based passage segmentation approaches with similar amounts of sentences retrieved. Table 2 shows the scores for retrieving 75 sentences, 20 paragraphs and 5 documents respectively.

We can see that document retrieval still yields the best MRR scores in the retrieval step. However, redundancy and coverage are much lower when reducing the number of documents retrieved. On the other hand, the coverage of the sentence retrieval approach is now identical to the paragraph approach and redundancy is much higher due to the higher number of individual units retrieved. However, paragraph retrieval still produces the best results in terms of question answering accuracy. Single sentences seem to be too small as a unit for information retrieval whereas documents are too broad for question answering.

## 4 Conclusions

Our experiments show that accurate passage retrieval is essential for question answering that integrates IR techniques as a one-step pre-filtering step. Not only coverage and redundancy are important for such a module but also the ranking and the size of the retrieval result have a large impact on the success of such a QA system. We could show that discourse-based segmentation into paragraphs works well with standard information retrieval techniques. Other segmentation approaches may improve coverage and redundancy but do not work well when looking at the overall performance of the QA system. Among the window-based approaches a segmentation into overlapping passages of variable-length performs best, in particular for passages with sizes of 1 to 10 sentences. With this, QA performs comparable to the paragraph retrieval approach. We could also show that paragraph retrieval is more effective than full document retrieval which is also much more efficient considering the expensive information extraction tools in subse-

quent modules of the QA system. Improvements to the discourse based segmentation remain to be investigated. For example, merging headers and other special units with proceeding paragraphs may lead to further improvements. Additionally, we want to look at combinations of several retrieval settings using various segmentation approaches. For example, we want to consider combinations of sentence-level evidence with paragraph retrieval and multi-step approaches in the form of zoom-in techniques.

## References

- [1] J. P. Callan. Passage-level evidence in document retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 302–310, New York, NY, USA, 1994. Springer-Verlag New York, Inc.
- [2] CLEF. Multilingual question answering at CLEF. <http://clef-qa.itc.it/>, 2005.
- [3] R. Gaizauskas, M. A. Greenwood, M. Hepple, I. Roberts, H. Saggion, and M. Sargaison. The university of sheffield's trec 2003 q&a experiments. In *Proceedings of the 12th Text REtrieval Conference*, 2003.
- [4] M. A. Greenwood. Using pertainyms to improve passage retrieval for questions requesting information about a location. In *Proceedings of the Workshop on Information Retrieval for Question Answering (SIGIR 2004)*, Sheffield, UK, 2004.
- [5] M. A. Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.
- [6] M. A. Hearst and C. Plaunt. Subtopic structuring for full-length document access. In *Research and Development in Information Retrieval*, pages 59–68, 1993.
- [7] M. Kaszkiel and J. Zobel. Passage retrieval revisited. In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, New York, NY, USA, 1997. ACM Press.
- [8] M. Kaszkiel and J. Zobel. Effective ranking with arbitrary passages. *Journal of the American Society of Information Science*, 52(4):344–364, 2001.
- [9] N. Lester, H. Williams, J. Zobel, F. Scholer, D. Bahle, J. Yianis, B. von Billerbeck, S. Garcia, and W. Webber. The Zettair search engine. <http://www.seg.rmit.edu.au/zettair/>, 2006.
- [10] F. Llopis, J. Vicedo, and A. Ferrández. Passage selection to improve question answering. In *Proceedings of the COLING 2002 Workshop on Multilingual Summarization and Question Answering*, 2002.
- [11] D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Girju, R. Goodrum, and V. Rus. The structure and performance of an open-domain question answering system, 2000.
- [12] C. Monz. *From Document Retrieval to Question Answering*. PhD thesis, University of Amsterdam, 2003.
- [13] I. Roberts and R. Gaizauskas. Evaluating passage retrieval approaches for question answering. In *Proceedings of 26th European Conference on Information Retrieval*, 2004.
- [14] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau. Okapi at TREC-3. In *Text REtrieval Conference*, pages 21–30, 1992.
- [15] S. Tellex, B. Katz, J. Lin, A. Fernandes, and G. Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the SIGIR conference on Research and development in information retrieval*, pages 41–47. ACM Press, 2003.
- [16] J. Zobel, A. Moffat, R. Wilkinson, and R. Sacks-Davis. Efficient retrieval of partial documents. *Information Processing and Management*, 31(3):361–377, 1995.

# Improved Sentence Alignment for Movie Subtitles

Jörg Tiedemann  
University of Groningen  
Alfa Informatica,  
P.O. Box 716  
9700 AS Groningen, the Netherlands  
*j.tiedemann@rug.nl*

## Abstract

Sentence alignment is an essential step in building a parallel corpus. In this paper a specialized approach for the alignment of movie subtitles based on time overlaps is introduced. It is used for creating an extensive multilingual parallel subtitle corpus currently containing about 21 million aligned sentence fragments in 29 languages. Our alignment approach yields significantly higher accuracies compared to standard length-based approaches on this data. Furthermore, we can show that simple heuristics for subtitle synchronization can be used to improve the alignment accuracy even further.

## Keywords

sentence alignment, parallel corpora, multilingual resources

## 1 Introduction

Sentence alignment is a well-known task applied to parallel corpora as a pre-requisite for many applications such as statistical machine translation [2] and multilingual terminology extraction [9]. It consists of finding a monotonic mapping between source and target language sentences allowing for deletions, insertions and some n:m alignments. Several algorithms have been proposed in the literature mainly based on translation consistency. We can distinguish between the following two main approaches: (1) sentence alignment based on similarity in length [1, 4], and, (2) alignment based on term translation consistency and anchor points [5, 3, 6]. Both techniques can also be combined [8, 7, 14]. It has been shown that these simple, often language independent techniques yield good results on various corpora (see, e.g. [12]) and the problem of sentence alignment is often regarded to as being solved at least to some reasonable degree.

In this paper, we focus on the alignment of movie subtitles, a valuable multilingual resource that is different to other parallel corpora in various aspects: Movie subtitles can be described as compressed transcriptions of spoken data. They contain many fragmental utterances rather than grammatical sentences. Translations of subtitles are often incomplete and very dense in the sense of compressing and summarizing utterances rather than literally transcribing them. They are often mixed with other information such as titles, trailers, and translations of visual data (like signs etc.).

The amount of compression and re-phrasing is different between various languages, also dependent on cultural differences and subtitle traditions. A special type are subtitles for the hearing impaired which are closer to literal transcriptions combined with extra information about other sounds (such as background noise etc.). All this causes many insertions, deletions and complex mappings when aligning subtitles. Some of the challenges are illustrated in figure 1.



Fig. 1: Alignment challenges: An example with English and Dutch subtitles.

The figure shows a short example of English subtitles and their Dutch correspondences. There are untranslated segments such as the English fragments shown in subtitle screen one, three and six. The latter two are even embedded in surrounding sentences which makes it impossible to find a proper alignment with sentences as the basic unit. Furthermore, automatic tokenization and sentence splitting causes further errors. Obviously, sentences may span over several subtitle screens as illustrated in figure 1. However, in the Dutch example the first subtitle line is attached to the preceding ones because the sentence splitter did not recognize a proper sentence boundary between line one and two. A sentence aligner has no other chance then to link the entire unit to corresponding ones in the other language even if the mapping is only partially correct. We can also see in the example that there is only one real 1:1 alignment whereas other types are more frequent than in other parallel resources.

From the discussion above, it seems obvious that traditional sentence alignment approaches are not ap-

propriate for this kind of data. Hence, we propose a new approach specifically designed for the alignment of subtitles. However, in the next section we firstly present the subtitle corpus we have collected including a brief discussion about pre-processing issues.

## 2 The Subtitle Corpus

Several databases are on-line that provide subtitles in various languages. All of them collect user uploads that can be searched in various ways. However, most of them are not very stable in the sense that they move to different locations and have a lot of down-time. This made us suspicious about their legal background. We found one provider, <http://www.opensubtitles.org>, that seems to be very reliable, which offers an extensive multilingual collection of subtitles without user registration necessary. They claim that their database only contains legal downloads that are free to distribute. Furthermore, we were pleased to obtain the entire database of about 308,000 files by the provider covering about 18,900 movies in 59 languages (status of July, 2006) for which we are very grateful.

In order to build our corpus, several pre-processing steps had to be taken. First of all, we had to identify the subtitle format and to convert it to a uniform corpus format. Several formats are used and we decided to support two popular ones, SubRip files (usually with extension '.srt') and microDVD files (usually with extension '.sub'). The latter were automatically converted to SubRip using a freely available script `sub2srt` (<http://www.robelix.com/sub2srt/>). Furthermore, subtitles use various character encodings. Unfortunately, we are not aware of a reliable tool for detecting character encodings and, therefore, we manually defined a conversion table (one encoding per language) after inspecting some sample data. We converted the subtitle files to a simple standalone XML format using Unicode UTF-8. An example is shown in figure 2.

Each subtitle file has been tokenized and marked with sentence boundaries as shown in figure 2. Both, tokenization and sentence splitting is done by means of regular expressions. The annotation is done automatically without any manual corrections and, therefore, contains errors especially for languages that do not use similar word and sentence boundaries as defined in our patterns. In future work, we would like to improve tokenization and especially sentence boundary detection which is crucial for the success of an alignment at the sentence level<sup>1</sup>.

Another issue with the database we obtained is that it contains erroneous files, for example, files with corrupt character encodings and subtitles tagged with the wrong language. In order to remove such noise as much as possible, we included a language classifier to check the contents of all subtitles. For this we used

```
<?xml version="1.0" encoding="utf-8"?>
<document>
  <s id="1">
    <time id="T1S" value="00:00:26,500" />
    <w id="1.1">Spend</w>
    <w id="1.2">all</w>
    <w id="1.3">day</w>
    <w id="1.4">with</w>
    <w id="1.5">us</w>
    <w id="1.6">.</w>
    <time id="T1E" value="00:00:28,434" />
  </s>
  <s id="2">
    <time id="T2S" value="00:00:28,502" />
    <w id="2.1">There</w>
    <w id="2.2">are</w>
    <w id="2.3">two</w>
    <w id="2.4">--</w>
    <w id="2.5">pardon</w>
    <w id="2.6">me</w>
    <w id="2.7">--</w>
    <time id="T2E" value="00:00:30,436" />
    <time id="T3S" value="00:00:30,504" />
    <w id="2.8">two</w>
    <w id="2.9">of</w>
    <w id="2.10">everything</w>
    <w id="2.11">in</w>
    <w id="2.12">every</w>
    <w id="2.13">Noah</w>
    <w id="2.14">s</w>
    <w id="2.15">arcade</w>
    <w id="2.16">.</w>
    <time id="T3E" value="00:00:34,440" />
  </s>
```

Fig. 2: Subtitles in XML

`textcat` a freely available and trainable classifier designed for language identification [13]. It uses N-gram models trained on example texts and, therefore, relies on the given encoding used in the training data. We applied the language checker after encoding conversion and, therefore, built language models for UTF-8 texts. For simplicity we used the training data from the `textcat` package converted to Unicode using the Unix tool `recode`. Altogether, we created 46 language models. The classifier predicts for each given input file the most likely language according to the known models. The output of `textcat` is one of the following: (1) a certain classification of one language, (2) a ranked list of likely languages (in cases where the decision is not clear-cut), and, (3) a “resign” message in cases where the language classifier does not find any language that matches sufficiently enough. We accepted subtitles only in the case where the language classifier is certain that the language is the same as specified in the database and disregarded all other files.

After pre-processing and language checking we retained 38,825 subtitle files in 29 languages. From that we selected 22,794 pairs of subtitles for alignment (selecting only the ones corresponding to the same physical video file) covering 2,780 movies in 361 language pairs. Altogether, this corresponds to about 22 million sentence alignments created by the approach described below.

## 3 Sentence alignment

One of the essential properties of parallel corpora is that they can be aligned at some segmentation level. A common segmentation is to split on sentence boundaries and to link sentences or sequences of sentences in the source language with corresponding ones in the target language. Sentence alignment is assumed to be

<sup>1</sup> Note that sentences may span several subtitle screens as also shown in figure 2. This makes it necessary to store time information (which we need for the alignment later on) in a special way to avoid crossing annotations that are not allowed in XML. Hence, *time slot* information is split into two time events, one for the starting time and one for the end of the slot.

monotonic, i.e. crossing links are not allowed. However, deletions and insertions are usually supported.

Subtitles can be aligned at various segmentation levels, for instance, mapping *subtitle screens* (text fragments shown together in one time slot on screen) or *sentences*. We opted for the latter for the following reasons: Sentences are linguistically motivated units and important for applications using the aligned data. Subtitle screens on the other hand often include various fragments by different speakers and their compilation highly depends on visual requirements and language dependent issues. The contents of these screens varies very much between different subtitles and, therefore, they are hard to align without partial overlaps with other screens. We therefore decided to align the data at the sentence level assuming that our sentence splitter works well for most of the languages included.

In the following, we first discuss a standard length-based approach applied to subtitles. Thereafter, we will present our new alignment approach based on time overlaps. Finally, some additional heuristics are discussed for further improvements.

### 3.1 Length-based approaches

One of the standard approaches to sentence alignment is the popular length-based approach proposed by [4]. It is based on the assumption that translations tend to be of similar lengths in characters (possibly factorized by a specific constant) with some variance. Using this assumption we can apply a dynamic algorithm to find the best alignment between sentences in one language and sentences in the other language. Alignments are restricted to the most common types (usually 1:1, 1:0, 0:1, 2:1, 1:2 and 2:2) with prior probabilities attached to them to make the algorithm more efficient and more accurate. In the default settings, there is a strong preference for 1:1 sentence alignments whereas the likelihood of the other types is very low. This is based on empirical studies of some example data [4].

It has been shown that this algorithm is very flexible and robust even without changing its parameters [12, 11]. However, looking at our data it is obvious that certain settings and assumptions of the algorithm are not appropriate. As discussed earlier, we can observe many insertions and deletions in subtitle pairs and typically, a length-based approach cannot deal with such cases very well. Even worse, such insertions and deletions may cause a lot of follow-up errors due to the dynamic algorithm trying to cover the entire text in both languages. In order to account for the special properties of subtitles we adjusted the prior probabilities set in the length-based alignment approach. For this we manually aligned a small subset of randomly selected subtitles from five movies in English, German, and Swedish. We aligned parts of all language combinations using the interactive sentence alignment tool ISA [10] resulting in a total of 1312 sentence alignment units. We used relative frequencies of each occurring alignment type to estimate the new parameters. For efficiency reasons we omitted alignment types with probabilities below 0.001. Table 1 lists the final settings used for the length-based approach.

Using the settings above, 1:1 sentence alignment are

alignment type	count	probability
1:1	896	0.6829
2:1	100	0.0762
0:1	91	0.0694
1:0	74	0.0564
1:2	72	0.0549
1:3	24	0.0183
3:1	16	0.0122

**Table 1:** Adjusted priors for various alignment types (with probability > 0.001)

still preferred but with a smaller likelihood (0.89 in the original settings). As expected, deletions and insertions (1:0 and 0:1 alignments) are more frequent in subtitles (0.0099 each in the original implementation) and two types are added: 1:3 and 3:1 alignments. On the other hand, 2:2 alignments are not considered in our model whereas they are in the original approach with a prior probability of 0.011). We are aware of the fact that there is a substantial variance among alignment types (depending on the language pair and other factors) and that our sample is not representative for the entire collection containing many more language pairs. However, we assume that these settings are still more appropriate than the default settings used in the original algorithm. Figure 3 shows example output of the approach with adjusted parameters.

English	Dutch
<i>Spend all day with us .</i>	<i>De wereld van Wayne Er</i>
<b>There are two – pardon me – two of everything in every Noah’s arcade .</b>	<b>zijn twee , excuseer me , twee van Zantar . ... gestoorde helicopters ...</b>
<i>That means two of Zantar ,</i>	<i>Het is goed om je weer te zien , Benjamin .</i>
<i>That means two of Zantar , Bay Wolf , Ninja Commando , Snake-azon , Psycho Chopper ...</i>	<i>Je bent al heel lang niet meer in Shakey’s geweest .</i>
<i>It’s really good seeing you , Benjamin .</i>	<i>Ik heb het heel erg druk .</i>
<i>You haven’t been into Shakey’s for so long .</i>	<i>Het zijn er twee voor jou , want eentje zal het niet doen .</i>
<i>Well , I’ve been real busy . It’s two for you ’ cause one won’t do .</i>	<i>De hele week , krijgen kinderen onder de zes elke vijfde ...</i>
<i>All this week , kids under 6 get every fifth – There’s a new pet .</i>	<b>Er is een nieuw huisdier</b> <i>Het Chia huisdier .</i>
<i>Ch- Ch- Chia Chia Pet – the pottery that grows .</i>	<i>Het aardewerk dat groeit .</i>
<b>They are very fast .</b>	<b>Zij zijn erg snel .</b>
<b>Simple .</b>	<b>Simpel .</b>
<b>Plug it in , and insert the plug from just about anything .</b>	<b>Plug het in .</b>

**Fig. 3:** Length-based sentence alignment - text in italics is wrongly aligned.

As the figure illustrates there are many erroneous alignments using the length-based approach. In fact, most of the alignments are wrong (in italics) and we can also see the typical problem of follow-up errors. For example, the alignment is shifted already in the beginning due to the deletion of some sentences fragments in Dutch.

### 3.2 Alignment with time overlaps

As seen in the previous sections, a length-based approach cannot deal very well with our data collec-

tion. Let us now consider a different approach directly incorporating the time information given in the subtitles. Subtitles should be synchronized with the original movie using the time values specified for each screen. Intuitively, corresponding segments in different translations should be shown at roughly the same time. Hence, we can use this information to map source language segments to target language segments. The timing is usually not exactly the same but the overlap in time in which they are shown should be a good predictor for correspondence. The main principle of an alignment approach based on time overlaps is illustrated in figure 4.



Fig. 4: Sentence alignment with time overlaps

The main problem with this approach is to deal with the differences in dividing texts into screens in various languages. The alignment is still done at the sentence level and, hence, we need time information for sentences instead of subtitle screens. Figure 4 illustrates some simple cases where sentences span over several screens but still start and end at screen boundaries. However, this is not always the case. Very often sentence boundaries are somewhere in the middle of a screen (even if they span over several screens) and, hence, start and end time are not explicitly given. In these cases we have to approximate the time boundaries to calculate overlaps between sentences in corresponding files. For this we used the nearest “time events” and calculated the time proportional to the strings in between. Computing a new time event  $t_{new}$  for a sentence boundary in this way is given by the following equation:

$$t_{new} = t_{before} + C_{before} * \frac{t_{after} - t_{before}}{C_{before} + C_{after}}$$

Here,  $t_{before}$  corresponds to the nearest time event before the current position and  $t_{after}$  is the time at the nearest time event after the current position. Similarly,  $C_{before}$  and  $C_{after}$  are the lengths of the strings before and after the current position up to the nearest time events. Hence, we interpolate the time linearly over the characters in the current segment. This is done dynamically from the beginning to the end of the subtitle file using approximated time values as well for further time estimations if necessary (in cases where more than one sentence boundary is found within one

subtitle screen). Additionally, consistency of the time values is checked. Due to errors in the subtitle files it can happen that time events have identical or even decreasing values. In these cases a dummy time of 0.0001 seconds is added to the previous time event overwriting the inconsistent one. This is done iteratively as long as necessary.

Now, with time values fixed for all sentences in the subtitle we need to find the best alignment between them. We still want to support deletions, insertions and n:m alignments. In our approach, we define a set of possible alignment types (as in the length-based approach) which are then considered as possible alternatives when looking for the best mapping. In our experiments, we simply applied the same types as used in the length-based approach (see table 1). However, prior probabilities are not used in this model. The comparison is purely based on absolute time overlaps. The algorithm runs through the pair of subtitles in a sliding window, comparing alternative alignments according to the pre-defined types and picking the one with the highest time overlap. Note that we do not need any recursion and the alignment can be done in linear time because of the use of absolute time values. The result of the alignment with time overlaps for our little example is shown in figure 5.

English	Dutch
Spend all day with us .	
There are two - pardon me - two of everything in every Noah's arcade .	De wereld van Wayne Er zijn twee , excuseer me , twee van Zantar . ...
That means two of Zantar , That means two of Zantar , Bay Wolf , Ninja Commando , Snake-azon , Psycho Chopper ...	gestoorde helicopters ...
It's really good seeing you , Benjamin .	Het is goed om je weer te zien , Benjamin .
You haven't been into Shakey's for so long .	Je bent al heel lang niet meer in Shakey's geweest .
Well , I've been real busy .	Ik heb het heel erg druk .
It's two for you ' cause one won't do .	Het zijn er twee voor jou , want eentje zal het niet doen .
All this week , kids under 6 get every fifth - There's a new pet .	De hele week , krijgen kinderen onder de zes elke vijfde ... Er is een nieuw huisdier Het Chia huisdier .
Ch- Ch- Chia Chia Pet - the pottery that grows .	Het aardewerk dat groeit .
They are very fast .	Zij zijn erg snel .
Simple .	
Simple . Plug it in , and insert the plug from just about anything .	Plug het in . Het is simple !

Fig. 5: Sentence alignment based on time overlaps - text in italics is wrongly aligned.

One of the big advantages of this approach is that it can easily handle insertions and deletions at any position as long as the timing is synchronized between the two subtitle files. Especially initial and final insertions often cause follow-up errors in length-based approaches but they do not cause any trouble in the time overlap approach (look for example at the first English sentence in the example in figure 4). Remaining errors mainly occur due to sentence splitting errors and timing differences. The latter will be discussed in the following section.

### 3.3 Movie synchronization

Intuitively, the alignment approach based on time overlaps ought to produce very accurate results assuming that subtitles are equally synchronized to the original movie. Surprisingly, this is not always the case. Time information in subtitles often varies slightly resulting in growing time gaps between corresponding segments. In preliminary evaluations we realized that the alignments produced by the time overlap approach either is very accurate or very poor. After inspecting some problematic cases it became obvious that the errors were due to timing issues: a difference in speed and a difference in starting times. Considering the fact that alignment is entirely based on time information already small timing differences have a large impact on this approach.

Fortunately, timing differences can be adjusted. Assuming that speed is constant in both subtitles we can compute two additional parameters, speed difference (*time ratio*) and *time offset* using two anchor points that correspond to true alignment points. The following equations are used to calculate the two parameters:

$$\begin{aligned} \textit{time}_{ratio} &= \frac{(trg_1 - trg_2)}{(src_1 - src_2)} \\ \textit{time}_{offset} &= trg_2 - src_2 * \textit{time}_{ratio} \end{aligned}$$

Here,  $src_1$  and  $src_2$  corresponds to the time values (in seconds) of the anchor points in the source language and  $trg_1$  and  $trg_2$  to the time values of corresponding points in the target language. Using *time ratio* and *time offset* we adjust all time values in the source language file and align sentences using the time overlap approach.

The time synchronization approach described above is very effective and yields significant improvements where timing differences occur. However, it requires two reliable anchor points that should also be far away from each other to produce accurate parameter estimations. One approach (and the most reliable one) is to define these anchor points manually. Again, ISA can be used to do this job simply by adding two break points to the subtitle pair, one at the beginning and one at the end. We then use the times at the beginning of each break point and synchronize. This approach is simple and requires minimal human intervention. However, it is not feasible to use it for all subtitle pairs in our corpus.

An alternative approach is to restrict human intervention to cases where erroneous alignments can be predicted using some simple heuristics. For example, we can count the ratio between empty sentence links (1:0 and 0:1) and non-empty ones.

$$\textit{algtype}_{ratio} = \frac{|\text{non-empty links}| + 1}{|\text{empty links}| + 1}$$

Assuming that an alignment should mainly consist of non-empty links we can use a threshold for this ratio (for example  $> 2.0$ ) to decide whether an alignment is likely to be correct or not. The latter can be inspected by humans and corrected using the anchor point approach.

Another approach for synchronization is to use cognates in form of similar strings to identify corresponding points in source and target language. For this, subtitle pairs are scanned in a sliding window from the beginning and from the end in order to find appropriate candidates. Using string similarity measures such as the longest common subsequence ratio (LCSR) and thresholds on similarity we can decide for the most relevant candidate pairs with the largest distance (it is also advisable to set a threshold for the minimal length of a possible candidate). Alternatively, we can restrict the search to identical strings and/or to strings with initial capital letters or we may include pairs from a given bilingual dictionary to find anchor points in the subtitle pairs. Note that a candidate does not have to be limited to a single word. Using these pairs of corresponding candidates we can use the time (start or end) of the sentences they appear in to compute the timing differences. Clearly, the cognate approach is restricted to related languages with more or less identical character sets. A solution for more distant language pairs would be to use existing bilingual dictionaries to select appropriate candidate pairs. This, however, requires corresponding resources for all language pairs included which are not available to us.

Furthermore, selecting candidate pairs is not straightforward especially in our subtitle data. Names are often spelled in a similar way in different languages and therefore, they will frequently be selected as anchor point by the string similarity measure. However, the use of names may differ significantly in various languages. As we discussed earlier, subtitles are not transcriptions of the spoken data and, hence, names are often left out or replaced by referring expressions. Therefore, we may find a lot of false hits when using a general search for cognate pairs. In our initial experiments we observed that a general synchronization based on cognate pairs for all subtitle pairs is harmful for the overall alignment quality. Hence, heuristics based on alignment type ratios as mentioned above are again useful for selecting potentially erroneously aligned subtitle pairs for which synchronization might be useful. Another strategy to reduce synchronization errors made by wrongly selected anchor points is to average over all candidate pairs. However, this can lead to other errors. Finally, we can also try all possible combinations of anchor point candidates and use them iteratively for synchronization. We then pick the one that performs best according to the alignment type ratio as defined above. Fortunately, the time overlap approach is fast enough to make it feasible to apply this approach (see section 4 below).

## 4 Evaluation

For evaluation we randomly selected 10 movies with subtitles in three languages, English, German and Dutch. We manually aligned parts of all pairs of Dutch-English and Dutch-German subtitles from this set using ISA<sup>2</sup>. In particular, we selected about 15 initial, 15 intermediate and 15 final sentences in each

<sup>2</sup> Note that the alignments are symmetric and the direction of the alignment as mentioned here is only due to alphabetic sorting of the language name



aligned subtitle pair to account for differences in alignment quality at different document positions. The exact number of sentences aligned varies slightly between all subtitle pairs due to the amount of insertion, deletions and n:m alignments necessary. In total, we included 988 alignment units in our evaluation set, 516 for Dutch-English and 472 for Dutch-German. The 10 movies are all originally in English and, therefore, it is interesting to compare the alignments for the two selected language pairs. English subtitles are mainly produced for the hearing impaired and, therefore, contain much more information than the two translations into Dutch and German. However, let us first look at the overall accuracy of the alignment for the following four alignment approaches: (1) length-based sentence alignment with adjusted priors (*length*), (2) standard time-overlap alignment (*time1*), (3) time-overlap alignment with a cognate filter (LCSR) using a threshold of 0.8 which is applied in cases where  $algtype_{ratio} < 2.0$  (*time2*), and, (4) time-overlap alignment with a cognate filter (threshold=0.6) and iterative selection of candidate pairs according to the alignment type ratio in cases the initial ratio is  $< 2.0$  (*time3*). The minimal string length for the cognate filter is set to five characters for both, *time2* and *time3*. The LCSR threshold for *time3* is lower than for *time2* to give it more flexibility when selecting anchor points. It is not recommendable to use such a relaxed threshold for *time2* because of the risk of finding false positives. *Time2* automatically selects the candidate pairs with the largest distance from each other and, therefore, the probability of selecting a wrong candidate pair is larger with lower thresholds for the cognate filter.

The results of the alignments measured on our evaluation data are shown in table 2. The scores are split into three categories: *correct* for exact matches, *partial* for partially correct alignments (some overlap with correct alignments in both, source and target language<sup>3</sup>), and *wrong* for all other alignments. Naturally, we count only scores for sentences included in the manually aligned data.

approach	correct	partial	wrong
length	0.515	0.119	0.365
time1	0.608	0.105	0.286
time2	0.672	0.136	0.192
time3	0.732	0.144	0.124

**Table 2:** *Different alignment approaches*

The scores in table 2 show that the alignment accuracy is significantly lower than otherwise reported for sentence alignment, which could be expected due to the difficulties in our data discussed earlier. However, the time-overlap approach yields major improvements compared to the length-based alignment. We can also see that the heuristics for enabling synchronization based on the type ratio is successful. The final approach using iterative candidate selection clearly outperforms the others. It is interesting to see where the

<sup>3</sup> Note that empty alignments are always mapped to 1:0 or 0:1 alignments (never 0:x or x:0 with  $x > 1$ ) and are either correct or wrong but never *partial*.

strengths of the time overlap approach can be found. For this, we computed accuracy scores for the different alignment types (see table 3). In order to make it easier to compare the results we counted partially correct links as 50% correct and added them accordingly to the scores of the correct links.

type	nr	length	time1	time2	time3
1:1	685	0.734	0.676	0.763	0.842
0:1	106	0.000	0.566	0.575	0.594
1:2	70	0.429	0.529	0.671	0.743
1:0	52	0.000	0.904	0.923	0.885
2:1	43	0.535	0.686	0.791	0.849
1:3	16	0.469	0.594	0.625	0.688
2:2	5	0.300	0.300	0.400	0.400
3:1	3	0.333	0.667	0.833	0.833

**Table 3:** *Accuracy per alignment type (skipping 8 alignments with more than four sentences involved)*

The strength of the time-overlap approach is certainly in the non-1:1 alignments. The length-based approach is rather good in finding proper 1:1 links and yields even better results than the standard time-overlap approach. However, using synchronization heuristics brings about a significant improvement beyond the accuracy of the baseline approach even for this alignment type. The largest difference can be seen in the empty alignments. The time-overlap approach can handle insertions and deletions much better than the length-based approach. It also yields better results for the other types. Synchronization is helpful for almost all types. One exception is the score for 1:0 alignments which actually drops a little bit when applying the iterative anchor point selection. A reason for this is that such empty alignments are taken as indicators for erroneous alignments even when they are correct. In some cases this assumption causes a degradation of performance. This can also be seen when looking at the results for the individual subtitle pairs (tables 4 and 5).

movie ( <i>dut-eng subtitles</i> )	length	time1	time2	time3
Location Production Footage: The Last Temptation of Christ	0.857	0.976	0.976	0.976
Finding Neverland	0.375	0.333	0.333	0.615
A Beautiful Mind	0.339	0.688	0.688	0.688
Under Fire	0.896	0.896	0.896	0.896
Batman Forever	0.976	0.988	0.988	0.988
The Last Samurai	0.043	0.928	0.942	0.928
Basic	0.737	1.000	1.000	1.000
Pulp Fiction	0.088	0.175	0.175	0.579
Return to Paradise	0.640	0.940	0.940	0.940
The Diary of Anne Frank	0.308	0.754	0.754	0.754
average (516 alignments)	0.476	0.754	0.756	0.825

**Table 4:** *Alignment accuracy per subtitle pair for Dutch-English*

There are indeed subtitle pairs for which the accuracy drops when using iterative anchor point selection as mentioned above. However, the overall performance is higher using this strategy compared to the fixed selection of the most distance candidates (*time2*). Tables 4 and 5 also include average accuracies per language pair. Here, we can observe a huge difference between the accuracy of the length-based ap-



movie ( <i>dut-ger subtitles</i> )	length	time1	time2	time3
Location Production Footage: The Last Temptation of Christ	0.963	1.000	1.000	1.000
Finding Neverland	0.523	0.047	0.791	0.756
A Beautiful Mind	0.796	0.092	0.663	0.643
Under Fire	0.890	0.900	0.900	0.900
Batman Forever	0.608	0.706	0.706	0.706
The Last Samurai	0.787	0.915	0.915	0.915
Basic	0.500	0.154	0.590	0.487
Pulp Fiction	0.637	0.049	0.049	0.716
Return to Paradise	0.798	0.915	0.915	0.915
The Diary of Anne Frank	0.361	0.759	0.759	0.759
average (472 alignments)	0.683	0.559	0.722	0.781

**Table 5:** Alignment accuracy per subtitle pair for Dutch-German

proach which is much higher for Dutch-German than for Dutch-English. This could also be expected due to the differences in style. The English subtitles are much more detailed whereas German and Dutch subtitles are more compressed. It looks like that this compression is rather similar for the two languages which favors the length-based approach. The time-overlap alignment is actually a bit worse for Dutch-German than for Dutch-English but this might be rather incidental. Many errors are due to sentence splitting mistakes which can be quite different for the various subtitles. In future work, preprocessing should be improved to reduce errors originating in tokenization and sentence splitting. Fortunately, the alignment is done automatically and, therefore, can easily be re-run after any preprocessing improvement. Another task for future work is to check the alignment quality for other language pairs especially more distant ones for which the synchronization approach using cognates is not applicable.

## 5 Conclusions

In this paper a sentence alignment approach for movie subtitles based on time overlaps has been introduced. It has been used to align an extensive multilingual corpus of about 38,000 subtitles in 29 languages. Its accuracy outperforms standard length-based alignment approaches especially by improving non-1:1 alignments that frequently occur in this kind of data. Furthermore, we presented additional techniques to synchronize subtitles to improve the alignment even further.

## References

- [1] P. F. Brown, J. C. Lai, and R. L. Mercer. Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Meeting of the ACL*, pages 169–176, 1991.
- [2] P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, June 1993.
- [3] S. F. Chen. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 9–16, Morristown, NJ, USA, 1993. Association for Computational Linguistics.
- [4] W. A. Gale and K. W. Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102, 1993.
- [5] M. Kay and M. Röscheisen. Text-translation alignment. *Computational Linguistics*, 19(1):121–142, 1993.
- [6] I. D. Melamed. Bitext Maps and Alignment via Pattern Recognition. *Computational Linguistics*, 25(1):107–130, 1999.
- [7] R. C. Moore. Fast and accurate sentence alignment of bilingual corpora. In *AMTA '02: Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users*, pages 135–144, London, UK, 2002. Springer-Verlag.
- [8] M. Simard, G. F. Foster, and P. Isabelle. Using cognates to align sentences in bilingual corpora. In *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, pages 67–81, Montreal, Canada, 1992.
- [9] F. Smadja, K. R. McKeown, and V. Hatzivassiloglou. Translating collocations for bilingual lexicons: A statistical approach. *Computational Linguistics*, 22(1), 1996.
- [10] J. Tiedemann. ISA & ICA - two web interfaces for interactive alignment of bitexts. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*, Genova, Italy, 2006.
- [11] J. Tiedemann and L. Nygard. The OPUS corpus - parallel and free. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'2004)*, Lisbon, Portugal, 2004.
- [12] E. F. Tjong Kim Sang. Aligning the Scania Corpus. Technical report, Department of Linguistics, University of Uppsala, 1996.
- [13] G. van Noord. Textcat – implementation of the algorithm presented in Cavnar, W. B. and J. M. Trenkle, “N-Gram-Based Text Categorization” In *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, NV, UNLV Publications/Reprographics, pp. 161–175, 1994. <http://www.let.rug.nl/~vannoord/TextCat/>, 2006.
- [14] D. Varga, L. Németh, P. Halácsy, A. Kornai, V. Trón, and V. Nagy. Parallel corpora for medium density languages. In *Proceedings of the Recent Advances in Natural Language Processing*, pages 590 – 596, 2005.

# A Comparative Study of Parsers Outputs for Spanish

Nevena Tinkova Tincheva  
Irene Castellón Masalles

Department of Linguistics, University of Barcelona, Spain  
{nevenatinkova,icastellon}@ub.edu

## Abstract

We present a comparative study of the outputs of three widely known available parsers implemented for Spanish – FreeLing, HISPAL and Connexor. Our prime goal is to develop subsequently a method for optimization of syntactic parsing for Spanish using the conclusions of the comparative analysis reported below. In this paper, we discuss the results of the comparison and investigate the error analysis in order to achieve further better performance and higher reliability in automatic parsing for Spanish. With this study, we attempt to demonstrate that it is worthwhile to identify the sources of several common errors for making progress in this direction.

## Keywords

Syntactic analysis, Syntactic parsers for Spanish, Error analysis, Parsing evaluation.

## 1 Introduction

A lot of recent works on automatic parsing of natural languages have shown the necessity of disposing of an appropriate grammar no matter how simple it may be and its usefulness in different areas of linguistic research. As until now a great part of the efforts has been concentrated on English, Spanish has recently become a new challenge for linguists because it does need mature and full-grown grammar capable of analysing unrestricted texts. With this objective, a number of syntactic parsers for Spanish based on different grammatical formalisms have been designed and implemented, but only after a close examination and comparison of their outputs, we can notice that there is still a lot of work left to do on this side. Therefore, we consider that their comparative study could contribute for improving the syntactic analysis for Spanish<sup>1</sup>.

Section 2 describes the three syntactic parsers for Spanish involved in the study – HISPAL (Bick, E. 2006), FreeLing (Carreras et al 2004; Atserias et al 2006) and Connexor (Tapanainen 1996). In Section 3 we examine some of the syntactic peculiarities of Spanish and the problems that these can entail in syntactic processing. Actually, we will further analyse these problems and the

way they are treated in the productions of the three analysers. Section 4 explains the methodology used in the study in order to proceed to analysing the results performed by the parsers in Section 5 and consequently, to set out the evaluation of the parsers outputs. Finally, in Section 6 we finish with some conclusions and trace out ideas for further research.

## 2 Syntactic Parsers for Spanish

The three available state-of-the-art and most accurate parsers used here in order to run the comparison are HISPAL, Connexor and FreeLing<sup>2</sup>. The first two rely on Constraint Grammar formalism whereas the latter begins using constituency structure (as it comes from the shallow parser TACAT) but later on it is developed within dependency grammar. Of special importance for the comparison is the fact that HISPAL, FreeLing and Connexor opt to use the dependency approach to syntax, in which words modify other words and the former adds details to the latter. Thus, the whole combination inherits the syntactic properties of the words that govern. Keeping in mind this basic circumstance, we can explore their outputs in Section 5.

As far as their performance is concerned, we can find only a detailed evaluation of HISPAL system (Bick 2006), since no other evaluation for other syntactic parsers for Spanish is reported. The results of the evaluation of HISPAL parsing performance are quite good achieving an overall syntactic accuracy of 95 – 96% on raw text, even though, as Bick (2006) states, it was not enough rigid because it did not use multiple annotators and manual revision was performed on top of an automatic analysis.

## 3 Syntactic Peculiarities of Spanish

As also discussed in Galicia-Haro et al (2002) there are certain features that depend on each language and that make simpler or more complicated the relation between groups of words in a sentence. Thus, to recognise possible combinations of verbs and their complements seems less difficult when we deal with languages with fixed word order than with flexible one. However, this may not happen, because sentence structures in languages have

<sup>1</sup> This research is supported by KNOW Project (MCyT TIN2006-1549-C03-02) and Nevena Tinkova's scholarship (FI Generalitat de Catalunya 2004FI-IQU1/00084). The authors thank Eckhard Bick, Mirkka Soiminen and Lluís Padró for providing the data of HISPAL, Connexor and FreeLing parsers, respectively, due to which was possible to carry out the comparison.

<sup>2</sup> HISPAL (<http://beta.visl.sdu.dk/visl/es/parsing/automatic/trees.php>), Connexor (<http://www.connexor.com/demo/>), FreeLing (<http://garraf.epsevg.upc.es/freeling/demo.php>)

different orders of constituents and present different degrees of freedom. For instance, Spanish is similar to English in the order of the constituents (subject, verb, direct object) but there is much more freedom in Spanish to go off this order. Moreover, constituents in SVO order can also be put in other, less common, orders depending on what is emphasized and what is treated as important information by the speaker. Thus, as Butt (1998) argues factors like context, psychology, register and rhythm affect the ordering of the main constituents of a Spanish sentence. Next, we present some of the most problematic cases for Spanish in parsing that we have found in our study.

As we have already mentioned that Spanish has a relatively flexible word order, we may find a number of cases where subjects and direct objects change their normal positions as illustrated in the below example:

(a)  
Papel fundamental han desempeñado en esta recuperación los evangelios... [Spanish]

\*A fundamental role have played in this recovery the gospels... [English]

Inversion of subjects and objects is a common phenomenon that takes place in Romance languages and is typical of flowery literary style. Thus, in Section 5 we will explain how the inversion of these main sentence constituents leads to misidentification of the syntactic function of both elements in parsing that could be avoided if several observations are taken into account.

Direct objects in many languages occupy the immediate position after verbs without any prepositions in between. This is not the case, however, of Spanish that has an accusative case marker (the preposition *a*<sup>3</sup>), though this is limited to direct objects referring to humans as in (b). Thus, direct objects are preceded by the preposition *a* even if they are animals, countries, collective nouns or social entities (political parties, companies) as in (c) and its omission would be a crass error. Consider the following examples:

(b)  
Pero el colmo es ver a Lucio subiendo al ataque... [Spanish]

But the last straw is to see Lucio getting into attack... [English]

(c)  
Adoro a mi perro pequeño. [Spanish]

I adore my little dog. [English]

In (b) we have a clear example of a direct object referring to a human being *a Lucio*, whereas in (c) we personify the animal because of its close relation with the speaker, that is, the more familiar the language, the more likely the use of *a*. As we may notice, animateness is obviously a

<sup>3</sup> As Demonte (1999) argues some grammarians consider *a* as a particle and not as a preposition because it does not behave as a real preposition when it is used to introduce direct objects.

syntactic trait but it has as well a shade of semantic meaning and in some cases it may help us to distinguish subjects from direct objects.

While the first use of the preposition *a* in (b) denotes a clear example of animate direct object, its second use in the same sentence *al ataque* should conceivably be interpreted as an adverbial. Nevertheless, this reading is not easily available in some parsing systems as we can observe in 5.3.

Overall, each semantic valency can be represented on syntactic level by only one element as is the case of most of the languages. In Spanish, instead, there is the possibility of doubling valencies which according to Demonte (1999) expresses the culmination of the event as in the next sample sentence:

(d)  
Anier García le dio el lunes a Cuba su primera medalla dorada... [Spanish]

\*Anier García it gave on Monday to Cuba his first golden medal... [English]

In (d) it may be claimed that there is a repetition of the indirect object *a Cuba* expressed by means of the use of the dative clitic *le*. In parsing such a sentence, we should try to achieve that systems analyse both elements as indirect object because, on the contrary, they will fail to assign correctly the syntactic function of each constituent (see 5.4).

Other typical phenomena that make difficult the syntactic analysis of Spanish are subject elision and omission of some arguments. Analysing them will be outside the scope in this article but we have taken them into consideration in our study.

## 4 Methodology of the Study

We experiment our methodology on a corpus that consists of approximately 70 sentences (1600 words) extracted from different Spanish electronic newspapers such as *El País*, *El Mundo Deportivo* and *La Vanguardia*. These sentences are of different sizes and we have selected them in such a way that they contain a wide variety of typical syntactic and semantic structures for Spanish that constitute common cases of the linguistic phenomena discussed in the study. Before we proceed to performing a qualitative comparison of the parsers outputs, we analyse the corpus with HISPAL, FreeLing and Connexor systems. We do that taking into account that each parser starts from a morphological analysis and an own tagger. Note that this is important in order to arrive at correct analysis because some of the syntactic errors are due to morphology as shown in the following examples:

(e)  
Animados con la cerveza para hacer frente al bochorno y el calor olvidaron... [Spanish]

Cheered up with the beer in order to stand up the close weather and the heat they forgot... [English]

(f) Irán dice que no aceptará... [Spanish]

Iran tells that will not accept ... [English]

The sentence in (e) is correctly analysed uniquely by HISPAL because FreeLing and Connexor assign a wrong tag to *animados* that in this case is the participle of the verb *animar* and not a noun in plural as described in Connexor or the imperative of the verb *animarse*<sup>4</sup> (animad+os) as FreeLing states. Similar is the situation in (f) where only FreeLing fails in assigning to *Irán* the tag of future tense of the verb *ir* (in English *to go*). As a consequence, wrong syntactic analysis is performed. Fewer errors in morphology in our corpus are observed in HISPAL system because as Bick (2006) argues HISPAL's morphological analyser is a multitagger assigning multiple possible readings to tokenized input. Thus, it reaches accuracy for non-name words of around 99.4%, whereas for FreeLing it is over 97% (Carmona et al 1998). Some of the still unresolved cases in the morphology of HISPAL that can be improved in further editions are on the one hand, the way baseforms are assigned when words end with *s* and we have them in plural and on the other, the treatment of some adverbs. The former can be illustrated with the example of *países* (countries): the singular is *pais* (country) and in order to form the plural we add *-es*, but HISPAL's morphological analyser fails to assign the correct singular form and forms it as *\*paise*. As far as the latter case is concerned, we have the adverb *allá* (there) that in HISPAL's morphology appears as an adjective.

Once the corpus is parsed, we compare the productions of each system, identify the errors and finally we group them according to their source. We believe that some of them can be improved and avoided in further developments of parsers.

## 5 Comparison of the Parsers Outputs

We perform a qualitative comparison using the parsers previously described in Section 2. As we have mentioned in the introduction of this paper, we will focus on some types of errors discussed in the next subsections that might shed light on the reasons for why it is important and useful to compare the performance of these accurate and robust parsers. Other errors such as misidentification of syntactic categories, bad coordination of clauses and phrases, structural ambiguities, misidentification of constituents in interrogative sentences, relative clauses introduced by prepositions (*de / en / con quien*) are also considered and discussed in the study.

### 5.1 Syntactic Function Misidentification

A very frequent case of errors in the corpus concerns function mistagging as for instance the errors observed

below – subjects and objects misidentified due to their inversion ((a) and (g)) and time adverbs occupying positions of the main sentence constituents. Consider example (g) and the analyses assigned to it by the three parsers in figures 1, 2 and 3, respectively:

(g) El partido se convirtió en la pesadilla que había pronosticado Luis. [Spanish]

\*The match turned into the nightmare that had predicted Luis. [English]

#### Figure 1. Connexor annotation

```
8 que que subj>9 @NH PRON Rel
9 había haber v-ch>10 @AUX V IND IMPF SG P3
10 pronosticado pronosticar mod>7@MAIN V PCP PERF MSC
11 Luis luis obj>10 @NH N MSC SG Prop
```

#### Figure 2. HISPAL annotation

```
que [que] <rel> SPEC MF SP @ACC> @#FS-N<
había [haber] V IMPF 1/3S IND VFIN @FAUX
pronosticado [pronosticar] V PCP M S @IMV @#ICL-AUX<
Luis [Luis] PROP M S @<SUBJ
```

#### Figure 3. FreeLing annotation

```
subord/modnomatch/(que que PROCN000)
grup-verb/modnorule/(pronosticado pronosticado VMP00SM)
vaux/modnomatch/(había haber VAIIS0)
sn/obj/(Luis luis NP00000)
```

If we take a closer look at this example, we will note that only HISPAL parses correctly the sentence and assigns to *Luis* the syntactic function of subject whereas Connexor and FreeLing fail to identify it. As far as this case is concerned we consider that this error can be avoided because, as it was previously claimed, animate direct objects in Spanish are always preceded by the preposition *a* and here it is absent.

Another important fact that should be stated in parsers in order not to mistag functions is that subjects agree with the main verb whereas objects need not. Observe example (a) where it becomes clear that Connexor and FreeLing do not apply this knowledge and consequently produce an error analysis of the sentence. On the contrary, HISPAL includes it and gets its correct analysis.

We suppose that the frequent misidentification of time adverbs with subjects / objects seen in (h) and (k) can be accounted for by the fact that Connexor does not specify in its lexicon all the possible time adverbs. As a result, it analyses *esta noche* in (h) as an adverbial, whereas in both cases, (h) and (k), HISPAL does arrive at its correct assignment because *esta tarde* (this afternoon) and *esta noche* are added as time adverbs in the lexicon. FreeLing fails in both cases misidentifying the adverbs in (h) as a subject and as an object in (k) depending on the position they occupy in each sentence.

(h) Esta noche se sabrán los detalles del lanzamiento europeo de PlayStation 3. [Spanish]

<sup>4</sup> In English “to cheer up”

This evening we will know details about the European launch of PlayStation 3. [English]

## 5.2 Prepositional Phrases with *a*

Apart from introducing direct and indirect objects, the preposition *a* in Spanish is also used with verbs denoting movement as in (b) and (k). In (b), both FreeLing and Connexor assign wrong tags to *al ataque* analysing it as an indirect object and an object, respectively. The former sends the same error in (k) whereas the latter parses it correctly. HISPAL, however, manages to produce a correct grammatical analysis of both examples indicating that the prepositional phrases in (b) and (k) are adverbials.

(k) Ven a la tienda oficial esta tarde. [Spanish]

Come to the official shop this afternoon. [English]

In order to solve these misanalyses, we consider that it is convenient to encode in subcategorization frames for verbs more information that will permit us to establish how many arguments are required by verbs, of what syntactic type and thus, distinguishing which constituents are their arguments and which are adjuncts.

## 5.3 Repetition of Valencies

Having exposed in 3.3 what a repetition of valencies is, here we will discuss how it is handled by the three parsers. Consider (d) and its analyses with HISPAL, Connexor and FreeLing in figures 4, 5 and 6, respectively:

Figure 4. HISPAL annotation

Anier=García	[Anier=García]	PROP MF SP	@SUBJ>
le	[le]	PERS MF 3S DAT	@DAT>
dio	[dar]	V PS 3S IND VFIN	@FMV
el	[el]	<art> <dem> DET M S	@>N
lunes	[lunes]	N M S	@<ADVL
a	[a]	PRP	@<ADVL
Cuba	[Cuba]	PROP F S	@P<

Figure 5. Connexor annotation

1	Anier	anier	attr>2	@PREMOD	Heur N SG Prop
2	García	garcía		@NH N MSC SG Prop	
3	le	lo	dat>4	@NH PRON Pers SG P3 DAT	
4	dio	dar	main>0	@MAIN V IND PRET SG P3	
5	el	el	det>6	@PREMOD DET MSC SG	
6	lunes	lunes	subj>4	@NH N MSC SG	
7	a	a	pm>8	@PREMARK PREP	
8	Cuba	cuba		@NH N MSC SG Prop	

Figure 6. FreeLing annotation

```

grup-verb/top/(dio dar VMIS3S0)
patons/modnomatch/(le él PP3CSD00)
sn/subj/(Anier_García anier_garcía NP00000)
data/modnomatch/(lunes [L:??/??:??:??:??] W)
espec-ms/modnorule/(el el DA0MS0)
grup-sp/iobj/(a a SPS00)
sn/head/(Cuba cuba NP00000)

```

All of them analyse correctly the clitic pronoun *le* as dative, but only FreeLing arrives at its best parse because it takes into account that the indirect object is realized once by a full noun phrase *a Cuba* and once by the clitic pronoun *le*. HISPAL and Connexor do not consider the possibility that direct and indirect objects can be doubled by a clitic pronoun and as a result, they analyse them as an adverbial and a prepositional phrase, respectively. Capturing such important differences in treating main sentence constituents justifies further research in this direction.

## 5.4 Multiword expressions

Many linguists (Sag 2001) claim that multiword expressions pose a key problem for the development of large-scale precise natural language processing technology because they are still insufficiently investigated as we can see in the analyses assigned to (l) by HISPAL, Connexor and FreeLing in figures 7, 8 and 9, respectively:

(l) Fernando Alonso da las gracias a sus seguidores... [Spanish]

Fernando Alonso gives thanks to his fans... [English]

Figure 7. HISPAL annotation

Fernando=Alonso	[Fernando=Alonso]	PROP MF SP	@SUBJ>
da	[dar]	V PR 3S IND VFIN	@FMV
las	[la]	<art> <dem> DET F P	@<ACC
gracias=a	[gracias=a]	PRP	@N<
sus	[su]	<poss 3S/P> <si> DET MF P	@>N
seguidores	[seguidor]	N M P	@P<

Figure 8. Connexor annotation

1	Fernando	fernando	ada:>2	@<Proper> N MSC SG
2	Alonso	alonso		@NH <Proper> N MSC SG
3	da	dar		@MAIN V IND PRES SG3
4	las	las	det:>5	@PREMOD DET FEM PL
6	gracias	gracia	subj:>3	@NH N FEM PL
7	a	a	pm:>8	@PREMARK PREP
8	sus	su	ada:>8	@<Poss> PRON COM PL
9	seguidores	seguidor	mod:>5	@NH N MSC PL

Figure 9. FreeLing annotation

```

grup-verb/top/(da dar VMIP3S0)
sn/subj/(Fernando_Alonso fernando_alonso NP00000)
sn/obj/(gracias_a gracias_a SPS00)
j-fp/modnomatch/(las el DA0FP0)
sn/modnomatch/(seguidores seguidor NCMP000)
espec-mp/espec/(sus su DP3CP0)

```

The example in (l) *dar las gracias a* (in English *to give thanks to*) constitutes a clear case of light-verb constructions, that is, the noun is used in a normal sense whereas the verb meaning appears to be bleached. In the same sentence, however, if we do not take into account its meaning, we can find the compound preposition *gracias a* (in English *thanks to*). Seemingly, *gracias a* is a phrase that tends to appear more frequently in Spanish than the expression *dar las gracias a* because both FreeLing and HISPAL have it in their lexicons as a preposition.

Consequently, they fail to assign the correct analysis to some of the constituents of the sentence. From the example parsed with Connexor, we may conclude that no such information is encoded in this system but, even though, it does not achieve to analyse the sentence correctly. Yet, it is also misled in parsing the proper name *Fernando Alonso*. Therefore, as also Sag (2001) argues, a possible solution of the problem would be the development of a lexical selection mechanism, where a sign associated with one word of the phrase selects for the other word.

## 6 Comparative Evaluation

As already mentioned, we have investigated in our study what language phenomena the parsers cover and what quality of the outputs is provided by each parser. Globally, the HISPAL parser leads to more accurate analyses than Connexor and FreeLing as we can explore this in Sections 4 and 5. Yet, it may be claimed that advantages of HISPAL are its capacity to identify subjects and objects when they are inverted, prepositional phrases as adverbials not confusing them with objects, constituents of questions, recognition of time adverbs (such as *esta tarde*, *esta noche*) and to coordinate properly phrases and clauses. On the other hand, FreeLing has trouble with phrases introduced by the preposition *a* analysing all of them as indirect objects whereas Connexor and HISPAL do make difference between objects and adverbials. It is also confused by grouping constituents of sentences and presents several shortcomings in morphology that lead to wrong parses (see (e) and (f)). Nevertheless, FreeLing handles properly, among others, some fixed expressions, verb periphrases, clitic pronouns and date-time expressions.

However, all parsers are misled by examples (i) and (j) where none has achieved to identify correctly the compound subjects of the sentences. They are unable as well to parse correctly clauses with ellipsis of the head of a phrase, noun phrases postmodified by several prepositional phrases and a great part of yes/no questions when they begin with a verb form that coincides with a noun.

Obviously, the qualitative comparison of the parsers outputs deserves to be studied; it is of value for the further improvement of syntactic analysis and can be taken as departure point of doing a quantitative comparison based on actual empirical evidence.

## 7 Conclusions and Further Work

In this paper we have presented a comparative study of the outputs performed by the available existing parsers HISPAL, FreeLing and Connexor. We have discussed also some of the most frequent errors found in the parsed corpus that should be further resolved. Now, as an immediate future work we have in mind to carry out a quantitative evaluation of the accuracy of the three parsers

in order to establish the current state of syntactic analysis for Spanish.

As another future line of work, we plan to use the results exhibited in this preliminary research for the development of a broad-coverage grammar for Spanish using the dependency approach to syntactic pattern analysis.

To sum up, we hope to have demonstrated how useful the study of the sources of errors by comparing the parsers productions can be for improving syntactic analysis for Spanish even when we use a small corpus. Thus, we believe that the results from this experiment validate the research in this direction.

## 8 References

- [1] Atserias, J., B. Casas, E. Comelles, M. González, L. Padró and M. Padró (2006). FreeLing 1.3: Syntactic and semantic services in an open-source NLP library. In: *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*. Genoa, Italy.
- [2] Bick, Eckhard (2006). A Constraint Grammar-Based Parser for Spanish. In: *Proceedings of TIL 2006 - 4th Workshop on Information and Human Language Technology*. Ribeirão Preto, Brasil.
- [3] Butt, J. and C. Benjamin (1998). *A New Reference Grammar of Modern Spanish*. Arnold ed., London.
- [4] Carmona, J., S. Cervell, L. Márquez, M. A. Martí, L. Padró, R. Placer, H. Rodríguez, M. Taulé, and J. Turmo (1998). An Environment for Morphosyntactic Processing of Unrestricted Spanish Text. In: *Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC'98)*. Granada, Spain.
- [5] Carreras, X., I. Chao, L. Padró and M. Padró (2004). FreeLing: An Open-Source Suite of Language Analyzers. In: *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal.
- [6] Demonte, V. and I. Bosque (1999). *Gramática descriptiva de la lengua española*. Espasa Calpe, Madrid.
- [7] Galicia-Haro, S., A. Gelbukh and I. Bolshako (2002). Análisis sintáctico para el español basado en el formalismo de la teoría Significado  $\Leftrightarrow$  Texto. In: *Sociedad Española para el Procesamiento del Lenguaje Natural*. Valladolid, Spain.
- [8] Sag, I., T. Baldwin, F. Bond, A. Copestake and D. Flickinger (2001). Multiword Expressions: A Pain in the Neck for NLP. In: *Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2002)*, Mexico City, Mexico.
- [9] Tapanainen, Pasi (1996). "The Constraint Grammar Parser CG-2". No 27, Publications of the Department of General Linguistics, University of Helsinki.

# Extracting Collocations in Context: the case of Romanian VN constructions

Amalia Todirascu

LILPA / Université Marc Bloch  
Strasbourg  
22, rue René Descartes, BP 80010  
67084 Strasbourg cedex  
todiras@umb.u-strasbg.fr

Christopher Gledhill

LILPA / Université Marc Bloch  
Strasbourg  
22, rue René Descartes, BP 80010  
67084 Strasbourg cedex  
todiras@umb.u-strasbg.fr

Dan Stefanescu

Research Institute for Artificial  
Intelligence, Romanian Academy  
Calea 13 Septembrie, 13,  
Bucharest 050711, Romania  
danstef@racai.ro

## Abstract

We present here a linguistic analysis of verbo-nominal (VN) constructions in Romanian with a view to developing a system for the extraction of lexical collocations from large tagged and annotated corpora. We identify the salient morpho-syntactic properties not only of the collocation but also of the context surrounding the expression.

## Keywords

VN constructions, collocation extraction.

## 1. Introduction

This paper presents an on-going project for the *Agence universitaire pour la Francophonie* (AUF), whose aim is to develop an extraction tool for a multilingual collocation dictionary (German, French, Romanian). We focus here on the specific properties of Romanian collocations and on the linguistic resources developed to extract them from texts. Collocations are sequences of frequently co-occurring words which have a specific syntactic behaviour and a specific sense. Their idiomatic use is difficult for non-native speakers, and especially for Natural Language Processing (NLP) systems. Few dictionaries, whether traditional or electronic, provide complete information about collocations. While most explain the sense of idiomatic expressions, they often do not give any information about the morpho-syntactic behaviour of the expression. However, several methods and tools for extracting collocations from text have been developed.

Several definitions have been proposed for ‘collocation’ and few definitions are appropriate for the purposes of NLP systems. Collocations have been seen as “frequent word co-occurrence” [5], “a conventional way of saying things” [17] or a “fixed phrase” [10] [11]. As proposed in [6], three interpretations of the notion of ‘collocation’ are: **cooccurrence**, a statistical view [25]; **construction** (or ‘colligation’), in terms of lexico-syntactic relations [12], and **expression**, a semiotic unit from the point of view of pragmatics [18],[8]. We adopted the lexico-grammatical view of collocation, assuming that a collocation is made up of a base and a collocate, and whose syntactic relations can be described in terms of a generic pattern (such as V + N, N + ADJ, ADV + ADJ etc.), used

to automatically extract collocations.

In this paper, we focus on verbo-nominal (VN) constructions such as *make a decision / a lua o decizie*, *to make an application / a pune în aplicare* etc. VN constructions are associated with a subset of morpho-syntactic properties, such as a preference for the definite article or zero-article, for singular or plural noun, for the presence of an indirect complement, etc. These subregularities are important for an automatic extraction tool, since by using contextual information of this type, an NLP system can filter out salient collocations from a larger set of candidates, identified by statistical measures.

There have been several approaches which only use statistical methods for collocation extraction ([19], [21]), while other approaches identify collocations by purely looking at syntactic relations [24] or using both syntactic and semantic properties [27] [4]. In this paper we adopt a hybrid approach to extract VN constructions, in that we use a statistical module to extract VN co-occurrences and then apply a set of language-specific filters. The linguistic filters we use here were defined as a result of comparative linguistic data, carried out on a parallel corpus.

## 2. Methodology

We have adopted here a method which has already been applied to extract collocations from German corpora [14], [20]. These studies assume that collocations have their own morpho-syntactic properties. Their methodology has been used to analyze a large corpus in which any relevant morpho-syntactic information (preference for DEF ART, specific PREPs, case in German) is taken into account from the surrounding context of the expression.

In our project, a similar analysis has been applied to Romanian and to French. First, we identify common morpho-syntactic properties in the three languages. This is necessary in order to develop parametrizable tools for the automatic identification of collocation candidates. The next step involves a statistical module to establish a complete list of candidates, from parallel, tagged corpora [28]. Next, non-salient candidates are filtered out, using morpho-syntactic information. We are currently adapting several tools which already exist for German [16], French, and Romanian [26]. However, this process is only semi-

automatic, a final manual check of candidates is necessary.

### 3. The Corpus

In order to identify language specific filters, we require tagged and preferably syntactically annotated corpora. We have used a parallel corpus available in the languages of the EU was used: the *AcquisCommunaire Corpus* (ACC) [22], containing all the main legal texts published by the EU member states since 1950. We selected a set of common documents from the ACC in French, German and Romanian (about 15 million words for each language). The style of the ACC is impersonal, and it contains many domain-specific terms and fixed expressions, typical of administrative texts. In order to compare and to select only relevant collocations, it is necessary to compare our specialized corpora with more general text archives.

We set up various reference corpora containing similar genres (literature, newspapers, technical papers), to adjust the set of properties extracted from the ACC. We cleared these corpora of tables, pictures, irrelevant structural elements, and applied a uniform encoding to each. For instance, the Romanian corpus about 10 million words: the RoCo corpus (newspapers); the NAACL corpus (newspapers, Romanian constitution and 2 novels); a philosophical treatise (Eliade); a medical corpus, the L4TE corpus (computer science). One problem was to select only texts with proper diacritics, because in Romanian the absence of diacritics might change the case or sense of the word, e.g. *fata* ‘the girl’ / *fața* ‘the face’.

In order to identify construction-specific morpho-syntactic properties, we use a tagged and syntactically annotated corpus. The French corpus has been tagged with a tagger trained on a corpus previously annotated using TreeTagger [23], while the Romanian corpus was tagged using the TTL platform [14].

Syntactic information is important to interpret the functional role played by a collocation or by various components co-occurring with the candidate. As the German corpus is annotated at chunk level, we annotated the French data at chunk level, using the Syntex parser [3]. and the Romanian data with the TTL platform.

### 4. V-N Collocations

As mentioned before, we hypothesize that VN constructions can be identified by finding collocations sharing several morpho-syntactic properties extracted from their immediate context. We are currently concentrating on Verb-Noun collocations, due to the productivity of this type of construction. For example, the light verbs [1] or support verbs that typically occur in VN constructions, such as *face* / *faire* / *make* or *lua* / *prendre* / *take*, have very different morpho-syntactic properties according to context, and a complete multilingual dictionary should explicitly represent this information. Generative grammarians [9] assume that these properties are determined by the specific type of

‘predicate noun’ alone, and they therefore minimize the role of the verb. Here we adopt a different perspective. As set out in [7], we propose that all VN constructions involve a ‘generic’ V which determines the argument structure of the predicate, and a ‘specific’ N which expresses the semantic process or ‘range’ ([2], [12]) of the predicate, as in *make a decision*, *take flight*, etc.

The most salient morpho-syntactic properties of VN constructions and the relation with the three levels of analysis can be seen in the following examples (from [7]):

**V1. Morphology.** Some VN constructions are related etymologically to a simple V (*to do work* / *to work*, *a se face noapte* / *a înnopta* ‘to get dark’). But this equivalence is not always possible (*take a break* / *a face o pauză* is not the same as or is unrelated to *to break* / *\*a pauza* )

**V2 Arguments.** Like simple Vs, VN constructions can take direct or indirect complements: *The candidate gave the electors a fright* / *Candidatul a băgat spaima în electorat*, *He put a brave face on the situation* / *A facut față situației*.

**V3 Passive test.** Some VNs can have passive forms (*Pierre made a decision* / *Pierre ia o decizie* vs. *The decision was made by Pierre* / *O decizie a fost luată de Pierre*), but others do not: *to take flight* / *?a flight is taken face obiectul*, *\*obiectul a fost făcut* ‘to be subject to ...’. We have to mention that these examples are not translations of each other; they are intended to show the differences between Romanian and English.

**V4. Aspect.** Some VN constructions express perfective aspect [29]: *She laughed* / *She gave a laugh* / *She laughed for hours* / *?She gave a laugh for hours*. In Romanian, this property is not available.

In addition, VN constructions also share some morpho-syntactic properties with Ns:

**N1 Determination.** The DET is often absent or fixed in many VN idioms (*take flight*, *a face obiectul* ‘to be subject to’). When the N can be identified in referential contexts, the DET often becomes more variable (*to take an important decision*, *a luat o decizie importantă*).

**N2 Clefting.** The N in some VN constructions cannot be extracted (*He took flight* / *\*It was the flight that he took* *El și-a luat zborul* / *\*Zborul pe care și l-a luat*).

**N3 Expansion.** The N sometimes cannot be modified by relative clauses or other qualifiers (*He took the decision which was necessary* / *\*He took the flight which was necessary*, *?El a luat decizia care se impunea*, *?He took the flight which was necessary* / *\*el și-a luat zborul care se impunea*).

**N4 Conversion.** Some VN constructions cannot be nominalized (*The commission takes measures* / *Comisia a luat măsurii*, *The taking of measures by the commission* / *Luarea măsurilor de către comisie*).

So far we have evaluated these properties (V1-V4, N1-N4) in relation to French. In the following section we examine to what extent they apply to Romanian data, and



we present some conclusions about the kinds of syntactic filters necessary to extract collocation candidates.

## 5. The Romanian Data

Romanian grammar is very close to Latin. Ns are characterized by the following properties: number, gender, and 5 cases. Case is marked by a specific ending (if the N is determined by an enclitic definite ART) or indefinite ART (*unei / unui / unor / unora* / = of some) or PREP (*pe-* literally ‘on’, for the accusative). The DEF ART is added as an ending for definite nouns (*omului, casei, oamenilor, caselor*). Verbal morphology is characterized by mode (indicative, subjunctive etc.), tense (present, past, future...), number and person. The subject is not mandatory as in other Romance languages, and the perfect is usually formed with the auxiliary ‘*a avea* / to have’.. The passive is always made up of the auxiliary *a fi* / ‘to be’ followed by the past participle form of ‘be’, and by the past participle of the verb. The order of syntactic components is free.

### 5.1 The Case of *a face* (to do or make)

In order to identify the specific properties of VN constructions in Romanian, we studied the specific contextual properties presented in section 4. We looked in particular at morphology (V1, N1), the syntactic functions of the V and of the N, as well as their semantic roles. We searched for relevant information in the Romanian ACC corpus and in the general Romanian corpus.

VN constructions have several V-specific properties in Romanian. While V1-V3 are still valid tests for VNs, V4 (aspect) could not be used. For example, V1 applies to Romanian (the predicate can be replaced by a simple V), as in *a se face noapte* > *a înnopta* (‘night falls’, literally ‘it makes dark’), *a face dovada* / > *a dovedi* / to prove). Several idiomatic expressions cannot be replaced by a simple verb (*a face față* / \*to make face > *a fața* / to face, *a face obiectul* / to be subject to but this is not the same meaning as ?*a obiecta* / to object. The passive test (V3) is used to show that many of these expressions are idiomatic.

If the properties V1-3 apply to Romanian, although in different ways, as we have seen, the nominal properties N1-4 present some specific features. Extraction is not possible in Romanian. Expansion of the complement (N3) is however possible by modifying nouns with relative clauses: *al cărui obiect îl face* (‘whose object is ...’), *a cărei dovadă este...* (‘whose proof is...’). The determiner (N1) is fixed in several idiomatic expressions: *a face obiectul* – ‘be subject to’, *a face dovada* – \*‘to make proof of’ (definite article), *a face față* – ‘to face’ (no definite article),

### 5.2 Semantic Properties

In systemic functional grammar [13], the semantic role played by many nouns in VN constructions is known as ‘process range’. The process range expresses the semantic process of the predicate, and is often integrated into the

verb group [7] (as in *a face obiectul* ‘to be subject to...’). Any indirect complement which follows this element then becomes the semantic object (or ‘goal’). In French and English, this indirect complement is usually introduced by a PREP, but in Romanian this role is filled by the genitive case. In (1), the complement expresses a simple relational process. However, in (2) we have more complex situation (subject reading):

(1)...*să facă obiectul unei proceduri administrative...*

‘is the subject of an administrative procedure’

(2) ...*la instituțiile financiare, care fac parte din categoria....*

‘in financial institutions which are part of this category’

The most frequent collocations of *face* in the Romanian Acquis Communautaire are VN constructions where the N has been integrated into the verb group (VG). In French, it is possible to establish a relation between specific types of ART (definite, indefinite and zero) and a specific process type (e.g. material processes tend to be definite) [7]. But again, this is not possible for Romanian; VN constructions with a definite suffix (*face obiectul, face dovada, face legătura*) are mostly relational process, and the process range is expressed by the indirect complement:

(3)...*Trece peste granița dintre statele membre și care face legătura între sistemele de transport...*

‘...crosses the border between member states and which joins the transportation system...’

In VN constructions where Ns have indefinite ART (*fac+un / o / unele / niște* + N) several semantic processes can be identified: mental (verbal communication, as (4) or material as in (5):

(4) *se face un proces verbal al fiecărei ședințe a ...*

‘Minutes shall be taken of all meetings’

(5) *Comisia poate să facă orice modificări la prezentul Regulament care ...*

‘The commission should make some changes in the present rules...’

Among VN constructions without articles, we found several relational process: (*a face față* / to face, *a face parte* / be part of, *a face obiectul* / is subject to) :

(6) *Pentru a putea face față unor situații de urgență*

‘in order to deal with emergency situations’...

Other VN constructions where the DET is absent are mostly material intransitive processes: *face vizite* / to pay visits, *face comerț* / \*to make trade.

We conclude that in Romanian, as with English and French, there is a certain tendency for groups of words to lexicalize with a corresponding rigidity of morpho-syntactic features (preference for indefinite ART, systematic use of some specific classes of PREP etc.). These features are relevant to a module for filtering such expressions.

## 6. Automatic Extraction

As presented in section 2, our extraction approach combines statistical techniques and pattern-based matching in order to filter candidates.

### 6.1 The Statistical Module

Verb Noun pairs co-occurring together frequently (separated by one or several words) are potential collocation candidates. We have applied a statistical module for extracting V-N pairs from the corpora, based on [21], using mean and variance. The mean is the average of the distances between the words forming the pair, while the variance measures the deviations of the distances with respect to the mean already computed. Collocations are pairs of words for which the standard deviations of distances are small. We computed the standard deviation for all V-N pairs (from the ACC corpus) within a window of 11 content words length for all the three languages involved in the project and we considered as good, all the pairs for which standard deviation was smaller than 2 [21].

We want to further filter out some of the pairs so that we keep only those composed by words which appear together more often than expected by chance, using Log-Likelihood (LL). The idea behind the LL score is finding the hypothesis which describes better the data:

$$H_0 : P(w_2|w_1) = p = P(w_2|\neg w_1)$$

(null hypothesis - independence)

$$H_1 : P(w_2|w_1) = p_1 \neq p_2 = P(w_2|\neg w_1)$$

(non-independence hypothesis)

The LL score formula is:

$$LL = 2 * \sum_{j=1}^2 \sum_{i=1}^2 n_{ij} * \log \frac{n_{ij} * n_{**}}{n_{i*} * n_{*j}}$$

where  $n_{ij}$  represents the number of occurrences when the words  $w_i$  and  $w_j$  appear together,  $n_{i*}$  is the number of occurrences for  $w_i$  together with any  $w_j$ , etc.

We computed the LL score for all the pairs obtained by the first method. We kept in a final list the pairs for which the LL score was higher than 9 (see the table for Chi-square distribution with one degree of freedom). Using LL filtering, we obtained a list of candidates for Romanian collocations (table.1) Among the top pairs extracted, we identify some valid candidates, expressing processes (*face obiectul*, *aduce atingere*, *intra în vigoare*, *face modificări*), but the other candidates are not collocations (morpho-syntactic properties are variable). The *face+noun* constructions identified among the first 20 candidates are collocations and have specific morpho-syntactic properties (no article or definite, preference for singular). For all these pairs, we apply linguistic filters to select valid candidates.

Fig.1 First LL score

$w_1 w_2$	dist	LL score	Process
<i>Aduce atingere</i> 'to affect/to prejudge'	1	51567.34864	Relation process
<i>inlocui text</i> 'replace text'	3	43992.3067	-
<i>intra vigoare</i> 'applied' (or literally 'placed in vigour')	2	42527.03736	Relational process
<i>Face apel la</i> 'call for' (or literally 'to make a call')	3	32050.11219	Relational process
<i>face obiect</i> 'be subject (to)'	1	30729.47663	Relational process
<i>Face modificări</i> 'make changes'	4	29141.39454	Material process

### 6.2 The Filtering Module

As we saw in section 5, some Romanian collocations have specific morpho-syntactic and semantic properties. We use these properties to extract relevant candidates from the statistical module output. We mainly use a set of patterns, manually defined, based on linguistic analysis.

One example of an extraction pattern identifies the sequence P (predicate) + C (complement) (direct) + C (indirect), or in tagged code «*a face NxRY \*{1,5} NxOY*»; NxRY means Noun (plural or singular), in direct case (Nominative or Accusative definite form); NSOY means Noun, singular, oblique case (Genitive or Dative case definite form); {1,5} means 1 up to 5 words. This sequence alone can identify four valid VN constructions among the candidates proposed by the statistical module: *face obiectul*, *face dovada*, *face subiectul*, *face transferul*. Another pattern for *face* constructions combined with the preposition *cu* (with) (*face NxRY \*{1,5} cu*) identifies some interesting candidates: *a face legătura cu* (makes a link with), *a face declarația cu privire la* (make a declaration in relation to...). These candidates involve various relational processes: *a face legătura cu* ('relate'), *a face transferul* ('transfer'), but also some communicative processes as well *a face declarația cu privire la* ('to declare'). In addition, *V+în / in* selects candidates as *înlocui în text* ('to place in text'), *intra în vigoare* ('to apply / to enter into force').

## 7. Conclusion

The paper has presented some features of VN constructions in Romanian. Generally speaking, Romanian shares most of the properties of VN constructions that have been identified for Western European languages. The difference is that the specific configuration for each VN construction is different. The verb *a face* (equivalent to French *faire*) operates syntactically in the same way as *faire*, but does not cover the same semantic ground. It is also clear from this study

that the relevant context for all of these expressions extends way beyond the basic V plus N collocation: in almost every case, the expression involves a specific morpho-syntactic configuration and has a phraseology and context of use which is highly consistent. Our conclusion must therefore be that the contextual features of VN constructions are crucial to the semi-automatic extraction of collocations.

## 8. Acknowledgements

This work has been funded by Agence Universitaire pour la Francophonie (AUF). We thank Rada Mihalcea (University of Texas, United States) for the NAACL corpus, Dan Tufiş (Romanian Academy) for the tagging tools, as well as Dan Cristea (University of Iasi, Romania) for the L4TE corpus.

## 9. References

- [1] Allerton, D., 2002. Stretched Verb Constructions in English, London, Routledge.
- [2] Banks, D., (2000). The Range of Range: A transitivity problem for systemic linguistics, *Anglophonia*, 8, 195-206.
- [3] Bourigault D. & Fabre C.(2000), Approche linguistique pour l'analyse syntaxique de corpus, *Cahiers de Grammaires*, n° 25, pp. 131-151.
- [4] Bolshakov, I.A., Gelbukh. A. (2001). A Very Large Database of Collocations and Semantic Links. NLDB'2000. Lecture Notes in Computer Science N 1959, Springer-Verlag, pp. 103–114
- [5] Cowie, A. P. (1981) The treatment of collocations and idioms in learner's dictionaries, in *Applied Linguistics*, 2(3), 223-235.
- [6] Gledhill, C., (2000). Collocations in Science Writing, Gunter Narr Verlag, Tübingen
- [7] Gledhill (2007) La Portée : seul dénominateur commun dans les constructions verbo-nominales. In Frath, P. Pauchard, J. & Gledhill, C. (Eds.), 2007, *Actes du 1er Colloque, Res Per Nomen*, Université De Reims-Champagne-Ardenne, 24-26 Mai 2007 : 113-125.
- [8] Gledhill, C, Frath, P., (2007) Collocation, phrasème, dénomination: vers une théorie de la créativité phraséologique, in *La Linguistique*. Vol 1/1.
- [9] Grimshaw, J. & Mester, A., (1988). Light Verbs and  $\theta$ -Marking, *Linguistic Inquiry*, 19, 205-232.
- [10] Gross, M. (1993) Les phrases figées en français. *L'information grammaticale* 59, Paris, 36-41.
- [11] Grossmann, F., Tutin, A.(eds.) (2003). Les collocations: analyse et traitement, Numéro special: *Travaux et Recherches en Linguistique Appliquée*.
- [12] Hausmann, F.J. (2004). Was sind eigentlich Kollokationen?, en K.Steyer (eds.) *Wortverbindungen – mehr oder weniger fest*, 309-334
- [13] Halliday, M., (1985). *An Introduction to Functional Grammar*. London, Arnold.
- [14] Heid, U., Ritz, J. (2005) Extracting collocations and their contexts from corpora, *Actes de COMPLEX-2005*, Budapest.
- [15] Ion, R. (2007). TTL: A portable framework for to-kenization, tagging and lemmatization of large corpora. Research Institute for Artificial Intelligence, Romanian Academy, Bucharest (in Romanian), 22p.
- [16] Kermes, H. (2003) *Off-line (and On-line) Text Analysis for Computational Lexicography*, Ph.D. thesis IMS, University of Stuttgart, AIMS, vol. 9, number 3.
- [17] Manning, C., Schütze, H. (1999), *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge.
- [18] Moon, R., (1998). *Fixed Expressions and Text*. Oxford, Oxford University Press.
- [19] Quasthoff, U. (1998). Tools for Automatic Lexicon Maintenance: Acquisition, Error Correction, and the Generation of Missing Values. In *Proceedings of LREC'02*, ELRA, S. 853-856.
- [20] Ritz, J., Heid, U. (2006) Extraction tools for collocations and their morphosyntactic specificities, in: *Proceedings of LREC-2006*, Genova, Italia, 2006.
- [21] Smadja, F. A., McKeown, K. R. (1990), Automatically extracting and representing collocations for language generation, *Proceedings of ACL*, 252-259, Pittsburgh..
- [22] Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C. Erjavec, T., Tufiş, D., Varga, D. (2006), The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages, *Proceedings of LREC'06*, pp.2142-2147.
- [23] Schmid, D. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of International Conference on New Methods in Language Processing*.
- [24] Seretan, V., Nerima, L., Wehrli, E. (2004). A tool for multiword collocation extraction and visualization in multilingual corpora, *Proceedings of EURALEX'2004*, Vol2, pp.755-766.
- [25] Sinclair, J., (1991). *Corpus, Concordance, Collocation*, Oxford, Oxford University Press.
- [26] Stefanescu D, Tufis, D, Irimia E. (2006) Extragerea colocatiilor dintr-un text, Atelierul « Resurse lingvistice si instrumente pentru prelucrarea limbii române », Iasi.
- [27] Tutin, A (2004). Pour une modélisation dynamique des collocations dans les textes, *Actes du congrès EURALEX'2004*, Lorient, France, 2004, Vol. 1, 207-221.
- [28] Tufiş, D., Ion, R., Ceauşu, A., Stefanescu D. (2005). Combined Aligners. In *Proceeding of the ACL2005 Workshop on "Building and Using Parallel Corpora: Data-driven Machine Translation and Beyond"*, Michigan, pp. 107-110.
- [29] Wierzbicka, A., (1982). 'Why can you Have a Drink when you can't Have an Eat?', *Language*, 58.

# Formalising and bottom-up enriching the Ontology of a Generative Lexicon

Antonio Toral and Monica Monachini  
Istituto di Linguistica Computazionale  
Consiglio Nazionale delle Ricerche  
Via G. Moruzzi 1 - 56124 Pisa, Italy  
*antonio.toral, monica.monachini@ilc.cnr.it*

## Abstract

This paper presents on-going research to formalise the ontology of a computational lexicon in OWL (W3C standard) as well as to enrich it by applying a bottom-up approach that extracts semantic information from the lexicon. The resource used follows the Generative Lexicon (GL) theory and therefore (1) puts a challenge to ontology design as its semantic types are multidimensional and (2) enables the acquisition of further knowledge on concepts from semantic units. The formalisation allows the ontology to be processed by Description Logics reasoners as well as to be employed in Semantic Web applications. Moreover, the lexicon-driven enrichment increases the semantic information present in the ontology making it appropriate for ontology-driven Natural Language Processing. Finally, the paper studies the application of these procedures to a subsequent GL-based biological resource.

## Keywords

Ontologies, OWL, Web Ontology Language, Generative Lexicon, Lexicon, Qualia Structure, Natural Language Processing, Semantic Web

## 1 Introduction

Ontologies are recognised as an important component in Natural Language Processing (NLP) systems that seek to deal with the semantic level of language. In fact, most, if not all of the semantic lexical resources within the area (e.g. WordNet [3], CYC [5], SIMPLE [6]), have in common the presence of an ontology as a core module.

The Web Ontology Language (OWL) is a W3C recommendation and a major technology for the Semantic Web. It is defined by [1] as “a semantic markup language for publishing and sharing ontologies on the World Wide Web”. OWL allows applications to process the content of information instead of just presenting it to the user [7].

The fact that OWL is the ontology language for the Semantic Web and that it provides a formal semantic representation as well as reasoning capabilities has encouraged the NLP community to convert existing resources to this language. Work in this area includes, for example, the conversion of WordNet [13] and MeSH

[12] and, moreover, the proposal of a general method for converting thesauri [14].

This paper deals with the conversion into OWL of the ontology of a lexico semantic resource based on the Generative Lexicon (GL) theory. The ontology design presents a challenge as the nodes of the ontology are not only defined by their formal dimension (taxonomic hierarchy), but also by additional dimensions: constitutive, telic and agentive. Besides, we take advantage of the generative possibilities of the resource in order to enrich the converted ontology with further semantic information extracted from the lexicon. The final objective of this research is to derive a formalised and semantically rich ontology which could be used for Information Extraction and Knowledge Acquisition tasks.

The rest of this paper is organised as follows. Section 2 introduces PAROLE-SIMPLE-CLIPS, the GL resource used in this research. Next, section 3 deals with the formalisation and enrichment of the ontology. Subsequently, section 4 discusses the application of these techniques to a GL-based biological resource. Finally, section 5 presents conclusions and future work lines.

## 2 PAROLE-SIMPLE-CLIPS: a computational Generative Lexicon

SIMPLE [6] is a large-scale project sponsored by the European Union devoted to the development of wide-coverage multipurposed and harmonised computational semantic lexica for twelve European languages (Catalan, Danish, Dutch, English, Finnish, French, German, Greek, Italian, Portuguese, Spanish and Swedish). A language-independent ontology of semantic types and a set of templates were designed and developed in order to guarantee uniformity and consistency among the monolingual dictionaries. In the framework of this project, 10,000 word meanings were annotated for each language.

SIMPLE should be considered as a follow up of a previous European project, PAROLE [10], as it adds a semantic layer to a subset of the morphological and syntactic layers that were developed by the latter. SIMPLE provides thus multi-layered lexica, as the information is encoded at different descriptive levels (morphological, syntactic and semantic). Although the information included for these levels is mutually independent, the layers are connected by one-to-one, one-

to-many or many-to-one links (e.g. a syntactic unit is linked with one or more semantic units depending on the number of meanings that the syntactic entry conveys).

CLIPS is an Italian national project which enlarged and refined the Italian PAROLE-SIMPLE lexicon [11]. The core data encoded within SIMPLE was extended in CLIPS with a new set of lexical units selected from the PAROLE corpus according to frequency-based criteria. The resulting lexical resource contains 387,267 phonetic units, 53,044 morphological units, 37,406 syntactic units and 28,346 semantic units.

From a theoretical point of view, the linguistic background of PAROLE-SIMPLE-CLIPS is based on the Generative Lexicon (GL) theory [9]. In the GL, the sense is viewed as a complex bundle of orthogonal dimensions that express the multidimensionality of word meaning. The most important component for representing the lexical semantics of a word sense is the qualia structure which consists of four qualia roles:

- Formal role. Makes it possible to identify an entity.
- Constitutive role. Expresses the constitution of an entity.
- Agentive role. Provides information about the origin of an entity.
- Telic role. Specifies the function of an entity.

Each qualia role can be considered as an independent element or dimension of the vocabulary for semantic description. The qualia structure enables to express different or orthogonal aspects of word sense whereas a one-dimensional inheritance can only capture standard hyperonymic relations. Within SIMPLE, the qualia structure was extended by assigning subtypes to each of the qualia roles (e.g. *Usedfor* is a subtype of the telic role).

The formal entities involved in the semantic description of PAROLE-SIMPLE-CLIPS which are significant for the purposes of the current research are semantic types, templates, qualia relations and features. The description of each of these elements follows.

The semantic types are the nodes that make up the ontology. A peculiar trait of the adopted ontology is the fact that it consists of both simple types, which identify only a one-dimensional aspect of meaning expressed by hyperonymic relations, and unified types, which express multidimensional aspects of meaning by combining subtyping relations and orthogonal semantic dimensions.

The ontology consists of 153 language-independent semantic types. The top types are mappable to the ontology of EuroWordNet [15]. The design of the ontology is highly influenced by the GL model. In fact, the top nodes are the semantic type *ENTITY* and three other types named after the agentive, constitutive and telic qualia roles (*AGENTIVE*, *CONSTITUTIVE* and *TELIC*). These three nodes are designed to include semantic units definable only in terms of qualia dimensions. The direct subtypes of the node *ENTITY* are the semantic types *CONCRETE\_ENTITY*, *PROPERTY*, *ABSTRACT\_ENTITY*, *REPRESENTATION* and *EVENT*.

**Table 1:** Relations encoded in the template for the semantic type *INSTRUMENT*

Relation	Qualia role	Constraint value
Isa	Formal	Yes
Hasaspart	Constitutive	RecNo
Madeof	Constitutive	RecNo
Usedfor	Telic	RecYes

Each semantic type is associated with one template. These act as *blueprints* for any given type in the ontology and provide the conditions of well-formedness and constraints for lexical items belonging to that type. The template structure is built like a schema that works as an interface between the lexicon and the ontology: it imposes conditions for the belonging of a given semantic unit to a semantic type. The template is then a help and a guide for the encoding of information referring to the ontology.

Relations and features are the elements that allow to assign properties to the semantic units. They can be applied as constraints within templates, in this case they act as type-defining (prototypical) for the semantic units included in these templates and can take one of the following values:

- Yes. The information is mandatory. I.e. every semantic unit that belongs to the semantic type should initialise this property.
- RecYes. The information is mandatory and the cardinality can be higher than one. I.e. a semantic unit can be linked to more than one element via this property.
- No. The information is optional.
- RecNo. The information is optional and the cardinality can be higher than one.

Table 1 provides an example on the encoding of relation constraints in templates. It presents the relations included in the definition of the template *INSTRUMENT* and for each of them the qualia type and the constraint value.

Features are used to characterise those attributes for which a closed range of values can be specified (e.g. edible = yes, sex = male, female). Features are useful to connect nodes across the ontology that share a given aspect and that otherwise would remain isolated. Relations, differently, link pairs of semantic units. There are different kinds of relations; there exist four types for the corresponding top roles of the extended qualia structure (formal, constitutive, agentive and telic), and others of non-qualia nature (e.g. synonymy). E.g. the semantic unit *bisturi* (*scalpel*) that belongs to the semantic type *INSTRUMENT* is linked to the semantic unit *incidere* (*engrave*) by the telic relation *Usedfor*.

### 3 Ontology modelling

This section describes our on-going research to formalise and enrich the ontology of PAROLE-SIMPLE-

**Table 2: Modelling cardinality restrictions**

Original Value	Cardinality restriction
Yes	min 1, max 1
RecYes	min 1
No	min 0, max 1
RecNo	min 0

CLIPS. Subsection 3.1 deals with the aspects regarding the formalisation in OWL while subsection 3.2 introduces the approach related to the semantic enrichment.

### 3.1 Formalisation

The elements of PAROLE-SIMPLE-CLIPS which have been considered to be modelled in OWL are those used to define the original ontology, i.e. the semantic types, the qualia relations and the features that apply to the templates to which the semantic types are associated. Further details on how each of these elements has been modelled follow.

Semantic types, as aforementioned, are the nodes that constitute the ontology. Therefore, they are modelled in OWL as classes. All sibling classes across the OWL ontology are made disjoint.

Relations are modelled as object properties. For qualia relations the domain and the range is made up of the classes *ENTITY* and the class that corresponds to the type of qualia relation (*AGENTIVE*, *CONSTITUTIVE* or *TELIC*). On the other hand, for non-qualia relations both the domain and the range are set to the top node of the ontology.

The application of relations to semantic types, as represented in the templates (see Table 1), is modelled with cardinality restrictions. To each value corresponds a different cardinality restriction, as shown in table 2.

The multidimensional nature of some semantic types is preserved in the OWL ontology by the inclusion of restrictions on qualia relations. The latter are in fact the elements that in PAROLE-SIMPLE-CLIPS allow to have multidimensional semantic types (also called unified types). If an ontology class contains a mandatory cardinality restriction on a qualia relation, then this class has as an additional defining dimension the corresponding qualia type. E.g. The class *INSTRUMENT* has a mandatory restriction on the constitutive qualia relation *Madeof*. Therefore, *INSTRUMENT* has as an additional defining dimension the constitutive one.

Figure 1 provides an example on the assignment of the relation constraints to the classes, the establishment of cardinalities and the role of inheritance (*INSTRUMENT* is a daughter of *ARTIFACT*). This figure is a snapshot of Protégé, a software that supports the edition of OWL ontologies [4], which shows the asserted conditions for the type *INSTRUMENT* in the formalised ontology.

Finally, features are modelled as *DataType* properties. Their domain is the union of the classes that share the feature (e.g. *FOOD*, *VEGETABLE*, etc. for the feature *PLUS\_EDIBLE*).

### 3.2 Bottom-up lexicon-driven enrichment

Besides formalising the ontology in OWL, we enrich it by following a bottom-up approach that extracts semantic information from the word senses of the lexicon by using the qualia structure as a generative device. This initial research enriches the ontology with constraints on relations and features extracted from the lexicon. On-going research will provide further enrichment of the ontology by extracting predicates and subclasses.

Each class (semantic type) of the ontology is enriched with additional constraints on relations and features which are extracted by exploring the word senses (semantic units) that belong to it. The procedure extracts all the relations and features that are defined for the word senses that belong to the semantic type. Afterwards, from these relations and features, it selects those that are considered to be representative of the class and proposes them to be modelled in the class definition as cardinality restrictions.

As the objective is to extract those relations and features that are relevant for the class definition, we consider discriminating by frequency of appearance, i.e. the percentage of word senses (semantic units) that belong to a class for which the given relation or feature is defined. As an initial experiment, we have established a threshold for each class to be the frequency of the least frequent relation/feature that is defined in the template of the class. Thus, those relations/features not defined in the template but whose frequency is higher than that of the threshold are proposed to be considered in the class definition.

The outlined procedure finds 218 relations and 229 features that are not considered in the class definitions but that, according to our hypothesis, could be included in the ontology as cardinality restrictions because they convey information that characterises the semantic units that belong to the semantic types. In order to make more comprehensible this matter, we examine some relations and features that are extracted to enrich the ontology.

Beginning with relations, let's consider the semantic type *INSTRUMENT*. The agentive relation *Createdby* is not included in the template definition but due to its high frequency of appearance is proposed to enrich the ontology by including it as a defining relation in this template. Clearly, an instrument is an artificial entity and therefore the relation *Createdby* applies and so is included in the node definition<sup>1</sup>.

Regarding features, we take *PLUS\_HUMAN*. This feature is not defined for any class of the ontology. However, it is applied to a high percentage of semantic units across several nodes of the ontology: *PROFESSION*, *HUMAN\_GROUP*, *HUMAN*, *PEOPLE*, etc. It is clear that any semantic unit that belongs to any of these semantic types would be of a human nature. Therefore, this feature is promoted to be type-defining in the aforementioned classes.

Another feature that could serve as an example is *PLUS\_EDIBLE*. This feature is included in the tem-

<sup>1</sup> As it can be seen in figure 1 this relation is already present in the formalised ontology as it is inherited from the class *ARTIFACT*.

Asserted Conditions	
	NECESSARY & SUFFICIENT
	NECESSARY
p1:Artifact	<input type="checkbox"/>
p1:hasHasaspart min 0	<input type="checkbox"/>
p1:hasMadeof min 0	<input type="checkbox"/>
	INHERITED
p1:hasCreatedBy min 1	[from p1:Artifact] <input type="checkbox"/>
p1:hasSynonym min 0	[from p1:Entity] <input type="checkbox"/>
p1:hasUsedfor min 1	[from p1:Artifact] <input type="checkbox"/>

Fig. 1: Asserted Conditions for the class INSTRUMENT

plate of several semantic types such as *FOOD* or *VEGETABLE*. However, although having a high frequency of appearance in the type *SUBSTANCE\_FOOD*, it is not included in the definition of this class. The bottom-up procedure incorporates this feature as a type-defining element for this node.

## 4 Application to the biological domain

This section introduces the application of the presented procedures to the BioLexicon, a lexicon for the biological domain designed in the framework of the BOOTStrep project<sup>2</sup> which is inspired by the Generative theory and to a wide extent builds on top of the structures introduced by the PSC model [8]. This lexicon, together with the BioOntology, constitute a terminological backbone by combining lexical and ontological information thus becoming an innovative integrated resource suitable for NLP tasks in the bio domain.

The BioLexicon semantic relations build on the 60 Extended Qualia Relations of the SIMPLE model. The Extended Qualia Relations allow modelling different meaning dimensions of a word sense and specifying its relations to other lexical units (either paradigmatic or syntagmatic). Most of these relations, also shared by well-known ontologies of the biological domain, prove to be suitable for the domain of interest and therefore are imported into the BioLexicon model. Clearly, there are relations not considered in the Qualia Structure that however are relevant for this domain. We have studied the Open Biomedical Ontologies (OBO) Relations, an ontology of core relations for the biomedical domain, in order to find relevant relations not present in the Qualia Structure. Each of these has been added to the BioLexicon model, some of them as new relations whereas some others as subtypes of existing qualia relations.

The BioLexicon and BioOntology have been separately designed and constructed. The BioLexicon has been automatically populated with terms gathered from available bio terminologies and augmented with linguistic information about terms extracted from

texts [2]; the BioOntology has been built integrating different ontological resources of the domain. This is why we hypothesise that the procedures introduced in this paper might be useful in this case in order to synchronise the information present in lexicon and in the ontology:

- From the data present in the lexicon, we can generalise constraints from instantiated relations and check whether or not they have been included in the ontology definition. In other words, the procedures can be useful in order to find definition gaps in the ontology as its design has been done separately of the lexicon population.
- On the other hand, the procedures can easily be used to guarantee that the data encoded in the lexicon is consistent with the constraints that are present in the BioOntology.

## 5 Conclusions

This paper has presented a proposal to formalise and enrich the ontology of a GL resource. The approach followed has proved to success to formalise the GL ontology in the standard OWL format. Moreover, we have applied a bottom-up procedure in order to enrich the converted ontology with further semantic information obtained from the lexicon.

The formalisation allows the ontology to be processed and checked by standard reasoners. This can be useful for building semantic applications as well as to enhance the quality of the resource by validating it (through reasoning we can look for inconsistencies or conflicts).

Besides, the paper has studied the feasibility of the procedures to be applied to a GL-based domain specific resource. Also in this case, the ontology can be enriched with additional semantic information and the resource can be checked and thus consistency be guaranteed.

The possible uses of the resulting formalised and enriched ontology are twofold. First, as it is an OWL ontology it could be used in Semantic Web applications. Second, as it is a semantically rich resource, it could be applied to semantic NLP tasks. In fact, we

<sup>2</sup> www.bootstrep.eu

plan to use it for semantic Information Extraction and Knowledge Acquisition.

As for future work, some aspects regarding the modelling are to be considered. On one hand, we plan to research on enriching the ontology with semantic predicates, an additional kind of semantic information which is encoded in the lexicon and which could play an important role when using the ontology for NLP purposes. Once this is done, we will investigate regarding the atomisation of the formalisation and the enrichment with all the considered information.

## Acknowledgements

This research is part of an European Ph.D. program. It has been partially funded by a research grant of the ILC-CNR and by the ECs 6th Framework Programme (4th call), conducted within the BOOTStrep consortium under grant FP6-028099. We also would like to thank Riccardo del Gratta for his ideas and valuable comments on the application of the procedures presented to the BioLexicon.

## References

- [1] M. Dean and G. Schreiber. OWL web ontology language reference. W3C recommendation, W3C, February 2004.
- [2] R. del Gratta, V. Quochi, E. Sassolini, M. Monachini, and N. Calzolari. Toward a Standard Lexical Resource in the Bio Domain (to appear). In *3rd Language and Technology Conference*, Poznan, Poland, October 2007.
- [3] C. Fellbaum. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May 1998.
- [4] H. Knublauch, R. W. Ferguson, N. F. Noy, and M. A. Musen. The protege owl plugin: An open development environment for semantic web applications. In *Proceedings of the Third International Semantic Web Conference*, 2004.
- [5] D. Lenat. *From 2001 to 2001: Common sense and the mind of HAL*, pages 193–208. MIT Press, Cambridge, MA, 1998.
- [6] A. Lenci, N. Bel, F. Busa, N. Calzolari, E. Gola, M. Monachini, A. Ogonowski, I. Peters, W. Peters, N. Ruimy, M. Villegas, and A. Zampolli. Simple: A general framework for the development of multilingual lexicons. *International Journal of Lexicography*, 13(4):249–263, 2000.
- [7] D. L. McGuinness and F. van Harmelen. Owl web ontology language overview, February 2004.
- [8] M. Monachini, V. Quochi, N. Ruimy, and N. Calzolari. Lexical Relations and Domain Knowledge: The Bio-Lexicon Meets the Qualia Structure. In *Fourth International Workshop on Generative Approaches to the Lexicon*, Paris, France, May 2007.
- [9] J. Pustejovsky. The generative lexicon. *Computational Linguistics*, 17(4):409–441, December 1991.
- [10] N. Ruimy, O. Corazzari, E. Gola, A. Spanu, N. Calzolari, and A. Zampolli. The european le-parole project: The italian syntactic lexicon. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC'98)*, Granada, Spain, 1998.
- [11] N. Ruimy, M. Monachini, R. Distanti, E. Guazzini, S. Molino, M. Ulivieri, N. Calzolari, and A. Zampolli. Clips, a multi-level italian computational lexicon: A glimpse to data. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas de Gran Canaria, Spain, 2002.
- [12] L. F. Soualmia, C. Golbreich, and S. J. Darmoni. Representing the mesh in owl: Towards a semi-automatic migration. In U. Hahn, editor, *KR-MED*, volume 102 of *CEUR Workshop Proceedings*, pages 81–87. CEUR-WS.org, 2004.
- [13] M. van Assem, A. Gangemi, and G. Schreiber. Conversion of WordNet to a standard RDF/OWL representation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006.
- [14] M. van Assem, M. R. Menken, G. Schreiber, J. Wielemaker, and B. Wielinga. A method for converting thesauri to rdf/owl. In *Proceedings of the Third International Semantic Web Conference (ISWC'04)*, number 3298 in Lecture Notes in Computer Science, pages 17–31, Hiroshima, Japan, November 2004.
- [15] P. Vossen. Introduction to EuroWordNet. *Computers and the Humanities*, 32:73–89, 1998.



# Towards a Named Entity WordNet (NEWN)

Antonio Toral  
Istituto di Linguistica Computazionale  
Consiglio Nazionale delle Ricerche  
Via G. Moruzzi 1 - 56124 Pisa, Italy  
*antonio.toral@ilc.cnr.it*

Rafael Muñoz  
Natural Language Processing and Information Systems Group  
Department of Software and Computing Systems  
University of Alicante, Spain  
*rafael@dlsi.ua.es*

## Abstract

This paper presents a method to automatically add named entities to the noun taxonomy of WordNet which are extracted from Wikipedia and the building of a new resource called Named Entity WordNet (NEWN) which adds named entities to the original WordNet. The paper motivates and demonstrates that knowledge acquisition can benefit from exploiting wiki texts due to some characteristics of these resources that provide advantages over other commonly used resources such as corpora or Machine Readable Dictionaries. In fact, a simple extraction approach, which is described, is able to enrich the English WordNet with more than 55,000 new instances (i.e. more than seven times the amount of instances that WordNet contains) with a precision over 93% thus providing a valuable resource for Natural Language Processing tasks and especially for Named Entity Recognition.

## Keywords

Named Entities, Proper Nouns, Knowledge Acquisition, WordNet, Wikipedia

## 1 Introduction

World knowledge is a requirement to deal with the semantic level of natural languages. Conceptualisations of reality occupy human beings from the early Greece, where the term Ontology (from the Greek *ὄν*, genitive *ὄντος*: *of being* (part. of *εἶναι*: *to be*) and *-λογία*: *science, study, theory*) was introduced by Aristotle [2]. Long time later, at the end of the XX century, the first attempts to give *common sense* to computers by building Knowledge Bases (KBs) were initiated. Examples of this are the CYC project [9], MindNet [14] and WordNet [11].

There have been manual approaches to build KBs, in which linguist experts manually build these resources. There are as well automatic proposals which build KBs from information which is extracted from unstructured textual sources such as unannotated corpora and also from structured ones such as Machine

Readable Dictionaries (MRDs). Both present disadvantages due to inherent characteristics of these resources.

However new types of text have emerged as a consequence of the appearance of new forms of communication [8]. One of these new kinds of text is known as wiki. Wikis can be defined as on-line texts that allow users to easily edit and change the contents. These characteristics make them an effective tool for collaborative authoring. The most widely known example of a wiki resource is Wikipedia, a multilingual encyclopedia that follows the wiki philosophy. Wikipedia could be an interesting textual source for the automatic creation of KBs because, being an encyclopedia, it contains facts dealing with the entire range of human knowledge and, because it is developed by a large amount of people<sup>1</sup>, and therefore reflects the variations of language and human thought.

WordNet is an on-line lexical database that contains nouns, verbs, adjectives and adverbs organised into sets of synonyms - called synsets- and containing several types of semantic relations among its nodes [11]. It is manually tagged by a team of linguists and its design is inspired by psycholinguistic theories of human lexical memory. This resource is widely used within the Natural Language Processing (NLP) community. In fact, it has become the *de facto* standard for several NLP tasks such as Word Sense Disambiguation.

Regarding nouns, from version 2.1., WordNet distinguishes between common nouns (classes) and proper nouns (instances) [12]. While WordNet's coverage about open domain common nouns is quite high, it contains very few proper nouns<sup>2</sup>. This is related with the following asseveration: "building a proper noun ontology is more difficult than building a common noun ontology as the set of proper nouns grows more rapidly" [10]. The problem is then that a proper noun resource should be constantly updated. Following with this matter, [13] states that "the need for machine-assisted ontology construction is stronger than ever" because "humans cannot manually structure the avail-

<sup>1</sup> On 2007/01/16 the English version has 3,247,299 registered users.

<sup>2</sup> 7,669 synsets are tagged as being instances in WordNet 2.1.

able knowledge at the same pace as it becomes available". The so called *knowledge acquisition bottleneck* is a recognised issue within the NLP community, and a lot of research effort is devoted nowadays to solve it.

Proper noun ontologies could be very useful for NLP. [10] shows how their use, even if the ontology used has a low coverage, improves the precision of a Question Answering system. Moreover, this kind of resources could play a crucial role in Named Entity Recognition systems that consider an extended hierarchy of entity types like that proposed in [16].

This paper presents a proposal to automatically extend WordNet with noun instances which are automatically extracted from Wikipedia. The rest of the paper is organised as follows. Next section summarises related work. This is followed by the presentation of our method and some discussion on the obtained results. Finally, we derive conclusions and outline future work proposals.

## 2 Related Work

This section presents related research work devoted to the automatic acquisition of lexical and semantics in order to incorporate it into structured knowledge resources and especially to the creation of structured resources that include proper nouns.

Referring to MRDs, Rigau [15] presents a detailed proposal to the massive acquisition of lexical knowledge from monolingual and bilingual MRDs. Apart from designing a productive methodology to build and validate a multilingual KB, a software system (called SEISD) is implemented. Previous research includes [3], [7] and [1].

[5] criticises the utilisation of MRDs in knowledge acquisition because of their fixed size and proposes to extract semantic knowledge from corpora by using lexical patterns. Six patterns are proposed together with a methodology to find new ones. Next, research based on using patterns for extracting semantic information related to proper nouns is presented.

[10] creates a proper noun ontology from newswire text. The proposal consists of extracting phrases from a 1 gigabyte corpus by applying a Part-of-Speech pattern (a common noun followed by a proper noun). This allows the author to gather 113,000 different proper nouns and to reach a precision of 60% (84% for proper nouns referring to people and 47% for the rest). The author points out also that the employed methodology is problematic with polysemous words and that it is not straight-forward to integrate the proper noun ontology created with the WordNet taxonomy of nouns.

[4] extracts concept-instance relations from 15 gigabytes of newspaper text by using two Part-of-Speech patterns (common nouns followed by a proper noun and appositions). Machine Learning techniques are applied to increase the precision of the extracted info. 500,000 unique instances (Bill Clinton and William Clinton are considered as two different instances) are extracted. A evaluation over 100 concept-instance items is carried out, achieving a precision of 93%.

## 3 Method

This section presents our proposal to extract instances from Wikipedia in order to integrate them into the WordNet noun taxonomy. First, our approach is compared to other proposals within this area. Afterwards our method is introduced and explained.

### 3.1 Comparison with other approaches

Several aspects make this research different from previous work within lexical and semantic knowledge acquisition. Compared to research that relies on corpora, our research avoids problems due to subjective judgements and inconsistencies due to calling instances in different manners whereas compared to research that uses MRDs, our method is not limited by the small size of the input resource.

According to [6], relations found in unrestricted text tend to be subjective judgements compared to the more established statements present in dictionaries and encyclopedias. Therefore, unless some post-process is carried out, methods that extract semantic relations from these kind of textual sources are not appropriate for an automatically acquisition process. Our method however, as relies on an encyclopedia, which moreover has strong policies regarding neutrality<sup>3</sup>, does not suffer from such problems.

Following with corpora based methods, they might, if no special treatment is applied, acquire the same instance with different lexical forms [4] (e.g. Bill Clinton and William Clinton) and therefore include them as different instances in the created resource. However, in Wikipedia, if an instance being an entry has different lexical ways of referring to it, all of them are linked so it is straight-forward to extract all of them as being different strings that refer to the same entity (e.g. William Clinton links to Bill Clinton).

It is also important to mention that our proposal manages to integrate the extracted proper nouns into a widely used structured knowledge resource (WordNet) whereas other proposals to acquire proper nouns do only provide concept-instance pairs without any relation among the different pairs [4] or do integrate the instances extracted into an own ontology but however state that the integration of this ontology with WordNet would require further study [10].

Regarding MRDs based research, [5] claims that although projects that exploit MRDs have been successful, they are limited as the amount of entries of dictionaries is fixed. Those projects were productive because the structured nature of MRDs makes it easier to extract valuable knowledge than to do it from plain text. Our method exploits a resource which has a structured nature, like MRDs, but it is not small<sup>4</sup> and its size is not fixed as it is continuously updated and growing<sup>5</sup>. Therefore, our approach overcomes the MRD's size limitation.

<sup>3</sup> See [http://en.wikipedia.org/wiki/Wikipedia:Neutral\\_point\\_of\\_view](http://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view)

<sup>4</sup> The database dump of Wikipedia used contains 1,496,097 encyclopedic entries.

<sup>5</sup> See <http://stats.wikimedia.org/EN/ChartsWikipediaEN.htm>

## 3.2 Method description

In a nutshell, the proposed method proceeds like follows. WordNet's synsets are linked to Wikipedia's categories and the instances present (as entries) in those categories of Wikipedia are incorporated into the linked synsets of WordNet as instance hyponyms of those synsets. Our method is carried out in two phases. The first one establishes a mapping that links WordNet's synsets to Wikipedia's categories whereas the second one gathers articles in Wikipedia from the categories mapped integrating those being instances into WordNet. A graphic depicting the method is presented in figure 1.

The mapping of WordNet's synsets to Wikipedia's categories is carried out only for the synsets in which at least one word<sup>6</sup> is monosemous. For this first attempt to extend WordNet with Wikipedia, we preferred to avoid the problem of disambiguating word senses as it is not, at this initial step of the research, the problem in which we want to focus.

For each synset in the noun taxonomy of WordNet, we consider those which are not instances but classes<sup>7</sup>. The synset is then associated to the related categories found in Wikipedia. A synset is linked to a category if the lemma of any of the words of the synset is lexically identical to the lemma of the category.

The second step takes as input this mapping. For each category mapped, the method extracts all the articles which are contained in the category. Those articles already present in WordNet, those which are disambiguation articles and those which are "List\_of" articles are discarded.

Afterwards, a web search based method to discard those articles not being instances is carried out. Each article's title is searched in the Web by using Google. The first 50 results where the title is found are returned and an algorithm calculates the number of times the article's title appears (i) with all the words beginning by capital letters, (ii) with some words beginning by capital letters and (iii) with no word beginning by capital letters in the websites description. Besides, a threshold between 0 and 1 is established in order to discard between articles being instances and non-instances according to the different models of capitalisation.

Finally those articles being instances are incorporated into WordNet as new synsets. Each of these new synsets has as name the main name of the article in Wikipedia plus the redirects. The gloss is the abstract of the article in Wikipedia<sup>8</sup>. Finally, each new synset is linked as an instance hyponym of the synset mapped to the Wikipedia's category to which the article belongs.

It follows an example of the method:

- WordNet's synset: screenwriter, film\_writer
- Wikipedia's linked category: screenwriters

<sup>6</sup> This refers to the word field in Wordnet's noun data file, see `wndb.5WN` in WordNet's documentation for details.

<sup>7</sup> The method incorporates to WordNet's synsets extracted instances as hyponyms, thus it would be nonsense to consider those synsets being instances because an instance, by definition, cannot have hyponyms [12].

<sup>8</sup> The Wikimedia Foundation provides a database dump containing abstracts of the articles present in Wikipedia.

- Wikipedia's extracted article:

- title: Tim\_Robbins
- abstract: "Timothy Francis Robbins (born October 16, 1958) is an American Academy Award-winning actor, screenwriter, director, producer, and small time musician. He is the longtime companion of actress Susan Sarandon, with whom he shares strong liberal political views."

- Web-Search: 96,67% of the times "Tim Robbins" appears in the description field of the first 50 results returned by google, all the words of the string begin by capital letters.

- WordNet's new synset (WordNet's lexicographic syntax): { Tim\_Robbins, noun.person:screenwriter,@i (Timothy Francis Robbins (born October 16, 1958) is an American Academy Award-winning actor, screenwriter, director, producer, and small time musician. He is the longtime companion of actress Susan Sarandon, with whom he shares strong liberal political views) }

## 4 Experimental results and discussion

In order to carry out the experiments which will be outlined later on in this section, we have used the noun taxonomy of WordNet 2.1. (`index.noun` and `data.noun` files) and a database dump of the English version of Wikipedia (`enwiki-20061104`)<sup>9</sup>. From this dump, we have used the page, pagelinks, categorylinks and abstract data.

There are 81,426 noun synsets in the version of WordNet used. Out of these, 73,757 refer to classes and the remaining 7,669 refer to instances. From the 73,757 synsets that refer to classes, 18,783 (25,47%) contain only polysemous words and 54,974 (74,53%) do contain at least one monosemous word and therefore are considered.

The mapping process links 10,125 synsets out of the 54,974 considered (18,41%). From the categories linked, 231,354 articles are extracted. Out of these, 29,554 are discarded because they are already present in WordNet, 55,573 because they are disambiguation entries and 2,512 because they are "List\_of" entries. Therefore, 143,715 articles do remain.

Regarding the web-search method employed to discard non-instances, two sets of randomly selected entries were manually tagged as being instances or classes. One set contains 200 entries and was used as training set whereas the other is made of 100 entries and is used as evaluation set. The training set was used to select the threshold and capitalisation model that obtain the best results. The threshold selected is 0.91 while the capitalisation model is the one that considers the number of times that the first word of the string begins by capital letters. Subsequently, the algorithm was evaluated over the evaluation set. Table 1 shows the results obtained for this set

<sup>9</sup> Available at <http://download.wikimedia.org>

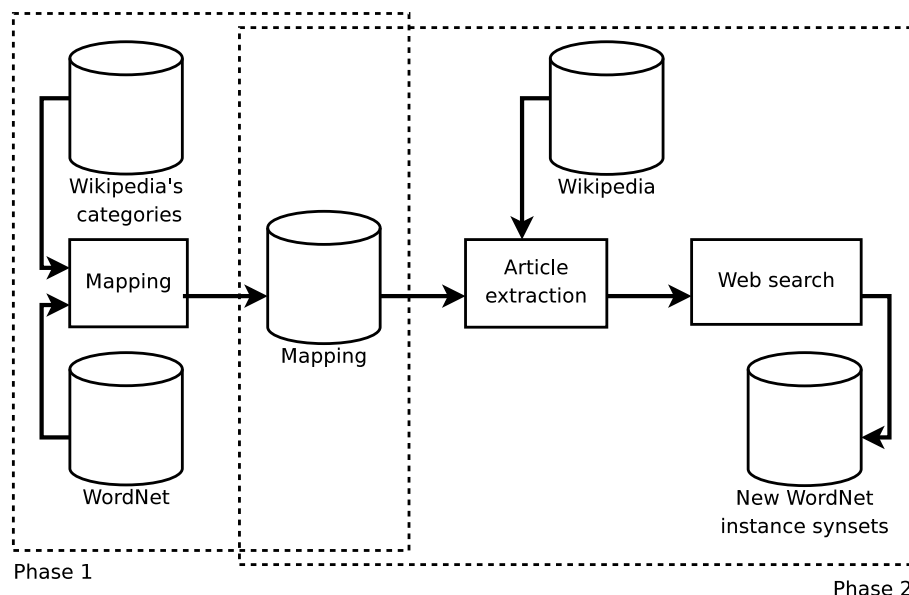


Figure 1: Method diagram

Table 1: Evaluation results of the web-search method

Threshold	Precision	Recall	$F_{\beta=1}$	$F_{\beta=0.5}$
0.81	87.75	74.14	80.37	82.69
0.83	89.13	70.69	78.84	82.00
0.85	91.11	70.69	79.61	83.11
0.87	90.90	68.97	78.43	82.19
0.89	90.90	68.97	78.43	82.19
0.91	93.02	68.97	79.20	83.33
0.93	92.86	67.24	78.00	82.39
0.95	94.29	56.89	70.97	77.34

regarding instances. For several values of the threshold, precision, recall and F-measure ( $\beta=1$  and  $\beta=0.5$ ) are included.  $F_{\beta=1}$  weights evenly precision and recall whereas  $F_{\beta=0.5}$  weights precision twice as much as recall.

It can be seen that the highest F-measure ( $\beta=0.5$ ) is obtained when the threshold is set to 0.91, reaching 83.33% and precision 93.02%. Although other values of the threshold provide higher values of F-measure ( $\beta=1$ ), as the aim of the approach is to extend a knowledge resource, we consider more important precision than recall as we think that it is better to add a lower number of synsets to WordNet but making sure that the quality of the final resource is good enough. With the configuration selected, 41,58% of the entries are tagged as instances, which means that WordNet is extended with more than 55,000 new instance synsets. This is, the number of instance synsets in WordNet is multiplied at least seven times.

The percentage of synsets that get mapped to Wikipedia's categories is quite low. We consider two possible reasons that can cause this: (i) Wikipedia's coverage on some parts of WordNet's noun taxonomy might be low, (ii) lexical pattern matching may not be an appropriate approach to find related cat-

egories in Wikipedia to WordNet synsets. Although the method to classify the extracted articles in instances and classes is fairly simple, the results obtained are reasonably high (precision 93,02%, recall 68,97%,  $F_{\beta=0.5}$  83,33%). In our opinion, this is due to the high quality of the data extracted. Therefore, even better results could be achieved by applying more elaborated approaches.

## 5 Conclusions

This paper has presented, to our knowledge, the first attempt to semantically exploit Wikipedia in order to extend WordNet with new synsets. We have introduced the motivation for our proposal, the method has been explained and results have been discussed. The practical result of this research consists of a new specific resource for Named Entity Recognition which will be available to the research community.

We have demonstrated that lexical and semantic knowledge acquisition could benefit from exploiting new text types such as wikis by showing the potential advantages over common approaches that rely on unrestricted corpora and MRDs. In fact, by using very simple techniques we have been able to multiply by more than seven times the amount of instances present in WordNet with a precision over 93%. Therefore, we think that there is a big room for research within this innovative approach which could lead to important advances in the automatic creation of knowledge bases and in the automatic extension of already existent lexico-semantic resources.

This research has produced a valuable resource for Named Entity Recognition tasks. In fact, the resource contains more than 55,000 named entities classified in a widely used noun taxonomy. Therefore, this could be exploited by systems that attempt to classify named entities across a high number of categories. Also, as we provide a classification of entities in nodes of a tax-

onomy instead of isolated lists of entities for each category, the resource can be used with different levels of granularity for entity recognition.

Another important feature of the proposed approach is its high degree of language independence. This method can be directly applied to any language if there is a version of Wikipedia, a version of WordNet and a lemmatiser<sup>10</sup>. Moreover, the multilingual knowledge encoded in Wikipedia could be exploited in order to build and extend multilingual Knowledge Resources.

As future work we plan to improve the automatic extraction of semantic knowledge from wiki texts by following three directions:

- Carry out a deep study about the mapping process in order to determine the reason why the percentage of synsets linked is low. Once we obtain a better understanding on this, we could apply the method proposed in this paper to extend with instances a broader part of WordNet.
- Analyse most advanced techniques to distinguish between instances and classes such as Machine-Learning based binary classifiers in order to increase the precision and the recall.
- Examine Word Sense Disambiguation techniques in order to be able to use our approach to extend synsets made up of polysemous words in WordNet and also to incorporate into WordNet those articles from Wikipedia that were discarded in the present research because they were polysemous (disambiguation entries).

## Acknowledgements

This research has been partially supported by the Spanish Government under project TEXT-MESS (TIN-2006-15265-C06-01).

## References

- [1] Hiyan Alshawi. Processing dictionary definitions with phrasal pattern hierarchies. *Computational Linguistics*, 13(3-4):195–202, 1987.
- [2] Aristotle. *Metaphysics*. In W. D. Ross, editor, *The Works of Aristotle translated into English, Volume VIII*. Oxford University Press, Oxford, 1908.
- [3] Nicoletta Calzolari. Acquiring and representing semantic information in a lexical knowledge base. In *Proceedings of the First SIGLEX Workshop on Lexical Semantics and Knowledge Representation*, pages 235–243, London, UK, 1992. Springer-Verlag.
- [4] M. Fleischman, A. Echihabi, and Eduard Hovy. Offline strategies for online question answering: Answering questions before they are asked. In *Proceedings of the ACL Conference. Sapporo, Japan*, 2003.

- [5] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING*, pages 539–545, 1992.
- [6] Marti A. Hearst. *Automated Discovery of Word-Net Relations*. MIT Press, Cambridge, MA, 1998.
- [7] Jun ichi Nakamura and Makoto Nagao. Extraction of semantic information from an ordinary english dictionary and its evaluation. *COLING-88*, pages 459–464, 1988.
- [8] J. Karlgren, editor. *NEW TEXT, Wikis and blogs and other dynamic text sources*, Trento, Italy, 2006.
- [9] D.B. Lenat. *From 2001 to 2001: Common sense and the mind of HAL*, pages 193–208. MIT Press, Cambridge, MA, 1998.
- [10] G. Mann. Fine-grained proper noun ontologies for question answering, 2002.
- [11] G. A. Miller. Wordnet: A lexical database for english. *Communications of ACM*, (11):39–41, 1995.
- [12] G. A. Miller and F. Hristea. Wordnet nouns: Classes and instances. *Computational Linguistics*, 32(1):1–3, 2006.
- [13] Andrew Philpot, Eduard Hovy, and Patrick Pantel. The omega ontology. In *IJCNLP Workshop on Ontologies and Lexical Resources (OntoLex-05)*, pages 59–66, Jeju Island, South Korea, 2005.
- [14] Stephen D. Richardson, William B. Dolan, and Lucy Vanderwende. Mindnet: Acquiring and structuring semantic information from text. In *COLING-ACL*, pages 1098–1102, 1998.
- [15] G. Rigau. *Automatic Acquisition of Lexical Knowledge from MRDs*. PhD thesis, Universitat Politècnica de Catalunya, 1998.
- [16] Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. Extended Named Entity Hierarchy. In *Proceedings of Third International Conference on Language Resources and Evaluation*, 2002.

<sup>10</sup> However, the web-search step as described here would not work for languages that follow different models of capitalisation (e.g. German, Arabic, Japanese)

# Specifying Properties of a Language with Regular Expressions

François Trouilleux  
Laboratoire de recherche sur le langage  
Université Blaise-Pascal  
29, boulevard Gergovia  
63037 Clermont-Ferrand cedex  
*trouilleux@lrl.univ-bpclermont.fr*

## Abstract

This article presents a translation of the Property formalism of [2] into the XFST regular expression formalism [6]. Besides offering at no cost a platform to use Properties in natural language processing, this operation allows us to clarify the interpretation of the Property formalism, which may be interpreted as strictly limited either to regular languages or to context-free languages, depending on the definition of the objects Properties apply to.

## Keywords

Properties, Regular expressions, Phrase structure grammars, Constraints, Xerox Finite-State Tool (XFST)

## 1 Introduction

In 1999, Gabriel G. Bès proposed a new formalism for the description of natural language syntax, called “Properties” [2], of which Blache afterwards proposed a variant, called “Property Grammar” [3]. The purpose of this paper is to offer a new point of view on this formalism, through its translation into a regular expression formalism, that of the Xerox Finite-State Tool [6]. This translation, developed in section 2, supports both theoretical and practical considerations: section 3 points out the specificity of the property formalism with respect to classical phrase structure grammars, and section 4 clarifies its interpretation in terms of regular or context-free language specification. Practical use of our translation scheme and relevance of the Property formalism are then discussed in section 5.

## 2 Properties as regular expressions

This section introduces a translation of Properties into regular expressions (2.1), followed by a discussion of some limitations of this translation (2.2).

### 2.1 A translation scheme

Properties are formulas of the form `pred(id, ...)`, where `pred` is a predicate corresponding to the name of the Property, `id` is the name of the language (or

category) the Property applies to, and `...` marks the place of one or several other arguments of the Property predicate `pred` (cf. [2]). These arguments are always symbols which refer to a category.

[2] specifies nine types of properties. Figure 1 presents the definition of six of them: *amod*, *uniq*, *oblig*, *exig*, *exclu* and *precede*, together with their translation into XFST regular expressions (below the dotted line)<sup>1</sup>. Limitations of our translation scheme, including the non-translation of some Properties, are discussed in the next section.

Full definition of the operators used in the regular expressions may be found in [1], as well as on the XRCE web site<sup>2</sup>. In short, `|` denotes union, `&` intersection, `<` precedence, `+` iteration (Kleene plus), `$` containment, `$$?` containment of at most one, and `~` complement.

In figure 1, we use letters as arguments of the Property predicates. These letters are to be interpreted as category names, *i.e.* as denoting regular languages. However, to keep the description short, we use expressions such as “an *a*” to mean “a string of category *a*” or “a string from language *a*”.

All Properties are presented as applying to a language called `id`. On the regular expression side, this language is defined as the result of the intersection of all the languages denoted by the Properties. Considering the Properties defined in figure 1, one would then have to write as a final definition:

```
define id [AMOD & UNIQ & OBLIG
          & EXIG & EXCLU & PRECEDE];
```

### 2.2 Limitations

In our translation scheme, we made a number of simplifications on the original definitions of [2], which we briefly justify here.

For space reasons, we do not present the translation of some parts of the formalism, *i.e.* variants for the *oblig*, *exig*, *exclu* and *precede* Property types, the *exigac* Property type, through which one will specify agreement constraints, and the fact that one category

<sup>1</sup> In XFST, the schema of a `define` command is `define variable regular-expression ;` the effect is to “invoke the compiler on the regular expression, create a network, and assign that network to the indicated variable. Once defined in this way, the variable [...] can be used in subsequent regular expressions.” [1, p. 85]

<sup>2</sup> [www.xrce.xerox.com/competencies/content-analysis/fsCompiler/home.en.html](http://www.xrce.xerox.com/competencies/content-analysis/fsCompiler/home.en.html)

<p><code>amod(id, [a, b, ..., z])</code> specifies that in a string of language <code>id</code>, one may only use words of category <code>a</code>, <code>b</code>, ..., or <code>z</code>.</p> <pre>..... define AMOD [ a   b   ...   z ]+ ;</pre>
<p><code>uniq(id, [a, b, ..., z])</code> specifies that a string of language <code>id</code> may contain at most one <code>a</code>, at most one <code>b</code>, ..., and at most one <code>z</code>.</p> <pre>..... define UNIQ [ \$?a &amp; \$?b &amp; ... &amp; \$?z ] ;</pre>
<p><code>oblig(id, [a, b, ..., z])</code> specifies that a string of language <code>id</code> must contain one <code>a</code>, or one <code>b</code>, ..., or one <code>z</code>.</p> <pre>..... define OBLIG [ \$a   \$b   ...   \$z ] ;</pre>
<p><code>exig(id, [a, b, c, ..., z])</code> specifies that in a string of language <code>id</code>, the presence of an <code>a</code> requires the presence of a <code>b</code>, or a <code>c</code>, ..., or a <code>z</code>.</p> <pre>..... define EXIG [ ~\$a   [ \$a &amp; \$[b   c   ...   z] ] ] ;</pre>
<p><code>exclu(id, [a, b, c, ..., z])</code> specifies that in a string of language <code>id</code>, the presence of an <code>a</code> forbids the presence of a <code>b</code>, of a <code>c</code>, ..., and of a <code>z</code>.</p> <pre>..... define EXCLU [ ~\$a   [ \$a &amp; ~\$[b   c   ...   z] ] ] ;</pre>
<p><code>precede(id, [a, [b, c, ..., z]])</code> specifies that, in a string of language <code>id</code>, if an <code>a</code> occurs with a <code>b</code>, or a <code>c</code>, ..., or a <code>z</code>, it must precede the <code>b</code>, <code>c</code>, ..., or <code>z</code>.</p> <pre>..... define PRECEDE [ a &lt; [b   c   ...   z] ] ;</pre>

Fig. 1: Translation table from Properties to XFST regular expressions.

may be characterized as the *nucleus* of the defined strings. The absence of a translation for these features does not affect the discussion developed below.

More interesting is the fact that two characteristics of the formalism *had* to be set aside, because they were impossible to express by regular expressions:

1. The original definition of the *amod* Property also specifies that for each category, there exists at least one string which contains a word from that category (*i.e.* all the categories are used at least once). This is a condition on the whole set of strings, not on the strings themselves, and cannot be expressed by regular expressions.
2. The original formalism includes a *fleche* Property type, which specifies relations between the words composing a valid string and cannot be translated by regular expressions. However, it must be noted it has a special status, compared to other Properties, as it allows the expression of statements over the strings defined by the other Properties, but not to modify this set of strings by addition or subtraction.

## 2.3 An example

The regular expressions in figure 2 illustrate the definition of a language with Properties written as XFST regular expressions. The first nine lines define nine categories. The word forms considered in that example appear between quotes: *is*, *do*, *does*, *sing*, *sings*,

*singing*. The words appearing next to a `define` command (e.g. `BE`) are category names. The language defined is `VC` (for “verb chunks”); it contains eight strings:  $\{do\ sing, do\ not\ sing, does\ sing, does\ not\ sing, is\ singing, is\ not\ singing, sing, sings\}$ <sup>3</sup>.

## 3 Specificity of the Property formalism

Compared with more classical approaches, Properties offer a different perspective. We here compare this formalism with classical phrase structure grammars and with Koskenniemi’s Finite-State Intersection Grammar (FSIG) [8].

### 3.1 Properties vs. phrase grammars

Like a regular expression or any phrase structure grammar, a set of Properties may be viewed as specifying a language. The novelty, when one compares Properties to classical phrase structure grammars, is that Properties systematically make use of *intersection* (as shown by the definitions of `id` at the end of section 2.1, and of `VC` in figure 2), and do not explicitly use *concatenation* (as shown by the absence of an explicit concatenation operator in the regular expressions of figure 1).

<sup>3</sup> With whitespaces added between the terminal symbols for readability.



```

define BE [ "is" ];
define DO [ "do" | "does" ];
define Aux [ BE | DO ];
define VInf [ "sing" ];
define V3Pr [ "sings" ];
define VIng [ "singing" ];
define VBase [ VInf | V3Pr ];
define V [ VBase | VIng ];
define Neg [ "not" ];

define AMOD [V | Aux | Neg]+;
define OBL [$V];
define UNIQ [$?V & $?Aux & $?Neg];
define RE1 [~$Neg | [$Neg & $Aux]];
define RE2 [~$VIng | [$VIng & $BE]];
define RE3 [~$DO | [$DO & $VInf]];
define EX1 [~$V3Pr | [$V3Pr & ~$DO]];
define EX2 [~$VBase | [$VBase & ~$BE]];
define PR1 [Aux < [Neg | V]];
define PR2 [Neg < V];

define VC [AMOD & OBL & UNIQ & RE1 & RE2
          & RE3 & EX1 & EX2 & PR1 & PR2];

```

**Fig. 2:** Definition of a small example language.

In contrast, phrase structure grammars favour *union* and *concatenation*. Typically, in a phrase structure grammar, for a given non terminal symbol  $A$ , one may have  $n$  rules with  $A$  on the left-hand side, which will be interpreted as stating that this symbol is to be rewritten as specified by rule 1, *or* rule 2, ... *or* rule  $n$ . In other words, the  $A$  language is the result of the *union* of the right-hand side specifications, where concatenation is the primary operation. As an example of this preferred use of union and concatenation, one would remark that the language VC of figure 2 would also, in a more classical manner, be defined by the following regular expression, *i.e.* as the union of three languages<sup>4</sup>:

```

define VC [VBase | BE (Neg) VIng
          | DO (Neg) VInf];

```

As this definition is more compact than that of figure 2, one might wonder what could be the advantage of using Properties. The advantage lies in the greater modularity of linguistic descriptions Properties offer. As noted by [10], Properties can be viewed as “a systematization of the decomposition of information initiated by the GPSG ID/LP formalism: the information expressed by the ID rules in GPSG are expressed by the conjunction of the *amod*, *uniq*, *oblig*, *exig* and *exclu* properties”. The consequence of this decomposition is that it will be easier to adjust linguistic descriptions to what is seen as variations within a data set (e.g. regional variations of a given language, or spelling errors in a written corpus)<sup>5</sup>.

### 3.2 Properties vs. FSIG

Properties contrast with classical phrase structure grammars in that they favor intersection, but [7] al-

<sup>4</sup> Not counting the use of union in the category definitions, which we assume to be that of the first nine lines of figure 2.

<sup>5</sup> Section 5.2 gives hints at how such adjustments could be implemented.

```

amod(S, [a, b, S])
uniq(S, [a, b, S])
oblig(S, [a])
exig(S, [a, b])
precede(S, [a, [b, S]])
precede(S, [S, [b]])

```

**Fig. 3:** Definition of the  $a^n b^n$  language.

ready described a parsing system based on constraints “implemented as finite-state machines” and where “the grammar as a whole is logically an intersection of all constraints”. The result of our translation is conceptually identical to that framework, but the Property formalism, however, does differ from its predecessor.

In practice, the preferred rule format in the grammar described in [8] is  $EXP \Rightarrow LC \_ RC$ , which specify that any occurrence of EXP must be surrounded by the given contexts LC and RC (all three parts of the rule being regular expressions). This kind of rules, like phrase grammar rules, in effect favours concatenation and union as the primary operations, as contexts are often specified as a disjunction of admissible strings.

In addition to this type of rules, the rule formalism of [8] gives the linguist the possibility to specify definitions of the form

```
name(param1, ..., paramn) = regex;
```

which could be used to define not only Properties, in the same manner as we did in figure 1<sup>6</sup>, but also *any* new predicate. The formalism of [8] allows one to use the full power of regular expressions, while Properties, in contrast, form a closed set of predefined constraint schemata.

## 4 Expressive power of the Property formalism

If it is possible, as we have shown, to translate Properties into regular expressions, then one must come to the conclusion that the expressive power of Properties is limited to the specification of regular languages. However, [3] gives an example of Properties specifying the context-free language  $a^n b^n$ , an example which is reproduced on figure 3. There is here an apparent contradiction, which deserves consideration.

We first examine the interpretation of Properties as specifying CF languages, and then discuss our stricter interpretation, in which they specify regular languages.

### 4.1 Properties specifying context-free languages

The understanding of Properties as specifying context-free or regular languages lies in the meaning one assigns to the symbols used as the arguments of the Properties. In our interpretation of Properties as

<sup>6</sup> Indeed, [8] give as an example definition the statement  $UNIQUE(FINV)$ .



equivalent to regular expressions, the symbols used as arguments of the properties are variables which are defined non recursively. Properties apply to the strings of the language, strings of terminal symbols. In the interpretation of [3], the symbols used as arguments of properties may be either terminal symbols or recursively defined non terminal symbols (e.g. symbol *S* in figure 3). Properties do not apply directly to the strings of the language, but to the strings of immediate constituents of a category. The description presupposes a phrase structure tree, and the Properties apply to levels in this tree.

The  $a^n b^n$  language example illustrates adequately this orientation: it is possible to say that the string *aabb* satisfies all the properties of figure 3 because the description implies the tree in figure 4, and the properties are about the immediate constituents of each *S* constituent, not about the string *aabb* itself (in which case, quite trivially, the property *uniq(a, b, S)* would *not* be satisfied, since the string contains several *a*s and several *b*s).

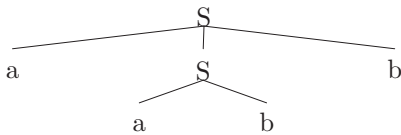


Fig. 4: Analysis tree for the string *aabb*.

[5] state that, in “Property Grammar”, “parse trees are no longer necessary”. [4] claims that “Property Grammar is a non-generative theory in the sense that no structure has to be build, only constraints are used both to represent linguistic information and to describe inputs.” As our analysis of the  $a^n b^n$  example shows, if Properties are to be interpreted as possibly applying to recursively defined non terminal symbols (as indeed they are in the cited articles), these statements are wrong.

## 4.2 Properties vs. regular expressions

Coming back to the interpretation of Properties of figure 1, one may question whether the Property formalism has the same expressive power as regular expressions. The answer is no.

Assuming categories defined using only the union operator (as in figure 1), the expressive power of Properties (understood as in figure 1) is strictly smaller than that of XFST regular expressions. For instance, it is impossible to define with Properties the language denoted by the regular expression  $[a (a)]$ , *i.e.* the set  $\{a, aa\}$ . As a rule, one cannot precisely control with Properties how many words of the same category are allowed in a string (e.g. say “one or two *a*s”). One can only state that such words may or must appear (with the *amod* and *oblig* Properties, and that there may only be one such word (with the *uniq* Property). The *exig* Property cannot help in that matter as a Property such as *exig(id, [a, a])* would be trivially satisfied.

## 4.3 (Dis)advantages of the two interpretations

At this point, we are left with two interpretations of one formalism. Choosing one rather than the other would presumably depend on one’s objectives and on the (dis)advantages of regular expressions vs. context-free grammars. As is well known, natural language has been shown to be non regular, but it has also been shown that some aspects of natural language syntax could be described by finite-state methods (e.g. chunks).

Appropriateness of Properties to such and such objectives (e.g. chunking vs. deep parsing) will be demonstrated by the development of effective linguistic descriptions. However, we would venture that the use of an underlying tree representation might suffer from two drawbacks: (1) interpretation of the Properties would sometimes be counter-intuitive (as when one reads that the string *aabb* satisfies the Property that there is only one *a*), and (2) it might make the Property formalism less relevant, as the tree tends to reduce the length of the strings Properties apply to. For instance, if one would work with binary branching trees, one might question whether Properties are not too sophisticated a system to describe two word strings.

## 5 Practical application of the translation scheme

Our translation of Properties into XFST regular expressions helped us to clarify the understanding of the Property formalim, but it also offers at no cost a platform to actually put Properties in practice, as one can use all the functionalities of the Xerox software. We here evoke some possible uses, which will be presented with the following definition as a reference:

```
define L [ P1 & P2 & ... & Pn ] ;
```

We will say that this formula defines the language *L* as well as the automaton *L*.

### 5.1 Analysis and generation

The most straightforward use of XFST will be to compile the automaton *L* and use it either to test whether a string belongs to the *L* language, or to generate strings of *L*, possibly *all* the strings of *L* if it is finite. In this latter case, XFST provides the “pattern generator” of [2, §5] and this shows that Properties may indeed be used for generation.

### 5.2 Multiple automata from a single set of Properties

[3] claims that Properties challenge generativity in that rather than parsing only grammatical sentences, one can take any input sentence and produce the lists of Properties it satisfies or not. This may be implemented within XFST by defining one automaton for each Property and analysing strings with each of these automata in turn. More generally, given a set *P* of

Properties defining  $L(P)$ , any subset  $P'$  of  $P$  can be used to define a language  $L(P')$  of which  $L(P)$  will be a subset.  $L(P')$  can be viewed as a language resulting from the relaxation of some constraints on  $L(P)$  (*i.e.* the subtraction of some Properties), a language which in effect contains sentences which would be judged ill-formed with respect to  $L(P)$ .

Properties offer modularity and an easy way to define from a single base set multiple languages included in each other. This quality, however, does not question the fact that the Property formalism in itself fully belongs to the generative grammar paradigm.

### 5.3 Testing the relevance of Properties

Another application of XFST is that it makes it possible to verify that in a set of Properties defining a language  $L$ , each Property is *relevant* to the definition of  $L$ . Given the definition of  $L$  above, the following XFST command sequence tests the relevance of any Property  $P_i$ :

```

regex L ;           Put the L network on the stack.
regex L - Pi ;      Put the L - Pi network on the
                    stack.
test equivalent    Test whether the top two net-
                    works on the stack are equiva-
                    lent.

```

A Property  $P_i$  is relevant to the definition of a language  $L$  iff  $L - P_i$  is not equivalent to  $L$ .

Note that this relevance testing procedure makes use of the subtraction operator. Unlike context-free languages, regular languages are closed under subtraction, as well as under intersection. We may then view this procedure as another advantage of our XFST interpretation of Properties.

### 5.4 Exploring the relevance of the Property formalism

Ultimately, the relevance of Properties will be demonstrated (or not) by effective descriptions of natural languages. Looking at our translation of Properties into regular expressions, one might wonder what would be the point of using Properties rather than regular expressions?

The weaker expressive power of Properties (*cf.* section 4.2) actually suggests a nice experimentation program: how far can we go into the description of natural languages with Properties understood as in figure 1? The objective would be to determine what, within finite-state expressivity, is needed or not to describe such and such aspect of a language, to find the appropriate position between a system using the full power of regular expressions and a system strictly limited to a specific set of constraint schemata.

Properties put constraints on the linguist's expression, but it might be to their benefit. [9] introduced a translation system from natural language to XFST regular expressions. This author, pointing out that the same thing could be said in a messy way as well as in a structured way, concluded his demonstration by advocating the importance of structured programming. We believe Properties are a good way to structure linguistic descriptions. Especially if, rather than

the sometimes cumbersome notation of regular expression Properties in figure 1, one considers the possibility of an interface to this notation.

## 6 Conclusion

We presented a translation of the Properties of [2] into regular expressions, a translation which we consider an *indirect* implementation of this formalism. To us, this indirectness is an advantage, because

- it helped to clarify the interpretation of the Property formalism,
- it provides at no cost a tool to actually analyse and generate strings defined by a set of Properties, as well as a tool to test the relevance of each Property,
- as it integrates Properties into a system with greater expressive power, it opens space to test the limits of the Property formalism on linguistic data.

With respect to the interpretation of Properties, our comparison shows to what extent they depart from classical phrase structure grammars, favouring the definition of a language by the intersection of sets rather than union, but also to what extent they do belong to this paradigm. In particular, as any set of Properties may indeed be translated into an equivalent regular or context-free grammar, they can be assigned the same interpretation as such grammars, *i.e.* they are expressions which denote languages.

## References

- [1] K. R. Beesley and L. Karttunen. *Finite State Morphology*. CSLI Studies in Computational Linguistics, 2003.
- [2] G. G. Bès. La phrase verbale noyau en français. *Recherches sur le français parlé*, 15:273–358, 1999.
- [3] P. Blache. *Les Grammaires de propriétés*. Hermès Science Publications, 2001.
- [4] P. Blache. Property grammars: A fully constraint-based theory. In *Constraint Solving and Language Processing*. 2005.
- [5] V. Dahl and P. Blache. Directly executable constraint based grammars. In *Proc. Journées Francophones de Programmation en Logique avec Contraintes*, Angers, France, June 2004.
- [6] L. Karttunen, T. Gaál, and A. Kempe. *Xerox Finite-State Tool*. The Document Company - Xerox, 1997.
- [7] K. Koskenniemi. Finite-state parsing and disambiguation. In H. Karlgren, editor, *Proceedings of COLING-90*, Helsinki, 1990.
- [8] K. Koskenniemi, P. Tapanainen, and A. Voutilainen. Compiling and using finite-state syntactic rules. In *Proceedings of COLING-92*, Nantes, 1992.
- [9] A. Ranta. A multilingual natural-language interface to regular expressions. In L. Karttunen and K. Oflazer, editors, *Proceedings of the International Workshop on Finite State Methods in Natural Language Processing*, pages 79–90, Bilkent University, Ankara, 1998.
- [10] F. Trouilleux. Note de lecture sur Philippe Blache, *Les Grammaires de propriétés*, Hermès Science Publications. *TAL*, 44(2):256–259, 2003.

# Consolidation and unification of dispersed multilingual terminology data

Andrejs Vasiljevs  
Tilde  
Vienibas gatve 75a  
Riga, LV1004, Latvia  
andrejs@tilde.lv

Signe Rirdance  
Tilde  
Vienibas gatve 75a  
Riga, LV1004, Latvia  
signe.rirdance@tilde.lv

## Abstract

This paper addresses the problem of consolidation of multilingual terminology resources that are dispersed among numerous collections, publications and databases. It proposes a standards-based approach for providing single unified web-based access to distributed multilingual terminology data. This federation approach supports consolidation of different established terminology databases and centrally stored resources. This paper introduces terminology entry compounding for identification and consolidation of matching multilingual entries from different collections. Practical results from using these approaches in EuroTermBank project are described as well as future development directions are pointed out for expanding EuroTermBank into the global access point for unified multilingual terminology.

## Keywords

Translation aids, terminology databases, multilingual terminology, terminology entry compounding.

## 1. Introduction

Globalization from the one side and growing language awareness from the other side dictate the need to consolidate terminology resources, harmonize international terminology, and provide online access to reliable multilingual terminology. Advances in language technologies and machine translation are about to change the traditional patterns of creation and use of language resources. New approaches and platforms are urgently required to support these requirements.

At the same time, terminology development and distribution continue to be fragmented across organizations, companies, industries and languages. As a result, existing resources are often not publicly available, reuse and application for research of large amount of accumulated data are extremely limited, quality is unreliable and proprietary formats are incompatible with international standards.

This article provides an overview of the foundations of EuroTermBank, a new type of multilingual terminology platform that proposes solutions to some of the inherent challenges of terminology management and application. First, it focuses on the existing and emerging standards in the realm of multilingual terminology, starting from terminology resource description to the data model and

exchange mechanisms. Then, it proposes terminology entry compounding as a new approach and a tool for identification and display of matching multilingual terminology entries across several terminology collections. Lastly, this article proposes a federated model of terminology consolidation and distribution, with terminology data from diverse sources made available through a central gateway, using distributed environment with a multitude of actors.

Complete account on various aspects of the EuroTermBank project is provided in a monograph based on EuroTermBank project deliverables [7]. A separate paper is dedicated to EuroTermBank's methodology of multilingual terminology work [6].

## 2. Overview of EuroTermBank project

An important initiative to address the above challenges, EuroTermBank project [1] was designed with the goal to collect, harmonize and disseminate dispersed terminology resources through an online terminology data bank.

EuroTermBank project was initiated in 2004 by 8 partners representing research institutions, terminology organizations and language technology companies from 7 European Union countries – Germany, Denmark, Latvia, Lithuania, Estonia, Poland, and Hungary.

Within the project, methodology for harmonization of terminology processes in new EU member countries and for ensuring compatibility of terminological resources for data interchange and resource sharing has been developed [6]. A web-based terminology data bank [www.eurotermbank.com](http://www.eurotermbank.com) has been created, to provide easy access to centralized terminology resources.

The objective of EuroTermBank is to integrate available terminology resources (not only from project partner countries) into the central EuroTermBank database or interlink them via EuroTermBank as a central gateway and a single point of service. The data bank works on a two-tier principle – as a central database and as an interlink node or gateway to other national and international terminology banks. Data exchange mechanisms have been developed to establish term import, export and exchange with other terminology databases.

A large number of terminology resources have been acquired and processed for inclusion into the

EuroTermBank database. The methodology developed in EuroTermBank project serves as the basis for content processing. The content passes several stages before integration into the database, including selection, prioritization, modification, and digitalization (for non-digital format).

The outcome is a reliable multilingual terminology resource, networked with other existing national and international resources available for users over the global network.

Currently EuroTermBank portal enables searching within approximately 600,000 terminology entries containing over 1.5 million terms in various languages and coming from about 100 terminology collections. A number of these collections were not available in digital format before this project; a few specialized term banks were not available to the general public at all. The initial focus of EuroTermBank has been on the “new Europe”, including Estonian, Hungarian, Latvian, Lithuanian, and Polish term collections.

### 3. Data consolidation based on standards

With the multitude of actors involved, including project partners and various types of terminology holders, implementation of applicable international standards, as described in this section, has been key to reaching the goals of the project, including a standards-based approach to describing terminology collections, defining the data model and ensuring a unified data exchange format.

#### 3.1 Terminology resource description

One of the major tasks of the EuroTermBank project was identification and description of terminology resources available in the new EU member countries. Due to a large number of resources to be described and different organizations in several countries involved in this process, it was important to use a common format for resource description. For this purpose the TeDIF format was chosen.

The Terminology Documentation Interchange Format TeDIF [4] is an SGML-based format for describing and exchanging metadata about terminology, developed in the framework of the TDCnet project – European Terminology Documentation Centre Network, with the purpose to establish a common format for bibliographical and factual data related to terminology.

For the purpose of the EuroTermBank project TeDIF was slightly adapted. TeDIF information types were limited to the description of term collections. Other modifications included: 1) a possibility to multiply the fields describing the author and copyright holder according to the number of persons/organizations, and 2) the addition of fields for the indication of the languages of definitions and context information.

TeDIF is used for importing terminology resource meta-data into EuroTermBank database, as well as for consolidation and analysis of data.

#### 3.2 Terminology data modeling

The data structure developed for EuroTermBank comprises up to 4 hierarchical levels based on ISO standards 12200 and 12620 [5]:

- The entry level provides concept-related data categories applying to all languages. It contains language-independent information like *entry identifier*, *subject information*, *data collection*; administrative information like *subset owner* identifying the institution responsible for the entry; *originator*, *origination date*, *updater*, *modification date* and a number of other fields.
- The language level provides concept-related data categories applying to the specific language. It contains language-specific information like *definition*, *reference*, *explanation* and others, as well as administrative information.
- The term level provides term-related data categories applying to the specific term. It includes term-related information like *term* in a particular language, *entry source*, *search term* containing related forms of the term to facilitate search, *reference* with source(s) of the term, *usage information*, and others.
- The word level provides word-related data categories applying to the specific words of a term. A term may be a multiword string, therefore this level is created to contain lexical information that concerns the individual words of a term. Data categories for lexical information are, for example, part of speech, grammatical number, grammatical gender etc.

#### 3.3 Terminology data exchange

Data exchange mechanisms are required to enable term import, export and exchange with other terminology databases. EuroTermBank data exchange format is based on TBX (TermBase eXchange) format, an open XML-based standard for terminological data exchange developed by LISA (Localization Industry Standards Association). TBX complies with the terminology markup framework defined by ISO 16642; it specifies a set of data categories from ISO 12620 and adopts an XML style compatible with ISO 12200.

The EuroTermBank system implements the TBX standard with required data categories to enable:

- data exchange between different EuroTermBank modules;
- data exchange between external terminology databases;
- data import and export to and from the EuroTermBank terminology database;
- data store in the EuroTermBank terminology database;
- data editing.



To import terminology data into the EuroTermBank database, data must be structured according to the EuroTermBank TBX-compliant data exchange format. To convert terminology resources to this specific format a number of conversion tools are created. As each resource is structured differently, an individual converter is developed or adapted for each resource type.

## 4. Entry compounding for unified representation

This section describes the approach taken in the EuroTermBank project for unification of potentially matching terminology entries from different resources.

Majority of terminology resources that are available in Eastern European countries are bilingual with the source language typically being English. A small number of resources are monolingual or have terms in three or more languages.

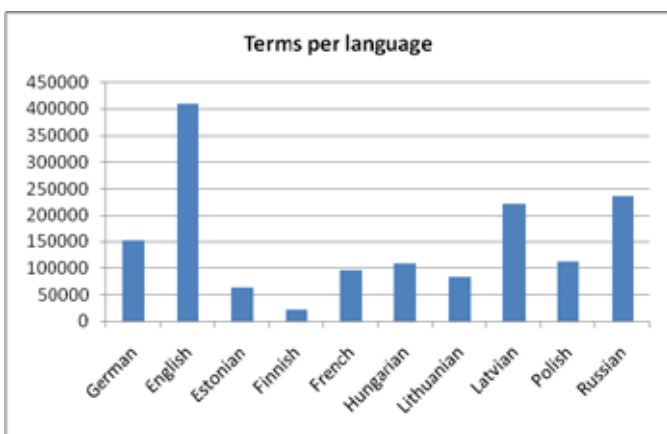


Table 1. EuroTermBank resources by languages covered.

As described earlier, EuroTermBank data structure is modeled according to concept-oriented approach to terminology. A terminology entry denotes an abstract concept that has designations or terms as well as definitions in one or more languages. If the terminology bank contains entries coming from different collections and designating the same concept we have an obvious interest to merge them into one unified multilingual entry.

For example, if there is a term pair *EN computer – LV dators* coming from a Latvian IT terminology resource and another term pair *EN computer – LT kompiuteris* from a Lithuanian IT terminology resource, it is possible to combine these two into a unified entry *EN computer – LV dators – LT kompiuteris*. This multilingual entry allows establishing a correspondence between language terms that is not directly available in any terminology resource (in this example, the new term pair is *LV dators – LT kompiuteris*).

However, merging entries just on the basis of matching term in one language that is common for the

given entries will lead to many erroneous term correspondences, resulting from frequent ambiguity of terms among subject fields or much rarer cases of ambiguity in the context within one subject field. The only error-free method for merging entries is evaluating manually whether these entries denote the same concept.

Unfortunately in practice it is often impossible or very expensive to make comparisons of cross-lingual terminology concepts. There is a lack of experts with sufficient knowledge of the respective languages and subject fields. The task is considerably hindered by the fact that most terminology collections do not provide definitions.

EuroTermBank proposes a practical solution by introducing the automated terminology *entry compounding* approach for matching terminology entries based on available data.

### 4.1 Entry compounding criteria

EuroTermBank uses the following entry compounding criteria used to match terminology entries across terminology collections:

- Unique concept identifiers:
  - ISO terminology identifiers
  - Latin in medicine and biology
- Identical English term with the same subject field classification

Other criteria were considered, such as using a second “lingua franca”, but not applied, to ensure high levels of reliability of the compounded results. Further research on entry compounding would be necessary to evaluate effectiveness of entry compounding methods for various types of terminological data.

The most reliable indication for matching entries is having unique and unambiguous concept identifiers. The best example here is terms from ISO terminology standards. These term entries have an identifier in the form [Standard\_identifier].[term\_number]. Accordingly, all national standards share the same identifier for corresponding entries and can be merged with a very high degree of reliability.

Another case of unique internationally applied identification is the usage of Latin names in medicine and biology (with a number of exceptions with different Latin names designating the same concept).

If there is no unique identification for concepts in collections, less precise matching criteria are used, namely, the English term and the subject field. English has been chosen as the most popular language in term resources.

For subject field classification, EuroTermBank uses Eurovoc, the multilingual thesaurus provided in all official languages of the EU that covers the fields in which the European Communities are active. Resources originally

having different classification (e.g. Lench) have been mapped to Eurovoc classes.

A number of terminology resources use only top classification levels of Eurovoc, although there are many resources with detailed classification using Eurovoc sublevels of different depth. For entry compounding, it was decided to take into account only the top classification level with 21 subject fields for entry compounding. This means that sublevels are equalized to the top classification level.

Some additional assumptions applied in the implementation of entry compounding for EuroTermBank resources:

- If an entry contains several English terms as synonyms, matching of at least one of them is sufficient.
- If an entry is classified under several subject fields, matching of at least one of them is sufficient.

It is important to understand that entry compounding is a data representation method that does not propose creation of new permanent term entries. It is a visualization aid that displays matching entries across collections. Much like in machine translation environment, the user is prompted about potential incompatibilities and errors.

#### 4.2 Initial evaluation of entry compounding

Entry compounding solves the problem of visual representation of multiple potentially overlapping term entries that are present in a consolidation of a huge number of multilingual terminology sources.

At present, the EuroTermBank database contains over 585,000 term entries. When applying entry compounding, over 135,000 or about 23% of entries get compounded. Hence entry compounding is a considerable aid for the user in finding the required term, for example, in the translation scenario between language pairs for which term equivalence is not established in existing collections.

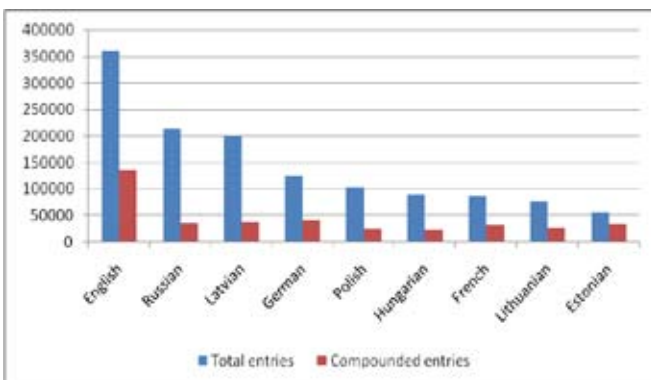


Table 2. Total and compounded entries per major languages of EuroTermbank.

At the same time, entry compounding may uncover incompatibilities and deficiencies across data and is therefore useful for further enhancement of the original data, but may be confusing for the immediate users.

The major source of problems for entry compounding lies in shortcomings of the subject field classification system and its application. While Eurovoc thesaurus was evaluated as the best practical solution, the lack of a universal, terminology-oriented classification system is evident. In addition, entry compounding problems may occur due to different interpretations or errors while applying the classification system across all term collections, or due to inherent differences across these collections. For example, errors may occur if a term within a subject field is used to denote several different concepts. This scenario contradicts to best practice methodology in terminology development, however, practice shows that existing term collections contain such deviant cases.

#### 5. Federated approach in consolidation of distributed resources

A convenient way of analyzing terminology consolidation practices is through a number of scenarios or frameworks [6]. Often, consolidation of terminology is a concept applied in the context of an organization or a company, for example, a company-wide terminology management system as implemented in IBM, or at the national level, for example, the Termbank of Lithuania. Beyond the company/local level and the national level is the international level, for example, the single IATE database [3] that consolidates multilingual terminology across a number of EU institutions; important steps towards the federated model of standards terminology development are taking place within ISO, which is an example of a single international organization working towards a federation of a number of internal databases.

EuroTermBank proposes terminology consolidation at this further level, uniting dispersed terminology databases in a federated system. To ensure viability of this system, inclusion of a termbank in the federated model requires it to be institutionally and technically supported and maintained.

The federated approach to terminology consolidation provides a solution to at least one inherent challenge of all terminology banks – maintenance of terminology is done at the local or national level, and the changes at the local or national level become instantaneously available for integration with other federated resources.

An important by-product of this approach is the promotion of a unified methodology for terminology work and application of industry standards.

Federation of terminology is a new phenomenon and there are a number of challenges yet to be faced, for

example, ensuring reliability of the sources or of the source data in case an important resource of the federation becomes unavailable, temporarily or ultimately, and ensuring a unified approach to change management on all levels, from data structure to the changing terminology content and preservation of legacy data. Another common challenge to terminology termbanks exacerbated in a federated model is the application and mapping of subject field classification systems. A major challenge is the implementation of a concept-oriented approach requiring a certain level of concept harmonization in a multilingual setting with diverse terminology creators.

However, these challenges are inherent to all terminology work, even on individual level. EuroTermBank's advantage lies in a more efficient and consolidated approach in solving these challenges, compared to the uncoordinated and oftentimes partial and incompatible solutions typical at the local level.

Several major external terminology databases are interlinked with EuroTermBank. An example of a national terminology database interconnected with EuroTermBank is online databank of Latvian official terminology [2]. An example of an international databank that could greatly benefit from interlinking with EuroTermBank is IATE, the termbank of European Union official institutions [3].

## 6. Conclusions

In today's world, market forces and technology developments dictate the need to consolidate dispersed language resources, including multilingual terminology data. Advances in machine translation and language search engines will radically change the traditional patterns of creation and use of language resources. New approaches and platforms enabling both human and machine use of diverse, dispersed resources in a consolidated environment are already emerging.

Representative of this new generation of collaborative environments, EuroTermBank proposes solutions to some fundamental challenges of handling multilingual terminology data. To summarize the most important points and lessons from the EuroTermBank project:

- observance and full application of standards in data consolidation are absolutely essential to interoperability and further applications of terminology data;
- entry compounding for representation of matching multilingual entries is an innovative mechanism for creation of a new type of automatically formed multilingual terminology entries;
- the federated approach in consolidation of resources enables distributed terminology to be accessible through a central gateway while it is maintained locally.

As a new type of terminology infrastructure providing access to diverse terminology resources, EuroTermBank is a model for further consolidation of terminology in Europe and beyond. Its rich and standards-based multilingual terminology resource collection, together with innovative instruments for analysis, can be used for research in terminology, lexicography, and computational linguistics, as well as applied in computer-assisted translation systems.

## 7. Acknowledgements

Many thanks to colleagues in all EuroTermBank project partner organizations: Tilde (Latvia), Institute for Information Management at Cologne University of Applied Science (Germany), Centre for Language Technology at University of Copenhagen (Denmark), Institute of Lithuanian Language (Lithuania), Terminology Commission of Latvian Academy of Science (Latvia), MorphoLogic (Hungary), University of Tartu (Estonia), Information Processing Centre (Poland). EuroTermBank Consortium would also like to acknowledge and thank the European Union eContent Programme for supporting the EuroTermBank project. Preparation of this paper was also supported by European Social Fund (ESF).

## 8. References

- [1] Vasiljevs A., Skadiņš R. 2005. EuroTermBank terminology database and cooperation network. In: Proceedings of the Second Baltic Conference on Human Language Technologies, Tallinn, pp. 347-352.
- [2] Skadiņš R., Vasiljevs A. 2004. Multilingual Terminology Portal – termini.letonika.lv. In: Proceedings of First Baltic Conference „Human Language Technologies – the Baltic Perspective”, Riga, 183-186.
- [3] Rummel, D., S. Ball 2001. The IATE Project – Towards a Single Terminology Database for the EU. In: Proceedings of ASLIB 2001, the 23rd International Conference on Translation and the Computer, London.
- [4] Betz A.; Schmitz K.-D. 1999. The Terminology Documentation Interchange Format TeDIF. In: Terminology and Knowledge Engineering TKE'99, Innsbruck, Wien, pp. 782-792.
- [5] Wright, Sue Ellen 2005. A Guide to Terminological Data Categories – Extracting the Essentials from the Maze. In: Proceedings of TKE 2005, the 7th International Conference on Terminology and Knowledge Engineering. Copenhagen, pp. 63-77.
- [6] Henriksen L., Povlsen C., Vasiljevs A. 2005. EuroTermBank – a Terminology Resource based on Best Practice. In: Proceedings of LREC 2006, the 5th International Conference on Language Resources and Evaluation, Genoa, on CD-ROM.
- [7] Aukšoriute, A., et al. 2006. Towards Consolidation of European Terminology Resources / Experience and Recommendations from EuroTermBank Project, Riga.

# Table classification: an application of machine learning to Web-hosted financial texts

Marc Vilain, Jonathan Gibson and Rob Quimby  
The MITRE Corporation  
202 Burlington Rd, Bedford, MA 01730 USA  
{mbv, jgibson, rquimby}@mitre.org

## Abstract

This paper presents learning-based techniques that support the processing of tables in HTML publications. We are concerned especially with classifying tables as to format and content, focusing on the domain of corporate financials. Towards this end, we define a range of new table classification tasks, present performance results for these tasks using multiple classification methods, and introduce a new evaluation corpus.

## Keywords

Text classification, information extraction, machine learning, maximum entropy models, SVM models, annotation.

## 1. Introduction

Tables matter. In a text document, they serve many purposes: they summarize, they aggregate, and they display change over time. The essence is this: a table provides for a compact and readable representation of relational or attributive information. Often, the most important information in a text is found in tables. This is certainly true in financial domains, where tables of financial figures are the lingua franca of accountants and investors alike.

Our work focuses on applying information extraction and data mining techniques to these kinds of financial documents. In so doing, we have investigated an array of issues regarding tables, namely their identification, classification, and de-structuring. We focus in this paper on the issue of classifying tables published in HTML, which is increasingly the medium of record for corporate financials (in 2005, for example, 96% of the Fortune 500 companies released their regulatory financial filings in HTML).

We have approached the problem from a machine-learning perspective, accepting as given several standard classification techniques. This paper therefore does not seek to advance the state of the art in machine learning, so much as to advance the state of the practice in the application of machine learning. In particular, we (i) applied the key classification test of *table genuineness* to the financial domain, besting earlier results for general text, (ii) extended the notion of fine-grained subject classification from free text to tables, and (iii) designed an efficient table annotation interface, and created a new table classification corpus.

## 2. Background: table processing

Much table processing research has focused on modeling the structural layout of a table, with the aim of exploiting

these layout models to automatically extract function-value relationships. Computational linguistic approaches typically cast the problem as a form of sequence labeling. Pinto *et al* [8], for instance, use conditional random field (CRF) models to label row headers, column headers, cells, and other table structure elements. The idea is that once a cell has been identified (say, with content “\$10,000,000”), along with its row and column headers (say, “earning” and “FY06” respectively), it is possible to extract such function-value tuples as *earnings(fy06)=\$10,000,000*.

Related current work in field segmentation has been applied to other semi-structured data, *e.g.*, bibliography entries ([2] [9]). In addition, an enormous body of work on table extraction falls under Kushemric’s rubric of wrapper induction [6], where the problem is cast as one of learning wrapper programs, *i.e.*, rules, procedures, or statistical models that de-structure a table extract the contents of its cells. Nearly every learning method in the book has been used: supervised discriminative methods [8], generative unsupervised approaches [2], rule-based supervised methods ([5] [7] *etc.*), synchronous CFGs [11], and so forth.

Behind the problem of learning table structure models or wrapper programs, however, is a (surprisingly) even more basic problem: namely determining whether a table actually represents tabular data in the first place! This problem has particularly become acute with the rise of Web-hosted tables formatted in HTML. The issue is that HTML’s <TABLE> construct provides a convenient framework for much more than just the creation of conventional numeric tables. In practice, <TABLE> is also widely used as a tabbing environment for the purpose of aligning columns of text, creating bullet lists, and so forth. This is particularly true given the widespread use of page layout programs, as these often compile everything from bullet lists to footnotes into <TABLE> constructs.

Wang and Hu consequently formulated the problem of identifying whether a table is *genuine* [10]. A genuine table is to be understood as a two-dimensional encoding of relational information, *e.g.*, bus schedules or stock market performance. A table is *non-genuine* if it only presents a two-dimensional layout of unrelated elements that share no underlying relation such as time of arrival or price. Cohen *et al.* [1] follow this notion, and attempt to identify genuine tables through standard classification methods: decision trees, Winnow, and maximum entropy models.



Category	Description
Data v. time	Cell values are (mostly) numbers, rows are indexed to a quantity (e.g., income), columns are indexed to a date
Time v. data	Ibid, but rows are date-indexed, and columns quantity-indexed
Other num	Number-valued cells with no chronological indexing of rows or columns
Text	Cell values are (mostly) text, meaning phrases, sentences, or even paragraphs

**Table 1. Major table categories.**

In our own work, we have found substantial value in this test of table genuineness. In financial reports, non-genuine tables are often filled with useful zones of free text, all of which are potential fodder for our text mining and information extraction methods. By excluding their genuine counterparts, we are able to run our text processing methods safely on these tables, without creating the gobbledegook that arises from (for example) applying a name tagger to actual financial tables.

In addition, we have also found that the conventional text-processing notion of subject categorization (Joachims [4] among many) has a corresponding utility with tables. In the case of financial reports, we have found that by further categorizing genuine tables as to financial subject matter, we can automatically produce a fine-grained *table of tables*. We have built a prototype application that exploits this to provide direct access to task-relevant information, e.g. income statement tables, listings of executives, etc.

This concern with genuine tables and table subject represents a significant departure from the bulk of earlier research directed at harvesting tables from the Web. Indeed, much work in wrapper induction has little need for these notions, as it presupposes a use case in which a user or robotic agent interacts with some Web-hosted data source by issuing information-seeking queries. The source typically responds by searching its internal database and posting the results as an HTML page: the job of a wrapper program is to de-structure this table and retrieve the sought-for data. Because of the highly targeted nature of the queries, there is little need to test for table genuineness or subject.

Much knowledge on the Web, however, does not fit this query-response model, but is closer to a conventional model of publication, and just happens to use the Web as a release medium. This is particularly true in our own area of corporate financial reports. Even though these are now almost entirely HTML-based, their fundamental format is still one intended for human perusal. For sources such as these, table classification becomes much more of an integral part of the text processing pipeline.

Category	Description
Numeric minor categories	
Income statement	The three major accounting views of a company's financials
Balance sheet	
Cash flows	
Consol. inc. stmt.	Consolidated versions of the above, typically produced by a company's auditors
Consol. bal. sh.	
Consol. cash fl.	Value, shares outstanding...
Stock info	
Pension plan	Stock/pension plans often have their own financials
Stock plan (esop)	
Misc. numeric	Anything else
Text-related minor categories	
Table of contents	For sections, attachments...
Bullets	Bulleted or numbered lists
Footnote	Often linked to num. tables
Signatures page	Auditor sign-off, etc.
Fat cats	Board members, executives
Stock info	Rare non-numeric tables
Text table	Tables such as this one
Formatting	Other alignment uses
Misc. text	Anything else

**Table 2. Key minor table categories: gray text shows low-count categories that were folded into other categories**

### 3. Financial table classification

For the purpose of this work, we defined a new table classification task oriented to the needs of the financial domain. As this is a novel task that we intend to share with other researchers, we will describe it here in some detail. We concentrated specifically on the annual securities filings (form 10-K) submitted by publicly-traded companies in the United States. This is a departure from earlier research in table classification that attempts to work with a general sampling of Web sources. It would not be practical, however, to design a meaningful fine-grained typology of table classes for the Web in general. By focusing on financial reports in particular, we were able to design non-trivial and useful typologies for both genuine and non-genuine tables.

In keeping with the preceding discussion, we defined two levels of table classification: a coarse distinction between genuine and non-genuine tables (a table's major category), as well as a fine-grained assignment of table type (its minor category). These typologies are shown in Tables 1 and 2.

### 3.1 Table categories

We recognize four major categories that broadly follow the division between genuine and non-genuine tables. The first three categories taken together (the numeric categories in Table 1) correspond to Wang and Hu's genuine tables. Text tables are our equivalent of non-genuine tables, as they mostly can not comfortably be construed as encoding an underlying semantic relation. The special case of tables of contents, which could arguably be typed as either *text* or *other num* is, for our purposes, taken to be text-typed.

Of the minor categories in Table 2, some are specific to numeric tables, and others to text tables. For the numeric tables, the minor categories are defined on a primarily semantic basis, providing a counterpart to subject classification in free text. They mostly include such accounting categories as *income statement* and the like. In contrast, the minor categories for text tables reflect our need to process specific table layouts in idiosyncratic ways. As such, their distinctions are primarily syntactic. Among them is the particular category of *Formatting*, an undistinguished catch-all that captures those cases where `<TABLE>` is exploited for alignment purposes alone.

Table 2 omits a number of particularly low-count categories, which we lumped into the two *Misc.* categories. Among these rare tables were some that the annotators had marked as belonging to multiple minor categories, *e.g.*, financial tables that were both a balance sheet and an income statement. In addition, rare examples of (*e.g.*) the *fat cats* minor category could arguably be seen as having either text or numeric major categories. We resolved these potential confusions by convention, so (*e.g.*) *fat cats* tables were all taken to be text-typed for their major category.

### 3.2 Corpus development

After setting guidelines for annotation, we created a corpus of 10-K filings in which all tables are marked with both their major and minor category. These documents vary a lot in length, though most are over 100 pages long. They also vary significantly in the number of tables they contain, ranging from a low of 22 tables per filing to a high of 363. This huge variance (avg= 110, sd= 94) is due to differences in reporting styles on the part of the filing corporations. In particular, some went to great length to provide multiple views of a particular financial table, broken down by geography, business, area, and so forth. Others provided only one such table, or simply referred the reader to their annual reports. For our purposes, this variance in reporting styles had implications relative to the practicality of using sequence models—more on this below.

For training and test purposes, we used 30 of these filings in the present study, with a total of 3,046 table instances. We defined a training-test split along document boundaries, collating enough 10-K's into the training camp to create a roughly 50-25-25 split, with 1,615 table instances for training, 723 for dev-test, and 710 for eval-test.

We deliberately chose this particular approach to dividing training and test sets over the more conventional approach of N-fold cross-validation. Because our source documents have such inconsistent table density, cross-validation cannot be readily performed without splitting some documents up and assigning one part of their tables to training, and the other to test. This is not representative of actual operational conditions, where the "test" tables are always drawn from documents with no overlap with the training set. In this case, cross-validation measures are likely to overestimate actual runtime performance.

To construct the corpus, we used a clever trick that allowed us to mark up the filings *in situ* while reading them in a web browser. This gave us a WYSIWYG annotation capability, which we found critically necessary, since HTML table code is unintelligible on its own. In our experience, efficiently and accurately judging the category of a table required that it be fully rendered by a browser. Wang and Hu describe a non-WYSIWYG annotation tool that would have been extremely cumbersome for our needs [10].

As to the clever trick, it worked as follows. We modified the HTML source code of our corpus documents, adding a pair of HTML `<SELECT>` menus in front of each HTML table. These pull-down menus allow the annotator to indicate the major and minor categories of the table with just two mouse clicks. The whole body of the document is then wrapped in some simple HTML boilerplate that allows the annotator to save or reload the mark-up. When the modified document is loaded into a Web browser, it is rendered exactly as the original document was, except for the crucial addition of the pull-down menus, with all the complexities of rendering the HTML tables being handled by the browser.

We performed a preliminary round of triple annotation, followed by adjudication to resolve inter-annotator disagreements. This helped identify annotation tough nuts that required further guidelines. The full corpus was then singly-annotated by our two most consistent annotators.

## 4. Experimental preparation

As noted, we cast the problem of identifying genuine financial tables as a classification task, and exploited a number of machine learning packages to learn a range of table classifiers. After pre-processing and tokenizing the texts, we performed two major steps to create experimental configurations: feature extraction and category mapping.

### 4.1 Feature extraction

We extracted a number of different classification features from the preprocessed documents. As with most previous work, we identified a number of structural features, such as the column or row count. In keeping with the standard text-classification literature, we also extracted bag-of-word lexical inventories, keeping separate tallies for column headers, row headers (if they could be heuristically inferred), and cells. We extracted a number of features that

	Majority	Max Ent	<i>Dev test</i>	SVM linear	<i>Dev test</i>	SVM rad	<i>Dev test</i>
Fine-grained	29%	70.3%	81.3%	82.1	80.0%	83.7%	80.5%
Fine-grained text vs. numeric	47%	91.7%	90.3%	87.6%	90.7%	90.6%	91.6%
Fine-grained numeric vs. text	45%	88.9%	89.9%	91.3%	87.6%	90.8%	88.2%
Major category	56%	94.9%	92.1%	94.1%	90.0%	94.6%	92.1%
Numeric vs. text	56%	98.0%	98.3%	98.5%	97.6%	98.5%	97.5%

**Table 3: Classifier performance (eval test followed by *dev test*); all runs performed without the preceding table label**

were aimed specifically at distinguishing some of our fine-grained categories, *e.g.*, counts of bullet-like or footnote-marking tokens, counts of date words or year numbers, and so forth. Finally we also identified a number of features that encode the lexical and structural context in which a table is found. In addition, for the maxent experiments, we attempted to capture sequence effects (where a table’s category might influence the category of the subsequent table), by including the category of any preceding table as a feature (using the gold standard for training and incremental system output for evaluation).

Curiously, preliminary experiments showed that monotonically adding even *task-informative* features did not always lead to monotonic improvements in classifier accuracy. To counter this chaotic trend, and to identify optimal feature configurations, we comprehensively varied feature configurations, separately toggling on and off various subsets of the features, *e.g.*, task specific tokenization, case-elimination on the bags of words, *etc.* We explored the full cross-product of these configurations by running a large number of training experiments on a modest computing grid (an 8-processor Sun workstation). We then used a held-out development test set to identify optimal feature configurations for each of several classification use cases.

## 4.2 Category mapping

The result of all this feature extraction was to produce a collection of feature vectors labeled with the annotator’s judgment of their major and minor categories. For actual experimentation, these categories were then remapped. The purpose of doing so is two-fold. First, as our original repertoire of minor text categories makes an impractically large set of distinctions (39, not all of which are shown in Table 2), we needed to reduce their number to a more manageable level. As noted, we re-mapped a number of similar low-count categories to each other. Second, we also took advantage of the category-remapping procedure to evaluate a range of possible use cases that do not require identifying the full set of minor categories. We considered five cases.

**Fine-grained (15 categories).** All the categories that remained after collapsing the low-count minor categories.

**Fine-grained text vs. numeric (8 categories).** A further round that collapses all the numeric minor categories together. This corresponds to a text processing use case

where one might ignore the numeric tables, but want to (*e.g.*) name-tag text in non-genuine (text) tables.

**Fine-grained numeric vs. text (8 categories).** A similar round collapsing the non-numeric text categories together.

**Major category only (4 categories).** All the minor categories were ignored, and only the four major categories were considered.

**Numeric vs text (2 categories).** The three number-oriented major categories were further collapsed together. This condition effectively corresponds to making the same genuine *vs.* non-genuine distinction as in previous work.

## 5. Experimental results

We first trained a multinomial maximum entropy classifier, using a Gaussian prior of 100.0 for all runs. We additionally repeated all these experiments with the LibSVM implementation of support vector machines. We used the linear, sigmoid, polynomial, and radial basis kernels, accepting LibSVM’s standard out-of-the-box parameter settings for the first two of these. For the polynomial and radial basis kernels, we first performed a round of parameter tuning, using the best configuration of features that we had found for the 15-way maxent classifier.

As noted, we used a held-out development test set to optimize feature configurations. For each use case, we trained a very large number of classifiers, based on different configurations of activated or disabled feature groups. The results we report here are for the best-performing configuration of features for each of our five use cases. We obtained separate sets of winning configurations for the Maxent and SVM classifiers, and each winning configuration was then separately tested on our evaluation test set.

Table 3 above reports the classification accuracy we measured using maximum entropy and support vector machine models. For SVMs, we only report on the linear and radial basis kernel models. The sigmoid kernel only performed competitively on the 2-way classification, and the polynomial kernel underperformed the others by an extremely large margin (we are still determining whether this is a real effect or just a training error).

All of the classifiers outperform a basic majority-category baseline by 30 to 40 points of accuracy. We also note that except for the 15-way classification of the fine-

grained use case, there is little measurable difference between the three classifiers. All three of them performed better on the use cases with fewer classification distinctions, which is in keeping with the generally accepted wisdom for free text classification (the more the labels, the lower the performance). All three classifiers reached upwards of 98% on the binary use case (a.k.a. the test of table genuineness), and all three reached 94% or more for the 4-way major category classification. The differences are somewhat greater for the 8-way and 15-way classifiers, but only the lower performance of maxent on the 15-way task is statistically significant (based on a randomized test of statistical significance on the dev test).

We should note that these results were measured for feature configurations that did not reference the label of the preceding table. It turned out that this particular feature did not consistently improve accuracy in our maxent experiments; it also proved too cumbersome to implement with LibSVM. The low utility of this feature surprised us, as table sequence information seems a priori useful to classification, given the nature of these texts. Our conjecture is that the variability in reporting style that we noted above may have made the feature less useful overall.

## 6. Discussion

We would particularly like to point out our final result for the Numeric vs. Text use case, as this is our closest point of comparison with previously published work. For instance, Wang and Hu [10] report a best F of 95.89 for their genuine vs. non-genuine classification task. Cohen et al [1] report a best F of 95.9 for a comparable task. Our accuracy for the numeric (genuine) vs. text (non-genuine) use case is 98.5%, using SVM models.

A significant difference between our financial table tasks and those attempted by these other authors is that our tasks are domain-specific where others have reported their results for the Web in general. Although their results and ours are not head-to-head comparable, at the very least we have proven that the notion of table classification is well founded in general, as it can be meaningfully specialized in particular domains, incorporate financials in this instance.

We were intrigued by the relatively high performance we found for our 15-way classifier, with SVM models getting above 80% accuracy on the eval test. In a standard text classification task, this level of performance for a 15-way classification is generally considered quite high. The restricted domain of corporate financials is not likely enough to account for this by itself: our own experiments with free-text classification in financial reports have not generally reached this level. The further restriction to tables must come into play here.

Another step forward with this work is to look at combining the binary and 8-way classifiers, as the product of their individual performance is suggestively higher than that of the 15-way classifiers.

We hope to pursue further analyses of our comprehensive feature optimization runs. Space considerations preclude giving details, but preliminary analyses suggest that some features are more stable than others, as they tend to consistently align with higher-performing classifiers.

Finally, we hope that other researchers will be drawn to the financial tables classification task, what with its multiple levels of classification and its domain intricacies. As our corpus is drawn from publicly-available materials, we are keen to share it with others, and we look forward to the dialogue that we hope will follow.

## 7. Acknowledgements

This work was internally supported by the MITRE Corporation under the the MITRE Technology Program.

## 8. References

- [1] Cohen, W, Hurst, M, & Jensen, L. (2003): A Flexible Learning System for Wrapping Tables and Lists in HTML Documents. In Antonacopoulos, A, & Hu, J. (eds.) *Web Document Analysis: Challenges and Opportunities*, World Scientific Publishing.
- [2] Grenager, T, Klein, D, & Manning C. 2005. Unsupervised learning of field segmentation models for information extraction. In *Proc. of ACL 2005*, pp. 371-378, Ann Arbor.
- [3] Hurst, M. 2001. Layout and language: Challenges for table understanding on the web. In *Proc. 1st Intl. Wkshp. on Web Document Analysis*, pp. 27-30, Seattle, WA.
- [4] Joachims, T. 2002. *Learning to Classify Text using Support Vector Machines*, Kluwer, 2002
- [5] Knoblock, C, Lerman, K, Minton, S, & Muslea, I. 2003. Accurately and reliably extracting data from the web: A machine learning approach, In Szczepaniak, P, Segovia, J, Kacprzyk, J, & Zadeh, L (eds.) *Intelligent Exploration of the Web*, Springer-Verlag.
- [6] Kushmerick, N, Weld D, & Doorenbos, R. 1997. Wrapper induction for information extraction, In *Proc. of IJCAI-97*.
- [7] Lerman, K, Getoot, L, Minton, S, & Knoblock, C. 2004. Using the structure of web sites for automatic segmentation of tables. In *Proc. of ACM SIG on Management of Data (SIGMOD-2004)*.
- [8] Pinto, D, McCallum, A, Wei, X, & Croft, W. B. 2003. Table extraction using conditional random fields. In *Proceedings of the 2003 ACM SIGIR Conference*.
- [9] Sutton, C, McCallum, A. 2005. Composition of CRFs for transfer learning. In *Proc. of EMNLP/HLT 2005*, Vancouver.
- [10] Wang, Y. and Hu, J. 2002 A machine learning based approach for table detection on the Web. In *Proc. of the WWW 2002 Conference*.
- [11] Wu, D.K. and Lee, K.W.K. 2006. A grammatical approach to understanding textual tables using two-dimensional SCFGs. In *Proceedings of ACL 2006*, Sydney.
- [12] Yoshida, M, Torisawa, K, & Tsujii, J. 2001. A method to integrate tables of the world wide web. In *Proc. 1st Intl. Wkshp. on Web Document Analysis*, pp. 31-34.

# A Knowledge-Light Approach to Query Translation in Cross-Language Information Retrieval

Jesús Vilares  
Dept. of Computer Science  
University of A Coruña  
Campus de Elviña s/n  
15071 – A Coruña (Spain)  
jvilares@udc.es

Michael P. Oakes  
School of Computing & Technology  
University of Sunderland  
St. Peter's Campus  
Sunderland – SR6 0DD (UK)  
Michael.Oakes@sunderland.ac.uk

Manuel Vilares  
Dept. of Computer Science  
University of Vigo  
Campus As Lagoas s/n  
32004 – Ourense (Spain)  
vilares@uvigo.es

## Abstract

This paper describes a new knowledge-light approach for query translation in Cross-Language Information Retrieval systems. This work has been inspired by previous work of the Johns Hopkins University Applied Physics Lab, preserving its advantages but avoiding its main drawbacks. Our work is also based on the direct translation of character  $n$ -grams, avoiding in this way the need for word normalization during indexing or translation, and also dealing with out-of-vocabulary words. Moreover, since such a solution does not rely on language-specific processing, it can be used with languages of very different natures even when linguistic information and resources are scarce or unavailable. In contrast with the original approach, our proposal is much faster and transparent. Our system has been tested using the CLEF evaluation corpus.

## Keywords

Cross-Language Information Retrieval, character  $n$ -grams, translation algorithms, alignment algorithms, association measures.

## 1 Introduction

Cross-Language Information Retrieval (CLIR) is a particular case of Information Retrieval (IR) where queries and documents are written in different languages. Machine Translation (MT) techniques are thus required for translating the queries into the language of the documents in order to allow the matching. Nevertheless, the needs of CLIR systems are different from those of MT systems [4].

One key characteristic of the translation systems integrated in CLIR applications is that, in contrast with classical MT systems, they do not need to respect the constraints of returning only one translation, and that such a translation must be syntactically correct. Thus many CLIR systems rely on some kind of simpler word-level translation approach for converting the source query into the target language. However, such approaches are sensitive to misspellings, out-of-vocabulary words, the lack of accurate linguistic resources, etc. In order to minimize the impact of these factors, the Johns Hopkins University Applied Physics

Lab (JHU/APL) proposed to go one step further by relaxing those constraints even more. They did not ask for complete translated words, but for character  $n$ -grams [8, 9].

The use of character  $n$ -grams for text conflation in IR offers interesting possibilities, particularly in the case of non-English languages. The use of these subwords provides a surrogate means to normalize word forms without relying on language-specific processing, which can be applied to very different languages, even when linguistic information and resources are scarce or unavailable.

Its use is quite simple, since both queries and documents are just tokenized into their compounding overlapping  $n$ -grams instead of words: the word *tomato*, for example, is split into: *-tom-*, *-oma-*, *-mat-* and *-ato-*. The resulting  $n$ -grams are then processed by the retrieval engine either for indexing or querying.

When extending its use to the case of CLIR, an extra translation phase is needed during querying. A first solution may simply consist of using any of the standard MT techniques usually used in CLIR for translating the source query; next, the output translated query would be split into its compounding  $n$ -grams [8]. However, we can go one step further by employing a direct  $n$ -gram translation algorithm which allows translation not at the word level but at the  $n$ -gram level [9]. This way, we can avoid some of the limitations of classic dictionary-based translation methods, such as the need for word normalization or the inability to handle out-of-vocabulary words. The original direct  $n$ -gram translation approach of the JHU/APL was found to be very slow, making the testing of new developments difficult: it could take several days in the case of working with 5-grams, for example [9].

This paper describes a new direct  $n$ -gram translation system we have developed both to speed up the process and to make the system more transparent. The article is structured as follows. Firstly, Sect. 2 describes our approach. Next, in Sect. 3, our proposal is evaluated. Finally, in Sect. 4, we present our conclusions and future work.

## 2 Description of the system

In contrast with the original system developed by JHU/APL, which relies mainly on ad-hoc resources,

our system has been built using freely available resources when possible in order to minimize effort and to make it more transparent. Instead of the ad-hoc retrieval system employed by the original design [8], we use the open-source retrieval platform TERRIER [1]. This decision was supported by the satisfactory results described in [13] when applying  $n$ -grams using different indexing engines.

The second point of difference with respect to the original approach comes from the translation resources to be used. JHU/APL employed bilingual word-lists extracted from a huge parallel corpus of their own [9]. In our case, the well-known EUROPARL parallel corpus [5] has been used. This corpus was extracted from the proceedings of the European Parliament, containing up to 28 million words per language. It includes versions in 11 European languages: Romance (French, Italian, Spanish, Portuguese), Germanic (English, Dutch, German, Danish, Swedish), Greek and Finnish.

Finally, with respect to the  $n$ -gram translation algorithm itself, it now consists of two phases. In the first phase, the slowest one, the input parallel corpus is aligned at the word-level using the well-known statistical tool GIZA++ [11], obtaining as output the translation probabilities between the different source and target language words. In our case, we have opted for a bidirectional alignment [6] which considers a  $(w_{EN}, w_{SP})$  English-to-Spanish word alignment only if there also exists a corresponding  $(w_{SP}, w_{EN})$  Spanish-to-English alignment. This way the subsequent processing will be focused only on those words whose translation seems less ambiguous. Next, in the second phase,  $n$ -gram translation scores are computed employing statistical association measures [7].

This approach increases the speed of the process by concentrating most of the complexity in the word-level alignment phase. This first step acts as a initial filter, since only those  $n$ -gram pairs corresponding to aligned words will be considered, whereas in the original JHU/APL approach all  $n$ -gram pairs corresponding to aligned paragraphs were considered.

Another advantage of this approach is that the  $n$ -gram alignment process can take as input previously existing lists of aligned words or even bilingual dictionaries, theoretically improving the results.

## 2.1 Word-level alignment using association measures

Our  $n$ -gram alignment algorithm is an extension of the way association measures can be used for creating bilingual word dictionaries taking as input parallel collections aligned at the paragraph level [14]. In this context, given a word pair  $(w_s, w_t)$  — $w_s$  standing for the source language word, and  $w_t$  for its candidate target language translation—, their cooccurrence frequency can be organized in a **contingency table** resulting from a cross-classification of their cooccurrences in the aligned corpus:

$T = w_t$   $T \neq w_t$			
$S = w_s$	$O_{11}$	$O_{12}$	= $R_1$
$S \neq w_s$	$O_{21}$	$O_{22}$	= $R_2$
	= $C_1$		= $C_2$   = $N$

As shown, the first row accounts for those instances where the source language paragraph contains  $w_s$ , while the first column accounts for those instances where the target language paragraph contains  $w_t$ . The cell counts are called the **observed frequencies**:  $O_{11}$ , for example, stands for the number of aligned paragraphs where the source language paragraph contains  $w_s$  and the target language paragraph contains  $w_t$ ;  $O_{12}$  stands for the number of aligned paragraphs where the source language paragraph contains  $w_s$  but the target language paragraph does not contain  $w_t$ ; and so on. The total number of word pairs considered —or **sample size**  $N$ — is the sum of the observed frequencies. The row totals,  $R_1$  and  $R_2$ , and the column totals,  $C_1$  and  $C_2$ , are also called **marginal frequencies** and  $O_{11}$  is called the **joint frequency**.

Once the contingency table has been built, different association measures can be easily calculated for each word pair. The most promising pairs, those with the highest association measures, are stored in the bilingual dictionary.

## 2.2 Adaptations for $n$ -gram-level alignment

We have described how to compute and use association measures for generating bilingual word dictionaries from parallel corpora. However, we do not start with aligned paragraphs composed of words, but aligned words —previously aligned through GIZA++— composed of character  $n$ -grams. A first choice could be just to adapt the contingency table to this context, by considering that we are managing  $n$ -gram pairs  $(g_s, g_t)$  cooccurring in aligned words instead of word pairs  $(w_s, w_t)$  cooccurring in aligned paragraphs. So, contingency tables should be adapted accordingly:  $O_{11}$ , for example, should be re-formulated as the number of aligned word pairs where the source language word contains  $n$ -gram  $g_s$  and the target language word contains  $n$ -gram  $g_t$ .

This solution seems logical, but is not completely accurate. In the case of aligned paragraphs, we had **real** instances of word cooccurrences at the paragraphs aligned. However, now we do not have **real** instances of  $n$ -gram cooccurrences at aligned words, but just **probable** ones, since GIZA++ uses a statistical alignment model which computes a translation probability for each cooccurring word pair [11]. So, the same word may be aligned with several translation candidates, each one with a given probability. Taking as example the case of the English words *milk* and *milky*, and the Spanish words *leche* (*milk*), *lechoso* (*milky*) and *tomate* (*tomato*), a possible output word-level alignment —with its corresponding probabilities— would be:

source word	candidate translation	prob.
milk	leche	0.98
milky	lechoso	0.92
milk	tomate	0.15

By considering the overlapping 4-grams that compose each word, we would obtain an alignment like this:

source word	candidate translation	prob.
-milk-	-lech- -eche-	0.98
-milk- -ilky-	-lech- -echo- -chos- -hoso-	0.92
-milk-	-toma- -omat- -mate-	0.15

This way, it may be considered that the source 4-gram **-milk-** does not **really** cooccur with the target 4-gram **-lech-**, since the alignment between its containing words **milk** and **leche**, and **milky** and **lechoso** is not certain. Nevertheless, it seems much more probable that the "translation" of **-milk-** is **-lech-** rather than **-toma-**, since the probability of the alignment of their containing words —**milk** and **tomate**— is much lower than that of the words containing **-milk-** and **-lech-** —the pairs **milk** and **leche** and **milky** and **lechoso**. Taking this idea as a basis, our proposal consists of weighting the likelihood of a cooccurrence according to the probability of its containing word alignments.

So, the resulting contingency tables corresponding to the  $n$ -gram pairs (**-milk-**, **-lech-**) and (**-milk-**, **-toma-**) are as follows:

$T = \text{-lech-}$   $T \neq \text{-lech-}$			
$S = \text{-milk-}$	$O_{11} = 1.90$	$O_{12} = 4.19$	$R_1 = 6.09$
$S \neq \text{-milk-}$	$O_{21} = 0.92$	$O_{22} = 2.76$	$R_2 = 3.68$
$C_1 = 2.82$   $C_2 = 6.95$   $N = 9.77$			
$T = \text{-toma-}$   $T \neq \text{-toma-}$			
$S = \text{-milk-}$	$O_{11} = 0.15$	$O_{12} = 5.94$	$R_1 = 6.09$
$S \neq \text{-milk-}$	$O_{21} = 0$	$O_{22} = 3.68$	$R_2 = 3.68$
$C_1 = 0.15$   $C_2 = 9.62$   $N = 9.77$			

Notice that, for example, the  $O_{11}$  frequency corresponding to (**-milk-**, **-lech-**) is not 2 as might be expected, but 1.90. This is because the pair appears in two word alignments —**milk-leche** and **milky-lechoso**—, but each cooccurrence in an alignment has been weighted according to its translation probability:

$$O_{11} = 0.98 \text{ (for milk-leche)} + 0.92 \text{ (for milky-lechoso)} = 1.90 .$$

In the case of the  $O_{12}$  frequency, it corresponds to  $n$ -gram pairs (**-milk-**,  $x$ ), with  $x$  different from **-lech-**. In our example, we find: a single pair (**-milk-**, **-eche-**) in the word alignment **milk-leche**; three pairs (**-milk-**, **-echo-**), (**-milk-**, **-chos-**) and (**-milk-**, **-hoso-**) in **milky-lechoso**; and three pairs (**-milk-**, **-toma-**), (**-milk-**, **-omat-**) and (**-milk-**, **-mate-**) in **milk-tomate**. By weighting each occurrence according to the translation probability of its containing word alignment, we obtain:

$$O_{12} = 0.98 \text{ (for milk-leche)} + 3*0.92 \text{ (for milky-lechoso)} + 3*0.15 \text{ (for milk-tomate)} = 4.19 .$$

The rest of the values can be calculated similarly.

Once the contingency tables have been generated, the association measures corresponding to each  $n$ -gram pair can be computed. In contrast with the original JHU/APL approach [8, 9], which used an ad-hoc measure, ours uses three of the most extensively used standard measures: the **Dice coefficient (Dice)**, **mutual information (MI)**, and **log-likelihood (log)**, which are defined by the following equations [7]:

$$Dice(g_s, g_t) = \frac{2O_{11}}{R_1 + C_1} . \quad (1)$$

$$MI(g_s, g_t) = \log \frac{NO_{11}}{R_1 C_1} . \quad (2)$$

$$\text{logl}(g_s, g_t) = 2 \sum_{i,j} O_{ij} \log \frac{NO_{ij}}{R_i C_j} . \quad (3)$$

If using the Dice coefficient, for example, we find that the association measure of the pair (**-milk-**, **-lech-**) —the correct one— is much higher than that of the pair (**-milk-**, **-toma-**) —the wrong one:

$$Dice(\text{-milk-}, \text{-lech-}) = \frac{2*1.90}{6.09+2.82} = 0.43 .$$

$$Dice(\text{-milk-}, \text{-toma-}) = \frac{2*0.15}{6.09+0.15} = 0.05 .$$

Notice that if we consider that a real existing cooccurrence instance corresponds to a 100% probability, we can think about the original word-based algorithm described in Sect. 2.1 as a particular case of the generalized  $n$ -gram-based algorithm we have proposed here with  $n=\infty$ .

### 3 Evaluation

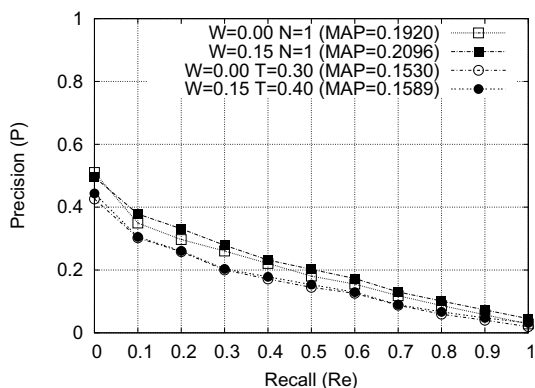
Before trying with less well-known languages with a greater lack of resources —which are the aim of this approach—, our system has to be tuned. For this purpose, our approach has been initially tested in English-to-Spanish bilingual runs using the English topics and the Spanish document collection of the CLEF 2006 **robust task** [2].<sup>1</sup> The Spanish data collection is formed by 454,045 news reports (1.06 GB), while the test set consists of 160 topics (C041–C200) divided into two subsets: a **training topics** subset to be used for training and tuning purposes and formed by 60 topics (C050–C059, C070–C079, C100–C109, C120–C129, C150–159, C180–189), and a **test topics** subset for testing purposes and formed by the 100 remaining topics. Since the goal of these experiments is the tuning and better understanding of the behavior of our system, we will only use the **training topics** subset.

These topics are formed by three fields: a brief **title** statement, a one-sentence **description**, and a more complex **narrative** specifying the relevance assessment criteria. However, only the **title** and **description** fields were used, to simulate the case of the "short" queries typically used in commercial engines [10].

During indexing, documents were lowercased and punctuation marks —but not diacritics— were removed. Finally, the texts were split into  $n$ -grams and indexed, using 4-grams as a compromise  $n$ -gram size after studying the previous results of the JHU/APL group [9]. The open-source TERRIER platform [1] has been employed as the retrieval engine, using a InL2<sup>2</sup> ranking model [3]. No stopword removal or query expansion were applied at this point.

<sup>1</sup> These experiments must be considered as *unofficial* experiments, since the results obtained have not been checked by the CLEF organization.

<sup>2</sup> Inverse Document Frequency model with Laplace after-effect and normalization 2.



**Fig. 1:** Summary precision vs. recall graph of the test runs performed using the Dice coefficient

For querying, the source language topic is firstly split into  $n$ -grams. Next, these  $n$ -grams are replaced by their candidate translations according to a selection algorithm, and the resulting translated topics are then submitted to the retrieval system. Two selection algorithms are currently available: a *top-rank-based* algorithm, that takes the  $N$  highest ranked  $n$ -gram alignments according to their association measure, and a *threshold-based* algorithm, that takes those alignments whose association measure is greater or equal than a threshold  $T$ .

Next, we present the results obtained with the association measures currently implemented in our system: the Dice coefficient, mutual information and log-likelihood.

### 3.1 Results using the Dice coefficient

Our first tests with the Dice coefficient used the top-rank-based selection algorithm, that is, by taking the target  $n$ -grams from the  $N$  top  $n$ -gram-level alignments with the highest association measures.<sup>3</sup> The best results were obtained when using a limited number of translations, those obtained with  $N=1$  being the best ones. Such results are displayed in the precision vs. recall graph of Fig. 1, labeled as 'W=0.00 N=1' — notice that mean average precision (MAP) values are also given.

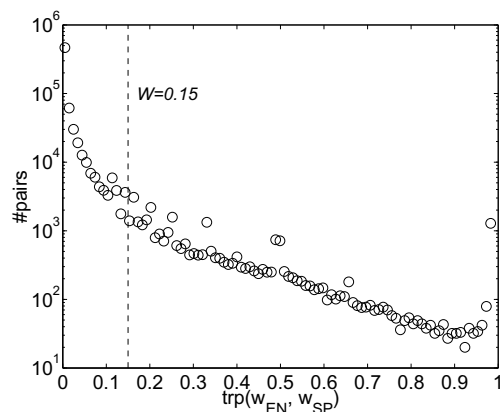
The next tests were made using the threshold-based selection algorithm, that is, by fixing a minimal association measure threshold  $T$ .<sup>4</sup> The best run, using  $T=0.30$ , is shown in the graph of Fig. 1 labeled as 'W=0.00 T=0.30'. As can be seen, the results obtained were significantly less good as the previous ones.<sup>5</sup>

After this initial set of experiments, we studied how to improve the  $n$ -gram alignment by reducing the noise introduced in the system by word-level translation ambiguities. In order to do this, we opted for removing from the input those least-probable word alignments. After studying the distribution of the input aligned word pairs across their translation probabilities

<sup>3</sup> With  $N \in \{1, 2, 3, 5, 10, 20, 30, 40, 50, 75, 100\}$ .

<sup>4</sup> With  $T \in \{0.00, 0.001, 0.01, 0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1.00\}$ .

<sup>5</sup> Two-tailed T-tests over MAPs with  $\alpha=0.05$  have been used along this work.



**Fig. 2:** Distribution of the original set of input aligned word pairs across their translation probabilities (trp)

	$W=0.00$	$W=0.15$	$\% \Delta$
#pairs	672,502	32,011	-95%
$\mu$	0.0287	0.3489	+1116%
$\sigma$	0.0887	0.2116	+139%

**Table 1:** General statistics of the distribution of the input aligned word pairs across their translation probabilities before ( $W=0.00$ ) and after ( $W=0.15$ ) pruning. Column  $\% \Delta$  shows the degree of variation

ties —shown in Fig. 2—, we decided to dismiss those pairs with a probability less than a threshold  $W=0.15$ . This way we reduced the number of input pairs processed by 95%, from 672,502 to 32,011 —see Table 1—, and by 91% the number of output  $n$ -gram pairs generated, from 6,828,044 to 600,120 —see Table 2. This resulted in a considerable reduction of processing and storage resources, including processing time.

On the other hand, regarding the level of ambiguity in the system, Tables 1 and 3 indicate that this refinement reduced the mean number of possible translations per input source word from 13.7427 translations with a mean probability of 0.0287, to 1.1336 translations with a probability of 0.3489. So, the mean number of translations in the input was reduced by 92% and their mean probability was increased by 1116%. Consequently, as is shown in Tables 2 and 3, the mean number of possible translations for the output  $n$ -grams

	$W=0.00$	$W=0.15$	$\% \Delta$	
#pairs	6,828,044	600,120	-91%	
Dice	$\mu$	0.0133	0.1439	+982%
	$\sigma$	0.0721	0.2252	+212%
MI	$\mu$	-0.1476	5.2094	+3629%
	$\sigma$	4.0581	2.4206	-68%
logl	$\mu$	0.6175	5.2995	+758%
	$\sigma$	3.9134	10.2329	+161%

**Table 2:** General distribution of output aligned  $n$ -gram pairs across their association measures before ( $W=0.00$ ) and after ( $W=0.15$ ) pruning. Column  $\% \Delta$  shows the degree of variation



	input word pairs			output $n$ -gram pairs		
	$W=0.00$	$W=0.15$	% $\Delta$	$W=0.00$	$W=0.15$	% $\Delta$
#terms	48,935	28,238	-42%	33,818	27,932	-17%
$\mu$	13.7427	1.1336	-92%	201.9056	21.4850	-89%
$\sigma$	43.1740	0.3858	-99%	502.7873	50.0478	-90%

**Table 3:** General distribution of source-language terms across their number of possible translations before ( $W=0.00$ ) and after ( $W=0.15$ ) pruning, in the case of the input aligned word pairs (left), and the output aligned  $n$ -gram pairs (right). Columns % $\Delta$  show the degree of variation

was reduced from 201.9056  $n$ -grams with a mean association measure of 0.0133, to 21.4850  $n$ -grams with an association measure of 0.1439, meaning a 89% reduction in the number of possible  $n$ -gram translations and a increase of 982% in their mean association measure.

The results obtained introducing this word-level pruning are not significantly different, in general, to those obtained without pruning, whatever the selection algorithm used. Those best results obtained for each selection approach —with  $N=1$  and  $T=0.40$ — are shown in Fig. 1. As can be seen, the top-rank-based selection algorithm keeps performing significantly better.

So, we can conclude that although the introduction of this minimal word-level probability threshold does not really improve the results, it considerably reduces those computing and storage resources required by the system, justifying its application. It can also be concluded that the system, when using the current configuration, seems to be robust against the noise introduced by the high percentage of low-probability alignments of the input.

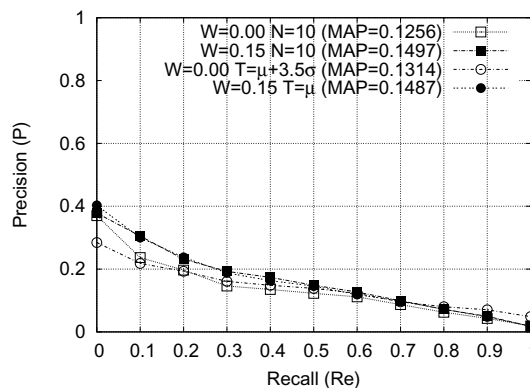
### 3.2 Results using mutual information

Our second series of experiments used mutual information (MI) as the association measure. The main difference with respect to the Dice coefficient is that the Dice coefficient takes values within the range  $[0..1]$ , while MI can take any value within  $(-\infty..+\infty)$ . Negative MI values correspond to pairs of terms avoiding each other, while positive values point out cooccurring terms. MI also tends to overestimate low-frequency data.

These features had to be taken into account in order to adapt our testing methodology. In the case of the top-rank-based selection algorithm, we continued taking the  $N$  top-ranked  $n$ -gram alignments, even if their MI value was negative. However, in the case of the threshold-based algorithm, since the range of MI values for each test run may vary considerably, the threshold values were fixed according to the following formula in order to homogenize the tests:

$$T_i = \mu + 0.5 i \sigma . \quad (4)$$

where  $T_i$  represents the  $i$ -th threshold —with  $i \in \mathbb{Z}$ —  $\mu$  represents the mean of the MI values of the  $n$ -gram pairs obtained for the present configuration, and  $\sigma$  represents their standard deviation. The resulting thresholds are as follows:



**Fig. 3:** Summary precision vs. recall graph of the test runs performed using mutual information

$$\dots \mu - \sigma, \mu - 0.5\sigma, \mu, \mu + 0.5\sigma, \mu + \sigma, \dots$$

In our case we worked only with those possible threshold values from  $T_0 = \mu$  upwards.

The first test run of this series corresponds to the use of the top-rank-based selection algorithm with no word-level pruning —i.e.,  $W=0.00$ . This time, the results obtained were not as good as those obtained using the Dice coefficient. The best run, using  $N=10$ , is presented in Fig. 3.

When introducing the word-level translation probability threshold  $W=0.15$ , the gains were the same as with the Dice coefficient, except for the mean association measure. This is because word-level gains —reduction of input word pairs and increment of the mean translation probability— only depend on the value of  $W$ , and are not affected by the association measure used. At the  $n$ -gram level, the reduction in the number of output  $n$ -gram pairs only depends on the input word pairs and, consequently, on  $W$  again. However, the mean association measures are different, since we are now using MI instead of the Dice coefficient. Mean values are given in Table 2, showing that they increased from -0.1476 to +5.2094, a 3629% improvement.

The results obtained were not significantly different from those obtained with  $W=0.00$ . The best ones, those for  $N=10$ , are shown in Fig. 3. As in the case of the Dice coefficient, the introduction of the threshold  $W$  did not damage the performance of the system, but reduced the computing and storage resources required. On the other hand, the system demonstrated again its robustness against the distortion introduced by low-probability inputs.

When using the threshold-based algorithm, results were slightly better than those obtained with the top-rank-based algorithm —except at the lowest recall levels—, although this difference was not significant. Results improved when raising the threshold, but continued being not as good as those previously obtained with the Dice coefficient. The results for the best run, with  $T = \mu + 3.5\sigma$ , are shown in Fig. 3.

When pruning the input data by applying the word-level probability threshold  $W=0.15$ , the results seemed to approach even more those obtained with the top-rank-based algorithm. As before, no significant difference was found with respect to the results obtained

without pruning. In this case the best threshold was  $T = \mu + \sigma$ , as shown in Fig. 3.

### 3.3 Results using log-likelihood

In our last series of experiments we used the log-likelihood as the association measure. As in the case of MI, it does not have a fixed range of possible values. As before, we will continue taking the  $N$  top-ranked  $n$ -gram alignments in the case of the top-rank-based selection algorithm. Regarding the threshold-based selection algorithm, we will continue fixing the threshold values according to the mean and the standard deviation of the association measure values obtained. Nevertheless, after studying the distribution of the output aligned  $n$ -gram pairs across their log-likelihood values, we realized that the variability of the measures around their mean value was minimal, and that it increased considerably when moving away after overtaking it. So, this time we decided to work with varying granularities, obtaining the following formula for calculating the threshold values:

$$T_i = \begin{cases} \mu + 0.05 i \sigma & -\infty < i \leq 2 \\ \mu + 0.50 (i - 2) \sigma & 2 < i < +\infty \end{cases} \quad (5)$$

where, as before,  $T_i$  represents the  $i$ -th threshold — with  $i \in \mathbb{Z}$ —,  $\mu$  represents the *mean* of the log-likelihood values of the  $n$ -gram pairs obtained for the present configuration, and  $\sigma$  represents their *standard deviation*. This way, the thresholds obtained are as follows:

...  $\mu - 0.05\sigma$ ,  $\mu$ ,  $\mu + 0.05\sigma$ ,  $\mu + 0.1\sigma$ ,  $\mu + 0.5\sigma$ ,  $\mu + \sigma$  ...

As before, the first runs of this last series correspond to those obtained using the top-rank-based selection algorithm. Once again, the pruning of the input word alignments by means of the introduction of a minimal translation probability threshold  $W=0.15$  did not allow us to significantly improve the results obtained — although the mean log-likelihood of the output alignments was improved by 758%, according to Table 2. Nevertheless, it allowed us again, on the one hand, to reduce drastically the resources needed by the system without damaging the performance, and on the other hand, to confirm the robustness of the system against inaccurate or ambiguous input word alignments. The best runs, those for  $N=1$ , are displayed in Fig. 4.

On the other hand, when applying the threshold-based selection algorithm, the results obtained were significantly worse than when using the top-rank-based algorithm, producing the lowest performance of all the association measures tested. In this case, the better results were obtained for  $T = \mu + \sigma$  with  $W=0.00$ , and  $T = \mu + 3\sigma$  with  $W=0.15$ , which can be seen in Fig. 4. As before, no significant difference was found between both runs.

Finally, in order to complete this evaluation, Fig. 5 shows the best results obtained for each association measure compared with several baselines: a monolingual Spanish run obtained by querying the Span-

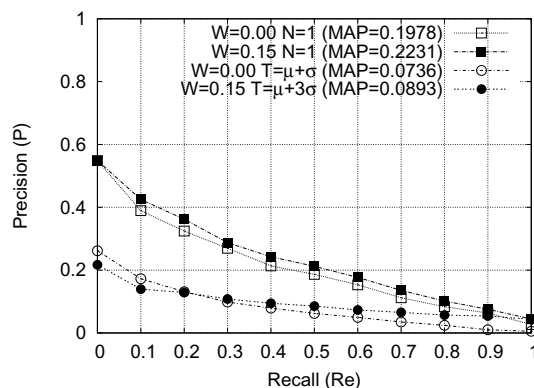


Fig. 4: Summary precision vs. recall graph of the test runs performed using log-likelihood

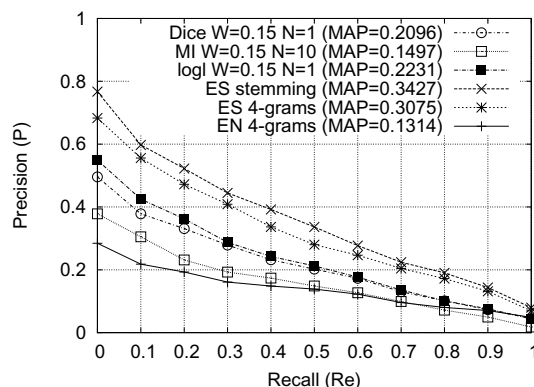


Fig. 5: Final summary precision vs. recall graph

ish index using the stemmed Spanish topics<sup>6</sup> (ES stemming), a second monolingual Spanish run obtained by querying the Spanish index using the Spanish topics split into 4-grams (ES 4-grams) —our ideal performance goal—, and a last run obtained by querying the Spanish index with the English topics split into 4-grams (EN 4-grams) —allowing us to measure the impact of casual matches. As can be seen, log-likelihood measure in combination with the top-rank-based selection algorithm obtained the best results, although no significant difference was found with respect to Dice. On the other hand, both approaches performed significantly better than mutual information.

Although we still need to improve our results in order to reach our ideal performance goal, our current results are encouraging, since it must be taken into account that these are our very first experiments, so the margin for improvement is still great.

## 4 Conclusions and future work

This paper describes an algorithm for character  $n$ -gram-level alignment in a parallel corpus and its use

<sup>6</sup> We have used the Snowball Spanish stemmer (<http://snowball.tartarus.org>), based on Porter's algorithm [12] and one of the most popular stemmers between the IR research community.

for the direct translation of  $n$ -grams during query translation in Cross-Language Information Retrieval tasks. Before trying with less well-known languages with a greater lack of resources, an initial set of experiments has been performed using English-to-Spanish bilingual runs in order to tune the system and to check its behavior.

The alignment algorithm proposed here consists of two phases. In the first phase, the slowest one, the input parallel corpus is aligned bidirectionally at the word-level using a statistical aligner. In the second phase, the association measures existing between the character  $n$ -grams compounding each aligned word pair are computed taking as input the translation probabilities calculated in the previous phase. This solution speeds up the training process, concentrating most of the complexity in the word-level alignment phase, making the testing of new association measures for  $n$ -gram alignment easier. Three of the most widely used association measures are currently implemented in the system: the Dice coefficient, mutual information and log-likelihood. Our experiments have shown that both the log-likelihood and the Dice coefficient outperform mutual information significantly, the former performing slightly better.

For the character  $n$ -gram translation itself, two algorithms for the selection of candidate translations have been also tested: a top-rank-based algorithm, which takes the  $N$  highest ranked  $n$ -gram alignments; and a threshold-based algorithm, which selects the alignments according to a minimal threshold  $T$ . In general, our tests showed the top-rank-based algorithm to be significantly better.

On the other hand, the introduction of a minimal word-level translation probability threshold have allowed us to reduce drastically both the number of input word alignments to be processed, and the number of output  $n$ -gram alignments, but without damaging the performance of the system. This way, we could reduce considerably the computing and storage resources required, including processing time. Moreover, these experiments have demonstrated the robustness of the system against noisy or ambiguous input alignments.

With respect to our future work, new tests with other languages of different characteristics are being prepared in order to complete the tune of the system. We will also focus our effort on the development of new algorithms for the selection of candidate translations, and the application of new association measures.

## Acknowledgments

This research has been partially funded by Ministerio de Educación y Ciencia and FEDER (TIN2004-07246-C03 and HUM2007-66607-C04), Xunta de Galicia (PGIDIT05PXIC30501PN, PGIDIT05SIN044E, *Rede Galega de Procesamento da Linguaxe e Recuperación de Información*, and *Programa de Recursos Humanos* grants), and Universidade da Coruña. The authors would also like to thank Prof. John Tait, from the University of Sunderland, for his support.

## References

- [1] <http://ir.dcs.gla.ac.uk/terrier/> (visited on March 2007).
- [2] <http://www.clef-campaign.org> (visited on March 2007).
- [3] G. Amati and C. J. van Rijsbergen. Probabilistic Models of Information Retrieval Based on Measuring Divergence from Randomness. *ACM Transactions on Information Systems*, 20(4):357–389, 2002.
- [4] G. Grefenstette, editor. *Cross-Language Information Retrieval*, volume 2 of *The Kluwer International Series on Information Retrieval*. Kluwer Academic Publishers, 1998.
- [5] P. Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the 10th Machine Translation Summit (MT Summit X), September 12-16, 2005: Phuket, Thailand*, pages 79–86, 2005. Corpus available in <http://www.iccs.inf.ed.ac.uk/~pkoehn/publications/europarl/> (visited on March 2007).
- [6] P. Koehn, F. J. Och, and D. Marcu. Statistical Phrase-Based Translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA, 2003. Association for Computational Linguistics.
- [7] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge (Massachusetts) and London (England), 1999.
- [8] P. McNamee and J. Mayfield. Character N-Gram Tokenization for European Language Text Retrieval. *Information Retrieval*, 7(1-2):73–97, 2004.
- [9] P. McNamee and J. Mayfield. JHU/APL Experiments in Tokenization and Non-Word Translation. Volume 3237 of *Lecture Notes in Computer Science*, pages 85–97. Springer-Verlag, Berlin-Heidelberg-New York, 2004.
- [10] A. Nardi, C. Peters, and J. L. Vicedo, editors. *Results of the CLEF 2006 Cross-Language System Evaluation Campaign, Working Notes of the CLEF 2006 Workshop, 20-22 September, Alicante, Spain*, 2006. Available at [2].
- [11] F. J. Och and H. Ney. A Systematic Comparison of Various Statistical Alignment Models, 2003. Source code available at <http://www.fjoch.com/GIZA++.html> (visited on March 2007).
- [12] M. F. Porter. An Algorithm for Suffix Stripping. *Program*, 14(3):130–137, 1980.
- [13] J. Savoy. Cross-Language Information Retrieval: Experiments Based on CLEF 2000 Corpora. *Information Processing and Management*, 39:75–115, 2003.
- [14] J. Vilares, M. P. Oakes, and J. I. Tait. CoLesIR at CLEF 2006: Rapid Prototyping of a N-Gram-Based CLIR System. In Nardi et al. [10]. Available at [2].

# Next-Generation Summarization: Contrastive, Focused and Update Summaries

René Witte  
Fakultät für Informatik  
Universität Karlsruhe (TH)  
Karlsruhe, Germany  
witte@ipd.uka.de

Sabine Bergler  
Department of Computer Science  
Concordia University  
Montréal, Canada  
bergler@cs.concordia.ca

## Abstract

Classical multi-document summaries focus on the common topics of a document set and omit distinctive themes particular to a single document—thereby often suppressing precisely that kind of information a user might need for a specific task. This can be avoided through advanced multi-document summaries that take a user’s context and history into account, by delivering focused, contrastive, or update summaries. To facilitate the generation of these different summaries, we propose to generate all types from a single data structure, *topic clusters*, which provide for an abstract representation of a set of documents. Evaluations carried out on five years’ worth of data from the DUC summarization competition prove the feasibility of this approach.

## 1 Introduction

As a much-noticed study attested, *information overload harms concentration more than marijuana*.<sup>1</sup> Internet search engines continue to deliver more and more information to users, when in fact they would rather have less [7]. One approach for mitigating information overload is to *compress* the information delivered from information retrieval (IR) engines through automatic summarization: Instead of displaying a list of relevant documents with keyword-specific highlights, a system can deliver a multi-document summary containing the most important information.

In recent years, extensive experiments on multi-document summarization has been carried out within the Document Understanding Conference (DUC) competition<sup>2</sup> sponsored by the U.S. NIST. In DUC, system developers participate in experiments based on common tasks and data, which allows a comparison of different approaches using various evaluation metrics.

In general, the purpose of a multi-summary is not to serve as a replacement for the real texts, rather, they aim to help a reader to find relevant *topics*. In combination with short (keyword-style) summaries for an individual document, a human reader should be able to quickly determine: (a) whether the set itself contains information on a relevant topic and (b) which of the individual text(s) should be read for an in-depth understanding of the topic.

However, this approach is not the most efficient one when more information is available concerning a user’s

*context* and *history*: Did he read some of the documents in the set before? Then he might only be interested in *updates*, in new information. Is he working on a specific task? Then he primarily needs information pertaining to the task at hand, not a general summary. These scenarios have been addressed in DUC with the introduction of focused and update summaries. In addition, within this paper we propose a third kind, *contrastive* summaries. These are designed for a differential analysis of a document set, showing first the *commonalities* of all texts and additionally the topics that are *unique* to each individual document.

In practice, a user (a.k.a. “knowledge worker”) might need all of these (and other) kinds of summaries while performing knowledge-intensive tasks, ideally embedded within a dynamic, semantic desktop environment that allows for changing the displayed content on-the-fly. Here, the concerns of language system engineers become important due to the growing number of required features. Developing, testing, and deploying individual summarization systems for each of these kinds of summary is not feasible. Thus, we propose a different approach: the generation of an abstracting data structure we call *Topic Clusters*, from which all of these summaries (and some additional) can immediately be generated.

Our research is significant for several reasons: (1) We revive the almost abandoned field of contrastive summarization with a contemporary application focus and a simple, practical approach for generating them; (2) The fact that we investigate automatic summarization for actual deployment within a user’s semantic desktop, deriving requirements that go beyond purely NLP issues by addressing software engineering concerns; and (3) We abstract from the generation of a single type of multi-document summary to arrive at a general data structure that can be used for computing *all* of them.

## 2 Summarization Tasks

As described above, a single kind of summary is not sufficient to adequately cover the information needs of a user performing a particular task. In this section, we motivate and define summaries that go beyond the classical, generic multi-document type.

### 2.1 Contrastive Summaries

Consider a user performing an analysis of a document set, e.g., on the top 50 list of hits delivered by an IR engine for a specific query. To avoid reading all of them, he instructs his semantic desktop to produce a multi-document summary of the whole set, as well as

<sup>1</sup> “Info-overload harms concentration more than marijuana.” *New Scientist*, April 30, 2005, p. 6. <http://www.newscientist.com/channel/being-human/mg18624973.400>.

<sup>2</sup> DUC, <http://duc.nist.gov>

short (ten words, keyword-style) single-document summaries of each text. If he is only interested in the most important topics of the set, this combination will help to detect those, as well as provide cues regarding a good candidate document to read in full. However, if an analysis requires finding the *differences* across the documents in a set, this technique will not work: Since both the multi-document summaries and the individual summaries have to focus on the most important and ubiquitous elements of the texts within the given space constraints, all *distinguishing* information is usually suppressed.

For example, one document set from the DUC 2004 competition contains texts regarding *Hurricane Mitch*. A topic-detecting summary generation algorithm would therefore generate or extract sentences about this natural disaster. Likewise, creating a very short per-document summary results in a similar task: find the most important topic(s) of each text. For the document cluster on Hurricane Mitch, the keywords *Hurricane* or *Central America* are extracted for every text, thus suppressing its distinguishing sub-topics (e.g., concerning EU relief efforts, military rescue operations, or the pope's appeal for aid).

The idea of homing in first on a cluster of multiple documents by their common topic, and then on the particular document of greatest interest within this cluster based on its distinctive topics leads us to propose *contrastive summaries*. We define a contrastive summary of multiple documents as a summary that indicates the common topics of all the articles as well as unique topics of each contained article.

While this idea is not entirely new, none of the current systems makes use of contrastive information. As Mani points out [5], "while similarity across documents is relatively well-understood, differences are not." We believe this is partly due to the lack of a suitable algorithm that can be easily implemented and works robustly even on large document sets (DUC requires summarization of 25–50 documents/set).

## 2.2 Focused Summaries

So far, we addressed the summarization of documents without additional, user-specific information. But in real life, nobody really wants to spend hours on *Google* searching for potentially relevant information. What users need is useful information pertaining to their task at hand, like writing a report, an email, or a research paper. Shouldn't a system be able to sense a user's current *context*, search for relevant information by itself, and present a summary thereof? Coupled with current information retrieval techniques and intelligent information system architectures [10], a new generation of language-aware information systems could proactively deliver the information users need, instead of requiring them to spend their limited time searching for them.

This leads to the idea of a *focused* summary, which only contains information relevant to the user's current context. This kind of summary essentially ignores information that does not contribute to the user's current task—a very useful property when trying to reduce the information overload.

Within the DUC competition, the context is modeled as a set of open-ended questions.<sup>3</sup> Being able

<sup>3</sup> An example for a DUC 2005 focused summary context is: "What countries are or have been involved in land or water boundary disputes with each other over oil resources or exploration? How have disputes been resolved, or towards what

to generate focused summaries has important practical applications for next-generation semantic desktop environments.

## 2.3 Update Summaries

The last kind of next-generation summary we address in this paper are *updates*. Here, the assumption is that a user has already read a number of documents on a certain topic and is only interested in new information that has not been covered before. A typical application scenario are newswire analysts that have to deal with multiple instances of the same or similar stories, as it is evolving over time.

Note that this kind of summary can be combined with both generic, focused, and contrastive summaries. In fact, the DUC 2007 competition defined the update task as a combination of generic updates with a context question, i.e., *focused update summaries*.

## 3 Topic Clusters

To generate contrastive, focused, and update summaries, we introduce a generic data structure that abstracts from individual tokens in a document collection: *topic clusters*. In the next subsection, we motivate this idea, followed by brief description of our approach for topic cluster generation.

### 3.1 Requirements

The target of our research is the individual user facing information overload caused by modern Internet/Intranet (IR) search engines: Rather than displaying a large list of documents with only keyword excerpts, we propose to condense the information contained in the result set through automatic summarization. A user should be able to switch between different kinds of summaries in a dynamic fashion, depending on his current work context and tasks.

From these observations, we can derive three main requirements for a data structure for summary generation:

**Requirement #1: Domain-Independence.** *The algorithm should work independently of an application domain.*

This follows directly from the intended application within a semantic desktop, where the summarization component acts a user's agent when interpreting results from Internet/Intranet searches.

**Requirement #2: Flexibility.** *The data structure needs to be flexible enough to generate all required kinds of summaries: single- vs. multi-document, general vs. focused, contrastive and update.*

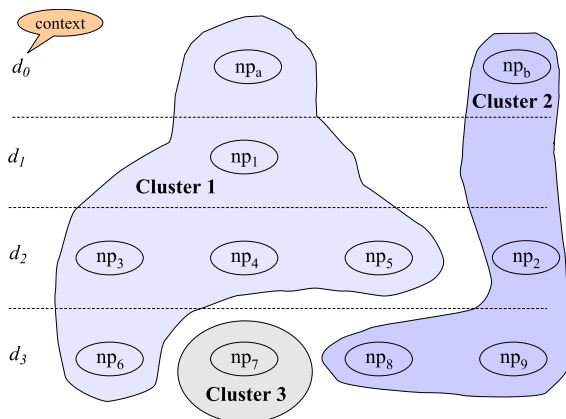
The first reason for this requirement is that a user needs to be able to dynamically switch between different summary views for a given document collection. Moreover, developing, implementing, and maintaining multiple algorithms would be prohibitively expensive from a software engineering perspective.

**Requirement #3: Efficiency.** *The data structure must be abstract enough for summary generation, while at the same time simple enough to be computed in a speedy and robust fashion.*

This requirement ensures the ineligibility of highly sophisticated proposals that are not possible to implement in contemporary desktop environments.

*kind of resolution are the countries moving? What other factors affect the disputes?"*





**Fig. 1:** Topic Clusters for a three-document set  $d_1-d_3$ , plus an additional context document  $d_0$  for focused summarization

### 3.2 Topic Cluster Definition

We now present our approach satisfying the requirements stated in the previous subsection. As we demonstrate below, it helps to abstract from the individual words within a document collection when generating summaries of various kinds. As a useful level of abstraction we use the notion of a *topic*, a particular theme within a set of documents.

We can now define *topic clusters* as an abstract representation of the topics occurring within a document collection. A single topic cluster represents the set of all entities in a document set pertaining to the topic.<sup>4</sup> All topic clusters together (i.e., a topic cluster set) represent the entirety of themes in a document set. Note that each topic cluster has a certain *size*, i.e., the number of contained entities, and spans a certain subset of documents, namely those containing the entities making up the cluster.

An example for topic clusters, generated on a hypothetical four-document set, is shown in Fig. 1. Here, an algorithm detected three topic clusters, using noun phrases (NPs) as the underlying entities. To show a possible implementation for generating topic clusters, we briefly present our algorithm in the next subsection.

### 3.3 Generating Topics Clusters

Our approach for generating topic clusters relies on a fuzzy set theory-based clustering algorithm working on coreference chains. The algorithm is described in detail in [12]. Within this paper, we only present a brief summary of the main steps. The input is a set of documents, as it could have been generated from a search engine. With this set, we perform three steps to obtain a topic cluster, which can then be used for summary generation as shown in Section 4.

**Noun Phrases.** The first step in our approach is the generation of noun phrase (NP) chunks for each document. This can be easily achieved with off-the-shelf part-of-speech taggers and transducer-based NP chunkers, which are available for most languages.

**Coreference Chains.** In a second step, we produce inter- and intra-document coreference chains from the generated NPs. Our approach relies on a fuzzy-set based approach [11], but in principle any coreference algorithm can be deployed for this step.

<sup>4</sup> As a useful entity size we empirically determined base noun phrases (NPs).

**Fuzzy Clustering.** The final step takes the computed inter- and intra-document chains and clusters them with a fuzzy algorithm [12]. The output of this algorithm is a set of NP clusters. Each cluster has a certain size (number of contained NPs) and spans a certain number of documents. Thus, the end result directly corresponds to the topic cluster data structure.

## 4 Summary Generation

In this section, we show how the various multi-document summaries defined in Section 2 can be generated based on topic clusters.

### 4.1 Generic Summaries

We begin by discussing the generation of generic multi-document summaries. Although not part of our list of next-generation summaries, this already illustrates the main points of summary generation based on topic clusters and also provides the foundation for the generation of advanced summary types.

The aim in generic multi-document summarization is to identify the most salient (shared) topics within a collection of documents. The summary, typically sentences (or sentence parts) extracted from the documents, should reflect as many common topics as space permits. This kind of summary can be immediately generated from the topic cluster data structure: Topics are identified by clusters, so by extracting those clusters that span all documents (or a sufficiently large subset thereof), a summarizer can obtain the common themes of all documents. In order to rank the topics by relevance, a summarizer can evaluate the *size* of each cluster: the larger a cluster, the more important the topic contained within.

Based on these ideas, we can define a strategy that generates multi-document summaries by selecting (at least) one candidate noun phrase from each topic cluster, in decreasing order of importance (topic cluster size), until a prescribed size limit has been reached or all topics are exhausted. The candidate NPs, in turn, can be used to select the sentences they appear in as a candidate text extract. Fig. 3 (top section) shows an example for a (roughly) 100-word summary generated with this strategy from DUC 2004 data [8].

Extractive sentence-based summarization typically involves additional techniques, i.e., replacing dangling pronominal references, eliminating duplicate noun phrases, or removing relative clauses. However, within the scope of this paper we are not concerned with this kind of post-processing, which is widely discussed in the literature (see e.g. [5] and the DUC proceedings).

### 4.2 Contrastive Summaries

Contrastive summaries consist of two parts: a summary of the *common themes* across all documents, and document-specific *contrastive themes*. The first part is identical to generic multi-document summarization as described in the previous subsection.

The topic cluster provides all information required for contrast detection: topics that span all documents (or a configurable percentage, e.g., > 90%) are common topics, as they are used for generic multi-document summaries. Topics covered only in a single document (or, again, in a subset, say, < 5%) indicate unique, distinguishing topics. For example, in Fig. 1, Cluster 3 would be such a (single-element) cluster representing a distinguishing topic for document  $d_3$ .

Common clusters	
Hurricane Mitch in Central America (31) – Honduras (21) – the country’s central coast (15) – last week’s storm (12)	
Distinctive clusters	
$D_1$	Gen. Mario Hung Pacheco – the shelves of some stores and some gasoline stations – mayor of Utila – a hurricane warning – the northwest Caribbean for five days
$D_2$	the western Caribbean on Wednesday – 165 kms – Honduras with 120 – west at only 2 mph – a resident of Guanaja Island
$D_3$	the center – emergency measures on the Caribbean coast of the Yucatan Peninsula – a boat – hotels – The storm’s power
$D_4$	the storm’s death toll in the region to 357 – 231 people have been confirmed dead
$D_5$	floods – the Guatemalan border – a state of emergency – 50 kph – late Sunday
$D_6$	area – the slopes of the Casita volcano in northern Nicaragua – Sunday night – a 32-square mile – addition
$D_7$	homes – The greatest losses – affiliate in San Miguel province – a statement – the EU
$D_8$	the audience – all public and private institutions and all men – the pope – a gift – six Russian cosmonauts
$D_9$	access to places – other countries – the recovery effort – More help – at least 300 children at the shelter for diarrhea, conjunctivitis and bacterial infections
$D_{10}$	Taiwan – aid and pledges of assistance – Residents – Cuba’s offer – the saddest thing

Fig. 2: Topic cluster results for a set of ten documents on “Hurricane Mitch”

By sorting these distinguishing clusters by their size, we can obtain a ranked list of topics that are *the most important for a document but not mentioned in any other documents*. Like before, we can select one or several candidate NPs from each cluster (for instance, based on their length or their position within the document) and use those NPs to select sentences for the final output. Of course, for a given document set a topic cluster algorithm might not detect any distinctive clusters. Based on our experiments, this typically happens for very short articles (2–5 sentences), or very homogenous document sets.

As a real-world example, consider the topic cluster generated from the DUC 2004 multi-document set d30002. This set contains ten documents, all on the “Hurricane Mitch” topic, each with slightly different information about the same natural disaster. After running our clustering algorithm, we obtained the topics shown in Fig. 2. The *common clusters* section shows the four biggest clusters (with their respective size in brackets), i.e., the most important topics spanning all documents, each identified by a candidate NP.<sup>5</sup> The *distinctive clusters* section shows the five biggest isolated topic clusters for each document with one selected noun phrase each.

How to present such contrastive summaries to the user is highly dependent on the integration within a desktop environment. In addition to the common topic summary as shown above, we currently give the per-document keywords as shown in Fig. 2, which can be expanded to view a sentence extract, like in Fig. 3.

### 4.3 Focused Summaries

The next type of summary we address here are *focused* summaries, which are not concerned with summarizing a document (set), but rather with collecting information on an explicit interest expressed through context information, like a *user profile*. Focused summaries have been evaluated on a large scale starting with Task 5 in DUC 2004 [8]; in DUC 2005 and 2006, it was the only task (DUC 2007 added the update task).

Topic clusters also allow to generate focused summaries, simply by including the context information as another, distinct document  $d_0$  when computing the topic cluster data structure. Then, all topics that overlap with document  $d_0$  also contain information relevant to the context. All other clusters, even if they are bigger, are discarded for this kind of summary, i.e., we *slice* the topic clusters with the context entities. As before, elements within the clusters have to be further ranked, extracted, and post-processed to create the final summary. Fig. 1 shows an example for this idea: both Cluster 1 and Cluster 2 overlap with the context

<sup>5</sup> Here, we simply used the longest NP, however, a targeted summarizer might apply additional strategies.

Common Topic Summary	
The Honduran president closed schools and public offices on the coast Monday and ordered all air force planes and helicopters to evacuate people from the Islas de la Bahia, a string of small islands off the country’s central coast. National police spokesman Ivan Mejia said the Coco, Segovia and Cruta rivers all overflowed their banks Monday along Honduras’ eastern coast. The European Union on Tuesday approved 6.4 million European currency units (dlrs 7.7 million) in aid for thousands of victims of the devastation caused by Hurricane Mitch in Central America. The greatest losses were reported in Honduras, where an estimated 5,000 people died and 600,000 people – 10 percent of the population – were forced to flee their homes after last week’s storm.	
Distinctive Topics Summaries	
$D_1$	: The head of the Honduran armed forces, Gen. Mario Hung Pacheco, said 5,000 soldiers were standing by to help victims of the storm, but he warned the military could not reach everyone.
$D_2$	: Hurricane Mitch paused in its whirl through the western Caribbean on Wednesday to punish Honduras with 120-mph (205-kph) winds, toppling trees, sweeping away bridges, flooding neighborhoods and killing at least 32 people.
$D_3$	: Hurricane-force winds whirled up to 30 miles (50 kilometers) from the center, with rain-laden tropical storm winds extending well beyond that.
$D_4$	: At least 231 people have been confirmed dead in Honduras from former-hurricane Mitch, bringing the storm’s death toll in the region to 357, the National Emergency Commission said Saturday.
$D_5$	: El Salvador – where 140 people died in flash floods – a state of emergency Saturday, as did Guatemala, where 21 people died when floods swept away their homes.
$D_6$	: Nicaraguan Vice President Enrique Bolanos said Sunday night that between 1,000 and 1,500 people were buried in a 32-square mile (82.88 square-kilometer) area below the slopes of the Casita volcano in northern Nicaragua.
$D_7$	: EU spokesman Pietro Petrucci said the funds will be used to provide basic care such as medicine, food, water sanitation and blankets to thousands of people whose homes were destroyed by torrential rains and mudslides.
$D_8$	: Among those attending the audience were six Russian cosmonauts taking a special course in Italy.
$D_9$	: Aid groups and governments have called for other countries to send medicine, water, canned food, roofing materials and equipment to help deliver supplies.
$D_{10}$	: Taiwan said today it will donate dlrs 2.6 million in relief to Honduras, Nicaragua, El Salvador and Guatemala.

Fig. 3: Topic cluster-generated contrastive multi-document summary

( $d_0$ ), hence they contain information relevant for the user’s focus. Cluster 3 does not overlap with the context and is therefore discarded (even if it were bigger than the other clusters).

To demonstrate that our approach works cross-lingual and cross-domain, we give an example for a focused summary generated from a German encyclopedia on building architecture. Here, we integrated the summarization component into a semantic desktop for building historians and architects [13]. Fig. 5 shows an excerpt of the result to the question “Welche Art von Putz bietet Schutz vor Witterung?” (*Which kind of plaster would be suitable to protect brickwork against weather influences?*). The first sentence of the automatic summarization means: *The joint filling is important for the resistance of the brickwork, especially for those parts exposed to the weather, as well as the quality of the bricks*. Within a concrete application scenario described in [13], this provides helpful information for an architect, who can now continue to access information in the encyclopedia concerning brick quality using the provided cross-links.

“Summarize Microsoft’s antitrust problems, including its alleged illegal behavior and antitrust proceedings against the company.”
Docsubset D0711C-A
Ever since ibm selected Microsoft to supply the operating system for the first ibm pc in 1981, ibm has been out-hustled and outsmarted by Microsoft in the pc business – not the sort of behavior the nation’s antitrust laws were meant to prohibit. Antitrust experts note, it is certainly legal for Microsoft to offer a distributor – as the ibm pc business was – incentives to aggressively distribute its product. If accepting those incentives means featuring Microsoft’s Windows over os/2 – made by another unit of ibm – that is mainly a business decision for ibm to make, they say.
Docsubset D0711C-B
The case is assigned to Jackson, who seeks to expedite the proceedings by setting an early trial date. Microsoft denies all charges of wrongdoing. To Microsoft in 1997, is a harsh critic of the company. Meanwhile, Jackson’s ruling could have an impact on other cases already pending against Microsoft. And Microsoft didn’t get to be Microsoft by shrinking from the battlefield. Tuesday’s courtroom statements – uncompromising attacks from each side – only seemed to underline the gap between the government and Microsoft. In the government’s view, Microsoft is trying to change the subject with such arguments.
Docsubset D0711C-C
The company said it would appeal. Klein said the ruling would bring positive change. He said the ruling showed how no company is above the law. The document was part of the public record from an older antitrust case against Microsoft, the Justice Department’s first suit against the company. The Justice Department and 17 state attorneys general proposed to break Microsoft into two companies. Local press reports said that attorneys for the Justice Department and the 19 states that successfully sued Microsoft for antitrust violations are considering ways to break up the company as a method to curb anticompetitive practices.

Fig. 4: Topic cluster-generated update summary for the DUC 2007 data set D0718D (context shown on top)

“Welche Art von Putz bietet Schutz vor Witterung?”
Ist das Dichten der Fugen für die Erhaltung der Mauerwerke, namentlich an den der Witterung ausgesetzten Stellen, von Wichtigkeit, so ist es nicht minder die Beschaffenheit der Steine selbst. Bei der früher allgemein üblichen Art der gleichzeitigen Ausführung von Verblendung und Hintermauerung war allerdings mannigfach Gelegenheit zur Beschmutzung und Beschädigung der Verblendsteine geboten. Will man einen dauerhaften Putz erzielen, so gilt für alle Arten von Mauerwerk die Regel, da die zu putzenden Flächen frei von Staub sein müssen, da dieser trennend zwischen Mauer und Putz wirken und das feste Anhaften des letzteren verhindern würde. . . .

Fig. 5: Excerpt from a focused summary generated based on a question (shown on top) from a German encyclopedia on architecture

#### 4.4 Update Summaries

To generate update summaries, we first generate the topic clusters based on the context and the current set of documents (including all previous documents, i.e., not just the new ones). For the first subset within an update cluster, summary generation is identical to a standard (main task) focused summary, as presented above. For each subsequent update subset, we re-generate the topic cluster, by adding the new documents to the current set.

To generate focused update summaries for the extended document sets, we again select sentences based on a ranking scheme: (1) The highest rank is given to sentences from clusters that overlap with the context (i.e., cover topics from the questions) but do not contain any elements from documents of a previous update (i.e., these are topical information *only* addressed in a new document). (2) A medium rank is given to sentences from clusters that overlap with the context and appear in the newly added (updated) set of documents (i.e., new information addressing a topic that has been addressed before). And (3) the lowest rank is given to all remaining sentences from clusters that overlap with the context (i.e., answer a question from the context).

In Fig. 1, Cluster 2 is an example for a highly ranked cluster after adding  $d_2$ , because it overlaps with the context ( $d_0$ ) and does not contain elements from a previous update ( $d_1$ ). Thus, the sentences picked from  $d_2$  will contain information regarding the focus question that has not been addressed in a previous document (subset), here,  $d_1$ . Note that generic update summaries (without a focus question) can be generated in the same fashion, by simply omitting the context slicing step.

Fig. 4 shows an example for an update summary generated from DUC 2007 data. Compared to a non-update summary of the same set (not shown here due to space constraints), the update summary clearly

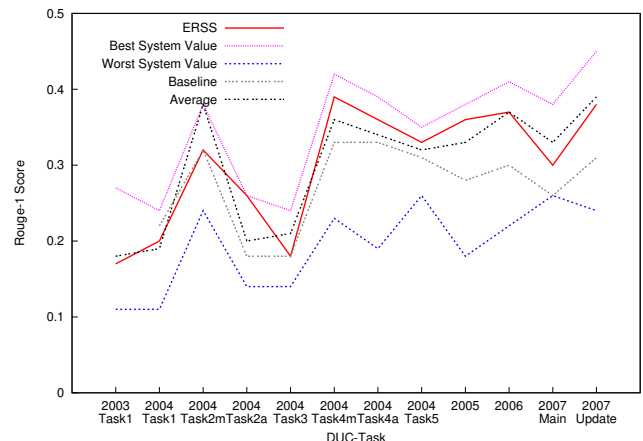


Fig. 6: Summarization based on topic clusters: ERSS performance on the DUC data from 2003–2007

shows the development of the topic through time—before the trial, during the trial, and its aftermath.

## 5 Evaluation

We evaluated our ideas with an implementation based on the fuzzy coreference cluster algorithm [12] for generating topic graphs, using the data of the DUC competitions from 2003–2007. These involved yearly changing tasks, including single-document (DUC 2003, Task 1 and DUC 2004, Task 1) and multi-document summaries (DUC 2004, Task 2), short (keyword) vs. long (sentence) summaries, generic (DUC 2003–2004) and focused summaries (DUC 2004, Task 5; DUC 2005–2006; DUC 2007 Main) as well as cross-lingual (DUC 2004, Tasks 3, 4) and update summaries (DUC 2007 Update). We generated the summaries for all of these different tasks with a single system (ERSS), based on the topic cluster as the only data structure.

We use the same evaluation method as in DUC, namely ROUGE<sup>6</sup> [4], to allow a direct comparison of our results with all other systems participating in DUC. Fig. 6 summarizes the results, comparing our system ERSS with the best, worst, average, and baseline system for each year and task.<sup>7</sup> For the detailed results from each year, we refer to reader to our DUC papers [14].

Overall, we can see that the topic graph algorithm performs very competitively with state-of-the-

<sup>6</sup> In this evaluation, we use the ROUGE-1 score only, to allow a comparison for all years and tasks.

<sup>7</sup> Note that the DUC competition so far included no contrastive summarization task, hence this kind of summary is not included in the evaluation.



art multi-document summarization systems. An analysis of the generated summaries showed that the biggest factor negatively impacting ERSS' scores is the current lack of any post-processing (removing dangling references, cleaning up redundancies, etc.).

## 6 Related Work

Clustering approaches have long been applied to document analysis (see e.g. [1] for an overview), including summarization (e.g., [9]), but our work differs in that we cluster entities (NPs) rather than individual (TF\*IDF-weighted) words.

With respect to contrastive summaries, a motivation related most closely to ours is given by [6] (with previous work in 1997), who also attempt to find both *similarities* and *differences* among related documents. However, Mani [5, p.188] describes this approach as “rather complex” and “recommended only for pairs of documents,” whereas we are concerned with finding contrasts in large document sets (up to 50 for the DUC 2005 data). Also, [6] are not concerned with what we call “contrastive summaries” (as in Fig. 3) but rather present their results in form of sentence extracts aligned between a document pair—which clearly does not help at all in reducing information overload.

In [15], the authors define the problem of “comparative text mining” (CTM) for a given text collection as “(1) discovering the different common themes across all the collections; (2) for each discovered theme, characterize what is in common among all the collections and what is unique to each collection.” They also apply a clustering strategy based on a cross-collection mixture model, but using only simple word-level statistics, which we believe is much less useful for creating summaries than our entity-based clustering approach.<sup>8</sup>

The research area of *change summarization* is concerned with tracking a single document (or a document collection) over time and extracting new/fading topics. [2] evaluate such changes, providing the result in form of web page ranking lists.

## 7 Conclusions and Future Work

In this paper, we investigated several types of multi-document summaries and their generation using a single abstracting data structure, topic clusters.

In particular, we revisited the notion of contrastive summaries, which show, at the same time, both topics *common* to all documents, as well as their *distinctive* information. Although this kind of summary has already been proposed ten years ago by Mani et al. and also alluded to in many other places (e.g., [3]), contrastive summaries are still virtually unknown. We believe this is partly due to the lack of a simple, robust, flexible algorithm, which allows to create this kind of summary from a given document collection. Contrastive summaries are in our view an important contribution to multi-document summarization, especially for less homogeneous collections where the individual documents contain different information only loosely coupled by a common topic. For these collections, a summary of the commonalities does not enable an information seeker to select a relevant document from the collection, and individual summaries are also not guaranteed to highlight the *differences* between the individual documents.

From a language engineering perspective, we essentially decoupled the generation of summaries from the generation of the topic cluster data structure. This allows for both, using different algorithms to compute the graph while keeping the summarization engine intact, as well as using the same data structure for generating multiple kinds of summaries. The evaluation we performed on multiple tasks over five years of data from the DUC competition show that this approach is feasible and delivers competitive performance.

More work is needed in determining efficient ways of integrating automatically created summaries in modern desktop environments. For example, a suitable, dynamic web interface could display topics in a hierarchical fashion, which would allow a user to “see” content that appears in a subset, but not in all documents. Summaries could be incrementally expanded, from keyword sets, like in Fig. 2, to complete summaries, with a single click, allowing a user to navigate from highly compressed views over summaries to the complete document. Creating summaries for dynamically changing document collections—like a newswire stream—can enhance the awareness of newly appearing topics (distinctive clusters) and fading topics. As [7] points out, “*What we find changes who we become.*”

## References

- [1] M. W. Berry. *Survey of Text Mining: Clustering, Classification, and Retrieval*. Springer, 2003.
- [2] A. Jatowt, K. K. Bun, and M. Ishizuka. Change Summarization in Web Collections. In *Innovations in Applied Artificial Intelligence: 17th Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, LNCS, pages 653–662, 2004.
- [3] M.-Y. Kan, K. R. McKeown, and J. L. Klavans. Domain-specific informative and indicative summarization for information retrieval. In *Proc. of the Document Understanding Conference*, New Orleans, U.S.A., 2001.
- [4] C.-Y. Lin. ROUGE: a Package for Automatic Evaluation of Summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, Barcelona, Spain, July 25–26 2004.
- [5] I. Mani. *Automatic Summarization*. John Benjamins B.V., 2001.
- [6] I. Mani and E. Bloedorn. Summarizing Similarities and Differences Among Related Documents. *Inf. Retr.*, 1(1-2):35–67, 1999.
- [7] P. Morville. *Ambient Findability*. O'Reilly, 2005.
- [8] NIST, editor. *DUC 2004 Workshop on Text Summarization*, Boston Park Plaza Hotel and Towers, Boston, USA, May 6–7 2004. <http://duc.nist.gov/pubs.html#2004>.
- [9] D. R. Radev, H. Jing, M. Styś, and D. Tam. Centroid-based summarization of multiple documents. *Inf. Process. Manage.*, 40(6):919–938, 2004.
- [10] R. Witte. An Integration Architecture for User-Centric Document Creation, Retrieval, and Analysis. In *Proceedings of the VLDB Workshop on Information Integration on the Web (IIWeb'04)*, pages 141–144, Toronto, Canada, August 30 2004. [http://rene-witte.net/downloads/witte\\_iiweb04.pdf](http://rene-witte.net/downloads/witte_iiweb04.pdf).
- [11] R. Witte and S. Bergler. Fuzzy Coreference Resolution for Summarization. In *Proceedings of 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS)*, pages 43–50, Venice, Italy, June 23–24 2003. Università Ca' Foscari.
- [12] R. Witte and S. Bergler. Fuzzy clustering for topic analysis and summarization of document collections. In Z. Kolti and D. Wu, editors, *Proc. of the 20th Canadian Conference on Artificial Intelligence (Canadian A.I. 2007)*, LNAI 4509, pages 476–488, Montréal, Québec, Canada, May 28–30 2007. Springer.
- [13] R. Witte, P. Gerlach, M. Joachim, T. Kappler, R. Krestel, and P. Perera. Engineering a Semantic Desktop for Building Historians and Architects. In *Proc. of the Semantic Desktop Workshop at the ISWC*, volume 175 of *CEUR Workshop Proceedings*, pages 138–152, Galway, Ireland, November 6 2005. [http://CEUR-WS.org/Vol-175/34\\_witte\\_engineeringsd\\_final.pdf](http://CEUR-WS.org/Vol-175/34_witte_engineeringsd_final.pdf).
- [14] R. Witte, R. Krestel, and S. Bergler. Generating Update Summaries for DUC 2007. In *Proc. of Document Understanding Workshop (DUC)*, Rochester, NY, USA, April 26–27 2007.
- [15] C. Zhai, A. Velivelli, and B. Yu. A Cross-Collection Mixture Model for Comparative Text Mining. In *Proc. of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'04)*, pages 743–748. ACM Press, 2004.

<sup>8</sup> A typical example cluster in [15] is the topic list “*port, jack, ports, will, your, warm, keep, down*”.

# Task-Dependent Visualization of Coreference Resolution Results

René Witte and Ting Tang  
Fakultät für Informatik

Institut für Programmstrukturen und Datenorganisation (IPD)  
Universität Karlsruhe (TH), Germany  
*witte@ipd.uka.de*

## Abstract

Graphical visualizations of coreference chains support a system developer in analyzing the behavior of a resolution algorithm. In this paper, we state explicit use cases for coreference chain visualizations and show how they can be resolved by transforming chains into other, standardized data formats, namely *Topic Maps* and *Ontologies*.

## 1 Introduction

The computation of coreference chains is an important task in natural language processing. Many high-level text analysis functions rely on coreferences, which makes it important to analyze the results of a particular resolution algorithm. The well-known coreference metrics like MUC [7] or CEAF [4] compute precision and recall values using a gold standard, which allows for a *quantitative* analysis of a system. However, these values only provide a conflated view of the performance; they do not allow for an in-depth analysis of the behavior of an algorithm, e.g., in order to find problematic entities that a coreferencer always “gets wrong.” Especially when developing a rule-based or hybrid coreference resolution system, a *qualitative* analysis becomes important, focusing on individual chains and their entities in order to identify error sources.

Yet the sheer amount of data produced by a coreferencer even on a moderately-sized text makes it infeasible to rely on a tabular or matrix-like representation for understanding an algorithm’s behavior. As humans are much better at analyzing images than numbers,<sup>1</sup> our idea is to transform coreference resolution results into dynamic graphical representations that can be explored and navigated by a user. Furthermore, coreference visualization should adapt to specific tasks, like chain and document navigation, error detection and analysis, or automatic summarization, in order to adequately support a developer.

However, graphical (2D/3D)-visualizations are notoriously difficult and costly to develop. Instead of building our own rendering pipelines from low-level graphical libraries, we investigated a different approach: The

translation of coreference resolution results into standardized data formats, for which a multitude of visualization interfaces exist. This not only allows us to reuse existing graphical tools for NLP, but even permits the application of newly developed visualizations as long as they can read one of the standardized data formats we provide.

## 2 Use Cases for Visualization

Our premise is that a single, generic visualization cannot provide adequate support for the different, varying tasks concerning coreference chains in NLP. Consequently, our approach is to define specific *use cases* based on the work of an NLP system developer, which result in different, task-specific visualizations:

**Chain and Document Navigation.** The visualization should provide for both a quick overview of all created coreference chains (inter- and intra-document), as well as navigational aids to analyze the chain members. Cross-document chain visualizations should additionally provide cues for the document range they span.

**Error Detection and Analysis.** Analyzing the behavior of a coreference resolution algorithm is a major task during system development. A visualization that contrasts computed chains with a manual gold standard should allow a developer to identify “weak spots” in the algorithm’s performance.

**Automatic Summarization.** Automatic summarization is an important application area of coreference chains and clusters. A visualization that shows summaries and their sentences together with the underlying coreferences can help the developer of a summarizer to discern and analyze the connections between a summary and its underlying coreferences.

## 3 Visualization Formats

As mentioned above, our goal is to transform coreference chains into external data formats that are supported by existing visualization tools. In this section, we first examine previous approaches to coreference visualization, and then discuss standard data formats for which suitable graphical tools exist.

<sup>1</sup> See, e.g., [8]: “Combining a computer-based information system with flexible human cognitive capabilities, such as pattern finding, and using a visualization as the interface between the two is far more powerful than an unaided human cognitive process.”

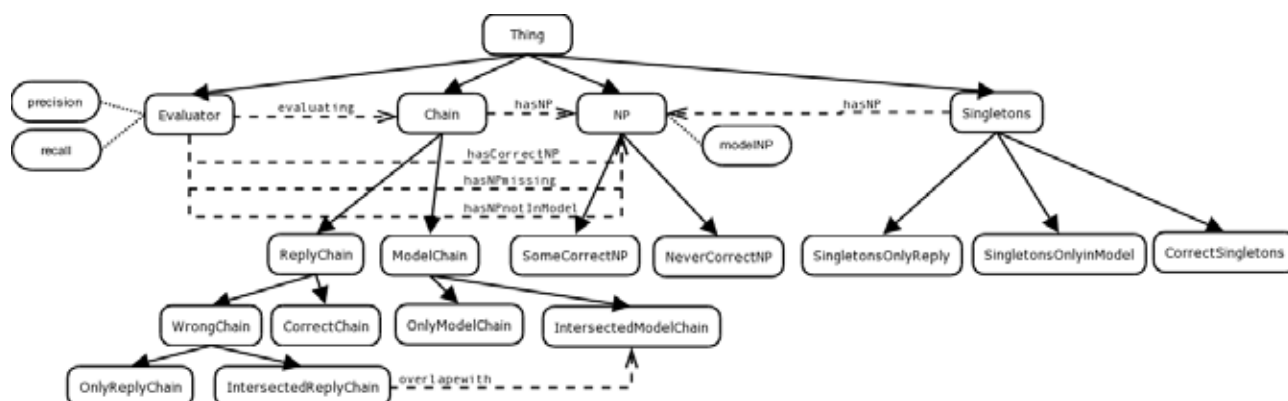


Fig. 1: Visualization ontology for coreference error detection and analysis

### 3.1 Existing Approaches

Little previous work exists on the visualization of coreference resolution results. However, most modern NLP development environments come with graphical user interfaces that are capable of displaying coreference chains as text overlays, e.g., by highlighting or drawing links between entities within a chain. This is also the only approach to coreference visualization discussed in the literature, e.g., within the GATE architecture [1], MMAX [5], or CorefDraw [2].

The main drawback of this approach is that only a part of a coreference chain—for the document text visible within the screen estate—can be viewed. Analyzing larger documents, or cross-document chains, requires permanent scrolling to cover the complete chain, which significantly slows down a developer attempting to gain an overview of all instances within a chain. Moreover, although several chains can potentially be visualized in parallel using e.g. different colors, this quickly becomes too visually complex to be useful.

None of the approaches in the literature suggest task-specific visualization strategies as we defined above.

### 3.2 Standardized Data Formats

We now review two standardized data formats that are expressive enough for visualizing coreference chains, *Topic Maps* and *OWL Ontologies*.

#### 3.2.1 Topic Maps

Topic Maps are an ISO standard<sup>2</sup> for representing knowledge. They have been designed with a particular emphasis on the findability of information, which makes them a promising target for coreference data.

A Topic Map represents information using *topics*, *associations*, and *occurrences*. A *topic* is a concept to represent any kind of entity, like a person or organization. *Associations* define the relationships between topics, while *occurrences* link topics with relevant information resources. Each of these three belong to a certain *Topic Type*, which in turn is a topic itself.

Topic Maps are stored and exchanged in an XML-based data format, XTM (XML Topic Maps).

<sup>2</sup> Topic Maps standard ISO/IEC 13250:2003

**Tool Support.** TM4J<sup>3</sup> is an open source topic map engine implemented in *Java*. It includes the graphical browser TMNav, which can display Topic Maps using different rendering pipelines, included a Swing-based and TouchGraph-based one.

#### 3.2.2 OWL Ontologies

Ontologies are a standard technique for representing domain knowledge, and expressive enough to model our domain of discourse, coreference chains. Formal ontologies based on description logics (DL) have been standardized by the W3C in form of the *Web Ontology Language* (OWL) [6].

**Tool Support.** GrOWL<sup>4</sup> is a visualization and editing tool for OWL. It has been specifically designed for visualizing large ontologies, by allowing a dynamic navigation showing a configurable amount of local context around a node (ABox or TBox). Other tools supporting OWL ontology browsing include Protégé<sup>5</sup> and SWOOP<sup>6</sup>.

#### 3.2.3 Discussion

Both formats have their strength and weaknesses when applied to coreference visualization. Topic Maps are a well-established format and nowadays supported by a whole range of mature visualization tools. In addition, they are easy to generate due to their simple structure. However, the simplicity is also their major downside, as more complex use cases cannot be directly represented using Topic Maps, as we will see below.

Ontologies in OWL-DL format, on the other hand, are much more expressive than Topic Maps, allowing to model complex use cases like coreference evaluation and coreference cluster-based summarization. But the structures expressible in OWL have been designed for machine readability rather than easy visualization for human users. However, the number of robust and scalable ontology visualization tools is steadily increasing, allowing us to upgrade our coreference visualization as they become available.

<sup>3</sup> Topic Maps For Java, <http://tm4j.org/>

<sup>4</sup> GrOWL, <http://ecoinformatics.uvm.edu/dmaps/growl/>

<sup>5</sup> Protégé ontology editor, <http://protege.stanford.edu/>

<sup>6</sup> SWOOP Hypermedia OWL Editor/Browser, <http://www.mindswap.org/2004/SWOOP/>



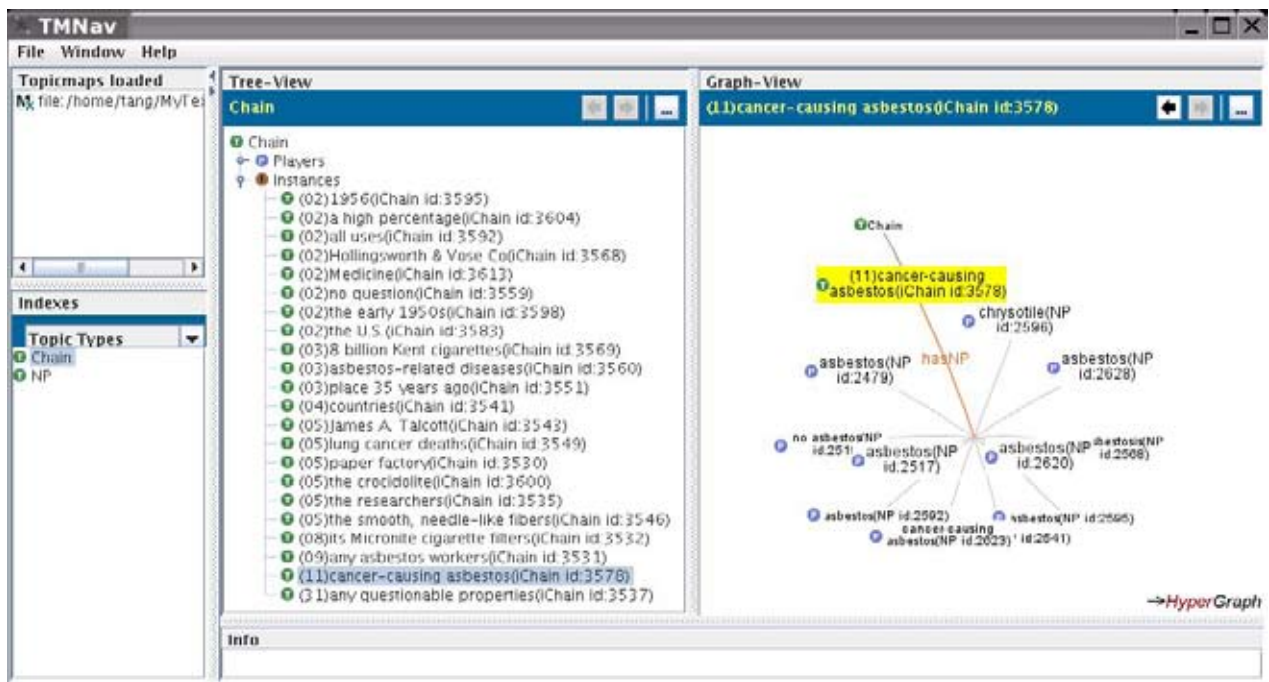


Fig. 2: Visualization of single-document coreference chains as a Topic Map using a HyperGraph renderer

## 4 Coreference Visualization

In this section, we show how to transform coreference chains into the two data formats discussed above, Topic Maps and OWL Ontologies.

We do not assume a particular data format for coreference chains. Within our visualization system, a coreference chain has a unique ID and is represented by a set of noun phrase (NP)<sup>7</sup> ID numbers. Each NP, referenced by its ID, holds meta information like position in the document (start/end) and containing document URI. This representation can be easily created from other formats, like the more implicit MUC style (ID/REF slots on NPs).

### 4.1 Chain and Document Visualization

Our first use case is to provide a visualization for all coreference chains within a document (set). We differentiate between *inter-document chains* that hold entities from a single document only, and *cross-document chains* that reference entities from two or more documents.

#### 4.1.1 Topic Maps

Coreference chains are transformed to the Topic Map format in the following way:

**Topic Type:** Three Topic Types are introduced, *Chain* (coreference chain), *NP* (noun phrase, a chain member), and *hasNP* (NP↔Chain relation).

<sup>7</sup> Although within the scope of this paper we only discuss coreference chains analyzing NPs, the visualization is not restricted to this type of entity. Other grammatical (e.g., VG) or semantical entities (e.g., Organizations, Proteins) can be visualized in the same fashion.

**Topic:** Each NP and each coreference chain becomes reified as a topic of its corresponding topic type.

**Association:** Navigation between chains and their NPs becomes possible through adding an association *is\_instance\_of\_hasNP*, which (unsurprisingly) is an instance of the Topic Type *hasNP*.

For visualizing cross-document chains, additional topic types are added to differentiate *IntraChains* from *InterChains*, as well as *Document* topics. Chains are linked with documents through two additional associations, *isIn* to connect chains with their document, and a *spanning* relation indicating which documents are touched by an inter-document chain.

An example of a generated Topic Map visualized using TMNav can be seen in Fig. 2.

#### 4.1.2 OWL Ontologies

Transformation of coreference chains into OWL ontologies is done in two steps: First, we pre-modeled concepts and their relations in an ontology, which is then populated with instances from a system's results:

**Classes:** Two main classes (TBoxes) are used, *CHAIN* for a coreference chain and *NP* for a noun phrase.

**Properties:** An object property *HASNP* with the domain *CHAIN* and the range *NP* models the connection between chains and NPs.

**Instances:** Each coreference chain becomes an instance (ABox) of class *CHAIN* and each noun phrase an instance of the class *NP*, adding their relations through the object property *HASNP*.

Fig. 4 shows a simple example for this, visualizing all coreference chains within a (single) document. Each of the white boxes represents a coreference chain, described with its id, a text label, and the number of

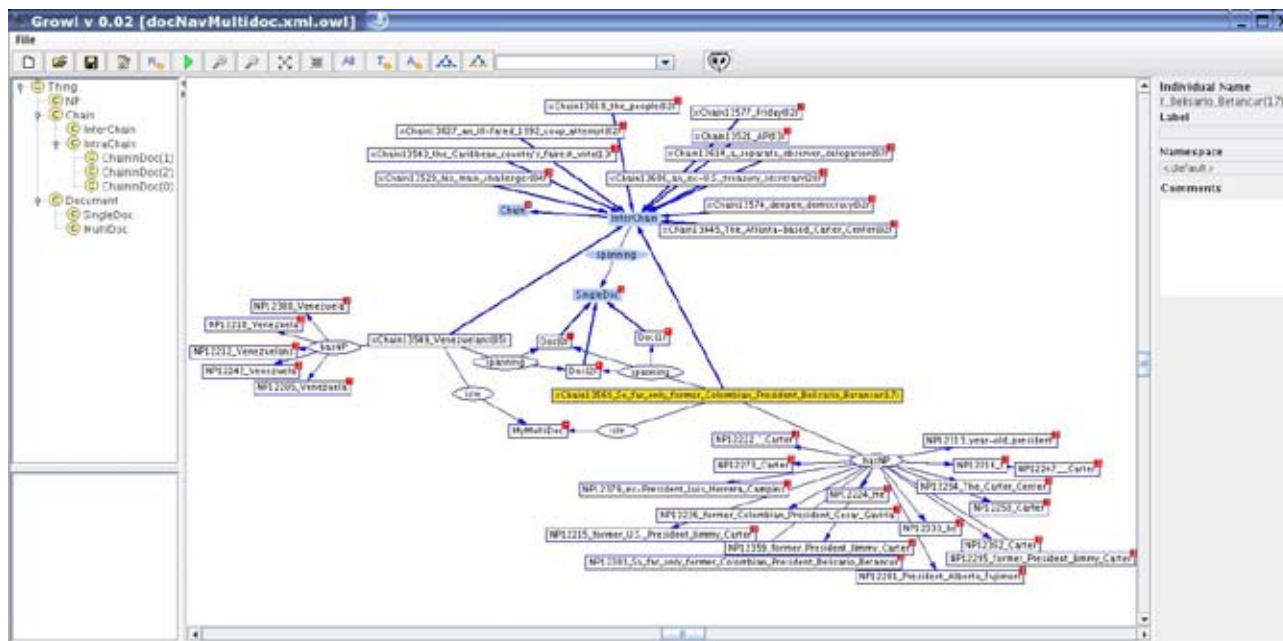


Fig. 3: Visualization of multi-document coreference chains as an ontology using GrOWL

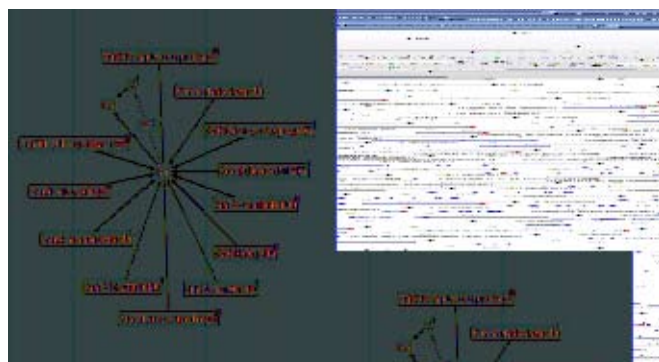


Fig. 4: Ontology-based visualization using GrOWL showing all coreference chains within a document

elements. By clicking on one of these boxes, the node can be expanded to show the chain's elements. The user can then navigate to the sentences and documents containing the chain elements. For multi-documents, the ontology is enhanced and further subclassed (see Fig. 1) to represent the connections between documents and intra-/inter-chains (Fig. 3).

## 4.2 Error Analysis Visualization

For this use case, we only developed an ontology-based visualization, as Topic Maps are not expressive enough to model the complex relationships needed for error analysis, in particular due to the lack of subsumption.

Fig. 1 shows our ontology for error detection and analysis. Besides CHAIN and NP classes we explicitly modeled SINGLETON chains (containing only one NP). In order to analyze coreference results, another entity EVALUATOR is needed that can compute precision/recall values based on a selected metric.<sup>8</sup> Given

<sup>8</sup> Here, we assume that manually annotated NPs correspond

a manually annotated document as gold standard, we can semantically enrich the visualization ontology with additional information: A CORRECTCHAIN is a chain where a MODELCHAIN and a REPLYCHAIN match exactly. Otherwise, the chain is a WRONGCHAIN that can exhibit several kinds of errors: An INTERSECTEDREPLYCHAIN overlaps with at least one MODELCHAIN (and vice versa), whereas an ONLYREPLYCHAIN does not overlap at all with any MODELCHAIN. Since a reply chain can overlap with multiple model chains, the evaluator computes these subsets for each overlapping model/reply combination.

Furthermore, we can push the semantic annotation for error analysis down to individual noun phrases: If an NP is correct with respect to a particular chain, we can tag it as CORRECTNP. Introducing a relation HASCORRECTNP allows a navigation from an evaluated chain to this class of NPs. Likewise, we can tag all NPs missing in a chain for a quick navigation via HASNPNOTINMODEL relation. We can also differentiate between NPs that have been correctly assigned to at least one reply chain, and NPs that are always wrongly assigned. If a NP has not been assigned to a reply chain for any overlapping model chain, we additionally tag it as NEVERCORRECTNP, which is a semantic class of particular interest to a system developer, showing possible serious error sources within a system.

An example for error visualization can be seen in Fig. 5. A user can also navigate starting from the NP class to see all “wrong” NPs, in order to analyze the error case with respect to the various coreference chains computed by the system (missing, additional). Such advanced semantic navigations and visualizations are currently not supported by any other system.

with automatically computed ones. If this is not the case, an additional alignment step has to be introduced, which we do not cover within this paper.

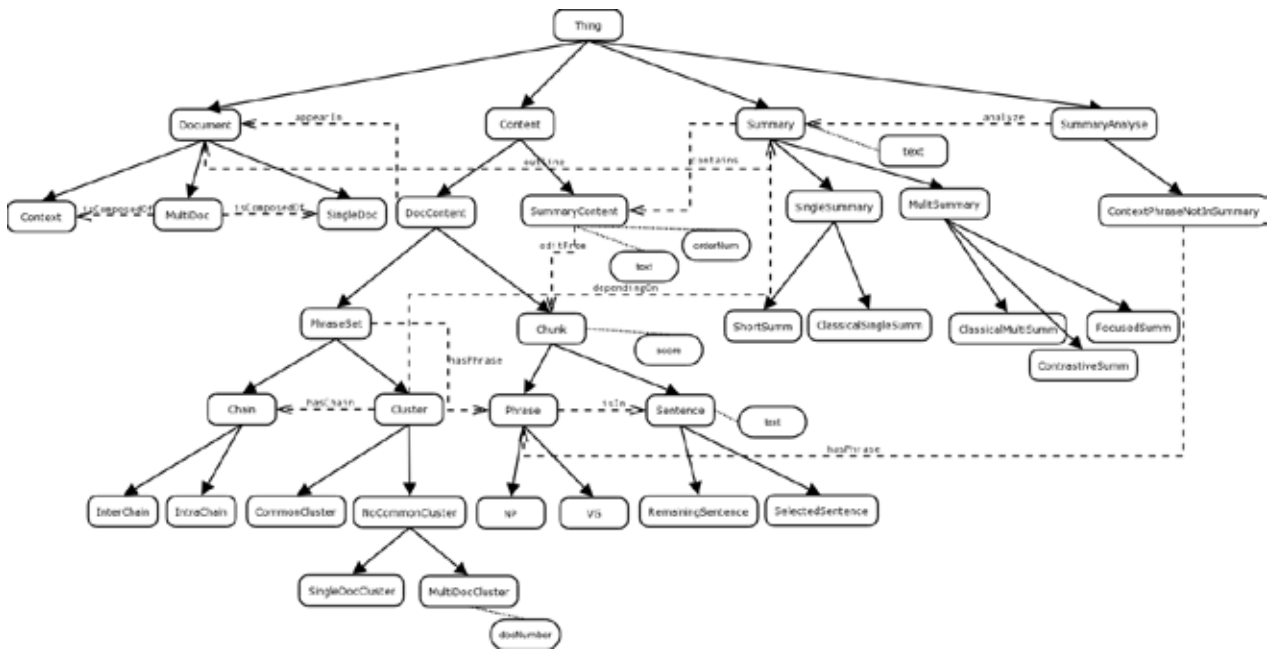


Fig. 6: Visualization ontology for coreference-based summarization generation analysis

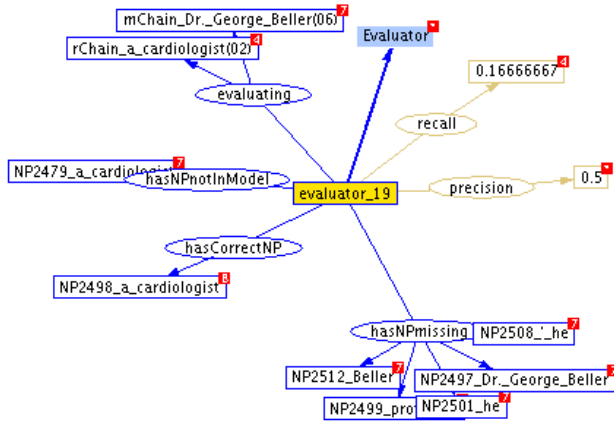


Fig. 5: Coreference chain error analysis using an ontology visualized in GrOWL

## 5 Summarization Visualization

We also investigated the visualization of automatic summaries that have been created based on coreference chains and (multi-document) coreference clusters [10]. For an NLP engineer, finding the interrelationships between generated summaries and their underlying coreference chains is another important task during system development. In our approach, the size of coreference chains and clusters determines the important topics in a document set. Sentences are then extracted and assembled to a summary based on these data structures. When performing a qualitative analysis of a generated summary, it becomes necessary to navigate between entities, chains, and sentences in a summary, analyzing their relationships in order to determine *how* and *why* a particular summary was generated.

By enhancing the ontology shown in Fig. 1 with classes for summaries and their constituents (sentences, NPs, etc., see Fig. 6), we can generate visualizations

for different kinds of summaries, including single- and multi-document, focused vs. generic, update, and contrastive summaries [11].

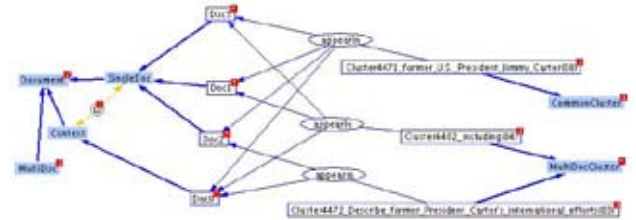


Fig. 7: Visualizing Coreference Chains and Clusters for the Analysis of a Focused Summary

An example is shown in Fig. 7, where coreference clusters (sets of coreference chains) are displayed together with the documents they are spanning. Here, the *context* used for generating the focused summary is shown (Doc0) together with the clusters overlapping with both the context and entities in the various documents. In this example, the engineer starts with the context given for the generation of the focused summary (stored in Doc0) and then examines clusters overlapping with the context. These cluster nodes can then be further expanded to display their related entities, including chains, NPs, and sentences selected for a summary.

## 6 Evaluation

We performed a preliminary evaluation of our work. To determine the impact of the coreference visualization when compared to a text-based output of the chains, we defined a number of tasks typically performed by an NLP engineer during the development and testing of a coreference algorithm. Here, we measure the time it takes a developer to identify certain information (e.g., NPs wrongly assigned to a coreference



chain) with and without our visualizations, in order to assess the impact of our approach. After filtering the resolution results for singletons, the developer had to perform the following tasks:

- Task 1:** Find all correctly computed chains (i.e., reply identical to model chains)
- Task 2:** Find all partially correct reply chains
- Task 3:** Find all reply chains that are completely wrong (i.e., no overlap with any model chain)
- Task 4:** For each partially correct chain from Task 2, identify the missing/superfluous entities with respect to each overlapping model chain
- Task 5:** Identify all those entities that are never correctly resolved (i.e., not part of any (partially) correct reply chain)

These tasks have been performed on a newspaper text containing 140 words, resulting in 54 entities. The manually annotated gold standard contains 44 chains, whereas our resolution system [9] computed 38 chains. Then, one of our group’s language engineers performed the defined tasks both using the plain system output and the developed visualization system. As can be seen in Table 1, the speedup for solving these tasks based on the visualization offers a dramatic improvement when compared to a text-based output.

	Manual	Visual
0: Remove singletons	4:26 min	n/a
1: Correct chains	1:45 min	10s
2: Partially correct chains	2:22 min	10s
3: Completely wrong chains	1:15 min	10s
4: Missing/superfluous chains	6:56 min	50s
5: Entities incorrect for all chains	16:14 min	10s

**Table 1:** Evaluation results comparing tasks performed on a text-based vs. the visualized output

Of course, it would be possible to develop a custom text-based output format for each of these specific tasks. The important point is that our visualization offers a single, interconnected representation to navigate the result space, allowing the NLP developer to dynamically analyze the results of a coreference algorithm from different perspectives.<sup>9</sup>

## 7 Conclusions and Future Work

In this paper, we presented two novel ideas for the visualization of coreference chains: (1) A task-centric approach that focuses on use cases of importance to an NLP system developer and (2) Visualization through transforming coreferences into external, standardized data formats supported by existing graphical interfaces. Our implementation demonstrated the feasibility of this approach. The preliminary evaluation results (as well as the practical experiences in our lab) show a dramatic improvement in analysis capabilities compared to state-of-the-art representations of coreference chains.

<sup>9</sup> See, e.g., [3]: “Compared with an informationally-equivalent textual description of an information a diagram may allow users to avoid having to explicitly compute information because users can extract information ‘at a glance’.”

The ideas stated here can also be applied to other areas in NLP, where complex structures are generated by analysis components, as our visualization extension to coreference-based summarization demonstrates. A major advantage of our approach is that language technology engineers can focus on building a conceptual model of the application domain and do not need to invest time in building the graphical renderings themselves. In addition to visualization, OWL-DL ontologies are also supported by powerful querying and reasoning tools, which provides for a completely new paradigm for the analysis of NLP results. When employed together with task-specific semantic visualizations, we expect a major impact on the productivity within the NLP development lifecycle.

## References

- [1] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proc. of the 40th Anniversary Meeting of the ACL*, 2002.
- [2] S. M. Harabagiu, R. C. Bunescu, and S. Trausan-Matu. COREFDRAW—A Tool for Annotation and Visualization of Coreference Data. In *ICTAI*, pages 273–279, 2001.
- [3] T. Keller and S.-O. Tergan. Visualizing Knowledge and Information: An Introduction. In *Knowledge and Information Visualization*, pages 1–23, 2005.
- [4] X. Luo. On Coreference Resolution Performance Metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada, October 2005. ACL.
- [5] C. Müller and M. Strube. A Tool for Multi-Level Annotation of Language Data. In *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Seattle, USA, 2001.
- [6] M. K. Smith, C. Welty, and D. L. McGuinness, editors. *OWL Web Ontology Language Guide*. World Wide Web Consortium, 2004.
- [7] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. A model-theoretic coreference scoring scheme. In *MUC6 '95: Proc. of the 6th conf. on Message understanding*, pages 45–52. ACL, 1995.
- [8] C. Ware. Visual Queries: The Foundation of Visual Thinking. In *Knowledge and Information Visualization*, pages 27–35, 2005.
- [9] R. Witte and S. Bergler. Fuzzy Coreference Resolution for Summarization. In *Proceedings of 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS)*, pages 43–50, Venice, Italy, June 23–24 2003. Università Ca’ Foscari.
- [10] R. Witte and S. Bergler. Fuzzy Clustering for Topic Analysis and Summarization of Document Collections. In Z. Kobti and D. Wu, editors, *Proc. of the 20th Canadian Conference on Artificial Intelligence (Canadian A.I. 2007)*, Springer LNAI 4509, pages 476–488, Montréal, Québec, Canada, May 28–30 2007.
- [11] R. Witte and S. Bergler. Next-Generation Summarization: Contrastive, Focused, and Update Summaries. In *Proc. of Recent Advances in Natural Language Processing (RANLP-2007)*, Borovets, Bulgaria, September 27–29 2007.

# Predicting the Perceived Quality of Web Forum Posts

Markus Weimer, Iryna Gurevych  
Ubiquitous Knowledge Processing Lab, Telecooperation Division  
Technische Universität Darmstadt, Germany  
<http://www.ukp.informatik.tu-darmstadt.de>  
[[mweimer](mailto:mweimer@tk.informatik.tu-darmstadt.de),[gurevych](mailto:gurevych@tk.informatik.tu-darmstadt.de)]

## Abstract

Assessing the quality of user generated content is an important problem of Web 2.0. Currently, most web sites need their users to rate content manually, which is labour intensive and thus happens rarely. The automatic systems in the literature are limited to one kind or domain of discourse.

We propose a system to assess the quality of user generated discourse *automatically*. Our system learns from human ratings by applying SVM classification based on features such as *Surface, Lexical, Syntactic, Forum specific and Similarity features*.

Our system has also shown to be adaptable to different domains of discourse in our experiments on three different web forum data sets. The system outperformed the majority class baseline for all three data sets. Our best performing system configuration achieves an accuracy of 89.1%, which is significantly higher than the baseline of 61.82%.

## 1 Introduction

User generated content is a significant part of Web 2.0. It is characterized by a low publication threshold and a general lack of editorial control. Content is not created by professionally trained authors, but by ordinary users. We focus on automatic quality assessment of *user generated discourse*, which is textual user generated content. User generated discourse occurs for example in systems like Blogs, Wikis, Forums, and Product Reviews.

The nature of its creation not only leads to huge amounts of user generated discourse being created, but also to a varying quality of the content: Much of it is of great value to users, while many parts of it are of bad quality. Thus, users have problems to navigate through these large repositories of information and find information of high quality quickly.

In order to address the information navigation problem outlined above, many web sites, like

Google Groups<sup>1</sup> and Nabble<sup>2</sup>, have introduced rating mechanisms. Users are asked to rate the content available on the site which has been submitted by other users of the forum. Typically, this rating is expressed on a five-star rating scale. The number of stars corresponds to categories such as *Poor Post* or *Excellent Post*. Table 1 shows the categories as used by Nabble.

User ratings have been shown to be consistent with the user community at large by Lampe and Resnick [2004]. They also showed that user ratings lead to the problem of *premature negative consent*, when combined with filtering based on these ratings. Posts that are once rated to fall below the filtering threshold are not shown to the users anymore. Thus, they can never be rated up again. Additionally, the percentage of manually rated posts is typically very low (about 0.1% in Nabble).

Addressing these issues and departing from pure manual ratings, the main idea explored in the present paper is to investigate the feasibility of automatically assessing the *perceived quality* of user generated discourse, as expressed by the ratings given by the users. The *perceived quality* is not an objective measure. Rather, it models how the community at large perceives quality. We evaluate a machine learning approach to automatically assess it.

The main contributions of the present paper are: (1) A domain-independent system for automatic quality assessment of forum posts that learns from human ratings. Thus, the system adapts itself to new domains of discourse. We evaluate the system on real web forum discussions extracted from Nabble.com. (2) An analysis of the usefulness of different classes of features for the prediction of post quality in different forums.

<sup>1</sup> <http://groups.google.com>

<sup>2</sup> <http://www.nabble.com>



## 2 Related work

**Quality assessment of user generated discourse** is a new field of research and has been addressed only recently by Weimer et al. [2007] in a first case study. The authors present a similar system to the one discussed in this paper. However, they only apply it to one domain of discussion and thus do not reach the broad applicability we focus on.

There has also been some work on automatic assessment of product review usefulness by Kim et al. [2006c]. They test their system on data from Amazon.com, where users can submit reviews of products. These reviews are then rated by other users for their helpfulness, by answering the clear question “Was this review helpful to you?” with the answer choices *Yes/No*. This study found that the dominant features to predict these ratings are the length of the reviews as well as the rating given to the product on a five star scale by the review. Please note that review helpfulness is a rather clearly defined term on the website. This is not the case for post ratings in web forums.

**Automatic essay scoring:** One closely related field is the area of automatic essay scoring (Valenti et al. [2003], Chodorow and Burstein [2004], Attali and Burstein [2006]). There, the goal is to automatically assess the grade of an essay written by students. This seems very similar to what we propose in the present paper. However, there exist well established guidelines that define what a good essay is. Thus, these systems do not need to adapt to the prevalent quality standards of the data they are applied to as our system has to. In web forums, different users cast their rating with possibly different quality criteria in mind.

**Web forum analysis:** Web forums have been in the focus of another track of research, in particular in the context of eLearning. Kim et al. [2006b] found that the relation between a student’s posting behavior and the grade obtained by that student can be predicted automatically. To do so, the number of posts, the average post length and the average number of replies to posts of the student have been shown to be the most important features.

In related research, Feng et al. [2006] describe a system to find the most authoritative answer in a forum thread, based amongst others on the author’s trustworthiness and lexical similarity. Kim et al. [2006a] add speech act analysis as a feature to their system. Finding the most authoritative post in a thread seems to be very closely related

to the task we focus on. However, it is definitely different, as we assess the perceived quality of a given post, currently based solely on its intrinsic features. Any discussion thread may contain an indefinite number of good posts, rather than a single authoritative one.

## 3 Experiments

The system that we propose should be able to adapt to the quality standards existing in a certain user community by learning the relation between a set of features and the perceived quality of posts. We currently employ features from five classes described in Table 2: *Surface, Lexical, Syntactic, Forum specific and Similarity features*.

### 3.1 Data

We evaluated our systems on three data sets extracted from discussions on Nabble.com. Nabble.com hosts forums, but also bridges conventional mailing lists into their system. Forums at Nabble.com are categorized. Analysis of the data showed that most of the rated posts are within the “Software” category.<sup>5</sup> As we seek to develop a system that is applicable to many domains of discussion, we extracted the following three data sets that allow us to assess its performance with that respect: **ALL:** All rated posts in the database. This is the broadest of all data sets. **SOFT:** All rated posts of forums that are in the software category. These are posts that concern closely related. This data set is the same as used by Weimer et al. [2007]. **MISC:** All posts that are in ALL, but not in SOFT. This data set is very diverse in topic, even more so than ALL, as half of ALL are posts from SOFT. Topics range from discussions amongst wikipedia community members to discussions of motor bikes.

At Nabble, posts can be rated by multiple users. Table 1 shows the distribution of average ratings on the five star scale employed by Nabble. From this statistics, it becomes evident that users at Nabble prefer extreme ratings. Therefore, we define the task of predicting the post quality as a binary classification task. Posts with less than three stars are rated as “bad”. Posts with more than three stars are “good”.

We removed the posts, where all ratings are exactly three stars. We also removed the posts that had contradictory ratings from different users. Manual analysis of those posts revealed that they were mostly spam, which was voted high for commercial interest and voted down for being spam.

<sup>5</sup> <http://www.nabble.com/Software-f94.html>

Stars	Label	ALL		SOFT		MISC	
*	Poor	1928	45%	1251	63%	677	29%
**	Below Avg.	120	3%	44	2%	76	3%
***	Average	185	4%	69	4%	116	5%
****	Above Avg	326	8%	183	9%	143	6%
*****	Excellent	1732	40%	421	21%	1311	56%

**Table 1:** *Categories and their usage frequency. Data on the SOFT data set taken from (Weimer et al. [2007]).*

Feature category	Feature name	Description
<b>Surface Features</b>	Length	The number of tokens in a post.
	Question Frequency	The percentage of sentences ending with “?”.
	Exclamation Frequency	The percentage of sentences ending with “!”.
	Capital Word Frequency	The percentage of words in CAPITAL, which is often associated with shouting.
<b>Lexical Features</b>	Spelling Error Frequency	The percentage of words that are not spelled correctly. <sup>3</sup>
	Swear Word Frequency	The percentage of words that are on a list of swear words we compiled from resources like WordNet and Wikipedia <sup>4</sup> , which contains more than eighty words like “asshole”, but also common transcriptions like “f*ckin”.
<b>Syntactic Features</b>		The percentage of part-of-speech tags as defined in the PENN Treebank tag set Marcus et al. [1994]. We used TreeTagger Schmid [1995] based on the english parameter files supplied with it.
<b>Forum specific features</b>	IsHTML	Whether or not a post contains HTML. In our data, this is encoded explicitly, but it can also be determined by regular expressions matching HTML tags.
	IsMail	Whether or not a post has been copied from a mailing list. This is encoded explicitly in our data.
	Quote Fraction	The fraction of characters that are inside quotes of other posts. These quotes are marked explicitly in our data.
	URL and Path Count	The number of URLs and filesystem paths. Post quality in the software domain may be influenced by the amount of tangible information, which is partly captured by these features.
<b>Similarity features</b>		Forums are focussed on a topic. The relatedness of a post to the topic of the forum may influence post quality. We capture this relatedness by the cosine between the posts unigram vector and the unigram vector of the forum.

**Table 2:** *Features used for the automatic quality assessment of posts.*

We also filtered out the posts that did not contain any text, but only attachments like pictures and program files. Finally, we removed non-English posts using a simple heuristics: Posts that contained a certain percentage of words above a pre-defined threshold, which are non-English according to an English dictionary, were considered to be non-English. The upper part of Table 3 shows how many posts were removed from the three data sets. Please note that we did the filtering independently for each filter. Thus, posts that matched several filtering criteria are listed more than once. The lower part of that table shows the distribution of good and bad posts after filtering.

### 3.2 Evaluation procedure

Using the features described in Table 2, we compiled a feature vector for each post. Feature values that were not normalized by definition were

scaled to the range  $[0.0, \dots, 1.0]$ . To classify the posts, we use support vector machines. In particular, we used a C-SVM with a gaussian RBF kernel as implemented by LibSVM in the YALE toolkit (Mierswa et al. [2006]) in all experiments. We did not perform model selection or fine-tuned the parameters of the SVM or the kernel. The parameters were fixed to  $C = 10$  and  $\gamma = 0.1$  for all experiments. We performed stratified ten-fold cross validation for performance evaluation.<sup>6</sup>

Several randomly chosen experiments were repeated using the leave one out evaluation scheme. They yielded comparable results to the ones obtained using cross validation. Thus, we only report the latter in this paper. Please note that it is inherently hard to compare the performance of different machine learning algorithms or algorithm configurations and that statistical signifi-

<sup>6</sup> (See (Bishop [2006]) for an in-depth description.

	ALL		SOFT		MISC	
Unfiltered Posts	4291		1968		2323	
All ratings three stars	135	3%	61	3%	74	3%
Contradictory ratings	70	2%	14	1%	56	2%
No text	56	1%	30	2%	26	1%
Non-English	668	15%	361	18%	307	13%
Remaining	3418	80%	1532	78%	1886	81%
Good Posts	1829	54%	947	62%	1244	66%
Bad Posts	1589	46%	585	38%	642	34%

**Table 3:** Number of posts filtered out in the different data sets.

cance of cross validation performance values can be forged to be arbitrarily high when comparing two algorithms or algorithm configurations (see Witten and Frank [2005], chapter 5.5). Thus, we do not report it.

### 3.3 Experimental Results

Table 4 shows the average cross validation accuracy for all combinations of feature and data sets, whereas we reproduce the results of Weimer et al. [2007] for the SOFT data set. The baseline is based on the majority class. All results but one (SIM/ALL) are equal to or better than the baseline. The usage of all features results in the best or close to best performance for all data sets. The results on the MISC data set are only slightly better than the baseline. The gains on the SOFT and ALL data sets over the baseline are significant. Naively, one may think that the performance on the ALL data set is the average between the performance on MISC and SOFT, as both form approximately one half of the data in ALL. Our results are different, and the performance on ALL is comparable to the performance on SOFT. Thus, the system is able to learn how to classify posts in MISC from posts in SOFT. This leads us to believe that the rating structure in some posts of the MISC data set is very close to the SOFT data set, while the overall rating structure is too diverse to be captured correctly by our system.

The difference in rating structure also shows in the analysis of the best performing feature categories, which are different for each data set. For MISC, the surface features perform best. For SOFT, the forum specific features work best, when only one feature category is used. Weimer et al. [2007] discuss in greater detail, which features from that category have the biggest impact on overall performance. For ALL, two categories share that position: lexical features as well as forum specific features.

It is useful to have a look at the performance

#### ALL:

	true good	true bad	sum
pred. good	1517	456	1973
pred. bad	312	1133	1445
sum	1829	1589	3418

#### SOFT:

	true good	true bad	sum
pred. good	490	72	562
pred. bad	95	875	970
sum	585	947	1532

#### MISC:

	true good	true bad	sum
pred. good	1231	516	1747
pred. bad	13	126	139
sum	1244	642	1886

**Table 5:** Confusion matrix for the system using all features on the three different datasets.

of all other feature categories, when the single best one is not present to assess the influence of the best feature category on the overall performance. For MISC, this leads to a performance on the baseline level. For SOFT, the drop in performance is much smaller, yet still measurable. For ALL, the effects are the smallest, being almost zero for the removal of the lexical features.

### 3.4 Error analysis

Table 5 contains the confusion matrix for the system using all features on the three data sets. The system produces approximately an equal amount of false positives and false negatives on the ALL and SOFT data sets. However, it has a tendency towards false positives on the MISC data set.

Below, we will give descriptions of common errors of our system as well as some examples from

SUF	LEX	SYN	FOR	SIM	ALL	SOFT	MISC
✓	✓	✓	✓	✓	77.53% (1.45)	89.10% (1.44)	71.95% (1.09)
✓	-	-	-	-	64.72% (1.21)	61.82% (1.00)	<b>71.31%</b> (1.08)
-	✓	-	-	-	<b>74.08%</b> (1.38)	71.82% (1.16)	65.96% (1.00)
-	-	✓	-	-	69.18% (1.29)	82.64% (1.34)	66.70% (1.01)
-	-	-	✓	-	<b>74.08%</b> (1.38)	<b>85.05%</b> (1.36)	65.96% (1.00)
-	-	-	-	✓	46.49% (0.87)	62.01% (1.00)	65.96% (1.00)
-	✓	✓	✓	✓	75.92% (1.42)	89.10% (1.44)	66.60% (1.01)
✓	-	✓	✓	✓	<b>77.39%</b> (1.45)	<b>89.36%</b> (1.46)	72.00% (1.09)
✓	✓	-	✓	✓	76.27% (1.43)	85.03% (1.38)	70.03% (1.06)
✓	✓	✓	-	✓	72.82% (1.36)	82.90% (1.34)	71.74% (1.08)
✓	✓	✓	✓	-	76.83% (1.44)	88.97% (1.44)	<b>72.43%</b> (1.10)
Baseline					53.51% (1.00)	61.82% (1.00)	65.96% (1.00)

**Table 4:** Accuracy with different feature sets. *SUF*: Surface, *LEX*: Lexical, *SYN*: Syntax, *FOR*: Forum specific, *SIM*: similarity. The baseline results from a majority class classifier.

the data. We will also provide conclusions on how to improve the current system to overcome the errors. Note that some of the problems were also discussed by Weimer et al. [2007]. We include their analysis, but group it with the errors on the other data sets and discuss means to overcome the limitations of the system.

**Ratings based on domain knowledge:** The following post from the SOFT data set shows no apparent reason to be rated badly. The human rating of this post seems to be dependent on deep domain knowledge, which is currently not present in our system.

```
> Thank You for the fast response, but I'm not
> sure if I understand you right. INTERRUPTS can
> be interrupted (by other interrupts or signals) and
> SIGNALS not.
```

```
Yup. And I responded faster than my brain could
shift gears and got my INTERRUPT and SIGNAL crossed.
```

```
> All my questions still remain!
```

```
Believe J"org addressed everything in full. That the
compiler simply can't know that other routines have
left _zero_reg_ alone and the compiler expects to
find zero there.
```

```
As for SREG, no telling what another routine was
doing with the status bits so it too has to be saved
and restored before any of its contents possibly get
modified. CISC CPUs do this for you when stacking
the IRQ, and on RTI.
```

**Automatically generated mails:** Sometimes, automatically generated mails like error messages end up on the mailing lists. These mails can be written very nicely and are thus misclassified by our system as good posts, while they are bad posts from the point of view of the users. One could deal with these posts by integrating features of the sender of the message, as they originate from addresses like `postmaster@domain.com`.

**Non-textual content:** Especially the SOFT data set contains posts that mainly consist of non-textual parts like source code, digital signatures and log messages from programs. This content

confuses our system to misclassify these posts as bad posts.

To overcome this problem, the non-textual parts need to be marked. They can then be ignored in the quality assessment of the textual content. Additionally, the presence and the amount of non-textual content can be used as an additional feature.

**Very short posts:** Posts which contain only a few words show up as false positives and false negatives equally, as for example a simple “yes” from the master of a certain field might be regarded as a very good post, while a short insult in another forum might be regarded as a very bad post. Domain knowledge from external sources might be helpful in rating these posts.

**Opinion based ratings:** Some ratings do not rate the *quality* of a post, but the *expressed opinion*. In these cases, the rating is an alternative to posting a reply to the message saying “I do not agree with you”.

Take for example the following post which is part of a discussion amongst Wikipedia community members from the MISC data which has been misclassified as a bad post:

```
> But you would impose US law even in a country where
> smoking weed is legal
Given that most of our users and most significant
press coverage is American, yes. That is why I drew
the line there.
Yes, I know it isn't perfect. But it's better than
anything else I've seen.
```

Such posts form a hard challenge for automatic systems. However, they may also form the upper bound for this task: Humans are unlikely to predict these ratings correctly without additional knowledge about the rater.

**Posts that could be rated based on the reply structure:** Most of the posts discussed

above could be classified correctly if the replies to them provided some cues to the quality of the post. The attractive property of integrating features of the replies into the features of a post is that it is domain independent. For example, the simple presence or absence of replies could be part of the perceived quality of a post.

## 4 Conclusions and future work

Assessing post quality is an important problem for web forums. Currently, most forums need their users to rate the posts manually, which is labour intensive and thus happens rarely.

We presented a system and evaluated it on different data sets from different domains of discussion. Our system has shown to be able to assess the quality of forum posts from very diverse discussion domains. The system applies SVM classification using features such as *Surface, Lexical, Syntactic, Forum specific and Similarity features* to do so. We evaluated our system on three data sets and it performed very well on two of them, while only slightly better than the baseline on the third, most challenging, one. Our best performing system configuration achieves an accuracy of 89.1%, which is significantly higher than the baseline of 61.82%.

Careful error analysis leads us to several future improvements to our system. First of all, the integration of the discourse structure promises improvements. Additionally, external knowledge sources can help to assess the information content of a post, which can be of influence on the perceived post quality.

After evaluating it on different domains of discussion within the same kind of user generated content, we seek to apply our system to other kinds of user generated discourse. The system can obviously be applied to other web forums, but we also seek to apply it to adjunct areas like blog comments and several kinds of user reviews of movies, products, websites.

We believe that this system will support important applications beyond content filtering like automatic summarization systems and user generated discourse specific search.

## References

- Y. Attali and J. Burstein. Automated essay scoring with e-rater v.2. *The Journal of Technology, Learning, and Assessment*, 4(3), February 2006.
- C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.
- M. Chodorow and J. Burstein. Beyond essay length: Evaluating e-raters performance on toefl essays. Technical report, ETS, 2004.
- D. Feng, E. Shaw, J. Kim, and E. Hovy. Learning to detect conversation focus of threaded discussions. In *Proceedings of HLT-NNACL*, 2006.
- J. Kim, G. Chern, D. Feng, E. Shaw, and E. Hovya. Mining and assessing discussions on the web through speech act analysis. In *Proc. of the Workshop on Web Content Mining with Human Language Technologies at ISWC*, 2006a.
- J. Kim, E. Shaw, D. Feng, C. Beal, and E. Hovy. Modeling and assessing student activities in on-line discussions. In *Proceedings of the Workshop on Educational Data Mining at AAAI*, Boston, MA, 2006b.
- S.-M. Kim, P. Pantel, T. Chklovski, and M. Penneacchiotti. Automatically assessing review helpfulness. In *Proceedings of EMNLP*, 2006c.
- C. Lampe and P. Resnick. Slash(dot) and burn: Distributed moderation in a large online conversation space. In *Proceedings of ACM CHI 2004 Conference on Human Factors in Computing Systems, Vienna Austria*, pages 543–550, 2004.
- M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2): 313–330, 1994.
- I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. YALE: Rapid prototyping for complex data mining tasks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 935–940, New York, NY, USA, 2006. ACM Press.
- H. Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, Manchester, UK, 1995.
- S. Valenti, F. Neri, and A. Cucchiarelli. An overview of current research on automated essay grading. *Journal of Information Technology Education*, 2:319–329, 2003.
- M. Weimer, I. Gurevych, and M. Mühlhäuser. Automatically assessing the post quality in online discussions on software. In *Companion Volume of the 45rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007.
- I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2 edition, 2005.

# A Bayesian approach combining surface clues and linguistic knowledge: Application to the anaphora resolution problem

Davy Weissenbacher, Adeline Nazarenko  
Université Paris-Nord - Laboratoire d'Informatique de Paris-Nord.  
99 av. J-B. Clément 93430 Villetaneuse, FRANCE  
*dw@lipn.univ-paris13.fr, nazarenko@lipn.univ-paris13.fr*

## Abstract

In NLP, one traditionally distinguishes the linguistically-based systems and the knowledge-poor ones which mainly rely on surface clues, but each approach has its drawbacks and its advantages. In this paper, we propose a new method, based on Bayes Networks, that combines both types of information. As a case study, we consider the specific task of pronominal anaphora resolution which is known as a difficult NLP problem. We show that our bayesian system performs better than state-of-the art anaphora resolution ones.

## Keywords

Bayesian Network, anaphora resolution, linguistic knowledge, surface clue

## 1 Introduction

One often opposes knowledge based and knowledge poor Natural Language Processing (NLP) systems. The first ones exploit complex knowledge pieces which are often manually built and not always reliable or available. The second ones, based on machine learning methods, take only surface clues into account and give mitigated results on complex NLP tasks.

We propose to overcome that opposition. Our approach relies on the Bayesian Network formalism, a probabilistic model designed for reasoning on uncertain, and lacking information, which is still little exploited in NLP.

We tested this approach on the resolution of the anaphoric pronoun *it*, which is a complex task involving different types of knowledge. We designed a system that relies on a Bayesian Network for the classification of antecedent candidates and we compare its performances with that of a state-of-the-art system, MARS, proposed by R. Mitkov [8]. MARS can be considered as a knowledge-poor system.

The next section presents the state-of-the-art in anaphoric pronoun resolution and the difference between rich and poor approaches. Section 3 describes the formalism of the Bayesian Networks, its advantages for NLP and our anaphora resolution classifier. In Section 4, we compare the performances of that Bayesian system and several other ones. The last section discusses the results.

## 2 The opposition between linguistic knowledge and surface clues

Anaphora is a linguistic relation that holds between two textual units. One (the *anaphor*) cannot get interpreted as such but refers to the other, which usually occurs before (the *antecedent*). As the presence of anaphors significantly degrades the performances of NLP tasks such as information extraction or text synthesis, a lot of work has been devoted to the automatic resolution of these anaphoric relationships, *i.e.* the identification of the antecedents of anaphoric pronouns. In this paper, we focus on the pronoun *it* in English texts, which is a well-known and frequent type of anaphors.

The traditional approach for anaphora resolution is composed of three steps: the distinction between anaphoric and impersonal occurrences of the pronoun (*it is known that...* vs. *it produced...*), the selection of antecedent candidates and the choice of the most plausible antecedent.

For each of these steps, the first systems relied on complex linguistic knowledge that reflected the deep syntactic and semantic constraints of anaphoric relations. These systems often relied on a set of manually designed rules, which required a thorough corpus analysis. During the 1990's, several systems relying on surface clues were proposed to face the need for robust and less expensive anaphora resolution methods. These systems tried to approximate the complex linguistic rules by simple clues that are presumably more reliable and easier to compute. For instance, the RAP algorithm [7] was simplified in [6] or the co-occurrence frequencies were used to approximate the semantic constraints proposed by [4].

The surface clues proposed during the 1990's enabled to build robust systems [8] but recent work has underlined their limits. Since the predicate-arguments schemata that improve the candidate filtering [9], are seldom available, they have been approximated by concurrence frequencies [4]. However, [2] shows that these frequencies do not really enhance the performances of a system that is already based on morpho-syntactic knowledge. The contribution of frequencies seems to pertain more to hazard than to semantics.

Such a conclusion brings back to the initial problem. Anaphora resolution involves complex syntactic

and semantic knowledge that is not always available and which is often not fully reliable. Previous works have tried to substitute linguistic knowledge by surface clues which are easier to compute and therefore more reliable. However these clues only partially reflect the linguistic constraints and may lead to erroneous decisions, when solving ambiguous cases.

The MARS system [8] relies on surface clues to identify the most salient element in the discourse fragment preceding a pronoun occurrence. This salient element is considered as the most probable pronoun antecedent. The system relies on a part-of-speech tagging (POS tagging) of the text and on some simple grammar rules to list the noun phrases (NPs) of the three sentences preceding a given pronoun occurrence (including the pronoun sentence). For each NP associated to the pronoun occurrence, a set of constraints and preferences is applied. The constraints filter out the impersonal pronoun occurrences and the NPs that cannot be antecedent. The preferences rank the remaining NP candidates. Each preference is associated with a score, either positive or negative, and the various scores of a candidate are summed up in a global score. The antecedent with the highest score is chosen. When two candidates end with the same score, additional heuristics are used to rank them<sup>1</sup>.

We propose a new system that combines the surface clues of MARS with some the linguistic constraints that the surface clues approximate, whenever some linguistic knowledge is available. We argue that combining both types of information is beneficial. For instance, the subject of a sentence is often the most salient element but, since the syntactic role analysis may be erroneous, it is useful to exploit in parallel the information relative to the NP location: the surface clue (the first NP of the sentence is very often the verb subject) corroborates the grammatical role hypothesis.

Our system is modeled as a Bayesian Network. This type of representation has been designed to reason on uncertain and incomplete knowledge. Its probabilistic approach unifies in a single representation deep linguistic constraints and surface clues. This unification allows to corroborate linguistic constraints with the surface properties observed in corpora and to correct the errors made by the systems based on surface clues.

### 3 A unified approach: the Bayesian model

As many other NLP tasks, distinguishing anaphoric and impersonal pronoun occurrences and more generally solving anaphors can be considered as classification problems [3].

A Bayesian Network is composed of a qualitative description of the attribute dependancies, an oriented acyclic graph, and of a quantitative description, a set of conditional probability tables, each random variable (RV) being associated to a graph node. A first parameterising step associates *a priori* conditional prob-

ability tables to each RV. The second inferring step modifies the RV values on the basis of corpus evidence (it updates the *a priori* probabilities into *a posteriori* ones). The observations made in corpus are propagated through the network, which leads to update the *a priori* values even for some unobserved variables.

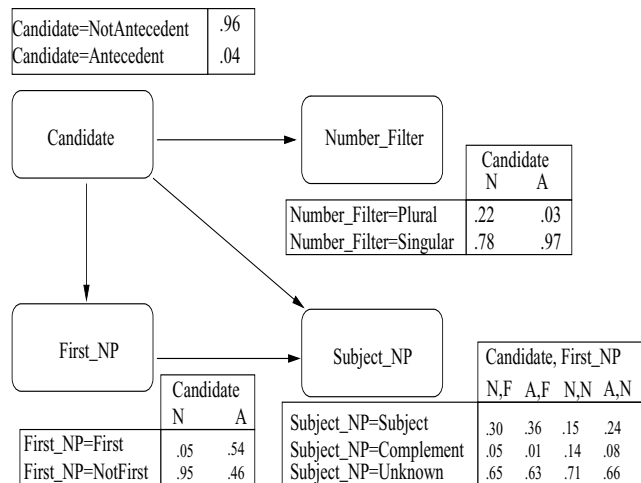


Fig. 1: Example of a Bayesian classifier represented by a Bayesian Network

Let us explain on a simplified example the inferring mechanism of the Bayesian Network represented on Figure 1. This network chooses the pronoun antecedent by ordering the various couples of candidates associated to a pronoun occurrence. The network is composed of 4 nodes, which respectively represent the probability for a candidate to be the antecedent of the pronoun occurrence (Candidate), to have some morphological properties regarding number (Number\_Filter), to be the first NP (First\_NP) or the subject (Subject\_NP) of the sentence.

The first parameterising step computes the *a priori* probability values. These probabilities are estimated on the basis of the frequencies computed on the set of couple examples extracted from a training corpus, for which all the attribute values are instantiated. From these observations, we state for instance that  $P(\text{Candidate}=\text{Antecedent})=0.04$  *i.e.* we consider that any candidate has *a priori* a probability of 4% to be the antecedent of an anaphoric pronoun occurrence.

The influence link between the variables Candidate and Number\_Filter indicates that a candidate is less likely to be plural if it is the antecedent of the pronoun *it* (reversely, it is less likely to be its antecedent if it is a plural noun). Similarly, the links between the variable Candidate and First\_NP on the one hand, Candidate and Subject\_NP on the other hand respectively indicate that the candidate is more likely to be the first NP of the preceding sentence and to be the subject of the verb if it is the pronoun antecedent. The link (First\_NP, Subject\_NP) connects two variables that are considered as dependant on each other on the basis of the training corpus and expert estimation. This means that the reliability of the subject syntactic role is increased if the candidate also occurs at the beginning of a sentence. This interdependency is measured through the table of conditional proba-

<sup>1</sup> The final ranking depends on the types of the preferences that have been used for each candidate and the most recent candidate is chosen, if nothing else applies.

bilities that is associated to the node `Subject_NP` on Figure 1. We also added a value *Unknown* to the RV of the `Subject_NP` node as the syntactic analysis quite often fails to associate a grammatical role to some NPs. This is a way to avoid taking into account incomplete data for the first evaluation of our system.

Once all the *a priori* conditional probabilities have been computed, the inferring step begins. Let's take as an example the couple (*citA transcription*, *it*<sub>1</sub>) extracted from the sentence *In minimal medium, [citA transcription]<sub>1</sub> was about 6-fold lower when glucose was the sole carbon source than [it]<sub>1</sub> was when succinate was the carbon source.* Our system computes the values of the attributes of that couple. The candidate is not a plural NP but it is the first NP of the sentence. Since these observations are very reliable, we can state that  $P(\text{Number\_Filter}=\text{Singular})=1$  and  $P(\text{First\_NP}=\text{First})=1$  (strong evidence). Even if the parser has produced a dependency analysis of that sentence in which the candidate is the subject of the verb, we know that this analysis may be erroneous and we consider that this third observation is only a soft-evidence:  $P(\text{Subject\_NP}=\text{Subject})=0.89$

On the basis of these observations, the probability for the candidate to be the pronoun antecedent can be computed:

$$P(\text{Candidate}=\text{Antecedent} | \text{Number\_Filter}=\text{Singular}, \text{First\_NP}=\text{First}, \text{Subject\_NP}=\text{Subject}) = 0.4$$

Our system similarly computes the probability for any other NP to be the antecedent of the pronoun *it*<sub>1</sub>. If none of the other candidates has a probability higher than 40%, *citA transcription* is considered to be antecedent of the pronoun.

We keep all the attributes of MARS, except the Command constraint that is mostly useful for demonstrative pronoun anaphors (*e.g. this*) and the preferences specifically designed for the technical type of corpora on which MARS has been initially tested<sup>2</sup>. We enrich that list with some additional clues that are relevant for salience calculus and which are used in several other systems described in the state of the art. Each property is modelled as a node in our Bayesian Network (see Figure 2<sup>3</sup>, where MARS attributes and the additional ones are distinguished. They are respectively coloured in black and grey):

## 4 Experiments and results

We have used 6 different classifiers for the anaphora resolution.

Three of them are used as baseline systems: *Random* system randomly chooses the antecedent among the candidate list; *First\_NP* system systematically selects the first NP of the preceding sentence as the pronoun antecedent; the *Bio\_MARS* is our version of Mitkov's MARS system. The solving algorithm of *Bio\_MARS*

is the same as that MARS but our system is specifically designed for genomics. The preprocessing includes the following steps: the NP list is extracted from a full constituent analysis of the corpus that is obtained thanks to a domain specific parser; for identifying the anaphoric occurrences, we exploit a filter that is based on a Bayesian Network and trained on a corpus of the same domain [12]; we rely on an extended and domain specific tagging of named entities and terms.

The three other systems have been designed to test various configurations of the Bayesian model. *NB\_Mars* system exploits the same attributes as *Bio\_MARS* but the final decision is based on a Naive Bayes classifier rather than on a global score. The fourth system is the Bayesian Network classifier itself (*BNC*): the choice of the attributes and the network structure are based on a linguistic analysis of a training corpus. The last system is the Naive Bayesian classifier (*NBC*), which has the same attributes as *BNC* but a simplified tree structure where the attributes are considered as independent of each other.

We tested our systems on a specialised corpus, *Transcript*. It is a collection of 2209 abstracts (around 800,000 words) of scientific papers that have been retrieved by querying the *Medline* bibliographical base with the keywords *bacillus subtilis*, *transcription*[1]. 697 occurrences of *it* have been identified in *Transcript*. Two different annotators have tagged each of these occurrences as either anaphoric or impersonal and have identified the corefering antecedent of the anaphoric pronoun occurrences.

In order to determine the attribute values of each candidate/pronoun couple, we have exploited the *Ogmios* platform [5] to analyse our corpus. *Ogmios* integrates TagEN, a named entity tagger specifically designed for genomics, to identify the biological named entities, and BioLG, a version of Link Grammar Parser adapted for biology [10], for the dependency and constituent syntactic analysis. It also exploits a large specialised terminology. For our first experiments, we have manually built the class of indicative verbs out of our training corpus.

Since our working corpus is relatively small, we have validated our results using a cross validation method. We have randomly selected 2/3 of our corpus to compute the *a priori* conditional probabilities and we have applied the resulting parameterised system to the remaining part of the corpus. We iterated these operations 20 times and we analysed the average performance of each classifier on our corpus.

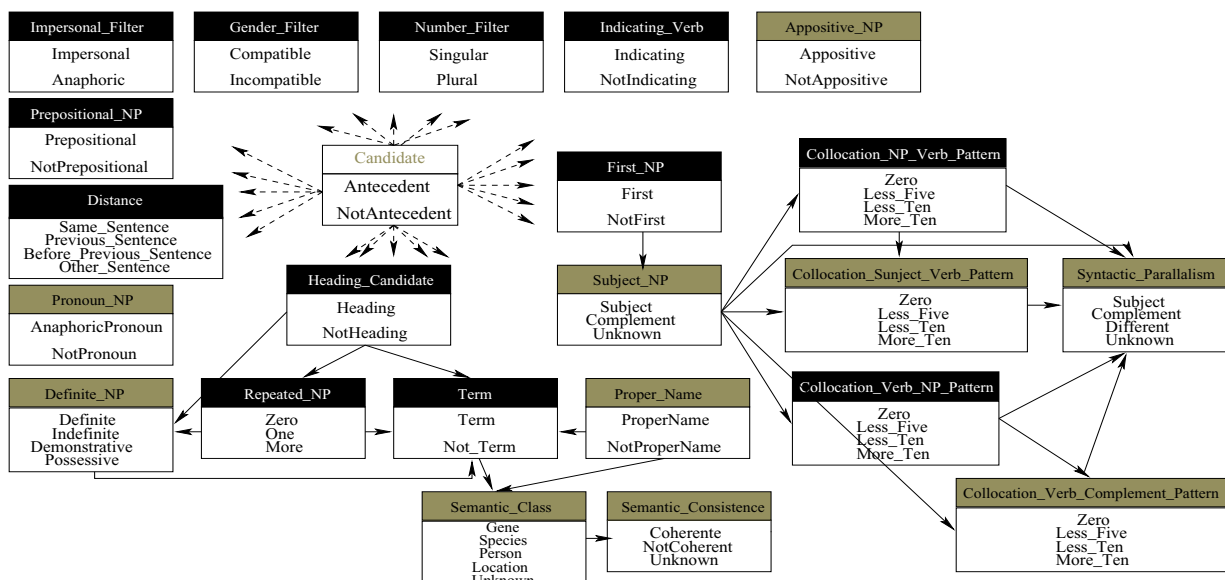
Table 1 summarises the performances of each system measured as a success rate (proportion of anaphors that have been correctly solved by the systems).

Two different measures are given for the last 6 lines: the strict and partial success rate which correspond to two different definitions of what a "correct" antecedent is. The strict success rate counts an anaphor as correctly solved only if the proposed NP exactly matches the phrase tagged as antecedent by the human annotators in the test corpus. The partial success score counts as correct an anaphor where the proposed NP only partially matches the phrases tagged as antecedent in the test corpus as soon as it can be substituted to the anaphoric pronoun without seman-

<sup>2</sup> Namely, the *immediate reference* and *sequential instruction* preferences.

<sup>3</sup> The prediction node is the node `Candidate`, at the centre of the network. It gives the probability for a given candidate to be the antecedent of a given pronoun occurrence. It is linked to all the other network nodes.





**Fig. 2:** A Bayesian Network for the ranking of the antecedent candidates of the anaphoric occurrences of the pronoun it.

tic inconsistency. For instance, in the sentence *[beta-Galactosidase expression from the spl-lacZ fusion] was silent during vegetative growth and was not DNA damage inducible, but [it] was activated at morphological stage III...* our system gives only *beta-Galactosidase expression* as antecedent instead of the whole NP but it can nevertheless be substituted to *it*: it is considered as partially correct resolution only.

Since there are some errors in the input NP list<sup>4</sup>, the anaphora resolution performance cannot reach 100%. The last row (MAX), which gives the highest reachable resolution score for comparison.

System	Results	
	Strict	Partial
Random	6%	-
First_NP	36.3%	51%
Bio_MARS	26.7%	43%
NB_MARS	39.9%	56%
Naive Bayes Classifier	43.1%	59%
Bayesian Network Classifier	44.0%	61%
MAX	93.3%	97.8%

**Table 1:** Anaphora Resolution Results (Success rate)

## 5 Discussion

The first striking observation that can be drawn from Table 1 is that Bio\_MARS performance is significantly lower than the success rate of First\_NP system on our corpus and also lower than the 45.81% score obtained by MARS on a different corpus made of technical manuals [8]. Most of the cases that are correctly

<sup>4</sup> BioLG does not parse sentences that are more than 70 words long or that do not contain any verb. When there is no parse available, we create a list of NPs on the basis of the POS-Tagging.

solved by First\_NP system and not by Bio\_MARS involve the terminological and collocation pattern attributes that are not sufficiently discriminating in our domain<sup>5</sup>: our platform tags as terms some elements that are not salient (e.g. *use*, *work*) and the collocation patterns have a weight too high to be corrected by other observations. In the probabilistic version of Bio\_MARS (NB\_MARS), the parameterising step adapts these scores for our corpus and therefore avoids the previous errors.

Comparing the systems NB\_MARS and BNC shows the importance of the complex linguistic constraints in the resolution process, even if the corresponding attributes are not fully reliable. These additional attributes help to distinguish among various candidates. Let us consider for instance the following sentences extracted from our corpus *[A grpE heat-shock gene]<sub>1</sub> was found by sequencing in [the genome of the methanogenic archaeon Methanosarcina mazei S-6]<sub>2</sub>. [It]<sub>1</sub> is the first example of grpE from the phylogenetic domain Archaea.* NB\_MARS gives the same probability for the candidates 1 and 2 and finally chooses the candidate 2, which is the most recent one. BNC classifier avoids this error: it exploits the syntactic role of the candidate 1 (subject) and its semantic type (*gene*), which increases the candidate probability to 0.73 and solves the ambiguity.

If surface clues are not always sufficient to decide between the candidates, their role is nevertheless important to correct the imperfectness of linguistic information. For instance, the syntactic and named entity information are not reliable enough to be used in isolation. BioLG parser has a fairly good precision (86%) but a low recall (55%) and the results of the named

<sup>5</sup> Our model allows to quantify this fact:  $P(\text{Term} = \text{Term} | \text{Candidate} = \text{Antecedent}) = 0.16$ ,  $P(\text{Collocation\_NP\_Verb\_Pattern} = \text{Less\_Five, Less\_Ten, More\_Ten} | \text{Antecedent}) = 0.08$ ,  $P(\text{Collocation\_Verb\_NP\_Pattern} = \text{Less\_Five, Less\_Ten, More\_Ten} | \text{Antecedent}) = 0.01$ .

entity tagging are noisy (71% of gene names are identified but only 68% of the tagged entities are really gene names, due to ambiguous gene names such as *not*, *All*, *similar*).

It is important to understand how the linguistic properties and the surface clues complement each other. In BNC system, these complementarity is represented and measured by the interdependency links that hold between two network nodes. These links express a set of reinforcement or invalidation constraints. NBC system, which does have such constraints, overestimates the attribute weights. It often puts the correct antecedent in the second or third position in the candidate list, whereas BNC chooses the correct candidate.

A detailed manual analysis of the BNC remaining errors shows the limits of the salience-based approach. 47% of the errors are due to an erroneous calculus of the salient element. BNC fails to find the element that is intuitively identified as the most salient by the human judge because a less salient element ends with a higher salient score than the actual antecedent.

In 21% of the cases, BNC actually finds the salient element but it is not the pronoun antecedent. For instance, in the sentence [*Amino acid sequence analysis*]<sub>1</sub> of [*the 33-kDa protein*]<sub>2</sub> revealed that it is a *sigma factor*, *sigma E.*, the most salient element is candidate 1 which is erroneously preferred to the candidate 2. Solving such anaphors would call for more complex semantic and domain knowledge to check the semantic compatibility of the candidate 2 and the pronoun occurrence.

The remaining errors are due to the corpus imperfect preprocessing (word segmentation errors and unidentified NPs) rather than to the resolution strategy itself.

## 6 Conclusion

In this paper, we have tried to show how interesting the Bayesian Network formalism is for NLP tasks, taking the complex problem of pronominal anaphora resolution as an example. This model allows to overcome the traditional opposition between systems based on linguistic knowledge and knowledge-poor systems. It appears that both approaches should rather be combined than opposed: linguistic knowledge is necessary but often lacking or not fully reliable; surface clues are easier to measure but fail to solve some ambiguities. By unifying both types of knowledge in a single representation, the Bayesian Network approach enables to exploit some information pieces to reinforce, invalidate or supplement others. This gives interesting results on the anaphora resolution task, in comparison with a state of the art system.

Our system can be further improved. We want to extend the set of clues that are exploited for anaphora resolution. For the moment, it only relies on the search of the most salient element to choose the pronoun antecedent and we have shown that this strategy sometimes fails. Our Bayesian Network can be enriched by integrating focused-based information [11]. It would also be interesting to learn the network structure from a training corpus, instead of relying of linguistic expertise as it is the case for the network of Figure 2.

Our first tests show that some nodes seem to be useless, actually. Finally, we would like to take into account the fact that the various candidate scores are not independent of each others. Actually, the choice of a candidate not only depends on the intrinsic properties of that candidate but also of alternative ones. This should lead us to exploit a specific extension of Bayesian Networks, the dynamic Bayesian Networks.

## References

- [1] E. Alphonse, S. Aubin, P. Bessieres, G. Bisson, T. Hamon, S. Lagarrigue, A. Nazarenko, A.-P. Manine, C. Nedellec, M. Vetah, T. Poibeau, and D. Weissenbacher. Event-based information extraction for the biomedical domain: the caderige project. In *Proceedings on International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (BioNLP/LNPBA), COLING'04*, pages 43–49, 2004.
- [2] L. T. Andrew. Kehler, Douglas. Appelt and A. Simma. The (non)utility of predicate-argument frequencies for pronoun interpretation. In *Proceedings of the Human Language Technology Conference*, pages 289–296, 2001.
- [3] R. Bouckaert. Low level information extraction, a bayesian network based approach. In *Workshop on Text Learning (TextML-2002)*, 2002.
- [4] I. Dagan and A. Itai. Automatic processing of large corpora for the resolution of anaphora references. In *Proceedings of COLING'90*, volume 3, pages 330–332, 1990.
- [5] J. Deriviere, T. Hamon, and A. Nazarenko. A scalable and distributed nlp architecture for web document annotation. In *Advances in Natural Language Processing (5th International Conference on NLP, FinTAL 2006)*, pages 56–67, 2006.
- [6] C. Kennedy and B. Boguraev. Anaphora for everyone: Pronominal anaphora resolution without a parser. In *Proceedings of COLING'96*, 1996.
- [7] S. Lappin and H. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561, 1994.
- [8] R. Mitkov. *Anaphora Resolution*. Longman, 2002.
- [9] S. Ponzetto and M. Strube. Semantic role labeling for coreference resolution. In *Companion Volume of the Proceedings of EACL'06*, pages 143–146, 2006.
- [10] S. Pyysalo, T. Salakoski, S. Aubin, and A. Nazarenko. Lexical adaptation of link grammar to the biomedical sublanguage: a comparative evaluation of three approaches. *BMC Bioinformatics*, 7(Suppl 3), November 2006.
- [11] M. Strube. Never look back: An alternative to centering. In *COLING-ACL*, pages 1251–1257, 1998.
- [12] D. Weissenbacher. Bayesian network, a model for nlp? In *Companion Volume of the Proceedings of EACL'06*, pages 195–198, 2006.

# Error Analysis to Translations by MT Systems

Xiaohong WU, Yujie ZHANG, Hitoshi ISAHARA  
Computational Linguistics Group  
3-5, Hikaridai, Seika-cho, Soraku-gun  
Kyoto 619-0289, Japan  
{[@nict.go.jp](mailto:xiaohongwu,yujie,isahara)}

Sylviane CARDEY  
Centre Tesniere  
Université de Franche-Comté  
25030 Besancon, France  
[sylviane.cardey@univ-fcomte.fr](mailto:sylviane.cardey@univ-fcomte.fr)

## Abstract

Machine translation (MT) as a new translation tool has achieved great progress during the past years in respect of text comprehension, technical feasibility and especially translation quality. While high quality output is still a far dream, most of the MT systems already commercialized are capable of reaching an understandable rendition to various subjects. Furthermore, more and more people begin to take advantage of various automatic translation tools as an aid in assembling information. However, a small investigation of several free MT systems on Internet reveals that these and also other systems still make errors while processing some common linguistic phenomena. These errors could be improved or avoided if more attention is paid on linguistic aspects such as certain language-specific structures, word order, etc. This paper based on a small investigation to the MT of some simplified medical texts discusses the common errors found in machine translations. By analyzing the errors from a linguistic point of view, it is suggested that many of these errors can be avoided or resolved by modifying grammar rules or adding some additional information to the databases. Finally, as a result of our own practice, some feasible solutions are also suggested.

## Keywords

Machine translation (MT), error analysis, sentential structures, language-specific phenomena, structural particle De (的), Ba-construction (把字句), syntactic orders

## 1. Introduction

Machine translation as a new translation tool has achieved great progress during the past years in respect of text comprehension, technical feasibility and especially translation quality, in particular in translating well defined domain texts between linguistically close languages [1]. However, as it is not easy to bridge the great gaps existing between different languages, especially between linguistically far languages [2], for example between English and Chinese, a good translation output is still hard to achieve, let alone a high quality rendition. This paper discusses an extended work on English-Chinese MT of medical texts. During our research we made a small investigation for the purpose of analyzing translation errors by MT systems to improve our output sentences. We tested our sentences with the following randomly selected MT systems: Systran [3]; WorldLingo [4]; Google translate [5] and Babel Fish Translation [6].

It is important to note here that the above systems are randomly selected because they are easier to access. We do not intend to make any judgments or criticisms of these MT systems. Instead, we made such an investigation for the purpose that by analyzing the errors found in the translations we could possibly get some useful information for our own work. It is purely for the purpose of research without any discrimination to one or the other. Therefore, while citing the sentences translated by these systems, we will not mention from which system they are translated.

For above mentioned purpose, we selected some of the sentences from our corpus which are constructed for our system that applies controlled language (CL) technique as a support [7] [8]. We apply CL aiming at reducing the linguistic complexity and further reaching a better translation quality. Therefore, the sentences to be tested are already well controlled in lexical usage, syntactic structures and text style. Our translation error-analysis reveals that whilst most of the translations done by these systems are understandable and some of the sentences are well transferred into the target language (TL) – Chinese, further improvements can be done by making some additional efforts on the syntactic constructions of the TL. Such improvements are not only necessary but also important as many of the errors are made while translating relatively simple sentences. It can be predicted that if a system fails to process the basic linguistic phenomenon in simple sentences, it will surely fail to treat long and complex sentences as errors might mount up with the growing complexity of the sentences. Our findings show that most of the errors made by the MT systems are linked to the failure to arrange correctly the word orders in the TL or to construct some structures which are language-specifically bound and are quite necessary in transferring the SL information into the TL. In fact, some of the Chinese language-specific phenomena, such as the use of the structural particle De (的), the Ba-construction (把字句) can be well transferred if such information is well generalized and properly processed by the systems.

## 2. The Chinese structural particle “的”

The uses of the particle De (的) in the Chinese language are very flexible like many other particles. It is generally considered as the marker of the attributives of nouns or the marker of the adjectives and it is often attached after the

attributives or adjectives. The particle De (的) has another special usage: to form a language unit: the De-structure (的字结构). In this case, it is also attached after a word or a phrase to substitute the central word or constituent and is no longer used as the marker of attributive (we will not discuss this in this paper). As the marker of adjectives, the best proof is that in all English-Chinese dictionaries the Chinese correspondences to the English adjectives have the particle De (的) attached after them, for example, “medical” as “医学的”; “terrible” as “可怕的”. However, this kind of practice is useless in many cases and might make the sentence ungrammatical if we keep “De (的)” all the way round while translating English attributives/adjectives into Chinese. For example, “parasitological monitoring” is usually translated as “寄生虫监护”; “serological control” as “血清对照”. Sometimes if we add “的” between the two constituents in Chinese, it might be misleading. For example “Korean friends 朝鲜朋友” does not necessary mean “the friends of/in Korea 朝鲜的朋友”. Furthermore, the word “寄生虫” in Chinese is always a noun, i.e. it will not become an adjective even if the word “的” is added after it. Besides, in Chinese grammar, we have special rules for the use of the structural particle “De (的)”. The employment of “的” after an attributive or an adjective is not mandatory. In many cases, we can leave out “的” or we must leave it out, for example:

- 1) She is very beautiful. (她很漂亮。)
- 2) This disease is very dangerous and contagious. (这种疾病很危险而且会传染。)

In example (1) we do not use the particle “的” after the adjective “beautiful (漂亮)” and it is the usual way of expressing the same idea in the SL. If we add a “的” after the adjective “漂亮”, semantically it has an emphasis in the fact that “*she is really beautiful*”. Furthermore this usage of adding “的” after the adjective is very informal and colloquial. As a matter of fact, when adjectives function as the predicate of the sentence, the particle “的” is seldom used, as shown in example (2).

- 3) a. China is a big country. (中国是个大国。)

Same evidence can be found for the attributives in Chinese. Example (3) shows that between the attributive “大 (big)” and “国 (country)” the particle “的” is not used either. In fact, if we add the attributive marker “的” between them, the whole sentence will look strange and ungrammatical:

- b. \*中国是个大的国。

Of course, there are other cases where the use of “的” is obligatory, for example:

- 4) This is a beautiful flower. (这是一朵漂亮的花。)

In example (4) “的” can not be left out, otherwise, the sentence will become ungrammatical.

Now let us show with some of the examples obtained from the MT systems to see what kind of characteristics these systems demonstrate while processing this linguistic phenomenon, compared with human translation.

**Example i:**

*Portable ultrasound equipment that has a 3.5 – 5 MHz probe*

**Human translation:**

带有一根 3.5 – 5 MHz探头的便携式B超设备

**Machine translation:**

有一根 3.5 – 5 兆赫探针的便携式的超声波设备

This example (we only choose one typical sentence as the example) showed an obvious trace of the word “的” being integrated as the marker of the adjectives in the lexicon similar to the already mentioned practice of English-Chinese dictionaries. Generally speaking, when an English adjective is translated into Chinese as “...式”, “的” should be left out. We found that most of the adjectives are translated into Chinese with “的” attached after them, except one system which makes fewer such errors but more errors on word orders. Let us look at another two examples of different situations:

**Example ii:**

*A Surgeon who deals with possible complications.*

**Human translation:**

处理可能出现并发症的外科医生

**Machine translation:**

- a) 应付可能的复杂化的外科医生。
- b) 外科医师处理可能的并发症。

**Example iii:**

*Antibiotic for plasmid selection*

**Human translation:**

用于筛选质粒的抗生素

**Machine translation:**

- a) 抗生素筛选质粒
- b) 抗生素为质粒选择

These two examples concern with the translation of the post-positioned English modifiers: a relative clause and a preposition phrase, which have to be pre-positioned while being translated into Chinese. In addition, the particle “的” has to be employed. In our test we found that in many cases, relative clauses are transferred into nearly good Chinese syntactic orders as shown in Example ii (a), with some exceptions (b). Besides, we found that “的” was correctly

employed. However, most post-positioned English PPs are not transferred into the right orders with “的” absent in the phrase as shown in Example iii.

**Example iv:**

*Safety and reliability of PAIR depend on the training of the medical staff, relevant indications, the observance of technical rules and safety rules.*

**Human translation:**

PAIR的安全性和可靠性取决对医疗人员的培训，相应的适应症，对技术规则及安全规则的遵守。

**Machine translations:**

对的安全性和可靠性取决于医疗职员的训练 ①，相关的征兆②，技术规则和安全规则遵守③

Example (iv) concerns with the employment of “的” between two nouns (between the modifier and the modified noun). The example is a well translated one except for the term “PAIR” (acronym of “Puncture, Aspiration, Injection and Re-aspiration”; it is wrongly matched to the word “pair” as in “a pair of”). Of course this is not a big problem as long as this term is included in the dictionary such kind of lexical errors can be avoided. The first ① and the second ② parts are nearly correctly translated into Chinese but for the third one ③ we can see that between the two conjoined noun-phrase (NPs) attributives “技术规则和安全规则 (technical rules and safety rules)” and the head word “遵守 (the observance)”, “的” is missing, resulting in a strange rendition: “技术规则和安全规则遵守”.

Usually when a noun modifies another noun, the use of “的” is also flexible. We can often leave out “的” unless the ellipsis might produce ambiguity or errors, for example, “工人榜样 (a model of the workers)” versus “工人的榜样 (a model for the workers)”; “父亲母亲 (father and mother)” versus “父亲的母亲 (father’s mother)”; “生物历史 (biology and history)” versus “生物的历史 (the history of the biology)”. If the relationship between the modifier and modified noun has to be clearly indicated as shown in example iv (医疗人员的培训，相应的适应症), the particle “的” cannot be left out.

However, if more than one noun or NP/adjective is used as a list of modifiers, we leave out “的” for the first few modifier(s) and add “的” between the last modifier and the head noun, for example:

5) a. *He is a tall①, strong② and handsome③ man.* (listed adjectives)

他是一个高大①、强壮②、英俊③的人。(he is a Ge (CLS-classifier) tall, strong, handsome person)

b. *He is a tall①, strong②, handsome③ and brave④ man.*

他是一个高大①、强壮②、英俊③并且勇敢④的人。(he is a Ge-CLS tall, strong, handsome and brave person)

6) a. *The students of Class One① and Class Two② are in this hall.*

一班①、二班②的学生在这个大厅里。(One Class, Two Class De student, in this hall)

b. *The students of Class One①, Class Two② and Class Three③ are in this hall.*

一班①、二班②和三班③的学生在这个大厅里。(One Class, Two Class and Three Class De student, in this hall)

The above usages can be regulated and further generalized into rules to support the MT systems. To solve such problems, we adopt the following approaches: first, while building the lexicon, we leave out “的” as the marker for the Chinese equivalent of the English adjectives and the usage of “的” are defined as independent rules; second, the usage of the structural particle “的” in the phrases are also treated as independent rules. In doing so, we have avoided errors such as attaching “的” after adjectives all the way round or “的” is missing when it should be used.

### 3. The Syntactic Positions of Chinese PPs

Syntactically, English and Chinese differ in many ways. One of the most important syntactic differences between these two languages is the position of the PPs in the sentences. In the following sections we will discuss the positions of the PPs of two major types, of which one refers to the post-positioned PP as modifier of the head noun in the SL, and the other is that of PPs functioning as adjuncts of verbs or sentences (we only discuss adjuncts of verbs).

#### 3.1 The positions of PPs as modifier of nouns

In both English and Chinese the PPs can function as the attributives of nouns. However, while an English PP functioning as an attributive has to be placed after the head noun, the Chinese attributive is always in front of the head noun, for example:

7) *a man with a book in his hand*

一个手里拿着本书的男人 (a Ge-CLS, hand in, take Zhe (AUX-auxiliary), Ben-CLS book De man)

8) *the windows of the classroom*

教室的窗户 (classroom De window)

As shown in example (7) and (8) both post-positioned PPs are fronted in the Chinese equivalents. It is also the case for English relative clauses. Our investigation shows that most PPs headed by “of” are well translated into Chinese but those headed by other prepositions rather than “of” are wrongly transferred into Chinese, often in the same structure as that of English. To save space, we cite only

three of the tested sentences to demonstrate this “PP position” related problem:

**Example v:**

*Lumbar puncture needles for percutaneous puncture*

**Human translation:**

用于经皮穿刺的腰穿针

**Machine translation:**

腰部刺针为经皮刺

**Example vi:**

*Intravenous (IV) catheter for resuscitation treatments*

**Human translation:**

用于复苏治疗的静脉导管 (IV)

**Machine translation:**

静脉注射 (iv) 导尿管为复活治疗

**Example vii:**

*Portable ultrasound equipment that has a 3.5 – 5 MHz probe*

**Human translation:**

带有一根 3.5 – 5 MHz 探头的便携式 B 超设备

**Machine translation:**

- a) 有一根 3.5 – 5 兆赫探头的便携式的超声波设备
- b) 便携式B超设备有 3.5-5 兆赫探头

Again we analyze only the structural errors here, leaving the lexical problems aside. In the above examples (v) and (vi), the PPs are concerned with the preposition “for” which is frequently used in our corpus. As is shown, the positions of both English PPs are not correctly arranged in Chinese. It seems that necessary information is missing for a right performance. However as shown in example (vii-a.), the relative clause is well transferred into the TL (pre-positioned), with some exceptions (Example vii-b.). As such arrangements of the PPs in both languages are quite regular and can be generalized into relatively constant rules. We propose that problems of the PP positions within a nominal phrase be modified by adding additional rules. A generalized rule could be: English N(P) + PP can be converted into Chinese as PP (VP) + de + N(P) with further semantic marks or rules to indicate the other possibilities or exceptions.

For noun phrases with more than one PP attachments and especially those PPs with different semantic contents, different rules have to be specified. We have observed that the semantically different PPs occupy different positions but in a relatively regular sequence in the phrase.

### 3.2 The position of PPs as adjuncts of verbs

Generally speaking, both English and Chinese allow adjuncts to be placed in front of the sentences. However, in

most cases the English PP adjuncts are placed at the end of the sentence whereas the same Chinese adjuncts are usually placed between the subject and the verb, e.g.:

9) *I saw him yesterday in the street.*

我昨天在街上看见了他。(I yesterday in street see Le (ASP-aspectual particle) he)

10) *Take out a few chairs from the classroom.*

从教室里拿出几把椅子。(from classroom in, take out a few Ba-CLS chair)

Example (9) shows that when the English sentence is translated into Chinese, the biggest difference is the different positions of the adjuncts. Example (10) shows the different positions of a PP in an imperative sentence which is our major concern of study. Of course the positions of the adjuncts are not as straightforward as the two examples we have illustrated. In this paper we will just discuss the basic positions of the adjuncts in both languages. In our investigation we find that in most cases, the adjuncts of the verbs are not correctly arranged in the TL resulting in ungrammatical renditions. Let us illustrate our observations of MT with a few of the tested sentences.

**Example viii:**

*Leave the mixture for 2 minutes at room temperature.*

**Human translation:**

于室温下放置混合物 2 分钟。

**Machine translation:**

留下混合物 2 分钟在室温。

**Example ix:**

*Begin albendazole therapy 4 hours before PAIR.*

**Human translation:**

PAIR 前 4 小时开始阿苯达唑治疗

**Machine translation:**

在对之前开始 albendazole 疗法 4 个小时

**Example x:**

*Check for protoscolecies in cystic fluid with the microscope.*

**Human translation:**

用显微镜在囊液中寻找原头蚴。

**Machine translation:**

检查 protoscolecies 在囊状流体里与显微镜。

**Example xi:**

*Store the tube on ice for 3-5 minutes.*

**Human translation:**

将试管存放在冰上 3-5 分钟。 Or:

在冰上存放试管 3-5 分钟。

### Machine translation:

存放管在冰 3-5 分钟。

As shown in the above examples, the systems fail to arrange correctly most of the adjuncts in Chinese with only two correct – “for 2 minutes” in example (viii) and “for 3-5 minutes” in example (xi). Furthermore it seems to us that another trace can be found in which the temporal adjuncts are always placed at the end of the sentence by these systems, for example in example (viii), (ix) and (xi). While in (viii) and (xi) they make a correct choice, “4 hours” in example (ix) is completely wrong as in this case the phrase “4 hours before PAIR” should be treated as one unit instead of two. These examples partially reveal the complex situation of the positions of the Chinese adjuncts in the sentence. However further observation of other analogical Chinese sentences can help find the regularities of the different positions of the adjuncts in the sentences. Let us take the above sentences as examples for further analysis. Let us begin with the temporal adjuncts.

It is true that in Chinese the temporal adjuncts can be put after the verbs. However, these adjuncts usually imply the duration of time as in example (viii) and (xi) but not for a specific time, for example:

- 11) *I have been here for ten years.* (duration)  
我来这里十年了。(I, come here, ten year Le-ASP)

But:

- 12) *Finish the work before 10 o'clock.* (ten o'clock before, finish work)

十点以前完成工作。

- 13) *Perform serological and US monitoring every year of the following 10 years.*

以后十年每年进行一次血清及 US 监护。

- 14) *I got here in 1990.*

我 1990 年来到这里。(I, 1990 year, come, here)

In these three examples (12, 13, 14) we can see that for a specific time, the usual way is still to put the adjuncts in front of the verbs (12) (13) or between the subject and the verb (14).

Concerning the adjunct indicating a location (or manner), there are also at least three flexible alternatives accordingly: in front of the sentence; between the subject and the verb (the most common) and following the verb (to be discussed in the following section), for example:

- 15) *He found the lost child in the woods.*

a. 他在树林里找到了丢失的孩子。(he, in woods, find Le-ASP, lost, De, child)

Or: b. 在树林里, 他找到了丢失的孩子。

- 16) *He found the lost child in the woods with a dog yesterday.*

a. 他昨天带着一条狗在树林子里找到了丢失的孩子。(he, yesterday, with a Tiao-CLS dog, in woods, find Le-ASP lost De child)

b. 昨天他带着一条狗在树林子里找到了丢失的孩子。(yesterday, he, with a Tiao-CLS dog, in woods, find Le-ASP lost De child)

c. 昨天在树林子里他带着一条狗找到了丢失的孩子。(yesterday, in woods, he, with a Tiao-CLS dog, find Le-ASP lost De child)

- 17) *Check for protoscoleces in cystic fluid.*

在囊液中寻找原头蚴

- 18) *Check for protoscoleces with the microscope*

用显微镜寻找原头蚴

- 19) *Check for protoscoleces in cystic fluid with the microscope.*

用显微镜在囊液中寻找原头蚴

Examples (15 – 19) show the different positions of the semantically different adjuncts in the sentences. No matter how flexibly these adjuncts are used, there are always some rules to follow. For example, the adjunct indicating the location can be placed in front of the sentence (15. b; 16. c; 17 and 19); between the subject and the verb (15. a; 16. a, b); and before the verb (if there is no subject) (17 and 19). If adjuncts indicating the time (not for duration) and the location appear together in the sentence, the adjunct indicating the time is before the adjunct indicating the location (16. a, b and c). When adjuncts indicating the location and the manner appear together in the sentence, the adjunct indicating the manner is before the adjunct indicating the location (19), and so on. As we can see, the adjuncts in example (16) can be placed very flexibly in both languages. To define rules for these variations in (16) is difficult and unnecessary for a MT system. However, it is quite necessary to have rules indicating the correct orders for semantically different adjuncts in the sentence as they have to follow regular sequences.

## 4. The Chinese Ba-construction

The Ba-construction (把字句) is a Chinese language-specific structure. It is very important yet quite complex. It is a complex structure because the grammatical status of the Ba is still controversial in Chinese linguistics [10]. Some linguists insist that the Ba is a preposition while others argue that the Ba should be considered as a verb. We adopt the idea that the Ba is a preposition with which the patient object is shifted to the front of the verb and the Ba structure functions as an adjunct of the verb like many other adjuncts that are often placed between the subject and the predicate verb. When the patient object is moved in front of the main verb, another interesting situation appears in which the adjuncts indicating the goal or location of the action is often placed after the verb instead of placing them in front



of the verb as is the common practice in sentences without the Ba. In this paper we will only focus on the usage of this structure which is obligatory in transferring some SL information into Chinese, in particular when translating imperatives into Chinese.

Schematically, a Ba-construction has the following linear configurations:

- a) NP\* + BA + NP + V + X
- b) NP\* + BA + NP + X + V

where the sentence can have an optional NP\* as subject, followed by the Ba and its NP complement, then followed by a transitive verb (V) and another constituent X (which might precede the verb as shown in (b), and usually is an adverb or a preposition phrase functioning as the adjunct). The BA-construction thus characterizes the pre-positioned object (usually a noun phrase) of a transitive verb followed by an adjunct, indicating a resultative or directional effect of the verb [11], for example:

20) *He tore up the photo..*

他把照片撕碎了。(Literally: He Ba photo, tear up, Le-ASP, resultative)

21) *Inject contrast medium into the cyst.*

把造影剂注射进囊肿中。(Literally: Ba contrast medium, inject into cyst in, directional)

In our investigation, only two sentences are found to be correctly transferred into the Ba-structure while most of the other sentences which should be constructed with the Ba-structure do not correspond to this structure. Here is one of the correct examples:

**Example xii:**

*Leave the contrast medium in the cyst as a substitute of protoscolicide agent.*

**Human translation:**

作为杀原头蚴剂的替代品，把造影剂留在囊肿里。

**Machine translation:**

把造影剂留在囊肿作为protoscolicide 代理替补。

This example can be considered as good translations in respect of the general structure of the whole sentence except that the adjunct “as a substitute of protoscolicide agent” is not arranged correctly in the target language (in front of the sentence). Other tested sentences which are to be constructed obligatorily with the Ba-structure in the TL are found understandable but structurally ungrammatical (we choose only those which have fewer lexical problems as examples, leaving aside most of the others which are hard to read), for example:

**Example xiii:**

*Inject contrast medium into the cyst.*

**Human translation:**

把造影剂注射进囊肿里。

**Machine translation:**

\* 注射造影剂入囊肿。

**Example vx:**

*If necessary, insert a catheter in the cyst.*

**Human translation:**

如有必要，把一支导管插入到囊肿里。

**Machine translation:**

\* 如果需要，插入导尿管在囊肿。

**Example xv:**

*Store the tube on the ice for three minutes.*

**Human translation:**

a). 把试管在冰上存放三分钟。(Literally: BA tube, on ice, store, three minute)

**Alternatives:**

- b). 在冰上存放三分钟试管。
- c). 在冰上存放 试管 三分钟。

**Machine translation:**

\* 存放管在冰 3-5 分钟。

In the above examples, all TL sentences follow the syntactic structures of the source language: to begin the sentence with the verb. It is true that like most English imperative sentences, the Chinese counterpart sentences start with verbs. However, in some cases, the Ba-construction has to be employed. This means that for the Ba-construction there exist two choices: obligatoriness and optionality.

Generally speaking, many of the sentences can be used in both ways: to start with a verb or start with the Ba-construction. This does not make a big difference in general. However, semantically the sentences starting with a verb tend to be more narrative while the Ba-construction is more firm and authoritative in expressing ideas. The obligatoriness and optionality can be tested by moving the adjuncts to check the grammaticality of the sentence, for example by moving the adjunct in front of the verb. Briefly, if the sentence stays grammatical when the adjunct is moved in front of the verb, it is optional; otherwise, the sentence should be constructed with the Ba-construction. We take the above sentences to exemplify this point:

**Obligatoriness:**

22) *Inject contrast medium into the cyst.*

We start the sentence by translating first the verb into Chinese just like the English counterpart sentence:

- a). 注射造影剂进囊肿里。(Same linear sequence)

It is a quite unnatural TL sentence and we can feel intuitively that it is structurally wrong. Then we move the adjunct “into the cyst” in front of the verb to see if it is better:

- b). \* 进囊肿里注射造影剂。(into cyst inject contrast medium)



Immediately we know that it is completely ungrammatical. By doing so, we know that it is obligatory to transfer this sentence into Chinese BA-construction:

c. 把造影剂注射进囊肿里。

It is the same situation for the other examples (vx and xv). This obligatoriness of the Ba-structure in the TL is decided by the verbs. It means that this information is closely connected with the sub-categorization of the verbs and should be integrated as additional information in the lexicon. As the Ba-structure conveys different semantic content compared with other similar structures, for example the same structure: V + NP + PP, it is quite feasible to define the rules for this construction.

#### Optionality:

23) Store the tube on the ice for three minutes.

We start translating the sentence again with the verb and with the same structure as shown in the SL:

a. 存放试管在冰上三分钟。

Like the above example (22 a.), this is a very unnatural rendition and is structurally wrong. But if we move the adjunct “on the ice” in front of the sentence:

b. 在冰上存放试管三分钟。

it becomes immediately grammatical. This structure corresponds to the examples shown in section 3.2 as another proof of placing the adjunct indicating the location in front of the verb, leaving another adjunct indicating the duration of time at the end of the sentence as it should be. For this sentence, we can still have two other alternatives:

c. 把试管在冰上存放三分钟。 (Literally: Ba tube, on ice, store, three minute; a BA-structure)

#### Alternative:

d. 在冰上存放三分钟 试管。 (Another way of arranging the adjunct or object)

In this case, we can conclude that the construction of the Ba-structure is optional for example (23). This optionality is also decided by the subcategorization framework of the verbs and can be formulated.

## 5. Conclusion

In this paper we have discussed frequently observed translation errors in English-Chinese MT systems. As we have shown in different examples that though these errors detected may look trivial when treated separately, they are in fact among the major sources of unacceptability and ungrammaticality of the output sentences. As illustrated in

the examples, many of the linguistically related problems can be improved or avoided by adding some additional linguistic information or by making some modifications to the existing databases. Despite of the fact that this investigation by its own limit can not cover many of the other error-producing problems, such error analysis is very useful and instructive in the improvement of the MT quality. This has been proved by our statistical study for other language resources, for example, examples extracted from texts on other general topics.

In conclusion, we state that some errors might look unimportant and are often ignored by MT builders as there are too many other complex situations to be improved. However, nobody can ignore the fact that in MT one error at one point will result in enormous increasing of errors. Therefore, whenever possible any errors should be taken seriously from the beginning.

## 6. References

- [1] BERNTH, Arendse and GDANIEC, Claudia, 2001. “MTranslatability”, Machine Translation **16**: 175–218, 2001.
- [2] HUTCHINS, W. John and SOMERS, Harold L., 1992. “An Introduction to Machine Translation”, London: Academic Press, 1992. [ISBN: 0-12-362830-X]
- [3] <http://www.systransoft.com/Corporate/SWS.html>
- [4] [http://www.worldlingo.com/en/products\\_services/worldlingo\\_translator.html](http://www.worldlingo.com/en/products_services/worldlingo_translator.html)
- [5] [http://www.google.com/trqnsqte\\_t](http://www.google.com/trqnsqte_t)
- [6] <http://world.altavista.com/tr>.
- [7] CARDEY, Sylviane. GREENFIELD, Peter; WU Xiaohong, 2004. “Desinging a Controlled Language for the Machine Translation of Medical Protocols: the Case of English to Chinese”, In *Proceedings of the AMTA 2004, LNAI 3265*, Springer-Verlag, pp. 37-47
- [8] WU Xiaohong, 2005. “Controlled Language – A Useful Technique to Facilitate Machine Translation of Technical Documents”, In *Lingvisticoe Investigationes* 28:1, 2005. John Benjamins Publishing Company, pp. 123-131
- [9] HU Yushu, 1987. “Modern Chinese” , Educational Publishing House, Shanghai. ISBN 7-5320-0547-X-G-466, p. 345
- [10] BENDER Emily, 2002. “The Syntax of Mandarin Ba: Reconsidering the Verbal Analysis”, *Journal of East Asian Linguistics*, 2002
- [11] ZHOU Jing and PU Kan, 1985. “Modern Chinese”, 华东师范大学出版社 , ISBN 7135 10

# Derivation of Macedonian Verbal Adjectives

Katerina Zdravkova  
University St Cyril and Methodius  
Arhimedova bb, Skopje, Macedonia  
keti@ii.edu.mk

Aleksandar Petrovski  
ElKomp  
Tetovo, Macedonia  
sise@mt.net.mk

## Abstract

Macedonian printed lexicons contain very few verbal adjectives, which results in regular collapsing of spelling checkers. In order to overcome this problem, we automatically derived the most common verbal adjectives. This paper describes the derivational process, which resulted in more than 30000 new adjectival lemmas derived from nearly 20000 verbs originating from printed dictionaries. Generation of all the inflections of derived adjectives is also presented. Both processes were performed with the linguistic development environment INTEX/NooJ. Accuracy of the obtained adjectival lemmas and their inflective forms was tested on the lexicon containing the most frequent word forms appearing in Macedonian daily newspapers and in recently published books. Even with this limited validation lexicon, many newly derived adjectives have been recognised in standard language.

## Keywords

Derivation, inflection, electronic lexicon, morphonology

## 1. Introduction

Macedonian language is a South-Eastern Slavic language spoken by approximately three million people, two million of them native speakers. It consists of 26 consonants, 5 vowels, and one, so called dark vowel. The alphabet is phonetic. It is represented by 65 characters in the Cyrillic script [1]. Latin script, which utilises only 57 characters, is implemented in parallel. Macedonian uses Unicode/UTF-8 encoding standard for the Cyrillic, and ISO Latin 5 for the Latin script.

Macedonian language is a moderately inflected language with nearly 200 inflectional types, predominantly related to nouns and adjectives. Contrasting Macedonian nominal inflections, which are always suffixes, adjectival inflections can be either suffixes or prefixes. A significant amount of Macedonian adjectives comprises both affixes in parallel, making the adjectival inflectional morphology more complex. However, adjectival derivational properties are more important for research purposes.

According to Koneski [2], very few Macedonian adjectives, such as бел (lat. bel, eng. white), голем (lat. golem, eng. big), чист (lat. čist, eng. clean) are not derived from other words. Adjectives are usually produced from nouns and from verbs, but also from other adjectives, and from adverbs. Adjectives derived from nouns, adjectives and adverbs exist in the printed dictionaries published so far [3], [4], and [5]. These three dictionaries are very inconsistent with verbal adjectives.

Grammatically, the core of verbal adjectives encloses participles. The most frequent is passive participle, e.g. игра – игран, спие – спан, скокна – скокнат, рипна – рипнат (lat. igra – igran, spie – span, skokna – skoknat, ripna – ripnat). However, there are many verbs capable of deriving active participles, such as: предизвикува – предизвикувачки, испукува – испукувачки (lat. predizvikuva – predizvikuvački, ispukuva - ispukuvački).

There are several adjectives derived from verbs which are not participles, such as: реши – решителен, задолжи – задолжителен; граби – граблив, работи – работлив; убеди – убедителен - убедлив (lat. reši – rešitelen, zadolži – zadolžitelen, grabi – grabliv, raboti – rabotliv, ubedi – ubeditelen - ubedliv). This implies that the number of verb adjectival lemmas should exceed the number of verbal lemmas.

Macedonian printed dictionaries we used contain 19985 verbs, and at the same time only 2083 adjectives, 1242 of them with a function of a passive participle. Therefore, spelling checkers usually collapse on adjectives and their inflective forms. This was a very good motivation for us to derive automatically all the verbal adjectives out of the verbs, and generate all their inflections afterwards.

This paper continues with the introduction of INTEX/NooJ, linguistic development environment used for both, the derivative and the inflectional process. Next two sections are devoted to these processes. In the forthcoming section of the paper, the obtained verbal adjectives are first compared with the adjectives existing in printed dictionaries, and afterwards with the set of the most frequent word forms originating from Macedonian search engine Najdi [6]. The paper ends with the implementation of automatically obtained lexicon and directions for further work.

## 2. Development Environment

Derivational and inflectional processes reported in this paper were made with INTEX/NooJ development tool. INTEX/NooJ is an extension of INTEX linguistic development environment, which has recently been redesigned with .NET object-oriented platform [7]. It is used to construct large-coverage formalized descriptions of natural languages and to apply these descriptions to very large corpora in real time. INTEX/NooJ is independent of the language and the alphabet, which makes it very convenient for Macedonian language and its Cyrillic script.

The descriptions of natural languages in INTEX/NooJ are formalized with electronic dictionaries, and with grammars represented by organised sets of graphs. Its Macedonian module currently consists of a huge morphological dictionary containing 67635 lemmas. They produce 1293946 word forms [8].

All the lemmas originate from dictionaries [3], [4], [5], which existed only in printed version. In absence of electronic dictionaries, the first step towards producing their digitised version was an exhaustive OCR scanning. After eliminating the errors, which were numerous, inflectional classes were developed and assigned to the lemmas. The information about Part of Speech (PoS) existing in [4], basic inflectional information of verbs taken from [3], as well as some syntactic features existing in all three dictionaries, particularly verbal aspect, made inflectional process much easier.

The research presented in this paper consisted of two complementary activities performed with INTEX/NooJ:

1. inflection of adjectives, and
2. derivation of verbal adjectives.

The set of all the inflections associated with one lemma define its inflectional paradigm. Adjectival inflectional paradigms depend on five properties (Table 1). They can simultaneously include prefixes and suffixes, making at most 64 different word-forms.

**Table 1. Adjectival morphosyntactic descriptions**

Attribute	Value	NooJ code	Example
Type	Qualificative	Aqlt	ubav
	Relative	Arel	zlaten
Gender	Masculine	m	ubav
	Feminine	f	ubava
	Neuter	n	ubavo
Number	Singular	s	ubav
	Plural	p	ubavi
Degree	Positive	pst	ubav
	Comparative	cmp	poubav
	Superlative	spt	najubav
	Elative	elt	preubav
Definiteness	No	Dn	ubav
	Yes	Dy	ubaviot
	Proximal	Dc	ubaviov
	Distal	Dd	ubavion

In parallel with word analysis and synthesis, INTEX/Nooj is capable of performing derivation [7]. Identifiable issue of the derivational process are morphosyntactic categories of each word form, which must be explicitly produced by derivational transducers. Derivations are initiated by the property DRV. For example, the following derivational description is used to describe the derivation of all the adjectives ending with suffix -ачки (lat. -ački, transliterated into -achki) from the corresponding verb: АЧКИ = <В>ачки/А.

This rule derives all the verbal adjectives ending in the suffix -ачки. For example, the verb предизвикува (lat. predizvikuva, eng. to provoke) produces the adjective предизвикув-ачки (lat. predizvikuv-ački, eng. provocative, or provoking). Adjectival lemmas derived with this suffix fall into inflectional paradigm SVETSKI. Therefore, inflections are produced according to the inflectional rule:

предизвикува, V+DRV=АЧКИ : SVETSKI

Although introduced in the order inflection → derivation, the process of obtaining verbal adjectives starts with the derivation. Therefore, it is explained the first.

### 3. Derivational and inflectional processes

As mentioned before, adjectives are almost always derived from other words, particularly from nouns and verbs. Derivational process is based on addition of appropriate suffixes. Here are the most frequent Macedonian suffixes for deriving adjectives out of verbs:

-ачки, -ечки (lat. -ački, -ečki). This suffix is widely used in the modern language as a replacement of the active participle in other languages. For example, движечки appearing in движ-ечка сила (lat. dviž-ečka sila, eng. moving force) is formed from the verb движи (lat. dviži, eng. to move); плетачка игла (lat. pletačka igla, eng. knitting needle) comes from the verb плете (lat. plete, eng. to knit). The decision which of both suffixes will be used in the derivational process is purely morphological.

-телен (lat. -telen). According to Koneski [2], this suffix is typical for standard, and omitted in traditional language. For example: внима-телен (lat. vnima-telen, eng. careful) is formed from the verb внимава (lat. vnimava, eng. to care).

-лив (lat. -liv). This suffix can be used to derive adjective from either, nouns and verbs. For example: работ-лив (lat. rabot-liv, eng. working), is an adjective derived either from the noun работа (lat. rabota, eng. work), or from the verb работи (lat. raboti, eng. to work).

-абилен, -ибилен (lat. -abilen, -ibilen). This suffix is used to derive adjectives from verbs with foreign origin, particularly those ending in -ира (lat. -ira). Some verbs form an adjective with only one of these suffixes, such as: дискут-абилен (lat. diskut-abilen, eng. discussable), and адапт-ибилен (lat. adapt-ibilen, eng. adaptable), but there is a big number of adjectives capable of combining with both suffixes. Such are two existing adjectives програм-абилен and програм-ибилен (program-abilen and program-ibilen, eng. programmable), derived from the verb програмира (lat. programira, to programme).

There are several other suffixes used to derive adjectives from verbs, but by far the most productive are those that form verbal adjectives ending in: -ан, -ен, -ат, -ет (lat. -an, -en, -at, -et). They replace the passive participle in other languages. Unlike other adjectives, they are derived by using an inflectional form.

**Table 2. Derivational suffixes and their occurrence in Macedonian printed dictionaries**

Suffix (Cyrillic)	Suffix (Latin)	appears	%
-ачки	-ački	80	0,8
-ечки	-ečki	6	0,6
-телен	-telen	79	0,8
-лив	-liv	503	5,1
-абилен	-abilen	3	0,3
-ибилен	-ibilen	3	0,3
In total		674	7,9

It is worth mentioning that the verb бepe (bere) is one of 75 verbs that produce two equally represented verbal adjectives: бeрeн and бpaн (lat. beren and bran, eng. picked). Although English translation is equal, both adjectives differ in the aspect: first is progressive, while the second is perfective. Progressive form is created by simply adding the suffix –eн (lat. -en) to the stem, but the creation of the perfective form is more complicated because it triggers stem alteration.

After an exhaustive examination of the derivational behaviour of Macedonian adjectives, 16 different derivational schemes were isolated. Their corresponding INTEX/NooJ rules are presented in Table 3.

**Table 3. NooJ expressions for forming verb adjectives with a function of passive participle**

Code	NooJ expression	Used
SUM	<BW> (бидeн/Vadj)	1
BRANI	<B> (eт/Vadj)	457
BELEZHI	<B> ((eн+ан)<E>/Vadj)	628
BROI	<B> ((eн+јан)<E>/Vadj)	124
SEDI	<B> (eн/Vadj)	5483
SLUZHI	<B> ((e+<B>гa)н/Vadj)	35
KINE	<B> (eт+aт/Vadj)	33
BERE	<B> ((e+<L><B><R>a)н/Vadj)	75
SPIE	<B> ((e+<B>a)н/Vadj)	17
PLACHE	<B> ((e+<B>кa)н/Vadj)	27
MELE	<B> ((e+<L><B><R>e)н/Vadj)	11
VRIE	<B> ((e+<B>e)н/Vadj)	8
SKINE	<B> (aт/Vadj)	1308
POMAZHE	<B> (<B>зaн/Vadj)	2
PISHE	<B> (aн/Vadj)	11740
POSTELE	<B> ((e+<L><B2><R>a)н/Vadj)	5
In total		19954

All the verbs from the dictionary have been assigned according to one of the above codes. After that, INTEX/NooJ machine generated thousands of potential verbal adjectives. It is interesting that only 6 of them derive only one verbal adjective, while other 10 derive two adjectives with different aspect. Thus, currently existing

19985 verbs in Macedonian INTEX/NooJ dictionary created 20948 different verbal adjectives in total. Compared to previously existing 2083 verbal adjectives, derivational process increased their number ten times. As previously stated, adjectives are highly inflective words. Therefore, in order to check their presence in real-life texts, in parallel with the derivative process, generation of adjectival inflective forms had to be performed.

Second class of Macedonian adjectives is the class of relative adjectives. They describe properties connected with another object, usually reflecting: the origin, or the material the object was made of, as in дpвeн (lat. drven, eng. wooden), or association with some category, e.g. гpyпeн (lat. grupen eng. grouping). These adjectives as a rule don't form degrees. Consequently, their inflectional paradigm is restricted to suffixes only, forming only 16 word forms.

## 4. Experimental results

Before derivational process started, it was interesting to check how many inflective forms could be generated from Macedonian adjectives originating from printed dictionaries. Inflected forms were obtained using basic inflectional rules UBAV and GRUPEN, which were assigned according to information inherited from dictionary [4]. Due to some morphological peculiarities found in some adjectives, inflectional paradigm was extended by additional 31 rules, as presented in Table 3.

Inflectional process ended up with barely half of million adjectival word forms. Compared with the initial number of 9907 lemmas, the increase was 47,27, which is in fact the inflectional factor, i.e. the ratio between number of word forms and number of lemmas.

At the moment, there is not an appropriate Macedonian corpus to prove the accuracy of produced word forms. The results of testing the results on small corpus are discussed in the next section of this paper. Even without an exact proof that inflectional process was faultless, we decided to implement the same approach to derived verbal adjectives.

In absence of any semantic information, only three types of adjectival derivations could be performed, without a fear that the derivational process would produce huge amount of obsolete lemmas and adjectival word forms.

### 4.1 Verbal adjectives corresponding to passive participle

This type of adjective derivation is the most frequent in Macedonian standard language. Although passive participle should be produced from transitive verbs only, i.e. from verbs that take direct objects, same word formation standard is used for perfective intransitive verbs, e.g. пaднe, тpгнe (lat. padne, trgne, eng. to fall down, to start). Verbal adjectives are presented in the dictionaries, but their number is rather low.

Derivational process of these adjectives can actually be treated as an inflectional process, but these word forms in Macedonian morphology are treated as separate canonical forms, in this case, as adjectives. They are derived using the suffixes -ан, -ен, -ат, -ет (lat. -an, -en, -at, -et). First two suffixes are predominant and they correspond to verbs ending in a vowel preceded by a consonant different from н (lat. n). Second two suffixes correspond to verbs ending in a vowel preceded with the last consonant н (lat. n).

If all the adjectives ending in -ан, -ен, -ат, -ет are considered to be verbal adjectives, in such case, Macedonian dictionaries contain only 1345 verbal adjectives corresponding to passive participle. But, this number is lower, because same suffixes are productive for deriving adjectives from nouns, as in: природен, јазичен (lat. prirodan, jazičen; eng. natural, lingual). Anyway, this information is useful to determine lower boundary of average increase factor, which is 14,57. Namely, number of these adjectives, presented in Table 6, gives in total 20942 lemmas, 19597 of them completely new. Compared with previous 9907 adjectives, new lemmas enrich adjectival part of the dictionary almost 200%.

#### 4.2 Derivation of adjectives from verbs ending in -ира (lat. -ira), -ачки (lat. -ački) and -ечки (lat. -ečki)

As mentioned before, progressive verbs of international origin can derive verbal adjectives by adding the suffixes -ибилен (lat. -ibilen) and -абилен (lat. -abilen).

Printed dictionaries are very poor regarding these adjectives. There are only 6 such adjectives presented, which produce 384 inflectional forms. After the derivational process, 923 potential adjectives ending in билен (lat. -bilen) have been obtained.

Similarly to later, adjectives ending in -ачки (lat. -ački) and -ечки (lat. -ečki) are derived from progressive verbs. Verbs belonging to morphological i-group, i.e. verbs with a base form ending in vowel и (lat. i), derive adjectives with the suffix -ечки, for example носи → носечки (lat. nosi → nos-ečki). Verbs belonging to a-group and e-group derive with the suffix -ачки (lat. -ački), for example ветува → ветув-ачки (lat. vetuva → vetuv-ački) and бере → бер-ачки (lat. bere → ber-ački). There are two derivational rules used to express this type of derivation:

ACHKI = <B>ачки/A and ECHKI = <B>ечки/A

Printed dictionaries contain only 86 of these adjectives. After the derivational process, their number increased to 10325, producing 10239 new potential adjectives.

#### 4.3 Inflectional process

Derivational process introduced more than 30000 new adjectival lemmas. In the very optimistic script, they are all eligible. In order to check their presence in real life language, we decided to check whether at least one

inflectional form of a particular lemma exists. In such case, newly derived adjective is considered to exist in the language. Similarly to inflectional process presented in section 3, one inflectional paradigm has been assigned to each new adjective. After that, INTEX/NooJ produced dictionaries with all the inflective forms.

Table 4. Number of adjectives ending in corresponding suffix

Suffix (Latin)	printed dictionaries	derived verbal adjectives	new verbal adjectives	increase factor
-an	290	12653	12363	42,63
-en	916	6458	5542	6,05
-at	95	1341	1246	13,12
-et	44	490	446	10,14
-bilen	6	923	917	152,83
-ački	80	9455	9375	117,19
-ečki	6	870	864	144,00
In total	1437	32190	30753	21,40

Table 5 presents the number of obtained inflectional forms for each type of derivation. Although verbal adjectives make 14,50% of all adjectives in printed dictionaries, their contribution in adjectival word forms is 18,76%. Newly derived verbal adjectives increase total amount of adjectives 2,11 times. Their inflections increase the number of existing adjectival word forms 3,34 times.

#### 5. Accuracy of obtained results

Creation of different Macedonian corpora is an ongoing project [9], and very few results could be used to evaluate the results of derivational and inflectional process. The most comprehensive set of real life words we had an access to was the lexicon extracted by search engine Najdi [6], available from www.najdi.org.mk. Najdi unites articles from daily newspapers, recently published books, and many Macedonian blogs. Although this engine advertises almost one million most frequent word forms, the set of word forms appearing at least twice in Najdi corpus contained only 416546 items.

Table 5. Existing and newly derived verbal adjectives

Adjectival type	Passive participle	BILEN	ACHKI ECHKI
Adjectival lemmas in printed dictionaries	1345	6	86
Word forms of existing adjectives	86080	384	1376
Newly derived verbal adjectives	20942	923	10239
Word forms of newly derived adjectives	1340672	59072	165200
Increase factor	40	154	120

Evaluation process was again done in INTEX/NooJ language environment by comparing word forms corresponding to new verbal adjectives with the words from Najdi lexicon. Initially, Najdi lexicon seemed to be sufficient to confirm the correctness of many new adjectives, but first results were worse than expected (Table 5). Therefore, evaluation was extended to verbal adjectives from printed dictionaries. Results were again poor. Our first assumption was that adjectives were not significantly represented in Najdi corpus, but after manual check of this lexicon, we concluded that it actually contained many spelling errors and colloquial forms, particularly because of its connection with blogs. This led to conclusion that the lexicon was actually not adequately big to deal with huge amount of adjectival word forms. Considering this fact, final result was actually very satisfactory.

First, very few adjectives appeared in more than three different inflectional forms. Compared with the average inflectional factor of 47,27, actual inflectional factor is much lower. Second, media based corpora, such as Najdi, encompass only the most frequent words in the language, while derivational process produced many infrequent lemmas.

**Table 6. Presence of verbal adjectives in the real-life corpus originating from Najdi.org.mk**

Adjectival type	Passive participle	BILEN	ACHKI ECHKI
Word forms of adjectives from printed dictionaries	6963	73	410
Percentage of found old word forms	8,09	19,01	29,80
Word forms of newly derived adjectives	16696	93	2055
Percentage of found new word forms	1,25	0,16	1,24
Increase factor	2,40	1,27	5,01

Nevertheless, if we look at the absolute figures, over 10,000 word forms in Najdi originated from derived dictionaries, and they could never be found in traditional ones. In relative figures, increase factor of real word forms is in average 2,53 times, which is actually higher than total increase of 2,11 presented in Table 6. This fact fully justifies the generation of derived dictionaries.

## 6. Implementation and further work

Research presented in this paper is a pioneering work not only for Macedonian, but also for most Slavic languages. It was initially motivated by poor performance of all Macedonian spelling checks. Verified automatically derived adjectival lemmas together with their inflectional forms, will considerably improve spelling performance.

Macedonian search engine nowadays cope with exact match of keywords with word forms found in documents. Inflective adjectival forms, together with other word forms generated with INTEX/NooJ will enable more sophisticated search including same words and phrases in different forms.

Further work will first be concentrated on proving the accuracy of obtained results. This is a very realistic goal, because monolingual dictionary is in final stage, and it will enable validation with current words. In parallel with monolingual dictionary, useful source of semantic information can be interpretative dictionary, which has been developing for several years. Addition of this information to our currently dictionary will exclude many artificial derivations. At the same time, it will enable derivation of adjectives with those suffixes that have been omitted.

Several teams currently build huge collections of written documents mentioning approximately two millions word forms. Comparison of our lexicon with theirs could also be very important to prove correctness of our approach.

Apart from strictly practical implementation, this research can be used to study the usage of verbal adjectives in composite verb tenses. This will be very important, because Macedonian tenses considerably differ from other Slavic languages, proving its usually disputed existence.

## 7. References

- [1] Christin, A.-M. (ed): A History of Writing from hieroglyph to multimedia, Paris, Flammarion (2002)
- [2] Конески, Б.: Граматика на македонскиот литературен јазик (Grammar of Standard Macedonian), Просветно дело, Скопје (2004)
- [3] Димитровски, Т., Корубин, Б., Стаматоски, Т.: Речник на македонскиот јазик: со српскохрватски толкувања (Macedonian Dictionary with Serbo-Croatian Interpretation), вол. 1-3, Дечка Радост, Скопје, (1961-1966)
- [4] Конески, К.: Правописен речник на македонскиот литературен јазик (Spelling Dictionary of Standard Macedonian), Просветно дело, Скопје (1999)
- [5] Микуновиќ, Љ.: Речник на странски зборови и изрази (Dictionary of Foreign Words and Phrases), Просветно дело, Скопје (2005)
- [6] Metamorphosis: Interview with Petar Kajeovski, Creator of Macedonian Search Engine Najdi!, available from [http://www.metamorphosis.org.mk/index.php?option=com\\_content&task=view&lang=en&id=211&Itemid=26](http://www.metamorphosis.org.mk/index.php?option=com_content&task=view&lang=en&id=211&Itemid=26) (2004)
- [7] Silberstein, M. INTEX/NooJ's Dictionaries, Vetulani Z. (ed.): Proceedings of LTC 2005, Poznan University (2005)
- [8] Petrovski, A. About a Macedonian Computational Dictionary, in Kon-Popovska, M., Zdravkova, K. (eds): Proceedings of the 2nd Balkan Conference in Informatics, Ohrid (2005) 76-83
- [9] Mitreski, G., Venovska-Antevska, S.: Македонски јазичен корпус (Macedonian language corpus: idea, possibilities, realization): Proceedings of the 37th Seminar on Macedonian Language, Literature and Culture, Skopje, Macedonia, (2005) 73-88