Detecting Emotions in Tweets Based on Hybrid Approach

Detecting emotions in tweets based on hybrid approach

Ivona Najdenkoska
Faculty of Computer Science and Engineering,University of Ss. Cyril and Methodius
Skopje, R. Macedonia
najdenkoska.ivona@students.finki.ukim.mk

Frosina Stojanovska
Faculty of Computer Science and Engineering, University of Ss. Cyril and Methodius
Skopje, R. Macedonia
stojanovska.frosina.1@students.finki.ukim.mk

Prof. Sonja Gievska Ph.D
Faculty of Computer Science and Engineering,University of Ss. Cyril and Methodius
Skopje, R. Macedonia
sonja.gievska@finki.ukim.mk

Abstract—**Emotion detection from text is increasingly popular nowadays, especially when it comes to human-computer interaction. It is one of the great areas for recognition of the human emotional state and it has a potential application in many other vast areas such as computer vision, psychology, physiology etc. In this paper, we will try to recognize emotions from posts on the popular social network Twitter also known as tweets. The emotions will be represented with four classes of emotions: anger, fear, joy, and sadness, with additional neutral class, and we will try to recognize them. For solving the problem, we will use a hybrid approach. This approach incorporates concepts of two major areas, natural language processing (NLP) with its linguistic models and more diverse machine learning (ML) algorithms**.

Keywords— Emotion detection, Tweets, WASSA dataset, Hybrid approach, Natural language processing, Machine learning

I. Introduction

Emotion detection is one of the great areas for recognizing of the human emotional state, that is being explored today and is intended for creating new methods when solving problems from this area. This field is part of the wider domain sentiment analysis. The goal of sentiment analysis is to identify positive and negative opinions in free text and to associate this opinion with relevant objects. The goal might be in the sense of identifying what and how something is discussed (e.g., which aspects of a car are liked or disliked), or the goal might be a judgement in the sense of diagnosing the nature and strength of opinion (e.g., diagnosing how much a reviewer liked a film from their online review) [1].

A. The definition of emotion

Emotions are psychical processes that influence the human's direction in carrying out his actions. They represent the mental state of the psychological stimulus and are manifested through the expression of somatic and autonomous responses. Emotion is often intertwined with mood, temperament, personality and motivation. The physiology of emotion is closely linked to arousal of the nervous system with various states and strengths of arousal relating to emotions.

B. Emotion detection from text

Emotion detection is the process of identifying human emotions from various sources like text, audio, image etc. It refers to something that humans do automatically, but also there are computational methodologies that are being developed. Humans show universal consistency in recognizing emotions, but also show great variability, which is a major topic of study in psychology. Sources used as data for this kind of problem are enormous. Starting with some documents, e-mails, blogs, comments on websites pages, and even posts, comments, messages through social networks. Concepts from natural language processing (NLP), as well as machine learning (ML) algorithms, are commonly used for analysing and extraction of emotion from a text [2]. The computers use a lot of different methods for interpreting emotions, usually according to the Paul Ekman's Facial Action Coding System.

C. Models for emotional representation

Emotions in this context are typically represented by two main models: emotional categories and emotional dimensions, shown in Fig. 1. With the emotional categories model, emotions are presented with distinct emotion classes or labels, like Ekman's approach [3] with six classes of emotions: anger, disgust, fear, happiness, sadness, and surprise. On the other hand, there are other models that represents the emotions in dimensional form where each emotion occupies a certain part of the space. That is the approach of Russell with his Circumplex model, where the emotions are represented in space with two dimensions: the valence of emotion, which indicates whether the emotion is positive or negative, and the arousal of the emotion, indicating the level of energy associated with the emotion. ([4], [5]).

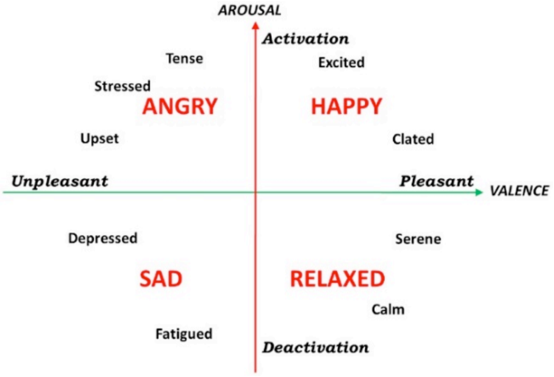

Fig 1. Representation of the Circumplex model

II. Methodology

A. Dataset

The dataset that we are using for the experiment is provided for the Workshop on Computational Approaches to Subjectivity, Sentiment, and Social Media (WASSA), by the researchers in computational linguistics Saif M. Mohammad and Felipe Bravo-Marquez [14]. There are four different training and test datasets, for the four emotions respectively. The four emotions that are used for the classification of the tweets are: anger, fear, sadness, and joy. These four types of emotions are the focus of this experiment. Every dataset consists of four attributes: tweet id, tweet, emotion type and intensity, shown in *Table 1*. Firstly, we decided to merge the four separate datasets into one, because our goal is to classify each tweet to an emotion. After merging the datasets, we have one training dataset which is consisted of 3613 tweets, labelled with their emotion intensity and emotion type as a class attribute. On the other hand, the test dataset consists of 3142 tweets labelled with the same attributes.

After reviewing some of the provided tweets and their labelled emotion and intensity we decided to remove certain number of them and add a neutral emotion. This decision is due to our notice that sometimes the emotion label does not provide the proper emotion for a specific tweet. For example, the following tweet: "*Don't #worry if you're not the best, if you are doing something you #love, you're heading in right direction ...*" is labelled as fear, but clearly, this sentence is wisdom quote and it would be more suitable to label it with a neutral emotion. The intensity of the labelled emotion for this tweet is 0.104. Because of the low intensity and the meaning of the sentence, we can easily conclude that it is better to classify it as a default class, i.e. neutral emotion, and so we are not taking this kind of tweets in consideration. To manage this challenge, we define a threshold of 0.34 for the emotion intensity. Each tweet that has an intensity above this threshold is retained in the dataset, and each tweet that has an intensity below this threshold is omitted. Adding an additional neutral emotion will help us to build more accurate classifier because if the tweet does not provide enough information for the detection of emotion, the classifier will choose the default class.

This dataset doesn't have a label for neutral emotion, so we decided to find proper tweets from some other dataset, and append them to the existing dataset. In addition, we found a dataset for sentiment classification from Sentiment140 [1], provided by Alec Go, Richa Bhayani, and Lei Huang [15]. This dataset consists of tweets with positive, negative and neutral class for the sentiment of the tweet, so for our purpose, we get the tweets from the test set with neutral class and add them to the initial dataset. After this, we ended up with a dataset containing 5348 tweets labelled with *five different classes* for the emotional state shown in Fig. 2 with their distribution.

Table 1. Dataset attributes and their types

Attribute	Type
Tweet id	Integer
Tweet	String
Emotion Intensity	Numeric
Emotion type	Nominal

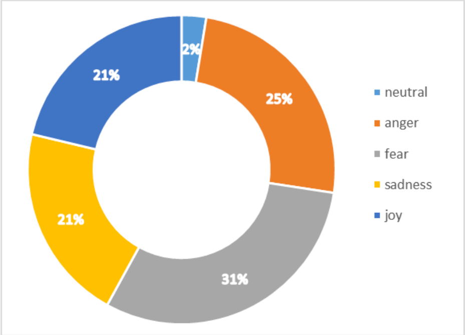

Fig. 2. Distribution of emotion classes in the final dataset

B. Hybrid approach

As mentioned before, usually concepts from the big fields of Natural Language Processing (NLP) and Machine Learning (ML) are used for text analysing and identifying emotions in text. The main idea of the hybrid approach is to combine methods from these two big fields and make a model that will join the advantages and try to improve the individual disadvantages from both separate approaches. Using NLP techniques, we are going to produce initial features of the tweets, and then after the feature selection, we will use the selected features to build a model with ML algorithms that can classify emotions of tweets.

C. Lexicon

Emotion lexicons are lists of words which have been labelled according to their emotional connotation. The label can simply be an emotion category the word is thought to belong to, or it can be a value representing the strength of a given emotional dimension reflected in the word [16].

After we finished our research in the field of emotion lexicons, we decided to use Warriner et al. extended ANEW (Affective Norms for English Words) lexicon[2]. This lexicon consists of affective norms for valence, arousal and dominance for 13,915 English words (lemmas), which were collected by Amazon Mechanical Turk [17]. The reason for choosing this lexicon is because it uses three dimensions (valence, arousal and dominance) for annotating the words. These dimensions are represented in the PAD (Valence - Pleasure, Arousal, Dominance) model [18]. Valence (also referred to as the pleasure dimension) refers to whether an emotion is positive or negative. Arousal refers to the intensity of which the emotion is experienced or expressed. Both dimensions are independent,

[1] http://help.sentiment140.com/home

[2] http://crr.ugent.be/archives/1003

in that the valence of an emotion does not affect its activation and vice versa. Dominance is a dimension that represents the controlling and dominant versus controlled or submissive one feels. In this paper, we will use the first two dimensions for generating the features.

D. Preprocessing

The first step in our approach is the preprocessing the text from the tweets. The tweets are presented as sentences that have mis-spellings and casual language used in Twitter. So, to "clean up" the text we preprocessed the tweets with the following rules:

- Tweets often contain usernames, words that start with the @ symbol. These words are removed from the tweet because they don't provide valuable information within our approach.
- Hash tags in tweets in most of the cases are representative to the emotion expressed in the text, so we decided to remove the # symbol from the hashtag and keep the rest of the hashtag.
- Another common incorrectness in tweets are words with repeated letters such as "loveeeee". Any letter occurring more than two times consecutively is replaced with one occurrence. For instance, the word "loveeeee" would be changed into "love".

Some of the tweets include several types of emojis which are annotated with special characters. We decided to ignore the emojis because they are not applicable within our approach.

E. Feature extraction

To train a model that would be able to classify emotions from tweets, we represent each tweet with a vector of features. This vector needs to capture the emotion expressed by each tweet. Therefore, in this paper, we explore the usage of valence and arousal dimensions of individual words, obtained from our chosen lexicon, as features in the vector. For the preprocessing of the dataset and feature extraction and selection, we are using the programming languages Java and Python. Also, we use few additional libraries. For example, to work with our dataset we are using the Pandas[3] library for data analysis in Python [19].

The steps of feature extraction from the tweets are:

1. Tokenization
2. Part-of-speech (POS) tagging
3. Lemmatization
4. Adding initial valence and arousal values from the lexicon for every word
5. Lexical dependency parsing and identifying grammatical relations between words
6. Changing initial valences with context shifters rules
7. Removing stop words and creating features from valence and arousal values of words.

In this section, we are going to describe these steps individually and then explain the process of selecting the best features, i.e. feature selection.

Often natural language processing tools require their input to be divided into tokens. So firstly, we applied tokenization [21] of the tweets. We divided each tweet into tokens using the tokenize method from the class TweetTokenizer in nltk.tokenize module [4] from the NLTK Python library. Separation of tokens is made by separating commas, quotation marks from words and disambiguating end-of-sentence punctuation (period, question mark, etc.). Before applying the lemmatization, we needed to tag the words with Part-of-speech (POS) tagging [22]. A POS-tagger processes a sequence of words and attaches a part of speech tag to each word. For this part, we use the method pos_tag[5] from NLTK library.

After that, the following step was the lemmatization of each word. In linguistic morphology and information retrieval, stemming is the process of reducing inflected (or sometimes derived) word to their word stem, base or root form. Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma [21]. This step was also achieved with NLTK Python library using the function lemmatize from the WordNetLemmatizer [6] class. This lemmatizer requires the tokens and their POS tag as parameters and returns the root of the words (lemmas).

With the three previously explained steps we had tokens and lemmas for each tweet. Using the lemmas, we assigned each word with the values of dimensions from the lexicon that we have chosen before. Initially, the lexicon had a scale with range 0-9 for the two dimensions. We shifted this scale to a range (-4.5)-4.5 to fit with our approach.

After the initial valence is assigned to the words from the tweets, we started with the process of identifying the grammatical dependencies and modifying the initial valence. This was important for implementation of the rules for contextual valence shifters. The base attitudinal valence of a lexical item is modified by lexical and discourse context, so we need to implement predefined rules as valence context shifters ([23], [6], [24]). The Stanford parser[7] [25] was utilized for performing the lexical dependency parsing and identifying grammatical relations between words. These dependencies were used to modify the initial valence of words with rules for contextual shifters, consisted of:

1. Negatives (negation) – If a word is in relation with negatives (e.g. not, never, nothing), then the initial valence of the word is shifted, i.e. is multiplied by -1.
2. Intensifiers – Intensifiers are adverbs or adverbial phrases that strengthen the meaning of other expressions and show emphasis. The words that are used in this paper as intensifiers are: deeply, always, absolutely, completely, extremely, highly, rather, really, so, too, totally, utterly, very, extraordinarily etc. If there is an intensifier in the tweet, then the valence of the word that is in relation with

[3] http://pandas.pydata.org

[4] http://www.nltk.org/api/nltk.tokenize.html

[5] http://www.nltk.org/book/ch05.html

[6] http://www.nltk.org/api/nltk.stem.html

[7] https://nlp.stanford.edu/software/lex-parser.shtml

the intensifier is increased by multiplying the initial valence with 1,5.

3. Mitigators (Downtoners) – Mitigators or downtoners are words that reduce the force of another word or phrase in the sentence. The words that are used in this paper as mitigators are: fairly, somewhat, rather, quiet, lack, least, less, slightly, a little etc. If there is a mitigator in the tweet, then the valence of the word that is in relation with the mitigator is decreased by multiplying the initial valence with 0,5.
4. Conjunctive adverbs – A conjunctive adverb is a word that connects two sentences together, making a new sentence. It is like the word "and", but it adds more meaning to the sentence. If there is a conjunctive adverb in the tweet, then the valences are neutralized by multiplication with 0.
5. Negative words – If the word is in a relation with a negative word, then it is multiplicated with -1 only if the word has positive valence. Otherwise, the valence remains the same.

After implementing these rules and calculating the valence of the words, we generate a vector of initial features. These features are valence and arousal values of every word appearing in every tweet. After removing the stop words using the set of stop words from NLTK library, we ended up with 8989 different words appearing in the tweets, so the vector was of length 17978, containing valence and arousal values for every word. The valence value was shifted by 10 units and the arousal value by 5 units, so these values are positive in every case. If some tweet doesn't include a particular word, then the valence and arousal values for this word are set to -1.

F. *Feature selection*

Feature selection is a process for automatically selecting those features in the data that contribute most to the prediction variable [26]. To perform feature selection we are using the sklearn.feature_selection [8] module from the Scikit Learn Python Library [27]. The choice of attributes is one of the most significant processes for reducing the dimensionality of data by ranking all possible attributes and selecting those with the highest value.

There are three general classes of feature selection algorithms: *filter methods*, *wrapper methods* and *embedded methods*. Tree-based estimators can be used to calculate features importance, which in turn can be used to dismiss unnecessary features. We are using this kind of estimator to compute importance of every valence feature and retain the most ranked words according to their importance. After the ranking of the words, we defined a threshold of 0.0005 and kept the words that have importance value above this threshold. The number of words that we had after the elimination was 290. Then for these words, we kept the valence and arousal features, and so we ended up with 580 features plus the class.

G. *Training the model*

After the feature selection, we have feature vector for every tweet. The feature vector is of length 581 including the class (290 valence features and 290 arousal features to the corresponding words). The next step is the training of the model for classification of emotions. In this paper, we will use Weka, a software for knowledge mining, for building the model [28]. To test the built model, we decided to use cross-validation with 10 folds.

For the classification, we used four different classifiers including, Linear SVM (Support Vector Machines), Multilayer perceptron with one hidden layer (approximate sigmoid as activation function and squared error as loss function), Random Forest and LDA. We decided to work with these classifiers because they are capable to handle high dimensional feature vectors and obtain good performance with them.

III. Evaluation and Results

The results obtained from the evaluation of the hybrid approach with the dataset are given in this section. As mentioned in the previous part, we are going to test the models with cross-validation with 10 folds. After the testing, we are evaluating the models and comparing their performances. The metrics that we use to evaluate our models are: Precision, Recall, and F-measure. In *Table 2* we can see number of correctly and incorrectly classified instances of the models. In *Table 3 – 5* are presented the results from the model evaluation according to the metrics. Generally, all the classifiers have a problem with classifying anger as fear furthermore, fear as sadness and vice versa. Also, surprisingly for us, there is a problem when the models classify joy as fear. This could be due to the dominance of the fear class in the dataset.

With the results obtained by the measures, we can conclude that in general SVM, Random Forest and LDA have approximately equivalent performances. Random Forest has the best performance for classification of the neutral class, and all the models have a difficulty by separating the negative emotions especially anger and fear. From all the classifiers LDA has the highest overall accuracy. Furthermore, the Multilayer perceptron model ignores the neutral class and has the lowest performance from all the models.

Table 2. Number of correctly and incorrectly classified instances of the models

	Correctly classified instances	Incorrectly classified instances
Linear SVM	4415	933
Random forest	4358	990
LDA	4453	895
Multilayer perceptron	4231	1117

[8] http://scikit-learn.org/stable/modules/feature_selection.html

Table 3. Precision metric for the model evaluation

	Joy	Neutral	Anger	Fear	Sadness
Linear SVM	0.927	0.85	0.884	0.729	0.84
Random forest	0.88	0.845	0.844	0.769	0.784
LDA	0.971	1	0.935	0.703	0.844
Multilayer perceptron	0.849	0	0.814	0.752	0.768

Table 4. Recall metric for the model evaluation

	Joy	Neutral	Anger	Fear	Sadness
Linear SVM	0.836	0.783	0.793	0.889	0.766
Random forest	0.882	0.79	0.799	0.843	0.726
LDA	0.833	0.71	0.781	0.916	0.786
Multilayer perceptron	0.856	0	0.784	0.834	0.768

Table 5. F-measure for the model evaluation

	Joy	Neutral	Anger	Fear	Sadness
Linear SVM	0.879	0.815	0.836	0.801	0.801
Random forest	0.881	0.816	0.821	0.804	0.754
LDA	0.897	0.831	0.851	0.795	0.814
Multilayer perceptron	0.853	0	0.799	0.791	0.768

IV. CONCULUSION AND FUTURE WORK

Emotion detection is one of the most attractive topics of research and experimentation in recent times. Although the definition of emotion is fuzzy, it is still evident that it has been recognized and proven that emotions affect people, especially in their reasoning and decision making about their actions.

In this paper, we have studied the problem of emotion detection from Twitter posts known as tweets. We present a hybrid approach which combines elements from the big fields Natural Language Processing and Machine Learning to classify emotions. The first step in our approach is the preprocessing of the tweets. This is a key step that cannot be excluded, especially because tweets are written with the casual language used in Twitter and so have misspellings and incorrect word writings. Also, tweets have their unique characteristics like usernames and hashtags that need to be handled. After the preprocessing, the feature extraction was made with several concepts from NLP, and then the dimension of the created vector with initial features was reduced with feature selection. For training a model and its evaluation we used four ML classifiers: Linear SVM (Support Vector Machines), Multilayer perceptron, Random Forest and LDA, and compared their performance. The results show that in general SVM, Random Forest and LDA have approximately equivalent results, and LDA has the highest overall accuracy. On the other hand, the Multilayer perceptron model ignores the neutral class and has the lowest performance.

We consider that this approach can be improved if we take the emojis and emoticons into consideration. Also, because tweets have extremely irregular language we can further explore approaches for fixing these irregularities. Another thing that could be applied to improve the performance is to split the hashtags. Hashtags are not always consisted of one word. They can have multiple joint words in them that need to be separated. We take the hashtags, that may consist multiple words, for calculating our features and by separating them we could end up with words and features that are important for the emotion of the tweet.

REFERENCES

[1] Mike Thelwall, David Wilkinson, Sukhvinder Uppal (2010), “Data Mining Emotion in Social Network Communication: Gender differences in MySpace”, in *Journal of the Association for Information Science and Technology*, p. 190-199

[2] Lea Canales and Patricio Martinez-Barco (2014), “Emotion Detection from text: A Survey”, in *Proceedings of the Workshop on Natural Language Processing in the 5th Information Systems Research Working Days (JISIC)*

[3] P. Ekman (1992), “An argument for basic emotions,” in *Cognition & Emotion*, Vol. 6, No. 3, pp. 169-200

[4] James A. Russell (1980), “A Circumplex Model of Affect”, in *Journal of Personality and Social Psychology*, Vol. 39, No. 6, pp. 1161- 1178

[5] S. Buechel, U. Hahn (2016), “Emotion Analysis as a Regression Problem – Dimensional Models and Their Implications”, in *European Conference on Artificial Intelligence (ECAI)*, pp. 1114-1122

[6] Sonja Gievska, Kiril Koroveshovski and Tatjana Chavdarova (2014), “A Hybrid Approach for Emotion Detection in Support of Affective Interaction”, in *IEEE International Conference on Data Mining Workshop*

[7] Saif M. Mohammad, Felipe Bravo-Marquez (2017), “Emotion Intensities in Tweets”, in *Proceedings of the Sixth Joint Conference on Lexical and Computational Semantics (*Sem)*

[8] Daniel Preotiuc-Pietro, H. Andrew Schwartz, Gregory J. Park, Johannes C. Eichstaedt, Margaret L. Kern, Lyle H. Ungar, Elisabeth Shulman (2016), “Modelling Valence and Arousal in Facebook posts”, in *WASSA@NAACL-HLT*

[9] Jonathan Posner, James A. Russell, Bradley S. Peterson (2005), “The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology”, in *Development and Psychopathology*, Vol. 17, pp. 715-734

[10] Amy Beth Warriner, Victor Kuperman, Marc Brysbaert (2013), “Norms of valence, arousal and dominance for 13,915 English lemmas”, in *Behavior Research Methods*, Vol. 45, pp. 1191–1207

[11] Taboada, M., J. Brooke, M. Tofiloski, K. Voll, M. Stede (2011), “Lexicon-Based Methods for Sentiment Analysis”, in *Computational Linguistics*, Vol. 37, pp. 267-307

[12] Peter D. Turney, Michael L. Littman (2003), “Measuring Praise and Criticism: Inference of Semantic Orientation from Association”, in *ACM Transactions on Information Systems (TOIS)*, Vol. 21, pp. 315-346

[13] Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani (2010), “”, in *Proceedings of the International Conference on Language Resources and Evaluation, LREC*

[14] Saif M. Mohammad, Felipe Bravo-Marquez (2017), "WASSA-2017 Shared Task on Emotion Intensity.", in *Proceedings of the EMNLP 2017 Workshop on Computational Approaches to Subjectivity, Sentiment, and Social Media (WASSA)*

[15] Go, A., Bhayani, R. & Huang, L. (2009), "Twitter Sentiment Classification using Distant Supervision", in *Processing*, 1-6

[16] Vaassen, F. (2014), "Measuring Emotion: Exploring the feasibility of automatically classifying emotional text", Thesis, University of Antwerp

[17] Warriner A., Kuperman V., Brysbaert M. (2013), "Norms of valence, arousal, and dominance for 13,915 english lemmas", in Behavior Research Methods, Vol. 45, pp. 1191-1207

[18] Albert Mehrabian (1996), "Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in Temperament", in *Current Psychology*, Vol. 14, pp. 261-292

[19] Pandas, Python data analysis library, http://pandas.pydata.org

[20] NLTK, Natural Language Toolkit, http://www.nltk.org

[21] Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze, "Introduction to Information Retrieval", Cambridge University Press, 2009

[22] Daniel Jurafsky, James H. Martin, "Speech and Language Processing", Chapter 10, 2016

[23] Livia Polanyi, Annie Zaenen (2006), "Contextual valence shifters", in *Computing Attitude and Affect in Text: Theory and Applications. The Information Retrieval Series*, Vol. 20, pp. 1-10

[24] Kiril Koroveshovski (2014), "System for emotion receognition in real time based on user written text analysis", Thesis, University of St. Cyril and Methodius

[25] M.C. De Marneffe, D.C. Manning (2012), "Stanford typed dependencies manual", v. 2.0.4. Technical report, Palo Alto

[26] Isabelle Guyon, André Elisseeff (2003), "An Introduction to Variable and Feature Selection", in *Journal of Machine Learning Research*, pp. 1157-1182

[27] Scikit-learn, Python Machine Learning library, http://scikit-learn.org/stable/index.html

[28] Weka 3: Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka