Twitter Sentiment Analysis: Lexicon Method, Machine Learning Method and Their Combination

Chapter · July 2015

Source: arXiv

4 authors:

Olga Kolchyna
University College London
4 PUBLICATIONS **43** CITATIONS
SEE PROFILE

Thársis T. P. Souza
University College London
16 PUBLICATIONS **50** CITATIONS
SEE PROFILE

Philip Treleaven
University College London
82 PUBLICATIONS **1,665** CITATIONS
SEE PROFILE

Tomaso Aste
University College London
179 PUBLICATIONS **4,073** CITATIONS
SEE PROFILE

Methodology for Twitter Sentiment Analysis

Olga Kolchyna[1], Thársis T. P. Souza[1], Philip C. Treleaven[12] and Tomaso Aste[12]

[1] Department of Computer Science, UCL, Gower Street, London, UK,
[2] Systemic Risk Centre, London School of Economics and Political Sciences, London, UK

Abstract. This paper presents a step-by-step methodology for Twitter sentiment analysis. Two approaches are tested to measure variations in the public opinion about retail brands. The first, a lexicon-based method, uses a dictionary of words with assigned to them semantic scores to calculate a final polarity of a tweet, and incorporates part of speech tagging. The second, machine learning approach, tackles the problem as a text classification task employing two supervised classifiers - Naive Bayes and Support Vector Machines. We show that combining the lexicon and machine learning approaches by using a lexicon score as a one of the features in Naive Bayes and SVM classifications improves the accuracy of classification by 5%.

Keywords: sentiment analysis, social media, Twitter, natural language processing

1 Introduction

Sentiment analysis is a line of research that allows to determine people's attitude and opinions in relation to different topics, products, services, events, and their attributes. L. Bing [1] highlights that in the research literature it is possible to see many different names, e.g. *" sentiment analysis, opinion mining, opinion extraction, sentiment mining, subjectivity analysis, affect analysis, emotion analysis, review mining"*, however all of them have similar purposes and belong to the subject of sentiment analysis or opinion mining.

The role of sentiment analysis has been growing significantly with the rapid spread of social networks, microblogging applications and forums. Today, almost every web page has a section for the users to leave their comments about products or services, and share it with friends on Facebook, Twitter or Pinterest - something that was not possible just few years ago. Mining this volume of opinions provides information for understanding collective human behaviour. An increasing amount of evidence is pointed out that by analyzing sentiment of the social-media content it might be possible to predict size of the markets, results of marketing campaigns and marketing ROI.

One of the most popular microblogging platforms is Twitter. It has been growing steadily for the last seven years and has become a meeting point for a diverse range of people: students, professionals, celebrities, companies, politicians. This popularity of Twitter results in the enormous amount of information being passed through the service, covering a wide range of topics from people well-being to the opinions about the brands, products, politicians and social events. In this contexts Twitter becomes a powerful tool for predictions. For example, the researchers at HP tried to predict the level of ticket sales for movies based on Twitter information [2]. The team managed to predict the revenue of the opening weekend with 97.3% accuracy, a prediction rate higher then the one achieved by the Hollywood Stock Exchange, a known prediction tool for the movies.

In this paper, we present a step-by-step approach for sentiment analysis of Twitter data. The methodology is applied to a dataset of more than 100 million Twitter messages that mention the names of 223 companies from fashion and sport sector collected during the period of one year, from November 1, 2013, to October 31, 2014. We propose a new approach of sentiment classification based on the combination of existing commonly-used techniques (lexicon-based and machine learning based) which outperforms standard benchmarks. Further, our method was successfully applied in the recent case study of [3], in which the authors found statically-significant information between retail brands' Twitter sentiment and movements in the stocks of the related companies.

The paper is organised as follows: section 2 covers literature review of the available techniques for sentiment analysis: lexicon-based and machine-learning based methods. Data pre-processing (tokenization, stop-words removal, part of speech tagging) is explained, including analysis of the impact of each of the pre-processing steps on the final sentiment score result. In the Lexicon-based sub-section the process of the lexicon creation and enhancement is described

in detail, with accuracy results presented for different lexicons combinations. Machine Learning sub-section explains the processes of feature generation and feature selection. Performance results of two algorithms are presented: Nave Bayes and Support Vector Machines.

2 Sentiment Analysis Methodology

The field of text categorization appeared long time ago [4], however categorization based on sentiment perhaps first appeared in ([5]; [6]; [7]; [8]; [9]; [10]).

The standard approach to text categorization [4] has been the bag-of-words method (BOW). According to the BOW model, the document gets a representation as a vector of words in Euclidian space where each word is independent from others. This bag of individual words is commonly called a collection of unigrams. The independence of unigrams means that the appearance of one unigram in the text will not influence the appearance of any other unigram.

Even though many advanced algorithms have been developed during the last thirty years (see [11]; [12]; [13]; [14]), the bag-of-words model is the most popular one, because it is easy to understand and it allows to achieve high performance (for example, the best results of multi-lable categorization for the Reuters-21578 dataset were produced using BOW approach, [15]; [16]).

The main two methods of sentiment analysis, lexicon-based method and machine learning based approach, both rely on the bag-of-words. In the machine learning supervised method the classifiers are using the unigrams as features. In the lexicon-based method the unigrams which are found in the lexicon are assigned a polarity score, the overall polarity score of the text is then computed as sum of the polarities of the unigrams.

In the recent years more advanced algorithms for sentiment analysis were developed that take in consideration not only the message itself, but the context in which the message is published, who is the author of the message, who are the friends of the author, what is the underlying structure of the network. For example, [17] showed that the quality of sentiment clustering for Twitter can be improved by joint clustering of tweets, users, and features. [18] exploited social connections for sentiment analysis by introducing a Sociological Approach to handling Noisy and short Texts (SANT). While [19] used visual content in addition to the textual information to improve sentiment classification accuracy, [20] achieved significant improvements of sentiment classification by using content-based features along with context-based features. [21] increased the feature space and as the result, the accuracy of classification, by adding semantics as additional features into the training dataset. [22] employed genetic algorithms to find subsets of words from a set of paradigm words that led to improvement of classification accuracy.

In this study we concentrate only on textual information of the message. We present a step-by-step methodology for both lexicon-based and machine learning based classification. We show that accuracy of the sentiment analysis can be improved by combining the two approaches: during the first stage a senti-

ment score is calculated based on the sentiment lexicon, during the second stage this score, called prior sentiment score, is used as a feature in machine learning classification. The results showed that the combined approach outperforms the two approaches working independently. Both lexicon based and machine learning based methodologies require data preparation step or pre-processing that is discussed in the following section.

2.1 Data Pre-processing

We perform the pre-processing steps before the actual methods of sentiment analysis are applied. Typical pre-processing procedure includes the following steps:

Tokenization or Bag-of-Words Creation. The incoming string is broken into tokens: comprising words and other elements, for example URL links. The common separator for identifying individual words is whitespace, however other symbols can also be used. Tokenization of social-media data is more difficult than tokenization of the general text.

We used ArkTweetNLP library which was developed by the team of researchers from Carnegie Mellon University [23] and was specially designed for working with twitter messages. ArkTweetNLP recognizes specific to Twitter symbols, such as hashtags, at-mentions, retweets, emoticons, commonly used abbreviations, and treats them as separate tokens.

N-grams Extraction. According to this approach accompanying words are being grouped into phrases called n-grams. N-grams method can decrease bias, but may increase statistical sparseness.

It has been shown that the use of n-grams improves the quality of text classification ([24]; [25]; [26]), however there is no a unique solution for which size of n-gram to use. Bigrams are collections of two neighboring words in a text, and trigrams are collections of three neighboring words. Caropreso et al. conducted an experiment of text categorization on the Reuters dataset [27]. They have reported that in general the use of bigrams helped to produce better results than the use of unigrams, however while using Rocchio classifier [28] the use of bigrams led to the decrease of classification quality in 28 out of 48 experiments. Tan et al. reported that use of bigrams on Yahoo!Science dataset allowed to improve the performance of text classification using Nave Bayes classifier from 65% to 70% break-even point [29]. On Reuters dataset the increase of accuracy was not significant. Trigrams were reported to generate poor performance [30].

In this study in order to extract unigrams and bigramswe we used a tokenizer from a free machine learning software WEKA. WEKA was developed in the university Wakaito and provides implementations of many machine learning algorithms. Since it is written in Java and has an API, WEKA algorithms can be easily embedded within other applications.

Stemming and lemmatisation. Stemming is a procedure of replacing words with their stems, or roots. The dimensionality of the BOW will be reduced when different words, such as read, reader and reading are mapped into one word read and are counted together. However, one should be careful when applying

stemming, since it can increase bias. For example, the biased effect of stemming is merging of distinct words experiment and experience into one word exper, or words which ought to be merged together (such as "adhere" and "adhesion") may remain distinct after stemming. These are examples of overstemming and understemming errors respectively. Overstemming lowers precision and understemming lowers recall.

We used WEKA package to perform stemming operation. WEKA contains implementation of a SnowballStemmer and LovinsStemmer. The overall impact of stemming depends on the dataset and stemming algorithm. After testing both implementations we discovered that stemming reduced the accuracy of our sentiment analysis, therefore stemming operation was avoided in the final implementation of the sentiment analysis algorithm.

Stop-words removal. Stop words are words which carry a connecting function in the sentence, such as prepositions, articles, etc. There is no definite list of stop words, but some search machines, are using some of the most common, short function words, such as the, is, at, which, and on. They can be removed since they have a high frequency of occurrence in the text, but do not affect the final sentiment of the sentence.

For the purpose of stop-words removal WEKA machine learning package is used. WEKA allows modifying the list of words, which should be considered as stop-words and removed.

Lowering the Case of the Letters. All tokens are lowered in case.

Part-of-Speech Tagging (POS). The process of part-of-speech tagging allows to automatically tag each word of text in terms of which part of speech it belongs to: noun, pronoun, adverb, adjective, verb, interjection, intensifier etc. The goal is to be able to extract patterns from analyzing frequency distributions of these part-of-speech tags and use it in the classification process as a feature. Pak and Paroubek analysed the distribution of POS tagging specifically for Twitter messages and identified the following patterns [30]:

- Subjective texts (carrying the sentiment) often contain more pronouns, rather than common and proper nouns;
- Creators of subjective texts talk from the first person point of view (describing themselves) or from the second person (addressing the audience), while the authors of objectives texts normally speak in the third person;
- Subjective messages often use past simple tense;
- Subjective texts contain many verbs in a base form and many modal verbs;
- Authors of subjective texts use superlative adjectives to express their emotions, while comparative adjectives are mostly used for expressing facts in the objective messages;
- Adverbs are mostly used in subjective texts;
- Very common for expressing a positive opinion in the text is the usage of superlative adverbs, for example the most, the biggest, the best;
- In the negative sentences there is a high rate of using verbs in the past tense, probably because the authors tend to express their negative emotions about losses or unsatisfactory experiences in the past. For example: lost, tired,

bored.

Let us note that there is no common opinion about whether POS tagging improves the results of classification or not. [31] got positive results using POS tagging, while [32] reported the decrease in performance.

In the current study we tested performance of multiple existing pos-taggers: Stanford Tagger[3], Illinois Tagger[4], OpenNLP[5], LingPipe POS Tagger[6], Unsupos[7], ArkTweetNLP[8], Berkeley NLP Group Tagger[9]. We chosed ArkTweetNLP [23] as a pos-tagger for our dataset since it was trained on Twitter data and learned to understand Twitters specific slang. Specifically for POS-tagging Twitter data unique POS tags were developed, some of them are presented in the Table 1.

Table 1: Example POS tags.

@ at-mentions	Is used to identify the user- recipient of the tweet
U	URL or email address
#	Hashtag to identify the topic of the discussion or a category
~	Discourse marker. Indicates, that message is the continuation of the previous tweet
E	Emoticons , , etc.
G	Abbreviations, shortenings of words

Overall 25 tags were developed. An example[10] of how ArkTweetNLP tagger works on practice is presented in a Table 2.

Negations Handling. Negation refers to the process of conversion of the sentiment of the text from positive to negative or from negative to positive by using special words: *no, not, dont* etc. These words are called negations.

The example negation words are presented in the Table 3.

Handling negation in the sentiment analysis task is a very important step as the whole sentiment of the text may be changed by the use of negation. It is important to identify the scope of negation. As the implementation of the negation handling in this study it was followed simple, but effective strategy: if negation word is found, the sentiment score of every word appearing between a negation and a clause-level punctuation mark (.,!?:;) is reversed ([7]).

[3] http://nlp.stanford.edu/software/index.shtml
[4] http://cogcomp.cs.illinois.edu/page/software_view/3
[5] http://opennlp.sourceforge.net/models-1.5
[6] http://alias-i.com/lingpipe/demos/tutorial/posTags/read-me.html
[7] http://wortschatz.uni-leipzig.de/~cbiemann/software/unsupos.html
[8] http://www.ark.cs.cmu.edu/TweetNLP
[9] http://nlp.cs.berkeley.edu/Software.shtml
[10] http://www.ark.cs.cmu.edu/TweetNLP/

Table 2: Example of ArkTweetNLP [23] tagger on practice.

ikr smh he asked fir yo last name so he can add u on fb lololol
word tag confidence
ikr ! 0.8143
smh G 0.9406
he O 0.9963
asked V 0.9979
fir P 0.5545
yo D 0.6272
last A 0.9871
name N 0.9998
so P 0.9838
he O 0.9981
can V 0.9997
add V 0.9997
u O 0.9978
on P 0.9426
fb ^ 0.9453
lololol ! 0.9664

"ikr" means "I know, right?", tagged as an interjection.
"so" is being used as a subordinating conjunction, which our coarse tagset denotes P.
"fb" means "Facebook", a very common proper noun (^).
"yo" is being used as equivalent to "your"; our coarse tagset has posessive pronouns as D. "fir" is a misspelling or spelling variant of the preposition for.
Perhaps the only debatable errors in this example are for ikr and smh ("shake my head"): should they be G for miscellaneous acronym, or ! for interjection?

Sometimes a negation term in a sentence does not have any scope. Some of these situations we implemented as exceptions:

Exception Situation 1: Whenever a negation term is a part of some special phrase without any negation sense, there is no scope for this negation term. Examples of these special phrases include not only, not just, no question, not to mention and no wonder.

Exception Situation 2: A negation term does not have a scope when it occurs in a negative rhetorical question. A negative rhetorical question is identified by the following heuristic. (1) It is a question; and (2) it has a negation term within the first three words of the question. For example:
Did not I enjoy it?
Wouldn't you like going to the cinema?

2.2 Lexicon-Based Approach

Lexicon-based approach is also called a dictionary approach and relies on a lexicon or dictionary of words with pre-calculated polarity. Sometimes this method is

Table 3: Example Negation Words

hardly	cannot	shouldnt	doesnt
lack	darent	wasnt	didnt
lacking	dont	wouldnt	hadnt
lacks	doesnt	werent	hasnt
neither	didnt	wont	havnt
nor	hadnt	without	havent

considered to be part of the Machine Learning Unsupervised approach, however we will describe it as an independent method, since the quality of classification in the lexicon-based approach depends solely on the quality of the lexicon.

The idea behind the lexicon-based approach is quite simple:

- Construction of the lexicon of words with assigned to them polarity scores (different ways of creating lexicons are described in detail in the following section).
- A bag-of-words is created from the text that needs to be analysed for sentiment.
- Preprocessing step: lowering of words in case, stop-words removal, stemming, part-of-speech tagging, negations handling.
- Sentiment score calculation: each word from the bag-of-words gets compared against the lexicon. If the word is found in the lexicon, the sentiment score of that word is added to the total sentiment score of the text. For example, the total score of the text:
  "*A masterful[+0.92] film from a master[+1] filmmaker , unique[+1] in its deceptive grimness , compelling[+1] in its fatalist[-0.84] worldview .*"
  will be calculated as follows:
  'total sentiment score' = +0.92 +1 +1 +1 -0.84 = 3.08, which means, that the text expresses a positive opinion.

Lexicon Creation Since creation of a lexicon is the central part of the lexicon-based approach, many researchers created their own sentiment lexicons. Among them:

1. Opinion Lexicon [33]
2. SentiWordNet [34]
3. AFINN Lexicon [35]
4. LoughranMcDonald Lexicon
5. NRC-Hashtag [36]
6. Harvard Inquirer Lexicon

Different ways of the lexicon creation process are described below:

Hand-tagged lexicons. The straightforward approach, but also the most time consuming, is to manually construct a lexicon and tag words in it as positive or negative. [5] constructed their lexicon by reading several thousands of

messages and manually selecting words, which were carrying sentiment. They then used a discriminant function to identify words from a training dataset, which can be used for sentiment classifier purposes. The remained words were expanded to include all potential forms of each word into the final lexicon.

Another example of hand-tagged lexicon is The Multi-Perspective-Question-Answering (MPQA) Opinion Corpus constructed by [37]. MPQA is publicly available and consists of 4,850 words, which were manually labeled as positive or negative and whether they have strong or weak subjectivity.

In order to create their lexicon, [38] extended a list of subjectivity clues created by [39]. 8000 words, comprising the lexicon, were then manually tagged with their prior polarity.

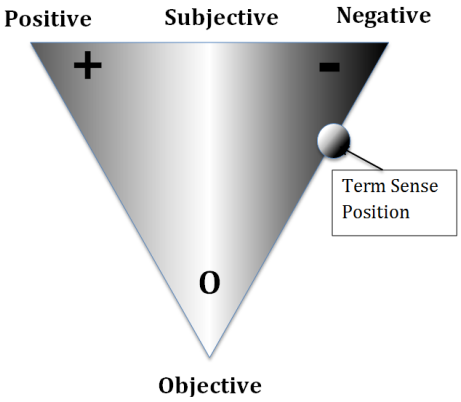

Fig. 1: The graphical representation of the opinion-related properties of a word in SentiWordNet.

Another resource is The SentiWordNet (Figure 1) created by Esuli and Sebastiani [34]. SentiWordNet extracted words from WordNet[11] and gave them probability of belonging to positive, negative or neutral classes, and subjectivity score. [40] demonstrated that SentiWordNet can be used as an important resource for sentiment calculation.

Constructing a lexicon from the trained data. This approach belongs to the category of the supervised methods, because a training dataset of labelled sentences is needed. According to the method the sentences from the training dataset get tokenised and a bag-of-words is created. The words are then filtered to exclude some parts-of-speech that do not carry sentiment, such as prepositions, for example. The prior polarity of words is calculated according to the occurrence of each word in positive and negative sentences. For example, if a word "success" is appearing more often in the sentences labelled as positive in the training data set, the prior polarity of this word will be assigned a positive

[11] http://wordnet.princeton.edu/

value. This way of creating the lexicon has shown good performance results, since the words and their polarity is more suited to a particular type of text.

Extending a small lexicon using bootstrapping techniques. Hazivassiloglou and McKeown proposed to extend a small lexicon comprised of adjectives by adding new adjectives which were conjoined with the words from the original lexicon [41]. The technique is based on the syntactic relationship between two adjectives conjoined with the "AND" it is established that "AND" usually joins words with the same semantic orientation.Example:

"The weather yesterday was nice and inspiring"

Since words "nice" and "inspiring" are conjoined with "AND", it is considered that both of them carry a positive sentiment. If only the word "nice" was present in the lexicon, a new word "inspiring" would be added to the lexicon. Other related work includes [42] and [43].

Similarly, [41] and [44] suggested to expand a small manually constructed lexicon with synonyms and antonyms obtained from NLP resources such as WordNet. The process can be repeated iteratively until it is not possible to find new synonyms and antonyms. It is important to conduct a manual inspection of newly added words to avoid errors.

[45] also created their lexicon by semi-automatically expanding manually-compiled lexicon with the words from WordNet 2.1.

Automatic Lexicon Generation For the purpose of this study we generated a lexicon using an approach "Constracting a lexicon from the trained data" described in the previous section. We used a Twitter dataset comprised by Mark Hall [46] to automatically generate our own sentiment lexicon. The messages are labelled as positive or negative. The method was implemented as follows:

1. The messages from the training datasets of Mark Hall were read from the database.
2. Words were extracted by tokenising all the sentences in the dataset; words in a BOW were assigned POS tags; stemming was performed (originally we performed stemming, but after analyzing the results and discovering that stemming reduced the accuracy of the analyser, we omitted this step); tokens were filtered based on their length (1-99); all words were lowered in case.
3. All the words obtained after tokenizing the instances of the training dataset and pre-processing were combined into one set called a bag-of-words.
4. The number of occurrences of each word in positive and negative sentences from the training dataset was calculated.
5. The positive polarity of each word was calculated according to the following formula: the number of occurrences in positive sentences divided by the number of all occurrences. For example, we calculated that the word "*pleasant*" appeared 122 times in the positive sentences and 44 times in the negative sentences. According to the formula, the "goodness" score of the word "*pleasant*" is:

$$sentScore = \frac{\#Positive\ sentences}{(\#Positive\ sentences + \#Negative\ sentences)} \quad [1]$$

$$sentScore = \frac{122}{(122+44)} = 0.73$$

Similarly, the negative score for the word "*pleasant*" can be calculated by dividing the number of occurrences in negative sentences by the total number of mentions:

$$sentScore = \frac{\#Negative\ sentences}{(\#Positive\ sentences + \#Negative\ sentences)} \quad [2]$$

$$sentScore = \frac{44}{(122+44)} = 0.27$$

Based on the negative and positive score of the word we can make a decision about its polarity. Thus, the word "pleasant" carries a positive sentiment. Some words, however, have negative and positive polarity scores very close to each other. These words are called "neutral" and do not have impact on the overall sentiment score of the message, since they tend to appear with equal frequency in both positive and negative texts. We checked the sentiment score of some words and presented the results in the table 4.

Table 4: Sentiment scores of words in the automatically generated lexicon

	GOOD	BAD	LIKE
Positive Score	0.675	0.213	0.457
Negative Score	0.325	0.787	0.543

As we can see from the table, the words "*GOOD*" and "BAD*"* have strongly defined polarity scores as we would expect. The word*" "LIKE"* have polarity scores in the range between 0.4 and 0.6 and is considered to be a neutral word. The word "*LIKE*" at a first sight creates an impression of a positive word and needs a deeper analysis to understand its "neutral" nature. The word "*LIKE*" generally plays one of two roles:

(a) Being a verb to express preference. For example: *"I like ice-cream"*.
(b) Being a preposition for the purpose of comparison. For example: "*This town looks like Brighton".*

The example sentence in case a) has positive sentiment, however can easily be transformed into a negative sentence: "*I don't like ice-cream*". This demonstrates that the word "*LIKE*" is being used with equal frequency for expressing positive and negative opinions. In the case b) the word "*LIKE*" is playing a role of preposition and does not play role in determining the overall polarity of the sentence.
All words from the Bag-of-Words with a polarity in the range [0.4; 0.6] were removed, since they do not help to classify the text as positive or negative.

As we can see from the table 4, both positive and negative scores have positive values, and in order to determine the polarity of the word both scores need to be compared. We mapped the "goodness" scores of the words into the range [-1;1], where negative scores were assigned for the words with negative polarity and positive scores indicated the words with positive polarity by. The following formula was used for mapping:

$$PolarityScore = 2 * GOODNESS - 1 \quad [3]$$

According to this formula, the word "LIKE", for example, will have a score 0.446 * 2 -1 = -0.1, which is very close to 0 and indicates the neutrality of the word. In case when the word was absolutely positive and had a "goodness" score of 1, the final score would also be positive: 1 * 2 – 1 = 1. If the word was absolutely negative and had the "goodness" score 0, the final score would be negative: 0 * 2 -1 = -1.

Lexicons Combinations Since emoticons role for expressing opinion online is continuously increasing, it is crucial to incorporate emoticons into lexicon datasets for automatic polarity analysis. However, most of the existing public lexicons do not contain emoticons, on contrary, lexicons are being removed as typographical symbols during the first stages of pre-processing.

[47] showed that incorporation of the emoticons into lexicon can significantly improve the accuracy of classification. In this study we manually constructed a lexicon of emoticons and abbreviations commonly used in social-media to express emotions (EMO) and analysed how performance of the classic opinion lexicon (OL) [33] can be improved by inclusion of EMO lexicon. We also expanded the lexicon further by incorporating words from the automatically created lexicon (AUTO) based on the training data [46]. The process of automatic lexicon creation was described in detail in the previous section.

With opinion lexicon (OL) serving as a baseline we compared performance of the following lexicon combinations:

Table 5: Combinations of lexicons tested

	Comprising lexicons
1.	OL
2.	OL + EMO
3.	OL + EMO + AUTO

The words from all lexicons were standarised to have the value in the range [-1;1].

Sentiment Scores Calculation The sentiment score is a numerical precise representation of the sentiment polarity. There are different ways to calculate

the final sentiment score of the sentence. The most common practice is to calculate the average of the sentiment scores of the individual words comprising the message.In this study we propose alternative measure based on the logarithm of the average score and compare it's performance with the standard score calculation technique.

1. *Simple Average* - sum of the scores of individual words caring the sentiment, divided by the number of such words:

$$Score_{AVG} = \frac{1}{m}\sum_{i=1}^{m} W_i, \qquad [4]$$

where W_i is the score of the sentiment word i

2. *Log10* - uses a logarithmic scale for calculating the sentiment.
This approach helps to create a better visual separation between the positive and negative classes. We normalise the score in such away that the values range between [-1; 1] with -1 being the most negative score and 1 being the most positive score.

$$Score_{Log10} = sign(Score_{AVG})Log_{10}(\mid Score_{AVG} \mid) \qquad [5]$$

Lexicon Approach Results The validation of the lexicon approach was performed on the test dataset available from **SemEval-2013 Competition** [48] (Figure 2). The dataset analysis showed that it contained many sarcastic messages which are known to be difficult to classify.

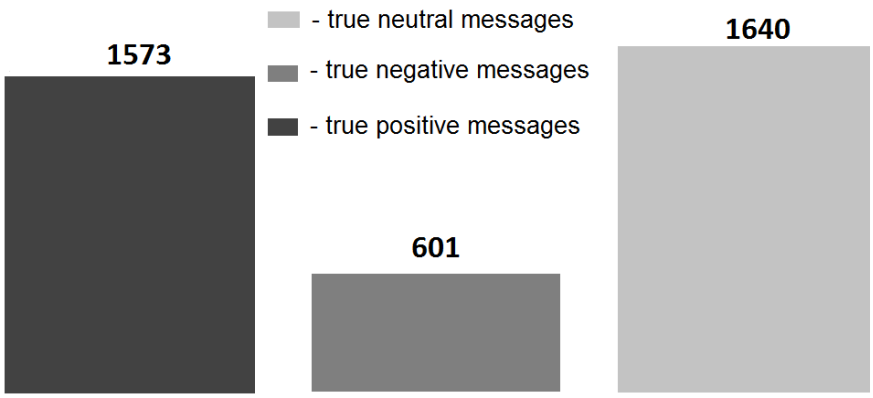

Fig. 2: Statistics of the test data set from SemEval-2013 competition [48]. The bar on the left represents the number of positive sentences in the dataset, the bar in the middle shows the number of negative sentences and the bar on the right reflects the number of neutral sentences.

For both types of sentiment scores described in the section above and every combined lexicon a histogram of scores distributions was created. Figure 2 presents a distribution diagram of the *Simple Average* score for the single OL lexicon. The colors represent true labels. The X-axis corresponds to the calculated score which ranges from -1 to 1, with -1 representing the most negative score, 0 indicating that the message is neutral and 1 standing for a positive message.

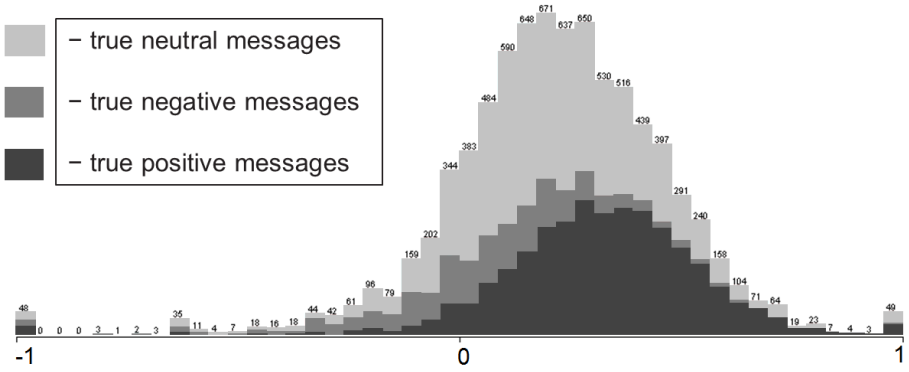

Fig. 3: The distribution of Simple Average Score values for OL lexicon. The colors represent true labels. In the case of ideal classification we would have a grey column (true negative message) on the left (-1 sentiment score), light grey (true neutral message) column in the middle (0 or close to 0 sentiment score) and dark grey (true positive score) on the right (sentiment score of value 1).

In the case of ideal classification we would have a grey column (true negative message) on the left (-1 sentiment score), light grey (true neutral message) column in the middle (0 or close to 0 sentiment score) and dark grey (true positive score) on the right (sentiment score of value 1).

Figure 3 represents the histogram of the sentiment Average Score for the baseline OL lexicon. We can observe that the positive, negative and neutral messages collapse together. Even though there is no clear separation between the classes, we can observe that true negative messages are located on the left side of the histogram, true positive messages are located on the right of the histogram and true neutral messages are widely spread along the axis. In order to assign class labels to the messages in the Lexicon approach we will have to set up the boundaries for each class: for example we can decide that all messages with the score below 0.1 will be considered negative, all messages with the scores between 0.1 and 0.3 will b considered neutral, and all messages above 0.3 will be considered positive. The goal is to set up the boundaries/tresholds which will allow to achieve the highest accuracy of classification.

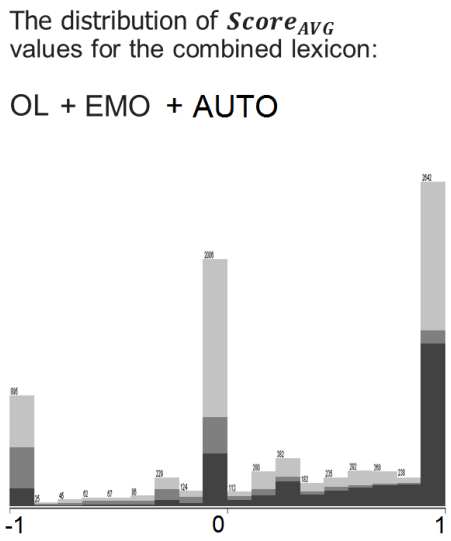

Fig. 4: Histogram of the sentiment Average Score for the combined lexicon OL + EMO + AUTO.

Inhancing the OL lexicon with emoticons and AUTO lexicon allowed to achieve much better separation of the classes (see 4).

We also compare how different ways of calculating the sentiment scores perform. Figure 5 shows the histograms for the Simple Average and Logarithmic scores for the lexicon OL + EMO + AUTO. We can see that the Logarithmic scale allowed to achieve better visual separation of the classes near the 0.

The analysis above showed that combined lexicon Ol + EMO + AUTO performed better than the baseline lexicon OL (see figure 3 vs figure 4). We also showed that use of logarithmic scale for sentiment score calculation allows to achieve better separation of the classes. In the next example we compare performance of OL + EMO + AUTO combined lexicon versus OL + EMO lexicon based on the logarithmic score (see figure 6)

We can observe that OL + EMO lexicon combination has a better separation of the classes. This is mainly achieved by the higher accuracy of predicting the neutral class. Neutral messages were found to be the most difficult to classify, since often they contain one or more words, that are registered in the lexicon as positive or negative. This causes the final score of the message to have a value different from 0 and to be classified as positive or negative. We observed that addition of AUTO lexicon introduced more ambiguity regarding the sentiment of individual words. In its turn the small OL consisted mainly of adjectives that carry strong positive or negative sentiment that are unlikely to cause ambiguity. OL dataset, enhanced with emoticons, even though smaller in its size, outperformed a bigger lexicon.

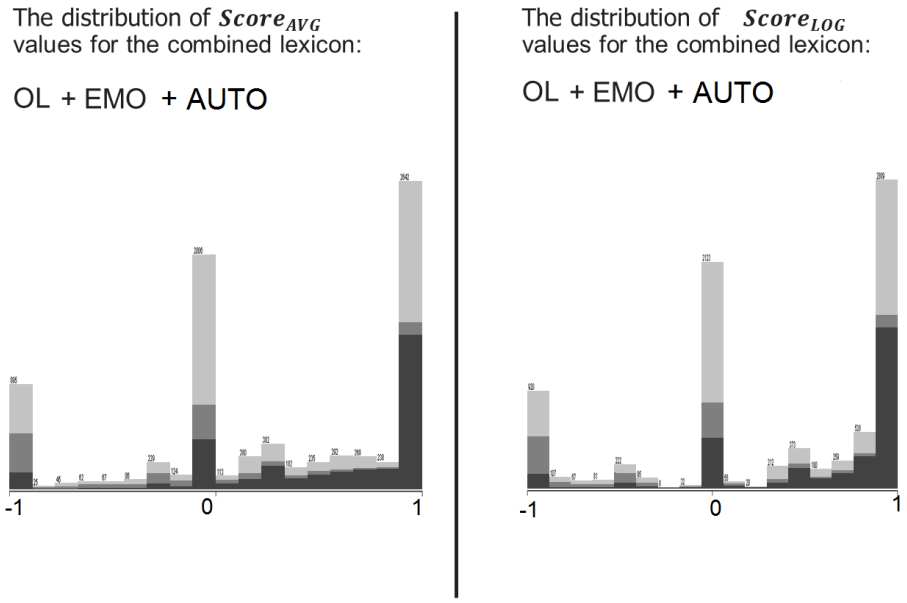

Fig. 5: Histogram of class predictions for the lexicon combination OL + EMO + AUTO for the different types of scores: Simple Average versus Logarithmic Score.

So far the analysis of different lexicons combinations and sentiment scores was based on visual analysis of the histograms. In order to get quantitative results it is necessary to choose threshold values for each lexicon combination and each type of the score. In order to avoid the task of thresholding we performed k-means clustering on the feature space, comprised of different combinations of scores and dictionaries.

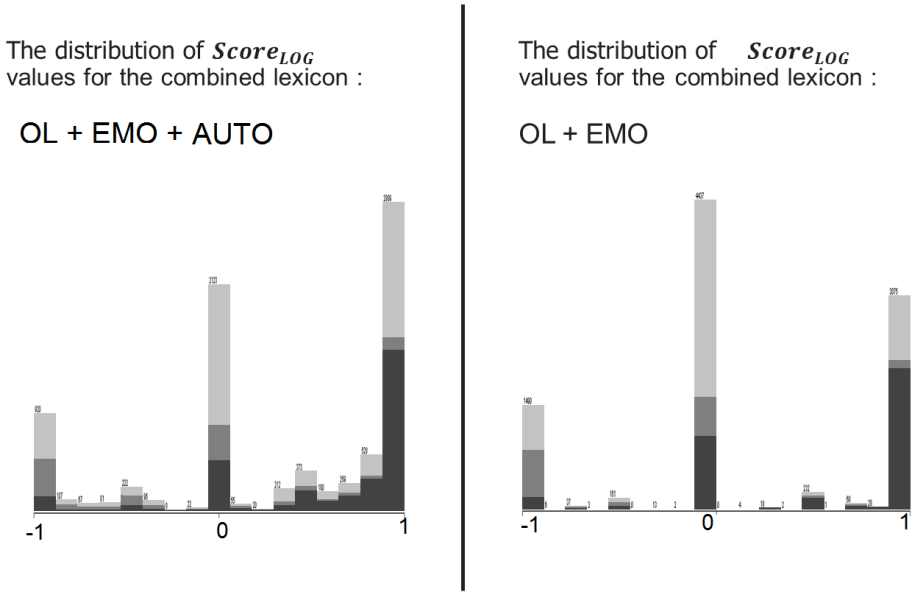

Fig. 6: Histograms of class predictions for the lexicon combinations OL + EMO + AUTO and OL + EMO based on the Logarithmic Score.

Table 6: Fscore results of SemEval Competition-2013 [48].

TEAM NAME	**F-SCORE**
NRC-Canada	0.6902
GUMLTLT	0.6527
teragram	0.6486
AVAYA	
BOUNCE	0.6353
KLUE	0.6306
AMI and ERIC	0.6255
FBM	0.6117
SAIL	
AVAYA	0.6084
SAIL	0.6014
UT-DB	0.5987
FBK-irst	0.5976

Table 7: Results of K-Means clustering for different lexicon combinations.

	OL	**OL + EMO**	**OL + EMO + AUTO**
Accuracy	52%	61%	58%

The results of clustering agreed with the visual analysis of he histogram. The highest accuracy of 61% was achieved when using the Logarithmic Sentiment Score and OL lexicon enhanced with emoticons and abbreviations (OL + EMO). Figure 7 presents the results for different lexicon combinations.

2.3 Machine Learning Based Approach

In this section we will cover supervised machine learning techniques used for text categorization. Supervised machine learning requires a labelled training dataset on which the classifier will be trained. Each example in the training dataset consists of an input object and a label or a class (also called supervised signal). The supervised algorithm analyses labelled data, extracts features that model the differences between different classes, and infers a function, which can be used for classifying new examples.

In the simplified form, the text classification task can be described as follows: if our training dataset of labelled data is $T_{train} = \{(t,\ l_1)\ ,\ \ldots,\ (t_n,\ l_n)\}$, where each text t_i belongs to a dataset T and the label $l_i = l_i(d_i)$ is a pre-set class within the group of classes $L = \{l_i,\ \ldots,\ l_m\}$, the goal is to build a learning algorithm that will receive as an input the training set T_{train} and will generate a model that will accurately classify unlabelled documents.

In terms of the individual classes, some researches [7] classified texts only as positive or negative, assuming that all the texts carry an opinion. Later [38], [30] and [31] showed that short messages like tweets and blogs comments often just state facts. Therefore, incorporation of the neutral class into the classification process was necessary.

As creation of a good lexicon is crucial for the lexicon-based method, the selection of features is essential for the success of the machine learning categorization. The most common features being used for his purpose include:

1. Unigrams and n-grams
2. Number of words with positive/nagative sentiment
3. Number of negations
4. Length of a message
5. Number of exclamation marks
6. Number of comparative and superlative adjectives

Even though the text categorisation problem belongs to the field of supervised learning, it can be distinguished as a separate study due to having some unique properties:

1. *High Dimensionality.* Since the main features of a text classifier are the unigrams and the number of individual words in a text can be very large, the dimensionality of the feature space grows proportionally to the size of the dataset;
2. *Statistical sparseness.* Even though the feature vector can become very long, not many of those words appear in each individual message;

3. *Domain knowledge*. In order to classify the text correctly the knowledge of natural language should be employed. [49] have demonstrated that it is important to pay attention to the inner structure of the document (part-of-speech tagging, presence of negation) in order to correctly perform the analysis of sentiment. Statistical properties of texts, such as adherence to Zipfs law can also be used;

The most popular classification algorithms are Support Vector Machines (SVMs) and the Naive Bayes. Classification accuracy of this algorithms is dependent on the list of features. Barbosa et al. reports better results for SVMs [31] while Pak et al. obtained better results for Naive Bayes [30].

For the purpose of Machine Learning based sentiment analysis in our study we used a free machine learning package WEKA. The following steps were taken for machine learning classification:

1. Pre-processing/cleaning the data.
2. Features generation.
3. Features selection.
4. Training the model and validation.

Pre-processing/cleaning the data Machine Learning Supervised approach requires labelled training and validation data. For this purposes we used publicly available training dataset from SemEval-2013 competition [48]. The dataset is highly unbalanced (Figure 7).

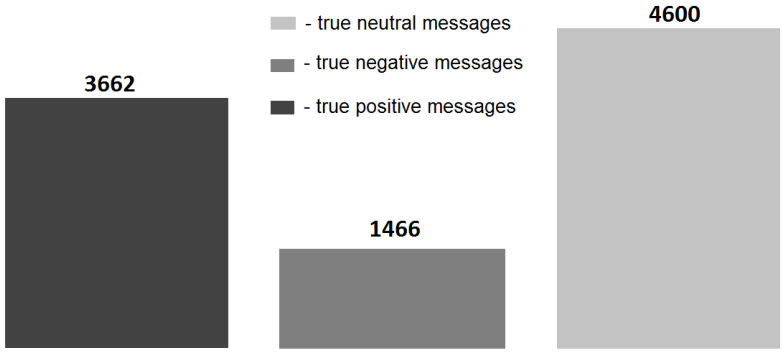

Fig. 7: Statistics of the training data set from SemEval-2013 competition [48]. The bar on the left represents the number of positive sentences in the dataset, the bar in the middle shows the number of negative sentences and the bar on the right reflects the number of neutral sentences.

Before training the classifiers the data needed to be pre-processed and this step was performed according to the process described in the section Pre-processing. Additional steps that were performed:

– *Filtering.* Some syntactic constructions used in Twitter messages are not useful for sentiment detection. These constructions include URLs, @-mentions, hashtags, RT-symbols and they were removed during the pre-processing step.
– *Tokens replacements.* The words that appeared to be under the affect of the negation words were modified by adding a suffix **_NEG** to the end of those words.
For example, the phrase *I don't want.* was modified to *I don't want_NEG.*
This modification is important, since each word in a sentence serves a purpose of a feature during the classification step. Words with **_NEG** suffixes increase the dimensionality of the feature space, but allow the classifier to distinguish between words used in the positive and in the negative context.
When performing tokenization, the symbols ():;, among others are considered to be delimiters, thus most of the emoticons could be lost after tokenization. To avoid this problem positive emoticons were replaces with **pos_emo**. and negative were replaced with **neg_emo**.
Since there are many variations of emoticons representing the same emotions depending on the language and community, the replacement of all positive lexicons by pos_emo and all negative emoticons by neg_emo also achieved the goal of significantly reducing the number of features.

Features Generation Features are the attributes of data which are the most useful for capturing patterns in the data. New features can be created which are expected to improve the overall quality of text classification. In this section we describe the features that were generated and used to train a classifier:

– **N-grams.** Very important features that are used in machine learning classification are the n-grams presence or n-grams frequency. In the presence-based representation for each instance a vector is created comprised of 0 and 1, where 0 means that a corresponding n-gram is not present in the instance and 1 means that the n-gram is present. In the frequency-based representation in case when the n-gram is present in the instance, the number of occurrences is used instead of a simple indication of presence.
For the work at hand, we have transformed the training dataset into the bag-of-ngrams. Bag-of-ngrams can be a bag-of-unigrams, bag-of-bigrams, bag-of-trigrams or a combination of all. In this study we have used a bag of unigrams and bigrams and "n-grams presence" approach. Using "n-grams frequency" would not be logical in this particular experiment, since Twitter messages are very short, and a term is unlikely to appear in the same message more than once.
– **Prior Sentiment.** One of the most important features of the presented algorithm is the sentiment score obtained through the lexicon-based classification. We called this score a "Prior Sentiment". Since the accuracy of the score is basically the accuracy of our lexicon approach, it can hypothesised that by using this score in the machine learning approach it's accuracy will be leveraged in the machine learning classification.

By using a lexicon score as one of the features in the machine learning model our study combined two approaches in one. The background literature analysis did not reveal any mentions of such combined approach in the previous practice of sentiment analysis. NRC research group reported that they used sentiment lexicons as features, but provided no further information or analysis of the contribution of those feature to the final classification accuracy. In this study we attempt to analyse the predictive power of the lexicon score feature for the machine learning classification.

- **Elongated words number**: the number of words with one character repeated more than 2 times, e.g. 'soooo';
- **Emoticons**: presence/absence of positive and negative emoticons at any position in the tweet;
- **Last token**: whether the last token is a positive or negative emoticon;
- **Negation**: the number of negated contexts. A negated context also affects the n-gram and lexicon features: each word and associated with it polarity in a negated context become negated (e.g., 'not perfect' becomes 'not perfect_NEG', 'POLARITY_positive' becomes 'POLARITY_positive_NEG');
- **POS**: the number of occurrences for each part-of-speech tag: verbs, nouns, adverbs, at-mentions, abbreviations, URLs, adjectives and others
- **Punctuation marks**: the number of occurrences of punctuation marks in a tweet;
- **Emoticons number**: the number of occurrences of positive and negative emoticons;
- **Negative tokens number**: total count of tokens in the tweet with score less than 0;
- **Positive tokens number**: total count of tokens in the tweet with score greater than 0;

Features Selection The number of possible n-grams exponentially increases as the n-gram size and the size of the training set increase. This exponential growth makes it infeasible to calculate all the features of a sample in a limited amount of time. Many features are redundant or irrelevant and do not significantly impact the classification of results.

Features selection is the process of selecting a subset of relevant features which have the highest predictive power. This step is crucial for the classification process, since elimination of irrelevant and redundant features allows to reduce the size of feature space tremendously for a speed increase, as well as contributes to improving the quality of classification.

From the original feature space of **1826** features a subset of **528** most relevant features were selected using "Information Gain" evaluation algorithm and a "Ranker" search method from WEKA package. Information Gain measures the decrease in entropy when the feature is present vs absent, while Ranker ranks the features based on the amount of reduction in the objective function [50]. Some of the top selected features are displayed in Figure8, revealing that the lexicon feature, described in the previous section as a "Prior score", indeed is

the top feature and have the highest predictive power. There are multiple lexicon features in the figure - one for each lexicon combination and one for each type of the score.

Other top features included: minimal and maximum scores, number of negated tokens, number of different parts of speech in the message. Appearance of these features at the top of the list demonstrates the relevance of the generated features described in the previous section.

Table 8: Top selected features.

TOP FEATURES
tweetPpriorSent
maxScore
posTokens
minScore
negTokens
good
posE
posR
posU
love
great
posV
happy
waitNeg
excited
can t
i
not
posA
posElongWords
best
fun
lastTokenScore
i love
don
don t
amazing
fuck
love you
can
awesome
bad
hope
thanks
luck
see you
i don
looking forward
sorry
didn t
hate

Most of the features after the position 28 are the n-grams. We can see that the top n-grams carry strong sentiment, for example : ”good” - position 21, ”great”

- position 27, "happy" - position 29, "cant" - position 33, "not" - position 35, "I love" - position 42, etc.

Model Training and Validation Once the features were generated and the most relevant were selected, each of the tweets from the training dataset was expressed in terms of the attributes. During the training process the presence of each attribute was checked for each of the classes (positive, negative and neutral).

We have trained multiple models using different WEKA classifiers: Nave Bayes, Support Vector Machines, J48 (base on decision trees). We have conducted experiments for the different sizes of the training datasets (1,000; 2,000 10,000) and different sizes of the feature space.

Before the final training of the model, the dataset was split into two complimentary subsets: one subset, called the training set, contained 2/3 of the data; the second subset, called the validation or testing set, was comprised of 1/3 of the data. The training subsets was used to train the model and the validation set was used for validating the performance of the classifiers. 2500 rounds of cross-validation were performed using different partitions, and the validation results were averaged over the rounds.

The results of classification of the independent test set are presented in the next section.

Machine Learning Approach Results The results of classification were tested on the same test set from SemEval - 2013 Competition [48] and compared agains the results of 44 teams that took part in the SemEval-2013 competition. While the classification was performed for 3 classes (pos, neg, neutral), the evaluation metric was Fscore between positive and negative classes.

We have trained the model using different WEKA classifiers and discovered that the Nave Bayes, Support Vector Machines showed the best results. The decision tree algorithm has the lowest accuracy. The reason to this may be a big size of the tree needed to incorporate all of the features. Because of the big tree, the algorithm needs to traverse multiple nodes until it reaches the leaf and predicts the class of the instance. This long path increases the probability of mistakes and thus decreases the accuracy of the classifier.

The best model was based on SVM and produced Fscore of 66% with Overall Accuracy 70%. The table 9 presents the official results of the competition.

Table 9: Fscore results of SemEval Competition-2013 [48].

TEAM NAME	**F-SCORE**
NRC-Canada	0.6902
GUMLTLT	0.6527
teragram	0.6486
AVAYA	
BOUNCE	0.6353
KLUE	0.6306
AMI and ERIC	0.6255
FBM	0.6117
SAIL	
AVAYA	0.6084
SAIL	0.6014
UT-DB	0.5987
FBK-irst	0.5976

We can see that our algorithm would score the second in the competition. Since the training and test datasets were highly unbalanced with the majority of the neutral class, the biggest error appeared in misclassifying positive and negative messages as neutral.

To address the problem of unbalanced dataset we used Cost-Sensitive Classifier provided by WEKA. Cost-Sensitive classifier allows to minimize the total cost of classification [51] This allowed us to achieve the value of **Fscore 73%**. In the SemEval Competition our CostSensitive classifier would have show the best results, scoring 4 points higher than the winner of competition.

It is important to highlight that using our classifier only for 2-class classification on the SemiEval test set results achieved 86% accuracy.

3 Conclusion

The interest in sentiment analysis as a field of research is growing rapidly. It has been shown that transformation of the huge volume of textual data from the web into meaningful information can be very useful. However, the task of accurate opinion extraction still remains challenging.

The performance of different combinations of lexicons was compared. We have experimentally proven the value of emoticons and abbreviations in sentiment analysis. A new approach of sentiment classification was proposed based on combining the two standard techniques (lexicon-based and machine learning based). In our combined approach a lexicon score is being used as a feature in machine learning classification. This innovation allowed to improve the accuracy of predictive model by 5%. Another contribution of our model is using a cost sensitive classifier for unbalanced training sets, which allowed to further increase accuracy by 7%. In the International Sentiment Classification compe-

tition SemEval-2013 our classifier would have shown the best results, scoring 4 points higher than the winner of the competition.

Acknowledgments

This work was supported by the company Certona Corporation. T.T.P.S. acknowledges financial support from CNPq - The Brazilian National Council for Scientific and Technological Development.

References

1. L. Bing, *Sentiment Analysis: A Fascinating Problem*, pp. 7–143. Morgan and Claypool Publishers, 2012.
2. S. Asur and B. A. Huberman, "Predicting the future with social media," in *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '10, (Washington, DC, USA), pp. 492–499, IEEE Computer Society, 2010.
3. T. T. P. Souza, O. Kolchyna, P. C. Treleaven, and T. Aste, "Twitter sentiment analysis applied to finance: A case study in the retail industry," 2015.
4. G. Salton and M. J. McGill. McGraw Hill Book Co., 1983.
5. S. Das and M. Chen, "Yahoo! for amazon: Extracting market sentiment from stock message boards," in *In Asia Pacific Finance Association Annual Conf. (APFA)*, 2001.
6. S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima, "Mining product reputations on the web," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, (New York, NY, USA), pp. 341–349, ACM, 2002.
7. B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, (Stroudsburg, PA, USA), pp. 79–86, Association for Computational Linguistics, 2002.
8. R. Tong, "An operational system for detecting and tracking opinions in on-line discussions," in *Working Notes of the SIGIR Workshop on Operational Text Classification*, (New Orleans, Louisianna), pp. 1–6, 2001.
9. P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, (Stroudsburg, PA, USA), pp. 417–424, Association for Computational Linguistics, 2002.
10. J. Wiebe, "Learning subjective adjectives from corpora," in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp. 735–740, AAAI Press, 2000.
11. P. S. Jacobs, "Joining statistics with nlp for text categorization.," in *ANLP*, pp. 178–185, 1992.
12. V. N. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, September 1998.

13. R. Basili, A. Moschitti, and M. T. Pazienza, "Language-Sensitive Text Classification," in *Proceeding of RIAO-00, 6th International Conference "Recherche d'Information Assistee par Ordinateur"*, (Paris, France), pp. 331–343, 2000.
14. R. E. Schapire and Y. Singer, "BoosTexter: A Boosting-based System for Text Categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.
15. S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in *CIKM '98: Proceedings of the seventh international conference on Information and knowledge management*, (New York, NY, USA), pp. 148–155, ACM, 1998.
16. S. M. Weiss, C. Apte, F. J. Damerau, D. E. Johnson, F. J. Oles, T. Goetz, and T. Hampp, "Maximizing text-mining performance," *IEEE Intelligent Systems*, vol. 14, pp. 63–69, July 1999.
17. L. Zhu, A. Galstyan, J. Cheng, and K. Lerman, "Tripartite graph clustering for dynamic sentiment analysis on social media," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, (New York, NY, USA), pp. 1531–1542, ACM, 2014.
18. X. Hu, L. Tang, J. Tang, and H. Liu, "Exploiting social relations for sentiment analysis in microblogging," in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, (New York, NY, USA), pp. 537–546, ACM, 2013.
19. Q. You and J. Luo, "Towards social imagematics: Sentiment analysis in social multimedia," in *Proceedings of the Thirteenth International Workshop on Multimedia Data Mining*, MDMKDD '13, (New York, NY, USA), pp. 3:1–3:8, ACM, 2013.
20. F. Aisopos, G. Papadakis, K. Tserpes, and T. Varvarigou, "Content vs. context for sentiment analysis: A comparative analysis over microblogs," in *Proceedings of the 23rd ACM Conference on Hypertext and Social Media*, HT '12, (New York, NY, USA), pp. 187–196, ACM, 2012.
21. H. Saif, Y. He, and H. Alani, "Semantic sentiment analysis of twitter," in *Proceedings of the 11th International Conference on The Semantic Web - Volume Part I*, ISWC'12, (Berlin, Heidelberg), pp. 508–524, Springer-Verlag, 2012.
22. J. Carvalho, A. Prado, and A. Plastino, "A statistical and evolutionary approach to sentiment analysis," in *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 02*, WI-IAT '14, (Washington, DC, USA), pp. 110–117, IEEE Computer Society, 2014.
23. K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith, "Part-of-speech tagging for twitter: Annotation, features, and experiments," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, (Stroudsburg, PA, USA), pp. 42–47, Association for Computational Linguistics, 2011.
24. B. Raskutti, H. L. Ferrá, and A. Kowalczyk, "Second order features for maximising text classification performance," in *Proceedings of the 12th European Conference on Machine Learning*, EMCL '01, (London, UK, UK), pp. 419–430, Springer-Verlag, 2001.
25. D. Zhang, "Question classification using support vector machines," in *In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 26–32, ACM Press, 2003.
26. J. Diederich, J. Kindermann, E. Leopold, and G. Paass, "Authorship attribution with support vector machines," *Applied Intelligence*, vol. 19, pp. 109–123, May 2003.

27. M. F. Caropreso, S. Matwin, and F. Sebastiani, "Text databases & document management," ch. A Learner-independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization, pp. 78–102, Hershey, PA, USA: IGI Global, 2001.
28. J. J. Rocchio, "Relevance feedback in information retrieval," in *The Smart retrieval system - experiments in automatic document processing* (G. Salton, ed.), pp. 313–323, Englewood Cliffs, NJ: Prentice-Hall, 1971.
29. C.-M. Tan, Y.-F. Wang, and C.-D. Lee, "The use of bigrams to enhance text categorization," in *INF. PROCESS. MANAGE*, pp. 529–546, 2002.
30. A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)* (N. C. C. Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, eds.), (Valletta, Malta), European Language Resources Association (ELRA), may 2010.
31. L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, (Stroudsburg, PA, USA), pp. 36–44, Association for Computational Linguistics, 2010.
32. E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!," in *ICWSM* (L. A. Adamic, R. A. Baeza-Yates, and S. Counts, eds.), The AAAI Press, 2011.
33. M. Hu and B. Liu, "Opinion lexicon," 2004.
34. A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," 2006.
35. F. . Nielsen, "A new anew: evaluation of a word list for sentiment analysis in microblogs," pp. 93–98, 2011.
36. S. M. Mohammad, S. Kiritchenko, and X. Zhu, "Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets," 2013.
37. J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language Resources and Evaluation*, vol. 39, no. 2/3, pp. 164–210, 2005.
38. T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *hltemnlp2005*, (Vancouver, Canada), pp. 347–354, 2005.
39. J. M. Wiebe and E. Riloff, "Creating subjective and objective sentence classifiers from unannotated texts," in *Proceedings of the Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*, no. 3406 in Lecture Notes in Computer Science, pp. 486–497, 2005.
40. B. Ohana and B. Tierney, "Sentiment classification of reviews using sentiwordnet," 2009.
41. V. Hatzivassiloglou and K. McKeown, "Predicting the semantic orientation of adjectives," (Madrid, Spain), pp. 174–181, 1997.
42. N. Kaji and M. Kitsuregawa, "Building lexicon for sentiment analysis from massive collection of html documents.," in *EMNLP-CoNLL*, pp. 1075–1083, ACL, 2007.
43. J. Wiebe and T. Wilson, "Learning to disambiguate potentially subjective expressions," (Taipei, Taiwan), pp. 112–118, 2002.
44. S.-M. Kim and E. Hovy, "Determining the sentiment of opinions," (Geneva, Switzerland), pp. 1267–1373, 2004.
45. K. Moilanen and S. Pulman, "Sentiment composition," in *Proceedings of Recent Advances in Natural Language Processing (RANLP 2007)*, pp. 378–382, September 27-29 2007.

46. M. Hall, "Twitter labelled dataset." http://markahall.blogspot.co.uk/2012/03/sentiment-analysis-with-weka.html, 2012. Accessed: 06-Feb-2013.
47. A. Hogenboom, D. Bal, F. Frasincar, M. Bal, F. de Jong, and U. Kaymak, "Exploiting emoticons in sentiment analysis," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, (New York, NY, USA), pp. 703–710, ACM, 2013.
48. P. Nakov, Z. Kozareva, A. Ritter, S. Rosenthal, V. Stoyanov, and T. Wilson, "Semeval-2013 task 2: Sentiment analysis in twitter," *Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, vol. 2, pp. 312–320, 2013.
49. C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: MIT Press, 1999.
50. L. Ladha and T. Deepa, "Feature selection methods and algorithms, international journal on computer science and engineering," *International Journal on Computer Science and Engineering*, vol. 3, pp. 1787–1800, 2011.
51. C. X. Ling and V. S. Sheng, *Cost-sensitive Learning and the Class Imbalanced Problem*. 2007.