ISSN 1870-4069

Combined Dictionary Approach to Opinion Analysis in Slovak

Martin Mikula[1], Xiaoying Gao[2],
Kristína Machová[1]

[1] Technical University of Košice,
Department of cybernetics and artificial intelligence,
Slovakia

[2] Victoria University of Wellington,
School of Mathematical and Computing Sciences,
New Zealand

{martin.mikula, kristina.machova}@tuke.sk
xiaoying.gao@ecs.vuw.ac.nz

Abstract. People produce more and more textual data every day. They speak with each other, write articles and comment on products and services. It is simple to analyze them manually, in case that we have a small amount of data. But when we have many data, it is very difficult to process them manually. We decided to use dictionary approach for the automatic analysis of comments in the Slovak language. The first algorithm achieved accuracy around 72%. One disadvantage was that the algorithm could not identify the polarity of all comments. More than 18% comments were not assigned polarity because they did not contain subjective words from the dictionary. The new approach combines the first dictionary approach with a probabilistic method which is used to create a new lexicon. The new dictionary was again used to analysis comments in the dataset. This new approach reduced the percentage of unidentified comments to 0.5%. The new approach outperformed the previous method and also achieved better results than Naïve Bayes classifier and Support Vector Machines (SVM) on the same dataset.

Keywords: Opinion analysis in Slovak, polarity of opinion, probability approach.

1 Introduction

People communicate through the Internet, talking about their feelings, comparing and rate the products and writing the blogs etc. All of these written texts are sources of very important information for people and companies who work with internet marketing and satisfaction surveys. If there are not a lot of data, it is easy to analyze them manually. When we have big volume of data, it is better to analyze them automatically. This is the main field of study of opinion classification.

Opinion classification is not a simple process. In this process, we try to identify the opinion of an author about a specific topic. An author is a person or a company, that has the specific opinion about the topic, who can also be called the opinion holder.

A topic means the object that the author talks about. It can be a product, a service or anything else. In some literature, opinion classification is called opinion mining. The opinion in opinion analysis consists of two parts. The first part is subjectivity/polarity. We know three basic types of subjectivity: positive, negative and neutral. The second part is the strength of this polarity. Some words can express polarity with higher strength than other, for example *the best* is more positive than *good*.

In this paper, we used the combination of two approaches to sentiment analysis. The first one is based on a dictionary. We created our own lexicon based on the English lexicon. This approach implements several functions (intensification, negation, stemming) to sentiment processing and classification. It achieved average results in the experiments. One of the main limitations was that the approach can not process around 18% of the comments. We decided to implement machine learning approach to classify these unclassified comments.

We chose multinomial Naïve Bayes classifier (NBC). It is a simple classifier based on probability theorem with good results for sentiment analysis. In the first phase, the combined approach used the lexicon approach to classify comments in the dataset. Then it split processed comments into two groups, classified and unclassified. The classified ones were used as the training dataset for Naïve Bayes classifier. In the last phase, the comments which were not classified by dictionary approach were classified by NBC. The final results of our combined approach outperformed other machine learning methods.

This paper is divided into five parts. The second part reviews the background and related work on sentiment analysis in English and Slovak language. In the third part, we described our combined which consists of the dictionary approach and the Naïve Bayes classifier. Experiments and results are in the fourth part. The last part is the conclusion and future work.

2 Background and Related Work

There are two basic types of methods for opinion analysis. The first type of methods are methods based on dictionaries. This method uses opinion words for text analysis. These words are stored in a sentiment lexicon. All words in the dictionary have been assigned polarity (positive or negative) and in some cases strength of polarity. In work by Hu and Liu [5], the authors used a dictionary created through a bootstrapping process using WordNet[1].

They counted the numbers of positive and negative words around the product feature. When there were more negative words than positive ones, then the final opinion was negative and otherwise positive. Benamara et al. [1] look for the best words, that can be used for sentiment analysis. The authors proposed three scoring methods based on adverb-adjective combinations (AACs).

The first scoring function was variable scoring (VS) which was used for adjectives. In the adjective priority scoring (APS) they selected a weight r which referred to the strength between adverb and adjective which it modified.

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

In the third scoring function, adverb first scoring (AFS) the authors used the same weight *r*, but it was applied to the adjective rather to the adverb. Their experiments show, that the APS achieved the best results which mean, that the adjectives are most useful ones for sentiment analysis. The influence of the other parts of speech was analyzed in work by Taboada et al. [14]. They created the dictionary manually. Their lexicon contained adjectives, adverbs, nouns and verbs. These types of words together achieved better results than individually.

The authors also studied the influence of different types of sentence processing. They analyzed the usage of two types of negation. The first was switch negation and the second was shift negation. The intensification was performed by using of percentage. The authors compared their lexicon with other lexicons such as General Inquirer, MPQA subjectivity lexicon, SentiWordNet etc. These dictionaries were compared on the dataset obtained from Epinion.com. The approach presented in this work which included all types of words, shift negation and intensification with other features achieved the best results.

An application for opinion classification in the Slovak language based on dictionaries is described in Machová and Krajč [6]. This application used dictionaries of positive and negative words. It was able to process intensification and negation. The range of influence of intensification and negation was determined by a dynamic coefficient. This coefficient was set as static at the beginning, but it was calculated base on sentence length later. This application achieved 86.2% precision for positive and 69.2% for negative comments.

The second type of methods are approaches based on machine learning. These approaches use machine learning methods for opinion classification. The most common methods are Naïve Bayes classifier, Support Vector Machines (SVM) and Maximum Entropy. An SVM as machine learning method was used in work [9]. They trained the SVM with linear kernel on various features such as N-grams, number of all-capitalized words, POS tags, polarity dictionaries, punctuation marks, emoticons, lengthened words, clustering and negation. From N-grams, from 1- to 4-grams were considered. The authors used five different polarity dictionaries, three were manually created and two created automatically.

They also used non-single punctuation marks (e.g. !! or !?) and words with a letter repeating more than twice. Negation was applied from the begging of a negation to the first punctuation. Their approach obtained 69.02% macro F-measure. A maximum entropy-based classifier was used in the work Proisl et al. [12]. They used N-grams, the length of tweet, polarity dictionary, emoticons and abbreviations and negation as a feature set. The authors applied only unigrams and bigrams. In contrast to previous approach, the frequency-weighted N-grams were used. The improved AFINN-111 lexicon [10] was implemented to this approach. The authors added 343 new words to this lexicon.

Also a list of emoticons and abbreviations was used. Negation was applied only on the next three words. This approach achieved 63.06% macro F-measure. The work by Habernal et al. [3] is dedicated to the sentiment analysis in Czech language. The authors used several pre-processing methods such as stemming, POS tagging, misspellings and grammatical corrections.

Table 1. Examples of words stored in the lexicon.

Word	Strength of polarity	Polarity
zlý (bad)	-1	n
dobrý (good)	1	p
najlepší (best)	3	p
veľmi (very)	2.0	i
dosť (enough)	1.5	i
nebol (wasn't)	-1	o
trochu (litte)	1.25	i

They experimented with five feature selection methods (N-grams, character N-gram features, POS-related features, emoticons and TFIDF variants). Their methods were tested with two classifiers: SVM and Maximum entropy. The best results were achieved by the maximum entropy using all types of features. This method was the best on all tested datasets.

In some approaches, the researchers use the two or more types of methods together and combine them. Sindhwani and Melville [13] combined the unsupervised and the semi-supervised learning method. The unsupervised method was based on the dictionary which contained 2986 human labeled words. This method was utilized for domain adaptation. After this step, the semi-supervised algorithm was used for lexical classification.

The bipartite representation of data (document-word bipartite) was used as the semi-supervised method. They used Regularized Least Squares (RLS) as the classification algorithm because they had sparse data. The prior knowledge of sentiment words was incorporated into the model by lexical RLS. This approach was tested on three different domains. The lexical RLS was compared with semi-supervised lexical RLS and achieved better results. The combination of four different approaches was used in the work Hagen et al. [4].

The authors reimplemented following methods: 1. an SVM classifier from work Mohammad et al. [9], 2. a stochastic gradient descent classifier [2], which was trained on unigrams, stems, clustering polarity dictionary and negation, 3. a maximum entropy-based classifier from work [12] and 4. a logistic regresion [8] as a supervised machine learning was used. It was trained on POS tags, N-grams, and polarity dictionaries. The final decision was made on the average probability (the confidence score) of each classifier for each of the three classes. This ensemble achieved F1-score 64.84%.

3 Our Combined Approach

In our approach, we first used the dictionary approach for sentiment analysis in Slovak. We created the lexicon which contained 1430 words. It achieved accuracy around 72%. But more than 18% comments were not classified because they did not contain words from the dictionary. This was the reason that we tried to implement a probability approach to generating a new lexicon. This lexicon was generated from comments, that were labeled by the previous dictionary based approach. Then we used this new lexicon for labeling the corpus again.

3.1 Dictionary Approach

Our lexicon was created by translating from an English lexicon used in work Hu and Liu [5]. The original lexicon contained 6789 words, 4783 negative and 2006 positive words. We used Slovak thesaurus for searching synonyms to every word which we translated. The Slovak lexicon contained 598 positive words, 772 negative words, 41 intensifiers and 19 negations. Every word in the dictionary had been assigned *polarity*, and *strength of polarity*. We used 4 types of polarity:

- p - positive word,
- n - negative word,
- i - intensifier,
- o - opposite.

For polarity strength, we decided to use a range from -3 (the most negative) to +3 (the most positive). The words were stored in two forms in the lexicon. If it was possible, the word was saved in the edited form. We could edit words which finish with vowels: ”a”, ”e”, ”i”, ”u”, ”y”, in their basic form. Other words were saved in a form without editing. Examples of words in the lexicon are in Table 2.

Our algorithm for sentiment analysis worked in several phases. In the pre-processing phase, it edited the incoming text. It removed diacritics, changed all letters to lowercase and edited words with the Lancaster stemming algorithm[2]. The algorithm split the text into sentences and words. Every word from comment was compared with words in the lexicon.

If the algorithm found the word in the dictionary, it updated the polarity of the sentence by adding the strength of polarity of the word. We implemented two functions (intensification and negation) to text processing into this algorithm. The first function was intensification. Our intensifier had an assigned strength of polarity from 1.0 to 2.0. This type of intensification allowed us to assign higher intensity for words with high strength of polarity and lower intensity for words with the small strength of polarity. For example, if we had a combination:

- Veľmy dobrý (very good) where *very* is intensifier with value 2.0 and *good* has value +1 $\rightarrow$ the phrase had value of polarity $1\text{x}2.00 = 2$,
- Dosť dobrý (enough good) where *enough* is intensifier with value 1.5 and *good* has value +1 $\rightarrow$ the phrase had value of polarity $1\text{x}1.5 = 1.5$.

The second function was a negation. The algorithm was able to process two types of negation by using combined negation processing [7]. The negation is processed by the following rules:

if word is opposite **then**
    search the first word with polarity
    if strength of polarity is +/-1 or +/-2 **then**
        use switch negation

[2] http://web.archive.org/web/20140725115219/http://www.comp.lancs.ac.uk/computing/research/stemming/index.htm

```
        else
            if strength of polarity is +/-3 or intensifier then
                use shift negation
            end if
        end if
    end if
```

The switch negation changes the value of the word polarity to the opposite with the same strength. The shift negation shifts the strength of polarity to the opposite way by an exact value. We used two/negative two as value for shift negation in our algorithm. Here are a few examples of our implementation of negation.

1. Switch negation: Film nebol dobrý. (The movie wasn't good.) $\rightarrow$ *wasn't* is opposite and *good* has value +1 $\rightarrow -1\text{x}1 = -1$.
2. Shift negation: Film nebol najlepší. (The movie wasn't the best.) $\rightarrow$ *wasn't* is opposite and *the best* has value +3 $\rightarrow -2 + 3 = -1$.
3. Shift negation: Film nebol veľmi dobrý. (The movie wasn't very good.) $\rightarrow$ *wasn't* is opposite, *very* is intensifier with value 2.0 and *good* has value +1 $\rightarrow -2+2.0\text{x}1 = 0$.

The described approach achieved accuracy around 72%. The big problem was that this version of the algorithm and the actual dictionary were not able to classify more than 18% comments in the dataset.

3.2 Probability Method

In order to classify comments which did not contain words from the dictionary, we decided to use a method which created a new dictionary from comments that had been classified by the dictionary approach. This method divided classified comments into the positive and negative group. These groups were sorted from the most positive/negative to the less positive/negative.

Then the two groups were compared to discover which one contained fewer comments. We join the two datasets together and form a new dataset. A new joined dataset was created using all comments in the smaller group and the same number of samples were chosen from the bigger group, so the new dataset is balanced. This method created a training dataset which contained 50% of positive and 50% of negative comment. This distribution was very important because, if we had more comments with one polarity, it could influence the results.

The training dataset was used to generate a new dictionary. The algorithm splits all comments in training dataset to sentences and words. Then it counted term frequency depending on the presence of each word in positive and negative comments. The new lexicon was based on term frequency of each word connected with each class (positive or negative).

Then if the word from dictionary was found in a new comment, the probability *P*, that this word *w* is from class *c* was computed by the simple probability method described in formula 1:

$$P(w_c) = \frac{\sum w_c}{\sum w}, \tag{1}$$

where:
$P(w_c)$ - the probability that the word is from class *c*,
$\sum w_c$ - the number of occurrences of word *w* in class *c*,
$\sum w$ - the number of occurrences of word *w* in the whole dataset.

In case that the word was not connected with a specific class and the probability would be zero, we implemented a method which returned a very low number instead of zero.

The value of polarity of the new comment consists of probabilities of each word connected to the positive and negative class. We computed the probability by formula 2:

$$P(sentence_c) = \frac{\sum P(w_c)}{\sum w}, \tag{2}$$

where:
$P(sentence_c)$ - the probability that sentence is from class *c*,
$\sum P(w_c)$ - the summed probabilities of each word from the sentence which is from class *c*,
$\sum w$ - the number of words in sentence.

The new comment was added to the class which had bigger final probability. The comment was positive if the positive probability was bigger than negative probability and vice-versa.

Whole process can be described by following:

classify the dataset by the dictionary approach
for all all comments in the dataset **do**
    if comment has label **then**
        add comment to the positive or the negative subset
    end if
end for
if size of the positive subset $>$ size of the negative subset **then**
    add to the training set whole negative subset and same number of comments from the positive subset
else
    add to the training set whole positive subset and same number of comments from the negative subset
end if
train classifier
for all all comments in the dataset **do**
    classify comment
end for

This new approach reduced the number of unclassified comment to 0.5%.

Table 2. Opinion classification using dictionary approach (DA).

Approach	F1(+)	F1(-)	Macro F1
DA with unclassified comments	0.742	0.645	0.694
DA without unclassified comments	0.89	0.824	0.857

Table 3. The comparison of different approaches for opinion analysis.

Approach	F1(+)	F1(-)	Macro F1
Our dictionary approach	0.742	0.645	0.694
Our combined approach	**0.872**	**0.863**	**0.868**
Naïve Bayes	0.811	0.812	0.812
SVM	0.845	**0.863**	0.854

4 Experiments and Results

The proposed approach was tested on a dataset with 5242 comments. The dataset contains 2573 positive and 2669 negative comments from different areas (politics decisions, electronics and books reviews, movie reviews, etc.). The comments contain 182 645 words. The neutral comments were removed.

The metrics which we used to evaluate our method were based on *precision* and *recall* to obtain *F1* measure per class (positive and negative). *F1* is the harmonic mean between precision and recall. Some datasets are unbalanced, so we decided to use *Macro-F1* that is computed by calculating *F1* values for each class, and then averaging over all classes. Thus, *Macro-F1* shows the effectiveness in each class, independently of the size of the class.

In our first experiment, we tested the dictionary approach. We present two set of results in Table 4. The first row shows the results with unclassified comments where the unclassified ones were treated as wrong results. The second row is the result with the only classified ones.

These results show that the unclassified comments degrade the performance. The accuracy of results with unclassified comments was around 72%, and without unclassified comments around 86%. This version of the algorithm was not able to classify 976 comments which are around 18%. The unclassified comments constituted 65% of the incorrectly categorized positive comments and 60% of the incorrectly categorized negative comments. This was the reason why we decided to create this combined approach.

In our next experiment, we compared our combined approach with other approaches. It was compared with previous dictionary approach and with two machine learning methods: Naïve Bayes classifier and SVM. The two machine learning methods (NB and SVM) are implemented in data mining tool RapidMiner[3] and used 10 fold cross-validation to got the results.

The results show that our combined approach achieved better results than the original dictionary approach and the Naïve Bayes classifier. The proposed combined approach highly outperformed the original dictionary based approach. The combined approach can classify 5226 comments from the dataset which represents 99.7%.

[3] https://rapidminer.com/

Table 4. The comparison of the dictionary approach (DA) and the combined approach (CA) on the different datasets.

Approach	F1(+)	F1(-)	Macro F1
DA on our dataset	0.742	0.645	0.694
CA on our dataset	0.872	0.863	0.868
DA on movie review	0.727	0.611	0.669
CA on movie review	0.741	0.599	0.670

The combined approach also achieved significantly better results than the Naïve Bayes classifier in all measures. This approach obtained slightly better results than SVM. In our next experiment, we compared our approach with a standard movie review dataset used in work Pang and Lee [11]. This dataset includes 1,000 positive and 1,000 negative movie reviews annotated by the human annotator. They were collected from rottentomatos.com. All text was translated to the Slovak language and converted to lowercase.

From the results in table 4 can be seen, that combined approach was better on our dataset. It was also slightly better on the standard dataset. The analysis on movie dataset was also compared with the results from the work by Taboada[14].

They achieved performance between 68.05% (a simple sentiment analysis using just words from the dictionary) and 76.37% (a sentiment analysis based on additional features). Our combined dictionary achieved slightly worse results which can be caused by misunderstanding during automatic translation process.

5 Conclusion

In this paper, we proposed a new combined approach to opinion analysis. It was created by the combination of a dictionary approach and a probability method. The dictionary approach had analyzed unlabeled data and classified comments into the positive and the negative group. But it can not analyze around 18% comments.

They were not labeled because these comments did not contain words from the dictionary. For this reason, we implemented a probabilistic approach which created new dictionary based on comments analyzed by the dictionary approach. The algorithm found new polarity words, typical for this dataset and added them to a new dictionary.

The dataset was again analyzed with this new dictionary and more than 99.6% comments were classified. This combined approach achieved better results than the previous dictionary approach and the Naïve Bayes classifier. It was compared also with SVM with which achieved comparable results. In the future research, we want to use this combined approach for domain adaptation and active learning in the data stream.

If this approach will be used in active learning, the dictionary approach can create training dataset. This dataset will be used for training classifier and the trained classifier will be then used for labeling other data from the stream. In case that the domain will be changed, the algorithm will use this combined approach for creating new training dataset, especially for this new domain. The machine learning algorithm can be retrained on this new training dataset.

Acknowledgments. The work presented in this paper was supported by the Slovak Grant Agency of the Ministry of Education and Academy of Science of the Slovak Republic under VEGA grant No. 1/0493/16 (70%) and by the European Commission through the Thelxinoe project (30%). This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

References

1. Benamara, F., Cesarano, C., Picariello, A., Reforgiato, D., Subrahmanian, V. S.: Sentiment analysis: Adjectives and adverbs are better than adjectives alone. In: Proceedings of the International Conference on Weblogs and Social Media (2007)
2. Günther, T., Furrer, L.: GU-MLT-LT: Sentiment analysis of short messages using linguistic features and stochastic gradient descent. In: Proceedings of the Seventh International Workshop on Semantic Evaluation, vol. 2, pp. 328–332 (2013)
3. Habernal, I., Ptáček, T., Steinberger, J.: Supervised sentiment analysis in Czech social media. Information Processing and Management, vol. 50, no. 5, pp. 693–707 (2014) doi: 10.1016/j.ipm.2014.05.001
4. Hagen, M., Potthast, M., Büchner, M., Stein, B.: Webis: An ensemble for twitter sentiment detection. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval '15), pp. 582–589 (2015) doi: 10.18653/v1/S15-2097
5. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177 (2004) doi: 10.1145/1014052.1014073
6. Machová, K., Krajč, M.: Opinion classification in threaded discussions on the web. In: 10th Annual International Conference Znalosti, pp. 136–147 (2011)
7. Mikula, M., Machová, K.: Spracovanie negácie pre klasifikáciu názorov v slovenskom jazyku. In: Data a znalosti, pp. 41–45 (2015)
8. Miura, Y., Sakaki, S., Hattori, K., Ohkuma, T.: TeamX: A sentiment analyzer with enhanced lexicon mapping and weighting scheme for unbalanced data. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval '14), pp. 628–632 (2014) doi: 10.3115/v1/S14-2111
9. Mohammad, S. M., Kiritchenko, S., Zhu, X.: NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In: Proceedings of the Seventh International Workshop on Semantic Evaluation, pp. 321–327 (2013) doi: 10.48550/arXiv.1308.6242
10. Nielsen, F. Å.: A new ANEW: evaluation of a word list for sentiment analysis in microblogs. pp. 93–98 (2011) doi: 10.48550/arXiv.cs/0409058
11. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (2004) doi: 10.3115/1218955.1218990
12. Proisl, T., Greiner, P., Evert, S., Kabashi, B.: KLUE: Simple and robust methods for polarity classification. In: Proceedings of the Seventh International Workshop on Semantic Evaluation, pp. 395–401 (2013)
13. Sindhwani, V., Melville, P.: Document-word co-regularization for semi-supervised sentiment analysis. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, pp. 1025–1030 (2008) doi: 10.1109/ICDM.2008.113
14. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Computational linguistics, vol. 37, no. 2, pp. 267–307 (2011) doi: 10.1162/COLI_a_00049