Albanian Syntactic Parsing

Arta Misini[0000−0002−3147−057X], Ercan Canhasi ✉[0000−0003−2295−1467], and Samedin Krrabaj[0000−0003−0954−7160]

University "Ukshin Hoti", Faculty of Computer Science, Str. "Rruga e Shkronjave" no. 1, Prizren 20000, Kosovo
{arta.misini, ercan.canhasi, samedin.krrabaj}@uni-prizren.com

Abstract. This paper is about computational processing of the natural language, and the goal of this study is to provide an appropriate method for syntactic parsing of Albanian. We describe the prior work by analyzing the current state of the field and presenting various algorithms used for parsing, using a hand-tagged corpus to build a model for the syntactic parser.
Parsing is the most proper approach to identify the syntactic structure that is useful in determining the meaning of a sentence. Research has shown that English language parsers are not useful in analyzing the sentences in Albanian because of some morphological, syntactic, and grammatical differences.
The algorithm splits the sentences into parts of speech and analyzes these sentences using the natural language's syntactic rules. We discuss the methodology and approaches for designing and implementing the parsing methods. Then, we describe the results of our research within these directions.

Keywords: Natural Language Processing · Syntactic Parsing · Sentence· Phrase · Part of Speech · Tagging · Grammar · Rule.

1 Introduction

Syntactic parsing is a fundamental issue in natural language processing and has a wide range of applications. This issue has been the subject of intensive research for decades, and as a result, there are parsers in specific fields. The vast amount of information in the natural language available for access by computers has introduced the need for the development of computer systems for processing the natural language [6]. Initially, extracting and interpreting the available content was difficult, but the progressive growth of natural language processing introduced easy and systematic solutions. In recent decades, natural language processing has emerged as an active research field, providing efficient applications such as machine translation, information retrieval, information extraction, text summarization, speech recognition, parsing, etc. Machine learning algorithms are used to solve these tasks. The way we share or communicate our feelings is of great importance in processing the text in the interest of analysis [31].

Parsing is the most appropriate approach to interpreting sentences of the natural language, to identifying and searching what possible expressions express. It is the process in which the syntactic structure of the sentence is identified using the lexical and syntactic rules. Identification of syntactic structure is useful in determining the meaning of a sentence [31, 17]. Parsing generates a parse tree of the sentence to eliminate the interpretation ambiguity [31]. Albanian is a language that has a free word order, like German, which means that the same forms of the word appearing in different positions in the sentence, often have different grammatical and/or semantic roles, so identification and summary sentences in Albanian is a challenging task.

In this study, we use two parsing algorithms: Cocke-Kasami-Younger (CKY) and Early algorithm to analyze the input sentences and returning the resulting parse tree. These algorithms are implemented based on the context-free grammatical rules of Albanian.

The main contribution of the described work is to an Albanian syntactic parser that can be useful in various types of applications in the field of natural language processing. The syntactic analysis of a sentence is appropriate to clarify the meaning of the words, to check if the sentence is syntactically correct according to the grammatical rules. Identifying the syntactic structure of a sentence is also appropriate to determine the meaning of the sentence regard to linguistic relations such as subject-verb, verb-object, and noun-modifier [22].

We have organized the remaining of the paper as follows. Section 2 presents the background of parsing in the natural language processing field and summarizes related works. Section 3 describes the Albanian part of speech tagging, syntax, and formal grammar for Albanian, and algorithms used to implement the Albanian syntactic parser. Section 4 presents the experimental results, and Section 5 concludes the paper and gives a direction to the future work.

2 Related Work

Parsing is a research field of natural language processing, an imperative basis for any advanced language processing and a key step in understanding a sentence. It is the process of analyzing a sentence to discover its phrase structure, according to the rules of a grammar.

Natural Language Processing is a field of computer science, artificial intelligence, and linguistics devoted to making computers understand the expressions and words written in natural languages. Natural Language Processing (NLP) encompasses everything a computer, or machine needs to understand the natural written or spoken language. The main goal of research in NLP is to analyze and understand the language [6].

Tagging is the task of mapping each word in a sentence with its part-of-speech (PoS); we decide whether any word is a noun, verb or adjective [25, 8]. Parts of speech are useful because of the large amount of information they provide about a word and its neighbors. Knowing that a word is a noun or verb tells us about possible neighboring words (nouns are preceded by determinants,

verbs by nouns) and about the syntactic structure around the word (nouns are generally part of the noun phrases), which makes the PoS tagging, a significant component of syntactic parsing [17]. The most popular method to tag sentences is the use of a large corpus of sentences marked with tags and then training a pattern on those tagged sentences [2, 4]. The two problems faced by all taggers are ambiguity when a single word has more than one tag, and unknown words related to the grammatical vocabulary used, given that even the largest dictionaries cannot include all the words used in real scripts. From the research of scientific literature, there have been encountered two scientific papers in the development of a computer model of the Albanian tagger. One paper [35] was developed based on the morphological rules of the Albanian grammar, while the other [3] based on the Hidden Markov Model (HMM).

Syntactic parsing is a fundamental field of research in computational linguistics. In the context of NLP, the term parsing refers to the automated analysis of a sentence as a sequence of words, to determine its potential syntactic structure from a formal grammar definition [7, 13, 14]. The formal grammar is generative and has answered many theoretical issues related to the linguistic structures [9, 12]. There are two forms of parsing. A top-down parser starts with the input sentence and tries to build a tree whose leaves match the given input. A bottom-up parser starts with input words and tries to build the trees by finding several derivations that give the input sentence, applying the grammatical rules [15, 16, 22].

Most parsers are partly statistical. They depend on a corpus of training data, which is parsed by hand. The statistical parsers select the most likely parse tree based on statistical information. The approaches include Probabilistic Context-Free-Grammar (PCFG), neural networks, and maximum entropy models [30].

An efficient solution to the problem of syntactic parsing can be the application of machine learning methods. The role of machine learning is to generalize the grammar for more accurate results, even in unknown sentences. However, these methods require a large number of training data and test data (parse trees). Such data are part of an annotated corpus, which is not available in Albanian. Creating an annotated corpus requires a large amount of data, which must be collected over a long period time (the collection of data for the Penn Treebank used for the English language has taken more than a decade).

In the absence of an annotated corpus, the only remaining solution to the problem is applying computational methods based on Context-Free Grammar. Dynamic Programming [23] provides a structure for solving this problem. In parsing, the dynamic programming tables store the parse subtrees for each constituent in the input. It solves the problem of re-parsing (the subtrees looked in the table are not re-parsed) and partially solves the problem of ambiguity (the dynamic programming table implicitly stores all possible parses and stores the constituents with the links that enable the reconstruction of the parse tree). The most used methods are the CKY algorithm and the Earley algorithm.

The basic algorithms implemented within the work are CKY [17, 13, 16] and Earley [15, 16, 23], which have been developed based on the grammatical rules

of Albanian. The grammar consists of a set of rules, which describe the syntax of the language [20]. Linguistic structures represent the structural relationship of the sentence through the phrase-structure rules (noun phrase [24, 27], verbal, adjectival, prepositional [19], and adverbial [33, 34]), which analyze a sentence as a composition of meaningful linguistic units.

Much effort has been made to develop syntactic analysis with different approaches [25]. There are many parsers for English and other languages such as French, Italian, German, etc. There are no models of syntactic parser for Albanian. This is because there is no ready-made annotation about the use of corpus resources [28] available for this language. Also, Albanian has a morphologically rich grammar, which makes it hard to build an efficient linguistic tool.

The Stanford Parser from the Stanford Natural Language Processing Group [32, 5] and the Berkeley Parser [32, 5] are the most popular parsers. The Stanford parser has its variants - PCFG Parser, Factored Parser - which improve the standard Stanford Parser for speed and accuracy. The Stanford Parser is used to parse sentences in languages, such as English [26], German [29], Arabic [10], Chinese [21], and French [11].

3 Albanian Syntactic Parser

A parser is a tool that is responsible for generating the parse tree. It is a procedural component, which remains the same during the parse tree generation regardless of language, but the grammar doesn't remain the same for all languages [31].In syntactic parsing, the parser seen as search through the space of all possibles parses to find the correct one for the sentence. The grammar defines the search space of the possible parse trees. Most of the parses rely on two search strategies: top-down search that is driven by grammar and bottom-up that is driven by the data [15, 16, 22]. Perhaps, the ambiguity is the most serious problem faced by syntactic parsers. At some point in one pass through a sentence, there will usually be some grammatical rules that can be applied [17].

The model of the syntactic parser consists of various levels of modules. The first level is the lexical analysis. The purpose of this level is to split the input into the sequence of tokens corresponding to the words. A further step in the analysis is to map each word with a part of speech tag. The second level, is the syntactic parsing, which analyzes the syntactic structure of the sentence. The analysis confirms that phrases are well-formed, and it determines a linguistic structure represented as a parse tree. The language analyzer uses knowledge of the language syntax (grammar), morphology (lexicon), and identifying the linguistic relationships provides a structure for semantic interpretation.

3.1 Albanian Part of Speech Tagger

The part of speech tagger used in this work is a trained model based on hand-tagged data. The tagging is done using a corpus of words labeled manually with tags. Texts tagged in the corpus include different word-forms of the same word,

because of the different grammatical characteristics that take a word within the sentence. The corpus is a 46,306-word collection of samples from written texts of different genres (novels, newspapers, etc.).

The corpus includes the 58-tag tagset. The tagset includes 10-word classes (nouns, verbs, pronouns, adjectives, numerals, conjunctions, prepositions, particles, adverbs, and interjections) that Albanian words fall into, along with the other grammatical features of each of them (including gender, number, case, form, person, tense). The tagset is shown in table 1.

Table 1. Part-of-Speech Tags (including punctuation)

Tag	**Description**	**Example**	**Tag**	**Description**	**Example**
ABBR	abbreviation	*LDK*, *etj*	PINT	interrogative pronoun	*çka*, *çfarë*
ADJFP	adj. feminine plural	*politike*	PPER	personal pronoun	*ajo*, *unë*
ADJFS	adj. feminine singular	*kryesore*	PPOS	possessive pronoun	*tyre*, *saj*
ADJMP	adj. masculine plural	*të ndryshëm*	PREF	reflexive pronoun	*vetë*, *vetvetes*
ADJMS	adj. masculine singular	*nismëtar*	PREL	relative pronoun	*që*, *të cilën*
ADJP	adjective plural	*gjysmë*	PRP	preposition	*në*, *për*, *nga*
ADV	adverb	*ndërkaq*	PRT	particle	*mbase*, *sidomos*
ADVQ	wh-adverb	*pse*, *ku*	PRTA	affirmative particle	*po*
APST	apostrophe	*'*	PRTN	negative particle	*nuk*, *mos*, *jo*
ART	article	*e*, *të*, *i*	PRTC	comparative particle	*më*, *se*
CNJC	coordin. conjuction	*dhe*, *por*	PRTQ	question particle	*apo*, *a*
CNJS	subordin. conjuction	*përderisa*, *nëse*	PRTR	r... particle	*u*
INTJ	interjection	*eh*, *ah*, *hë*	PRTS	s... particle	*të*, *t'ua*
NFP	noun feminine plural	*fëmijët*	PRTV	verb particle	*do*, *duke*
NFS	noun feminine singular	*bazë*	VMOD	modal verb	*duhet*, *mund*
NMP	noun masculine plural	*sytë*, *tjetri*	VIMP	verb, imperative	*ecni*, *tregojeni*
NMS	noun masculine singular	*laps*	VPCP	verb, past participle	*thënë*, *ardhur*
NP	proper noun	*Akademia*	VAUXP1	auxiliary verb, 1-pl	*kemi*, *jemi*
NLE	%	*%*	VAUXP2	auxiliary verb, 2-pl	*keni*, *ishit*
PNTE	end punctuation	*.*, *!*, *?*	VAUXP3	auxiliary verb, 3-pl	*janë*, *kanë*
PNTS	mid-sentence punc.	*,*, *-*	VAUXS1	auxiliary verb, 1-sg	*kam*, *jam*
NUMC	cardinal number	*një*, *dy*, *tre*	VAUXS2	auxiliary verb, 2-sg	*ke*
NUMD	decimal number	*1*, *2*	VAUXS3	auxiliary verb, 3-sg	*kishte*, *është*
PCL1	clit pro-1	*i*, *e*, *të*	VP1	verb, 1-pl	*ja*, *ia*, *t'i*
PCL2	clit pro-2	*ta*, *ua*	VP2	verb, 2-pl	*shkoni*, *mbeti*
PFS	prefix	*super-*, *ish-*	VP3	verb, 3-pl	*shihnin*, *vijnë*
SFS	suffix	*-ja*, *-në*, *-së*	VS1	verb, 1-sg	*kryen*, *bëj*
PDEM	demons. pronoun	*kjo*, *atyre*, *atë*	VS2	verb, 2-sg	*thuaj*, *punon*
PIND	indefin. pronoun	*njëri*, *ndonjëri*	VS3	verb, 3-sg	*pëlqente*, *shkonte*

We have made an experiment using several sentences from different texts written in Albanian. The experiment has 1000 words in which 912 of them are known words. To quantify the accuracy, we use the precision (the standard measure), that is the number of correct token-tag pairs that are produced, divided by the total number of token-tag pairs that are produced. According to this experiment, the accuracy of the tagger is 98.1. The results of this experiment are shown in table 2.

Table 2. The accuracy of POS tagger used in syntactic analysis.

	Number of words	**Accuracy**
Known words	912	0.988
Unknown words	88	0.914
Tagged words	1000	0.981

3.2 Syntax and Formal Grammar for Albanian

Studying the structure of the sentence is referred to as syntax. The word syntax comes from the Greek word *s'yntaxis*, which means "putting together or arranging", and refers to how the words are arranged together in a sentence, in the sense of the grammatical nature with the relationships between the units formed by them, with the related regularities [20].

Parsing requires a mathematical model of the syntax of the respective language, which is supposed to be a formal grammar. The most common mathematical system for modeling the phrasal structure in natural languages is Context-Free Grammar (CFG) [16]. Since its introduction by Noam A. Chomsky (1956), CFG has been the most influential formalism of the grammar to describe the syntax of the language and is often used as a basic formalism when describing the parsing algorithms [13]. Its purpose is not to describe the details of a particular language, but to formulate the basic principles that define the natural languages, grammar, and their common characteristics [24, 9, 12].

The standard way of defining a context-free grammar G is like a tuple with four parameters $G = \langle \sum, N, S, R \rangle$, where $\sum$ and N are finite disjoint sets of the terminal and non-terminal symbols, respectively, and S $\in N$ is the start symbol (Table 3).

We can formally define the language $\mathscr{L}_G$ generated by a grammar G as the set of strings consisting of terminal symbols that may be derived from the designated start symbol S.

$$\mathscr{L}_{\mathrm{G}} = \{w \mid w \text{ is in } \sum * \text{ and } S \stackrel{*}{\Rightarrow} w\} \tag{1}$$

The problem of mapping a sequence of words to its parse tree is called syntactic parsing [16, 13].

Table 3. Parameters of the context-free grammar

Parameter	Description
N	a set of non-terminal symbols (or variables)
$\sum$	a set of terminal symbols (disjoint from N)
R	a set of rules, each of the form $A \rightarrow \beta$, where A is a non-terminal, β a string of symbols from the set $(\sum \cup N)^*$
S	a designated start symbol

The most significant unit in describing natural languages is the phrase [34]. The term phrase in Albanian doesn't match that of English. The relevant term in Albanian of what the term phrase in English means, is the term syntagm [24]. Syntagm is the constructive, meaningful unit between the word (which performs the syntactic function) and the sentence (which contains meaning) [34]. A formal grammar consists of a set of rules, each of which expresses ways how the language symbols can be grouped and ordered together [16]. The syntagmatic structure is the way to handle the structural relationship of the sentence through the concept "consists of ($\rightarrow$)". One sentence may consist of one nominative syntagm followed by one verbal syntagm:

$$\text{S} \rightarrow \text{NPh VPh} \qquad (\textit{Ajo është duke lexuar një libër})$$

The syntagmatic structure rules are used to describe the syntax of a specific language. They analyze the sentence as a hierarchical relation of syntactic components known as syntagmatic categories. A constituent element of the syntagm is the head which defines its syntactical characteristics. The elements preceding the head of the syntagm are called premodifiers (specifiers), and the elements following the head of the syntagm are called postmodifiers (complements). The syntagmatic categories based on the lexical nature of the syntagm head are classified into:

Nominative syntagm (NPh): Nominative syntagms are the structures headed by a noun, pronoun, numeral, or any other nominative structure [27]. The nominative syntagm structure consists of the head, which is generally a noun that is preceded or followed by other dependent elements. As dependent elements, there may be premodifiers eg, ***ajo*** *heshtje*, ***im*** *vëlla*, or postmodifiers eg, *një mungesë* ***ajri***, *dritarja* ***matanë*** [24].

Verbal syntagm (VPh): Verbal syntagms are the structures headed by a verb. The head of a verbal syntagm is always a verb, like: *ai e* ***pësoi*** *jashtëzakonisht keq.* As dependent elements, there may be premodifiers eg, ***qëllimisht*** *e bëri*, or postmodifiers eg, *erdhi* ***furishëm*** [33].

Adjectival syntagm (AdjPh): The structure of the adjectival syntagms is similar to that of the nominative syntagms. The adjectival syntagm structure consists of an adjective, which is also the head of the syntagm. There may be premodifiers eg, ***gjithmonë*** *i lumtur*, and/or postmodifiers eg, *e lodhur* ***nga rruga***. We should note that the place of the adjective differs depending on the language; in English, unlike Albanian, the adjective is placed before the noun [34, 33].

Adverbial syntagm (AdvPh): Adverbial syntagms are the structures consisting of the syntagm head that is an adverb and/or dependent elements. As dependent elements there may be premodifiers eg, *krejt papritur*, or postmodifiers eg, *nesër* ***në mëngjes*** [33].

Prepositional syntagm (PrpPh): The Albanian prepositional syntagms are a special syntagmatic category in the hierarchical structure of the sentence. Their basic structure consists of a preposition, which is the basic element of the syntagm, eg, *në* ***fshat***, in function of the syntactic head and by a compulsory complement to it. Prepositional syntagms can only have postmodifiers. The premodifiers can be preceding the nominative and verbal forms eg, ***pikërisht*** *në kohë* [33, 19, 34].

The standard way to represent the syntactic structure of a sentence is like a parse tree, which is a representation of all derivation steps of the sentence from the root node. Each internal node on the tree represents an implementation of a grammatical rule [13]. The parse tree for the sentence '*ajo është duke lexuar një libër*' is shown in Figure 1.

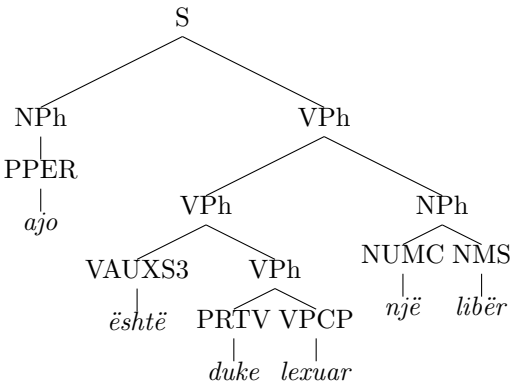

Fig. 1. The tree of the syntagmatic structure for the sentence '*Ajo është duke lexuar një libër*'

The existence of a parse tree proves that a sentence is legal in the grammar, and determines the structure of the sentence. The syntagm structure defines the

most profound linguistic organization of the language. For example, the partition of a sentence into a nominative and verbal syntagm determines the relationship between an action and its agent. This structure plays an essential role in the semantic interpretation [22].

In this paper we use a single model grammar in our examples, which is shown in Figure 2.

Grammar	**Lexicon**
S → NPh VPh	NMS → *libër*, *gjyshi*
NPh → NMS	PPER → *ajo*
NPh → PPER	NFS → *dhembjen*, *ndarjen*, *mbesa*
NPh → NUMC NMS	VS3 → *fshihte*, *ndiente*
NPh → PCL1 NFS	VAUXS3 → *është*
VPh → PRTV VPCP	VPCP → *lexuar*
VPh → PCL1 VS3	PRTV → *duke*, *po*
VPh → VAUXS3 VPh	NUMC → *një*
VPh → PRTN VPh	PRTN → *nuk*
VPh → PRTV VPh	PRP → *për*, *nga*
VPh → VPh NFS	CNJS → *që*
VPh → VPh CNJS VS3	PCL1→ *e*
VPh → VPh PrpPh	
VPh → PrpPh VPh	
VPh → VPh NPh	
PrpPh → PRP NFS	
PrpPh → PRP NPh	

Fig. 2. Selected grammatical rules used in the paper

3.3 Computational methods

The sentences in the natural language are not easily analyzed by the computational programs, as there is substantial ambiguity in the structure of the natural language. It is difficult to prepare formal rules to describe informal behavior even though it is clear that some rules are being implemented [14].

Ongoing, there are the Cocke-Kasami-Younger (CKY) and Earley algorithms that combine knowledge from bottom-up and top-down analysis with dynamic programming to build a context-free parser that recognizes word strings as sentence components and efficiently handle complex inputs [22, 15, 16, 23].

CKY algorithm. The Cocke-Kasami-Younger (CKY) algorithm, first described in 1960 (Kasami 1965, Younger 1967) is one of the simplest algorithms of the context-free parsing. One reason for his simplicity is that he works only with grammars in the Chomsky Normal Form (CNF). CKY is a parser that implements the bottom-up search in its working structure (table). For a sentence of length n, the CKY algorithm works with the triangular upper part of the two-dimensional square matrix τ. First, builds lexical cells $\tau_{i,\,i}$ with the words of the input sentence w_i applying lexical grammatical rules to determine their part of speech tags, then non-lexical cells $\tau_{i,\,k}$ $(i < k - 1)$ are filled up applying binary grammatical rules. In this way, the algorithm starts from the bottom of the parse tree, to reach its root, that is the input sentence itself. The sentence is recognized by the algorithm if $S \in \tau_{0,\,n}$, where S is the start symbol of the grammar. The CKY algorithm requires that the grammar used to perform the analysis be in the Chomsky Normal Form (CNF). Figure 3 gives the complete algorithm.

Algorithm 1: CKY parser

```
1  function CKY-PARSE(words, grammar):
2      for j ← 1 to LENGTH(words) do
3          table[j, j] ← {A|A → words[j] ∈ grammar};
4          for i ← j − 2 downto 0 do
5              for k ← i + 1 to j − 1 do
6                  table[i, j] ← table[i, j] ∪ {A|A → B   C ∈ grammar, B ∈
                     table[i, k], C ∈ table[k, j]};
7              end
8          end
9      end
10 return table;
```

Fig. 3. Pseudocode for CKY algorithm

The algorithm given in the figure 3, is a recognizer, not a parser. To succeed, it simply needs to find a S in cell $\tau_{0,\,n}$, and it simply says if a sentence can be generated by a grammar, but it does not say what the analyzes are. To retirieve the analyzes, additional information on the cells needs to be coded: backpointers pointing back to the two nonterminals that they lead to. The return of an analysis consists of selecting an S from the cell $\tau_{0,\,n}$, and then recursively returning its constituent components from the table.

For this section, we are referred to [17, 13, 16], the pseudocode is adapted from (Daniel & Jurafsky, 2008) [16], from which the practical implementation of the algorithm in JAVA is based.

Earley algorithm. In contrast to the bottom-up search implemented by the CKY algorithm, Earley is a parser that implements the top-down search in its working structure (chart). The essence of the Earley algorithm is a single

pass from left to right, which fills the table with $n + 1$ entries. It begins by generating an initial state with the sentence symbol on the right-hand side of the grammatical rule and generates all the states that correspond to the possible parses for each prediction made, the possible derivations of each of the sentence components (non-terminals for which the grammatical rules exist). For each position of the word in the sentence, the table contains a list of states that represent the partial parse trees that have been created so far. Then, it continues to successively analyze each of the words in the sentence to match the predictions made in the preliminary steps. By the end of the sentence, the table creates all possible analyses of the input. Each of the possible trees is represented only once and can thus be used by all the analyzes it needs. The algorithm considers three procedures (predictor, scanner, and completer [15–17]) to process the states in the table. Each takes a single state as input and derives new states from it. These new states are then added to the table provided they are not already present. The Earley algorithm works for any grammar. Figure 4 gives the complete algorithm.

Algorithm 2: Earley parser

```
1  function EARLEY-PARSE(words, grammar):
2      chart := empty;
3      ENQUEUE(($ → • S, [0, 0]), chart[0]);
4      for i ← 0 to LENGTH(words) do
5          foreach state ∈ chart[i] do
6              if INCOMPLETE?(state) and NEXT-CAT(state) is not a POS
                then
7                  foreach B → γ ∈ GRAMMAR-RULES-FOR(B, grammar)
                    do
8                      ENQUEUE((B → • γ, [j, j]), chart[j]);
9                  end
10             end
11             else if INCOMPLETE?(state) and NEXT-CAT(state) is a
                POS then
12                 if B ∈ PARTS-OF-SPEECH(words[j]) then
13                     ENQUEUE((B → word[j], [j, j + 1]), chart[j + 1]);
14                 end
15             end
16             else
17                 foreach A → α • B β, [i, j] ∈ chart[j] do
18                     ENQUEUE((A → α B • β, [i, k]), chart[k]);
19                 end
20             end
21         end
22     end
23 return chart;
```

Fig. 4. Pseudocode for EARLEY algorithm

The described version of the Earley algorithm is a recognizer, not a parser. After processing, legitimate sentences will leave the state ($\$ \rightarrow S \bullet$, $[0, n]$) in the table. Unfortunately, as it is we have no way to get the structure of this S. To turn this algorithm into a parser, we should be able to extract individual analyses from the table.

For this section, we are referred to [15, 16, 23], the pseudocode is adapted from (Daniel & Jurafsky, 2008) [16] and (Luger & Stubblefield, 2009) [23], from which the practical implementation of the algorithm in JAVA is based.

The algorithms presented here are practically implemented in JAVA, based on the previously referred parts. These algorithms represent the computational method of the syntactic parsing for Albanian, which is developed using the context-free grammar rules. There are currently 1150 grammatical rules (phrasal structure). The algorithms were used to develop an experimental analysis to compare the two algorithms in their functional performance based on the results obtained from the analysis of the sentences in Albanian. The experiment was developed using the data (some example sentences in Albanian) that give the results from where the comparison of the gained results and the conclusion for this analysis is done.

4 Results

The work in the syntactic parser is still in progress. To test the accuracy of the algorithms, we parsed 500 sentences from two texts (a part of a novel [18] and from Albanian literature [1]) by hand and compared the results to the parse trees produced by the parser. These sentences are splited into two groups: simple, and complex sentences. A sample, representing the data in the analysis, is shown in figure 5.

To quantify accuracy, we use the standard measures *precision* (the number of correct results divided by the number oof all returned results) and *recall* (the number of correct resuts divided by the number of results that should have returned). The results were analyzed based on two groups of data for both algorithms.

Table 4 shows the results, and the accuracy of two algorithms.

Table 4. The accuracy of analysis from CKY and Earley algorithms using precision and recall measures

Category	**Sentences**	**CKY**		**Earley**	
		Precision	**Recall**	**Precision**	**Recall**
Simple sentences	300	94%	92%	92%	91%
Complex sentences	200	88%	85%	93%	90%
Parsed sentences	500	91%	89%	93%	91%

In figure 6 the results are displayed graphically.

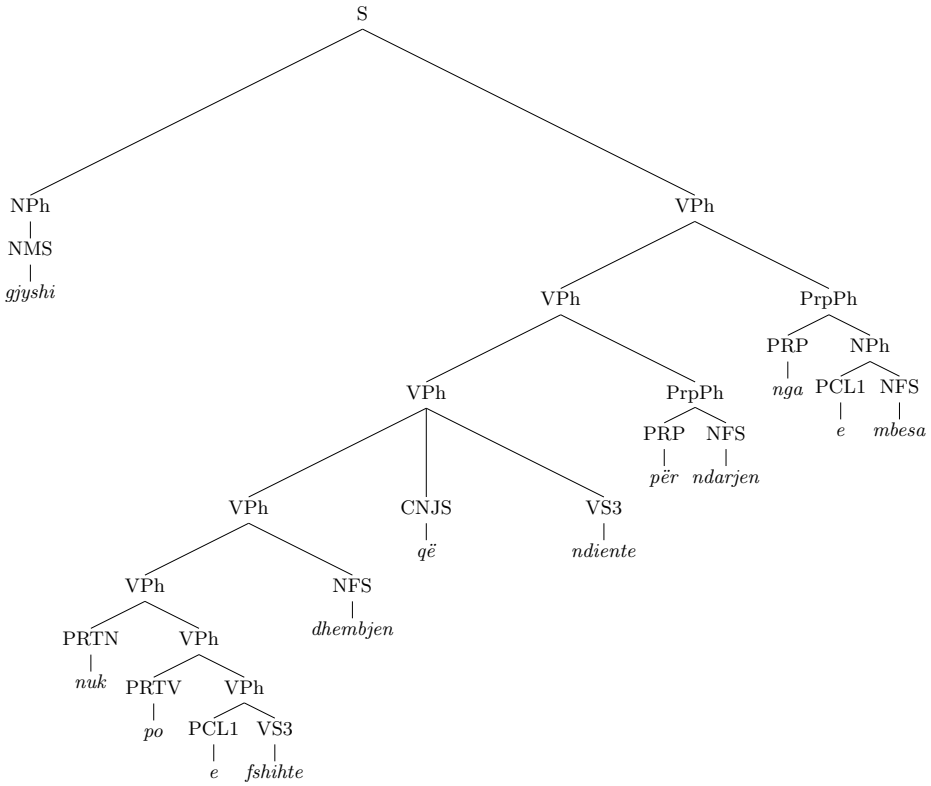

Fig. 5. The tree of the syntagmatic structure for the sentence *Gjyshi nuk po e fshihte dhembjen që ndiente për ndarjen nga e mbesa.*

The numerical results of the analysis for both algorithms, table 4, show that the accuracy of both algorithms falls according to the complexity of the sentences used in the evaluation. Simple sentences have higher accuracy because they do not have different types of complex phrases. Complex sentences have the lowest accuracy, due to the variety of complex constituents.

As we can see from table 4, the CKY algorithm has higher accuracy in simple sentences, and the Earley algorithm performs more accurate analyses in complex sentences. Why does this happen?

Since the Earley algorithm uses the top-down analysis knowledge, it begins by analyzing a wider set of analyses. This has the advantage in the case of coordination constructions. In the case of the CKY algorithm, due to its dependence on binary rules, it can return the incorrect analysis in cases where the binary combination in coordinate constructions is grammatically legitimate.

The Earley algorithm examines the trees that may result in the input sentence and it generates those trees only. On the other hand, top-down analysis spends much effort on trees that are inconsistent with the input. This weakness in top-

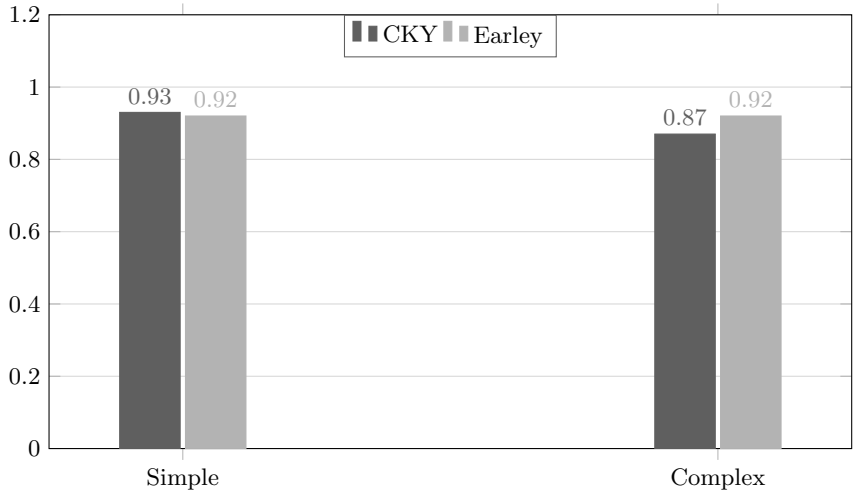

Fig. 6. Results of algorithms in graphic form

down parsers stems from the fact that they can generate trees before considering the input, and so analyses can be provided where the order of the grammatical constituents in the analysis is different.

On the other hand, top-down analysis in the case of the Earley algorithm shows a weakness due to the structural ambiguity.

The analysis of input sentences as binary combinations has shown to be effective due to the legitimate grammatical components that form them. Bottom-up parsers do not suggest trees that are not based on the current entry and they remove the trees that cannot lead to an S.

From the results showed in table 4, we conclude that in the test set, the Earley algorithm has higher accuracy due to the higher accuracy provided in complex sentences.

Also, each of the two algorithms implement different search strategies in syntax parsing, which with their advantages and disadvantages, affect the result of the evaluation.

The algorithms used here work accurately in most cases, but we should consider the usage of ambiguous grammar in phrasal combinations with the application of grammatical rules. In some cases, it gives the wrong parse tree because the tagger does not parse correctly.

The way algorithms perform in the most complex cases is very efficient in terms of accuracy. Although it is a problem when returning the result, the number of parses that the algorithm can build is amazing, which humans couldn't find evidently by applying the natural language's grammar and for which they are not even aware.

5 Conclusion and Future Work

The paper gives a basic study for syntactic parsing, implemented for Albanian, considering that such research has not been developed earlier in this field, for Albanian. Therefore, it is possible to improve and develop a more general model of the problem, using more efficient methods of solution.

Given that we are dealing with natural language accompanied by the enrichment of the language lexicon, the data given to the machine for learning constitutes a deficient amount in the generalization of grammar with wide coverage, for any natural language. Consequently, the development of an efficient linguistic tool is a challenging task for every researcher in this field. However, it's still possible to give more general and effective solutions to the problem. An efficient solution is the development of an analyzer that tries to produce useful output, such as a partial analysis, even if the input is not covered by the grammar.

The application of machine learning methods can give more efficient solutions to the problem of syntactic parsing, which was impossible for us to develop due to the lack of large amounts of data.

References

1. Beci, B.: Gramatika e gjuhës shqipe. Focus, Prishtina (2005)
2. Butt, M., King, T.H.: chap. Grammar Writing, Testing, and Evaluation. CSLI Publications (9 2002)
3. Caka, A., Neziri, V.: Computer model algorithm of the albanian language tagger. Conference of Engineering Sciences and Information Technology (8 2011)
4. Callison-Burch, C., Osborne, M.: chap. Statistical Natural Language Processing. CSLI Publications (2 2003)
5. Cer, D., de Marneffe, M.C., Jurafsky, D., Manning, C.D.: Parsing to stanford dependencies: Trade-offs between speed and accuracy. Computer Science Department, Stanford University, Stanford, CA 94305, USA
6. Chopra, A., Prashar, A., Sain, C.: Natural language processing. International Journal of Technology Enhancements and Emerging Engineering Research **1**(4), 131–134 (2013)
7. Clark, A., Fox, C., Lappin, S.: The Handbook of Computational Linguistics and Natural Language Processing. Wiley-Blackwell, Singapore (2010)
8. Collins, M.: chap. Tagging Problems and Hidden Markov Models. Columbia
9. Dushi, O.: Terma e koncepte gjenerative në sintaksën e gjuhës shqipe. Journal of Institute Alb-Shkenca **6**(4), 119–123 (2013), www.alb-shkenca.org
10. Green, S., Manning, C.D.: Better arabic parsing: Baselines, evaluations, and analysis. Computer Science Department, Stanford University, Stanford, CA 94305, USA
11. Green, S., de Marneffe, M.C., Bauer, J., Manning, C.D.: Multiword expression identification with tree substitution grammars: A parsing tour de force with french. Computer Science Department, Stanford University, Stanford, CA 94305, USA
12. Hadaj, G., Hysaj, G.: An overview of generative grammar in albanian. Academic Journal of Interdisciplinary Studies **4**(2), 231–235 (8 2015)
13. Indurkhya, N., Damerau, F.J.: Handbook of Natural Language Processing. CRC Press, United States of America, 2 edn. (2010)

14. Jurafsky, D.: A probabilistic model of lexical and syntactic access and disambiguation. Cognitive science **20**(2), 137–194 (4 1996)
15. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice-Hall, United States of America (2000)
16. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice-Hall, United States of America (2008)
17. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. 3 edn. (2017)
18. Kadare, I.: Ura me tri harqe. Onufri, Albania (1978)
19. Koskoviku, B.: Struktura dhe funksionet gramatikore të sintagmës parafjalore në gjuhën shqipe. In: Paçarizi, R. (ed.) Seminari XXXIV Ndërkombëtar për Gjuhën, Letërsinë dhe Kulturën Shqiptare. pp. 459–464 (8 2015)
20. i Gjuhësisë dhe Letërsisë, I.: Gramatika e Gjuhës Shqipe, vol. 2. The Academy of Sciences, Tirana (2002)
21. Levy, R., Manning, C.D.: Is it harder to parse chinese, or chinese treebank? Computer Science Department, Stanford University, Stanford, CA 94305, USA
22. Luger, G.F.: Artificial Intelligence: Structures and Strategies for Complex Problem Solving. Pearson, United States of America, 6 edn. (2009)
23. Luger, G.F., A.Stubblefield, W.: AI Algorithms, Data Structures and Idioms in Prolog, Lisp, and Java. Pearson, Boston (2009)
24. Lumezi, L.: Përqasje e ndërtimeve me sintagma emërore dhe parafjalore në gjuhët angleze dhe shqipe. Ph.D. thesis, The University of Tirana, Tirana (6 2012)
25. Manning, C.D., Schutze, H.: Foundations of Statistical Natural Language Processing. The MIT Press, England (2000)
26. de Marneffe, M.C., MacCartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure parses. Computer Science Department, Stanford University, Stanford, CA 94305, USA
27. Millaku, S.: The noun phrases. Anglisticum Journal **2**(6), 38–47 (12 2013)
28. Morozova, M., Rusakov, A.: Korpusi elektronik i shqipes: Përpunimi, përmbajtja dhe përdorimi. In: Rugova, B. (ed.) Seminari XXXII Ndërkombëtar për Gjuhën, Letërsinë dhe Kulturën Shqiptare. pp. 85–95. Faculty of Philology, Prishtina (8 2013)
29. Rafferty, A.N., Manning, C.D.: Parsing three german treebanks: Lexicalized and unlexicalized baselines. Computer Science Department, Stanford University, Stanford, CA 94305, USA
30. Ratnaparkhi, A.: Maximum Entropy Models for Natural Language. Ph.D. thesis, The University of Pennsylvania, Pennsylvania (1998)
31. Raza, A.A., Habib, A., Ashraf, J., Javed, M.: A review on urdu language parsing. International Journal of Advanced Computer Science and Applications **8**(4), 93–97 (2017), www.ijacsa.thesai.org
32. Socher, R., Bauer, J., Manning, C.D., Ng, A.Y.: Parsing with compositional vector grammars. Computer Science Department, Stanford University, Stanford, CA 94305, USA
33. Spahiu, A.: Togfjalëshi dhe sintagma (10 2002)
34. Taçi, J.: Caktimi i rasave në sintagmat emërore të gjuhës angleze në përqasje me gjuhën shqipe. Ph.D. thesis, The University of Tirana, Tirana (2015)
35. Trommer, J., Kallulli, D.: A morphological tagger for standard albanian. 4th International Conference on Language Resources and Evaluation (1 2004)