Sentiment Analysis of Movie Reviews Written in Macedonian Language

Vasilija Uzunova and Andrea Kulakov

Computer Science and Engineering Department
University Sts. Cyril and Methodius
Skopje, Macedonia
vasilijauzunova@gmail.com, andrea.kulakov@finki.ukim.mk
http://www.finki.ukim.mk

Abstract. Identifying sentiments is a natural language processing problem that became popular lately with the advent of various forums and social networks on the Internet. In this paper the analysis will focus on opinions that can evoke either positive or negative feelings in people. Most of the existing researches on textual information processing focus on data mining and fact analysis such as information retrieval, web search, text classification, clustering and many other types of natural language processing, unlike opinion analysis, especially for texts written in Macedonian.

Keywords: Sentiment analysis, movie reviews, Macedonian, Naive Bayes.

1 Introduction

Plenty of time before the expansion of the Internet people asked for an advice from friends for various products or services. Today a source of such information is the Internet because it dramatically changed the way people convey their thoughts and opinions. They can now post reviews of products and express their views on almost every topic on the Internet through forums, discussion groups or blogs that are based on user generated content. There is a huge potential for processing such data that can provide additional value for both users and the companies that make public opinion analysis about any product or service [6]. However, finding sources of opinions and monitoring them can still be a difficult task because there are a number of different sources, and each source can also have a huge amount of reviews. That's why it is required to have a system that summarizes all these posts. This can save a lot of resources and time in search of this information online. In this paper, we will propose a solution for this problem for sentiment analysis of film reviews written in Macedonian, using a Naive Bayes classifier.

As a beginning of this analysis, we introduce the notion of sentiment polarity. Suppose we want to classify a given comment text whether it is positive or negative, based on the opinion of the author. Is it going to be an easy task? In

A. Madevska Bogdanova and D. Gjorgjevikj, *ICT Innovations 2014*,
Advances in Intelligent Systems and Computing 311, DOI: 10.1007/978-3-319-09879-1_28

response to the question, we will take an example that consists of one sentence: ”Човекот X Y беше многу нервозен поради лошиот проект” (”Person X Y was very nervous because of the bad project”). The theme of this segment can be identified with the phrase ”X Y”, but the presence of the words ”нервозен” (”nervous”) and лош (”bad”) suggest a negative meaning. One would suppose that this task is really easy, and the polarity of opinions generally can be distinguished by a set of words.

However, the results of an analysis made by Pang and Lee for movie reviews point out that the suggestions coming through a set of keywords can be less trivial apart from the originally thought [9]. In order to get those keywords, opinion is taken by two human subjects to question whether what they think is positive or negative.

The main goal of our experiments is to confirm that the incorporation of some additional word processing into the polarity classification can significantly improve the results. In this paper, we first shortly describe some related work in the field of sentiment analysis and classification. Section 3 explains the most important challenges of making this analysis. Section 6 deals with sentiment analysis and text classification. In this section we talk about the use of a classifier and the data preprocessing as an important step in the sentiment analysis process. Finally we provide the obtained results using different machine learning algorithms and end up by reaching some conclusions, discussions and suggestions for further development.

2 Related Work

Sentiment analysis is currently receiving a lot of attention from the research community. Since 2001 till now there was rapid expansion and several papers along the subject because of the outstanding research and commercial potential [8]. The focus on this area is to solve the problem of computer processing of messages, sentiment and subjectivity in text.

Opinion mining is part of the area near to Web search and information retrieval. The opinion mining tool processes a set of search results for a given term or product, generates a number of product attributes (quality, features) and aggregates the opinions for each attribute e.g. bad, good [8].

Sentiment is a term used for the automatic evaluation of text and track predictions. Number of papers have placed their focus on sentiment analysis. Subjectivity analysis is a term also used for classification, in which documents are classified into two classes by their objectivity or subjectivity [12]. Thus, ”sentiment analysis” and ”opinion mining” can be considered as sub-areas of subjectivity analysis.

The rest of this section surveys previous work in sentiment analysis classification. In [7], the authors have focused on defining the polarity on Twitter posts by extracting a vector of weighted nodes from the graph of WordNet. For a supervised polarity they build a labeled corpus of tweets written in English. Therefore, they used positive and negative emoticons to label the tweets, ”:)”

returns tweets with positive smileys, and ":(" with negative. They used total number of 376,296 tweets, and the reported accuracy level of this approach was 62%. Another interesting approach is presented by Turney, 2002. This paper presents a simple unsupervised algorithm for classifying text using sentiment orientation of the phrases [11]. The algorithm used different review domains and it achieved an average accuracy of 74%. In [9], the authors Pang, Lee and Vaithyanathan have used three machine learning methods (Naive Bayes, maximum entropy classification and support vector machines) for classifying reviews. As a data source they used the Internet Movie Database (IMDb) archive, and the reviews were collected by stars or some numerical value. The achieved results were very good using all methods. The Naive Bayes approach had a high classification rate of 82.9%.

3 Challenges

What other people think is always important information during the decision making process. Thus, this is the most important challenge for sentiment analysis. Another key challenges are:

1. *Named Entity Recognition* - it is an important stage for sentiment analysis, it is locating and classifying atomic elements in text into predefined categories [2]. (What is the person actually talking about in the sentence?)
2. *Sarcasm/Irony* - a statement with a certain structure, which, actually means the opposite of what that particular statement really means. Sarcasm could be wrongly interpreted as a positive sentiment.
3. *Metaphor* - it can be a replacement of the meaning of a word with another meaning.
4. *Language complexity, spelling and slang words*

4 Classification Based on Supervised Learning

Since sentimental analysis is a special case of text classification, we use algorithms that are used for classification. The classification is solved using supervised algorithms for machine learning.

In supervised machine learning there are training data on which the algorithm learns how to act when it gets new data, and how to classify it. All algorithms have two phases. The beginning phase is the learning phase, in which the algorithm learns how to classify the information. The second stage is the prediction. At this stage, the algorithm gets new, unfamiliar text and based on the training data and some other text analysis, predicts which class should be assigned. Sentiment analysis is a problem that can be solved by supervised learning with two classes, positive and negative. There are several algorithms that can be used for this analysis, of which the most common are Naive Bayes, Support Vector Machines (SVM), Entropy Classification etc [6].

5 Naive Bayes Classifier

Training any classifier requires labeled training examples and a model that will fit. In this paper, we describe a sentiment analysis solution using Naive Bayes classifier. The Naive Bayes classifier is a simple probabilistic classifier. A more descriptive term for this model would be ”independent functional model”. Under normal conditions, the Naive Bayes classifier assumes that the presence (or absence) of certain characteristic of a class is unrelated to the presence (or absence) of any other feature, so in this case the probability of a word appearing in the document does not affect the probability of another word. Probability of occurrence of words wi in class cj is equal to the frequency of appearance of word wi in class cj divided by the total numbers of words in class cj.

$$p(wi|c) = \frac{number\, of\, times\, wi\, occures\, in\, c}{total\, number\, of\, words\, in\, class\, c} \tag{1}$$

$$p(ci) = \frac{training\, documents\, in\, class\, ci}{total\, number\, of\, training\, documents} \tag{2}$$

$$p(ci) = \frac{Ni}{N} \tag{3}$$

If after the training the test data shows up a new word, which has not appeared before in the training data, it will result with a problem for calculating the probability. In this case, the cumulative probability is equal to 0. This problem is solved with Laplace smoothing. Laplace alignment introduces the assumption that a new word has appeared in the training set once. In order not to disrupt the possibility with this kind of assumption, the number of occurrences of all words must be increased by 1. The formula to calculate the probability of words in a class is given by [5]:

$$p(wi|cj) = \frac{1 + count(wi, cj)}{|V| + Ni} \tag{4}$$

In this way, all the words will have a certain probability greater than 0. Also, the probability of certain words will be reduced, however, their relationship is still going to remain the same. In order to achieve higher accuracy of the algorithm, it is necessary to introduce some additional processing. The first processing task that improves the results of sentiment analysis and many other language processing tasks is stemming [10]. It removes the suffixes in words in order that the words with same meaning and different inflection were treated as one. In our analysis, the stemming is avoided. The second processing is handling negation. This is an important concern in sentiment-related analysis [8].

6 Experimental Analysis

The application for sentiment analysis is developed using Groovy programming language powered by the Grails framework. The application consists of a service that is doing the sentiment analysis for specified text. The learning algorithm as its warehouse uses a MySQL database in which are written and updated all the statistics for words obtained in the learning phase using Naive Bayes algorithm. The main table in the database is the table *Word* that contains information about the number of occurrences of all the words in particular class. The table contains the word, the class (positive/negative), the number of occurrences of a word in a given class (both positive and negative) and probability of that word to be in particular class.

The keywords, the important words that determine sentiment significantly, differ for positive and negative sentiment. These words are saved in two different tables, table *PositiveWord* for all the words that have a positive meaning, and table *NegativeWord* for all the words that have a negative meaning. Words that are not crucial to determine the sentiment in the test phase are stored in another table as neutral words - the table *NeutralWord*. Positive and negative words are obtained by conducting a survey of few people. Some of the words are shown in the table below.

Table 1. Word list

Positive	добар, одличен, убав, среќен, фантастичен, еуфоричен, живописен
Negative	лош, грозен, одвратен, вулгарен, грд, груб

We are using public movie review data from three Macedonian forums[1]. The data consists of 200 positive and 200 negative reviews, and they are divided in two categories, positive and negative. Neutral reviews are not included. The data is stored in two directories in the file system. The total number of words from these reviews is 13617 and this is the vocabulary |V| in equation (4). In order to enroll all the words with number of occurrences and calculated probability in the corresponding tables, first we are testing the Naive Bayes classifier on this dataset.

6.1 Handling Negation

The representation of these two sentences ”Ми се допадна овој филм” (”I like this movie ”) and ”Не ми се допадна овој филм” (”I don’t like this movie”) are considered to be very similar, but in fact they have opposite meaning. The only different word is the negation term. To solve this problem we created simple algorithm that analyzes the words to the first punctuation mark. If the word ”не” (”no”) appears in the sentence, it means that the sentence will have opposite meaning, so we take the smaller value of possibility for all the words to the first

[1] http://forum.femina.mk/filmovi/
http://forum.kajgana.com/forums/
http://forum.crnobelo.com/forums

punctuation mark. This is not a perfect model, since there are negations that are not related with the text to the first punctuation mark, but it is good enough for this analysis.

Words that have no meaning and often emerge in the text have to be removed. Such words are called stop-terms. In our analysis, if there are such words in the text, they are saved in a different table for neutral words, e.g. conjunctions, exclamations, particles. . Such words in Macedonian language are: ” без, во, врз, за, зад, и, кај, каде, бидејќи, . . . ” (” without, in, on top of, for, behind, and, where, when, because, . . . ”). These words are removed from the classification because we want the processed text to contain only the substantive words that have meaning for the sentiment analysis.

6.2 Spelling

Because of the fact that all the words in the database are in Cyrillic, and we want also to process text written in Latin, which is common on Internet forums, we use transliteration rules for mapping Cyrillic letters to Latin.

The spelling of the Macedonian language is phonetically based, which means that almost every word is written exactly as it is pronounced. Yet, on Internet forums, there are no clear rules for transliterating Cyrillic letters to Latin, but rather everybody uses different rules, even they can be changed in the same text. Like to write ”sh” for ”ш” and then later to write only ”s” for the same Cyrillic letter. This created difficulties for identifying individual words and we have solved it by using two-steps transliteration, one of the most common and simple way of transliteration using the following table:

Table 2. Transliteration rules

Latin	Cyrillic	Transliteration process	
Gj gj	Ѓ ѓ	gjavol	ѓавол
Zh zh	Ж ж	zhivot	живот
Dz dz	Ѕ ѕ	dzid	ѕид
Lj lj	Љ љ	ljubov	љубов
Nj nj	Њ њ	konj	коњ
Kj kj	Ќ ќ	kjumur	ќумур
Ch ch	Ч ч	chovek	човек
Dj dj	Џ џ	djudje	џуџе
Sh sh	Ш ш	shal	шал

Every word is preprocessed and in the process of transliteration each letter is mapped respectively, giving priority to the letters from the table above. Let’s consider the word ”sushtina” which should be mapped to ”суштина”, in the first step of transliteration the letters ”s” and ”h” - ”sh” are mapped into ”ш” and the word becomes ”suшtina”, in the second step, all the other letters are mapped respectively, so the word finally becomes ”суштина”.

7 Results with Different Approaches

7.1 Results with k-Fold Cross Validation

The k-fold cross validation is a process used for model selection and error estimation of classifiers [1]. In this analysis, we are using a 10 fold cross validation, so we are dividing the data into 10 sections, then train and test the classifier 10 times, each time choosing a different section as the test set and the other 9 sections as the training set. Since we have data for two different classes, we split the data from each class into 10 subsets.

With given 200 examples of each class we have successive blocks of 20 files that are used for cross-validation as test data. The results obtained in this analysis are shown in the table below.

Table 3. Results for every fold

Fold	Accuracy
0	0.475
1	0.875
2	0.975
3	1.0
4	1.0
5	1.0
6	1.0
7	1.0
8	1.0
9	0.925

The average Accuracy is:

$$Accuracy = 0.925 \tag{5}$$

7.2 Results with New Test Data from the Web Interface

All the words from the dataset are listed in the table Word with positive and negative occurrences for each class. When we enter new test data from the web interface, we are analyzing each word from a sentence and calculate the possibility whether it is positive or negative. In this case, we are also using a Naive Bayes classifier.

We are testing 30 new positive and 30 new negative movie reviews. Now the training set contains all the words from the collected movie reviews, 200 positive and 200 negative. The performance of this model is evaluated using the performance measures precision and recall [3]. The following table shows the confusion matrix for our analysis:

Table 4. Confusion matrix

		Predicted label	
		Positive	Negative
Known label	Positive	Tp = 25	Fp = 5
	Negative	Fn = 9	Tn = 21

$$Precision = 0,8333 \tag{6}$$

$$Recall = 0,7452 \tag{7}$$

$$Accuracy = 0,7666 \tag{8}$$

This is a small set of test data and in this case we are talking about very good results. If we have a larger test set, the probability of accuracy certainly would be less.

7.3 Testing with Other Classifiers

In the second experimental phase we used WEKA (Waikato Environment for Knowledge Analysis) to obtain the results from other classification techniques. WEKA is an open source data-mining tool [4]. WEKA has implementations of numerous classification and prediction algorithms. We are using the two decision tree algorithms J48 (C4.5) and ADTree with default values. The results are shown in the table below.

Table 5. Comparison results from each algorithm

Test scenario	Algorithm	Precision	Recall	Accuracy
Training set	J48	0,96	0,96	0,96
	ADTree	0,789	0,725	0,725
Cross-validation	J48	0,658	0,658	0,657
	ADTree	0,568	0,568	0,567

From the results in the table 5, we can assume that the Decision Tree classifier with accuracy rate around 60% in the two test scenarios does not perform the classification as well as the Naive Bayes classifier.

8 Discussion

Comparing the obtained results, we can assume that the Naive Bayes classifier works better with the collected movie reviews. We believe that we have good results because of the nature of our language and the size of our reviews. The algorithm is improved by processing new data as neutral words, which are excluded from the classification. In order to understand the misclassification of some reviews, we analyzed their content and found some of the following problems.

1. Neutral reviews are randomly classified according to the dominant sentiment of the contained words.
2. Some of the words that are positive are more frequent in negative reviews because of the negation.
3. Some of the words that are negative are more frequent in positive reviews because of the negation.
4. Irony is classified as negative sentiment.

There are specific cases in the Macedonian language for which this algorithm, most certainly does not make sense. Irony, Sarcasm and Metaphor are typical examples.

Movie reviews are great way of expressing an opinion of a movie. The reviews give enough details about the movie, and from it the reader can make an informed decision. In this paper, we selected the reviews that were expressed with numerical value or strength positive and negative meaning. The authors of the reviews are not public, and each of them is a member of the online forum from which reviews are extracted. The collected movie reviews are with extremely positive or negative sentiment and that's why the accuracy with k-fold validation is so high.

9 Conclusion

The increased interest in sentimental analysis is partly due to the potential use in applications available online. Equally important are the new intellectual challenges to the research community. There is a huge potential for processing data that make analysis of public opinions on a product or service. In this paper, a supervised approach to sentiment analysis of Macedonian movie reviews was described. Sentiment analysis using a Naive Bayes algorithm showed significant results, considering even small training data set. Despite the good results, there is a lot of space for improvements. The accuracy of the analyzer can be improved by providing a larger set of testing data. It is impossible for sentiment analysis to ever be 100 % accurate, but we will take new phrases and identify them, for example sarcasm and irony, as much as possible. As further improvement, we plan to add neutral sentiment as a separate category, and we suppose it will also affect the results.

References

1. Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L., Ridella, S.: The “K"in K-fold Cross Validation. In: ESANN (2012)
2. Çelikkaya, G., Torunoğlu, D., Eryiğit, G.: Named Entity Recognition on Real Data: A Preliminary Investigation for Turkish. In: Proceedings of the 7th International Conference on Application of Information and Communication Technologies, AICT 2013. IEEE, Baku (2013)
3. Davis, J., Goadrich, M.: The relationship between Precision-Recall and ROC curves. In: ICML, pp. 233–240 (2006)
4. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explorations 11(1), 10–18 (2009)
5. Liang, A.: Rotten Tomatoes: Sentiment Classification in Movie Reviews. CS 229 (15 2006)
6. Liu, B.: Sentiment analysis and subjectivity. In: Handbook of Natural Language Processing, 2nd edn. Taylor and Francis Group, Boca (2010)
7. Montejo-Ráez, A., Martìnez-Cámara, E., Martìn-Valdivia, T.M., Ureña-López, A.L.: Random Walk Weighting over SentiWordNet for Sentiment Polarity Detection on Twitter. In: Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis, pp. 3–10. Association for Computational Linguistics (2012)
8. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2007)
9. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. CoRR cs.CL/0205070 (2002)
10. Smirnov, I.: Overview of Stemming Algorithms. Mechanical Translation (December 2008)
11. Turney, P.D.: Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In: ACL, pp. 417–424 (2002)
12. Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., Cardie, C., Riloff, E., Patwardhan, S.: OpinionFinder: A System for Subjectivity Analysis. In: HLT/EMNLP (2005)