Two-stage SVM sentiment and subjectivity classification

Jakub Dutkiewicz, Czesław Jędrzejek
Institute of Control and Information Engineering, Poznań University of Technology
Pl. M. Skłodowskiej-Curie 5, 60-965 Poznań, Poland
czeslaw.jedrzejek@put.poznan.pl, kubadt@gmail.com

Abstract

In this paper, we present the method of two stage linear classifier for NLP tasks. The method bases on division of the training data into two subsets. Division is performed with simple rules, but the messages are placed in cross subsets. We include a brief description of related works. We present the algorithm of the method. We present the results for two corpora used by Pang and Lee. The results are better than any literature results within the categories of methods not oriented on NLP.

1 Introduction

Since the pioneering works of (Pang et al., 2002) on polarity dataset v2.0, and (Pang and Lee, 2004), on subjectivity dataset v1.0, many approaches were used for accuracy classification. The most successful NLP method (Socher et al., 2013) analyses whole sentences. It learns vector space representations for multi-word phrases. In sentiment prediction tasks these representations outperform other state-of-the-art approaches on commonly used datasets, such as movie reviews, without using any pre-defined sentiment lexica or polarity shifting rules. Several works extended standard tf-idf weighting scheme for general extension, most notably, (Ghag and Shah, 2014) by classifying a term as positive or negative based on its proportional frequency count distribution and proportional presence count distribution across positively tagged documents in comparison with negatively tagged documents. Yet another approach used a multistage linear SVM classification. This method showed significant improvement of classification measures for selected datasets (Xia et al., 2012), however with regard to sentiment classification the gain was marginal (Nguyen et al., 2013).

In this work the method is extended to the two-stage sentiment and subjectivity classification.

2 Multi stage classification

The mentioned multistage classifier (Nguyen et al., 2013) uses two stage classification. There is a primary classification/rejection stage and a secondary classification stage. It uses classifiers of two different types, first one being Naive Bayes classifier and a second one being SVM classifier. If a message gets rejected at the first stage, it is being classified by the next classifier. In this work, we present a different approach. We use three different classifiers for the two stages. The first stage determines, which classifier will be used at the second stage - the second stage is the actual classification. This approach is similar to (Xia et al., 2012), however the latter one consists of multiple stages and uses different data to train certain classifiers.

3 Our approach

We divide the training set into two subsets. We use the set of words *s* to determine the content of the subsets. Words within the set *s* should have high impact on the misclassification. Division of the training set is illustrated in Table 1.

Table 1. Division of the subsets

	Positive messages	Negative messages
Messages which contain at least one word from *s*	Subset 1	Subset 2'
Messages which do not contain words from *s*	Subset 2	Subset 1'

We train classifiers *c1* (operating on the sum of the Subset 1 and Subset 1') and *c2* (operating on the sum of the Subset 2 and Subset 2') *and* we train classifier *c* for

the entire corpus. Afterwards, we train the classifier *ck*, which is responsible for classifying the message as an element of the sum of the Subset 1 and Subset 1' (later labeled as Subset I) or an element of the subset sum of the Subset 2 and Subset 2' (later labeled as Subset II). The classifiers *c1* and *c2* relate to sum of subsets on diagonal of the square in Table 1. The strength of the method comes from the fact that upon classification by *c* misclassified messages are reclassified by either *c1* or *c2*.

We use the classifier *ck* to determine if a message from the test set should be classified with the *c1* classifier or the *c2* classifier. We noticed, that classifiers *c1* and *c2* perform significantly better if the messages contain words from the *s* subset, while the performance of classifier *c* is not diminished for messages which does not contain words from the *s* subset. We have designed the classification process accordingly.

The classification process uses the following rules to classify messages:

1. If the message does contain any word from the *s* subset, use classifier *c*
2. If the message does not contain any word from the *s* subset and classifier *ck* classified the message as an element of Subset 1, use *c1* classifier to classify the message.
3. If the message does not contain any word from the *s* subset and classifier *ck* classified the message as an element of Subset 2, use *c2* classifier to classify the message.

Logical explanation. Let us use a sample corpus of 4 messages.

1. The movie is good.
2. The movie is not good.
3. The movie is bad.
4. The movie is not bad.

Initially, the difference between the messages within the entire corpus is statistically not significant. To divide the corpus into subsets we use the set *s* containing {"not"}. The Subset I consists of messages 3. and 4. The Subset II consists of messages 1. and 2. A division of messages in such a fashion makes success probability using the second stage much more likely.

The elements of the set *s* became good descriptors because the difference between positive and negative messages is more significant. In this particular case *good* and *bad* adjectives are good descriptors for the *ck* classifier.

Additional advantages of this method. It should be noted, that the method applies to any classification method (for example a combination of Bayes, maximum entropy, SVM or rule systems).

4 Experimental setup

To determine effectiveness of our approach we use the standard SVM classifier. We use unigrams as features. The SVM light (Joachim, 2002) implementation is used for training and testing. The sources of data are two standard corpora: introduced to the research by (Pang and Lee, 2004) - 1000 positive and 1000 negative reviews from movie databases and (Pang et al., 2002). The first corpus regards subjectivity of messages, let us refer to it as Subjectivity corpus. Second corpus regards sentiment polarity, let us refer to it as Sentiment corpus. We have divided corpora into test and training sets. The number of messages within those sets is presented in Table 2.

Table 2. Contents of corpora.

	Sentiment Corpus		Subjectivity corpus	
Training set	Pos	Neg	Subj	Obj
	500	500	1000	1000
Test set	200	200	4000	4000

Preprocessing. We have removed every punctuation mark, besides apostrophes from the messages. We have tokenized messages. We have used the Porter Stemmer for stemming. Every stem in the training set is being used as a unique feature.

5 Experiments and results

Experiment A. In this experiment, we use the Sentiment corpus. First, we design the *s* set for this corpus. We assume that words which represent negation are likely to cause misclassification. As the messages within the corpus are fairly long, we assign only a word *"not"* alone into the set *s* for this experiment. Within the 400 messages from test set, 331 contain the word "not". Afterwards, we have trained classifiers *c, c1*, *c2* and *ck*. For the 331 messages with the word "not" we have used combination of *c1,c2* and *ck* classifiers. For the remaining 69 messages we have used *c* classifier. We achieve a clear improvement. Accuracy for the classifiers is presented on Table 3.

Experiment B. In this experiment we study the Subjectivity corpus. Experiment B is similar to Experiment A. Messages in the corpus are shorter, so we use larger set *s*. Because it is fairly hard to determine which words have high impact on misclassification in the subjectivity corpus, we have chosen 5 stems which represent negation *{"not", "doesn't", "isn't", "aren't", "don't"}*. If we look at the Table 1, we can notice, that the *ck* classifier distinguishes the difference between positive and negative messages within two corpora. Instead of using newly trained *ck* classifier, we use *c* classifier to perform its job. We present accuracy of classifiers for the experiment B on the Table 4.

Table 3. Accuracy of classifiers for Sentiment test set

	Negative messages with word from *s*		Positive messages with word from *s*
c1	13,56%		100,00%
c2	100,00%		20,24%
ck	82,10%		84,62%
	All messages	Messages with word from *s*	Messages without any word from *s*
*c**	83,50%	83,07%	84,21%
final	85,02%	85,52%	84,21%
**	91,4%		
+	87,15		

*The same as (Pang et al, 2002) for all messages – the baseline method
** (Gamon and Aue, 2005)
+ The same as (Pang and Lee, 2004)

In fact, classifier *c* as the one, which classifies messages as elements of Subset 1 or Subset 2 outperformed classifier *ck* (93% vs 92% final accuracy).

It is seen that our method is better than the baseline (one stage) SVM, however it gives worse results than methods fully NLP oriented.

Discussion. The strength of the method comes from *c1* and *c2* classifiers. Accuracy of those classifiers exceeds 99% when messages fulfil condition that the classifier *c1* classifies the Subset I, and the classifier *c2* classifies the Subset II. Contrary, accuracy of those classifiers varies between 10% and 50% when the classifiers are used for the other subset (for example *c2* for the Subset 1). Let us use regular classifier *c* to choose the between *c1* and *c2* classifiers. When the *c* classifier makes a mistake, there is 10%-50% chance that the classification is going to be correct anyways. Basically, as long as the *c1* and *c2* have the accuracy of nearly 100% for designated subset and accuracy of over 1% for the other subset, there is going to be an improvement.

Table 4. Accuracy of classifiers for Subjectivity test set

	Negative messages with word from *s*		Positive messages with word from *s*
c1	48,73%		98,97%
c2	99,75%		42,99%
c	92,31%		92,23%
	All messages	Messages with word from *s*	Messages without any word from *s*
*c**	90,30%	92,26%	88,86%
final	93,12%	95,17%	88,86%

*The same as (Pang and Lee, 2004) for all messages

6 Conclusions and future work

In this work we have shown that using a two-stage classifier gives better results than these obtained with a single classifier. Previously, two-stage classifiers used by (Pang and Lee, 2004), (Aman, Szpakowicz, 2007), (Ptaszynski et al., 2014) were independent of each other. In contrast to these works in method classifiers at each stage depend on each other. The strength of this method

comes from reclassification of misclassified messages with use of classifier designed for the other subset. We expect that a combination of this method and an NLP oriented approach would give better result than pure NLP oriented approach.

In this work, we have not presented a statistical method or rule based method of choosing the set *s*. Because of the large number of possible subsets we cannot check every possible pair of subsets out. In future we will show the influence of distribution of words within the corpus on the evaluation measures. We plan to use this study for development of the algorithm for choosing words for the set *s*.

We plan to employ various sentiment classifiers for each stage of the algorithm. In particular, it would be favourable to use SentiTFIDF (Ghag and Shah, 2014), RNTN (Socher et al., 2013) and a rule based approach. We are going to improve the evaluation the method with usage of cross validation.

Our method allows for incorporating improvements used in other approaches.

It is straightforward to remove objective sentences from messages as done by (Pang and Lee, 2004) and apply the LLR feature selection method (Aue and Gamon, 2005). Then significant improvement of accuracy is expected.

We plan to extend the number of studied corpora to 10, including twitter corpus, auction portal comments corpora, movie reviews corpora, subjectivity corpus and at least three different languages for the evaluation.

Acknowledgements This work was supported by the Polish PUT 04/45/DSPB/0136 grant.

References

Aman, S., Szpakowicz, S. (2007). *Identifying expressions of emotion in text.* In: Proceedings of the 10th International Conference on Text, Speech, and Dialogue (TSD-2007), Lecture Notes in Computer Science (LNCS), Springer-Verlag.

Aue, A., and Gamon, M. (2005). *Customizing Sentiment Classifiers to New Domains: a Case Study*. In: Proceedings of RANLP-05, the International Conference on Recent Advances in Natural Language Processing 2005.

Gamon, M. and Aue, A. (2005). *Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms*. In: Proceedings of the ACL-05 Workshop on Feature Engineering for Machine Learning in Natural Language Processing, Association for Computational Linguistics, Ann Arbor.

Ghag, K. and Shah K. (2014). *SentiTFIDF. Sentiment Classification using Relative Term Frequency Inverse Document Frequency* International Journal of Advanced Computer Science and Applications(IJACSA), Volume 5 Issue 2, 2014, pp. 36-43

Joachims, T. (2002). *Learning to Classify Text Using Support Vector Machines.* Dissertation, Kluwer.

Nguyen, D. Q. and Pham, S. B., (2013). *A Two-Stage Classifier for Sentiment Analysis.* In: Proceedings of the 6th International Joint Conference on Natural Language Processing, IJCNLP'13, pp. 897-901.

Pang, B., and Lee, L. and Vaithyanathan S. (2002). *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), Vol. 10. Association for Computational Linguistics, Stroudsburg, PA, USA, 79-86.

Pang, B. and Lee, L. (2004). *A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts.* ACL 2004: 271-278

Ptaszynski, M., Masui, F., Rzepka, R., and Araki K. (2014). *Emotive or Non-emotive: That is The Question.* ACL 2014.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning C., Ng, A. and Potts, C. (2013). *Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank.* Conference on Empirical Methods in Natural Language Processing (EMNLP 2013).

Xia, Q., Shaikh, M. J., Ersoy, O. and Moskowitz H. (2012). *Multistage Linear SVM Classification.* Purdue technical reports, TR-ECE-07-12.