A Two-Stage Classifier for Sentiment Analysis

Dai Quoc Nguyen and **Dat Quoc Nguyen** and **Son Bao Pham**
Faculty of Information Technology
University of Engineering and Technology
Vietnam National University, Hanoi
{dainq, datnq, sonpb}@vnu.edu.vn

Abstract

In this paper, we present a study applying reject option to build a two-stage sentiment polarity classification system. We construct a Naive Bayes classifier at the first stage and a Support Vector Machine at the second stage, in which documents rejected at the first stage are forwarded to be classified at the second stage. The obtained accuracies are comparable to other state-of-the-art results. Furthermore, experiments show that our classifier requires less training data while still maintaining reasonable classification accuracy.

1 Introduction

The rapid growth of the Web supports human users to easily express their reviews about such entities as products, services, events and their properties as well as to find and evaluate the others' opinions. This brings new challenges for building systems to categorize and understand the sentiments in those reviews.

In particular, document-level sentiment classification systems aim to determine either a positive or negative opinion in a given opinionated document (Turney, 2002; Liu, 2010). In order to construct these systems, classification-based approaches (Pang et al., 2002; Pang and Lee, 2004; Mullen and Collier, 2004; Whitelaw et al., 2005; Kennedy and Inkpen, 2006; Martineau and Finin, 2009; Maas et al., 2011; Tu et al., 2012; Wang and Manning, 2012) utilizing machine learning to automatically identify document-level sentiment polarity are still mainstream methods obtaining state-of-the-art performances. It is because of possibly combining various features such as: bag of words, syntactic and semantic representations as well as exploiting lexicon resources (Wilson et al., 2005; Ng et al., 2006; Taboada et al., 2011) like SentiWordNet (Baccianella et al., 2010). In these systems, Naive Bayes (NB) and Support Vector Machine (SVM) are often applied for training learning models as they are frequently used as baseline methods in task of text classification (Wang and Manning, 2012). Although NBs are very fast classifiers requiring a small amount training data, there is a loss of accuracy due to the NBs' conditional independence assumption. On the other hand, SVMs achieve state-of-the-art results in various classification tasks; however, they may be slow in the training and testing phases.

In pattern recognition systems, reject option (Chow, 1970; Pudil et al., 1992; Fumera et al., 2000; Fumera et al., 2004) is introduced to improve classification reliability. Although it is very useful to apply reject option in many pattern recognition/classification systems, it has not been considered in a sentiment classification application so far.

In this paper, we introduce a study combining the advantages of both NB and SVM classifiers into a two-stage system by applying reject option for document-level sentiment classification. In the first stage of our system, a NB classifier, which is trained based on a feature representing the difference between numbers of positive and negative sentiment orientation phrases in a document review, deals with easy-to-classify documents. Remaining documents, that are detected as "*hard* to be correctly classified" by the NB classifier in the use of rejection decision, are forwarded to process in a SVM classifier at the second stage, where the *hard* documents are represented by additional bag-of-words and topic-based features.

2 Our approach

This section is to describe our two-stage system for sentiment classification. Figure 1 details an overview of our system's architecture.

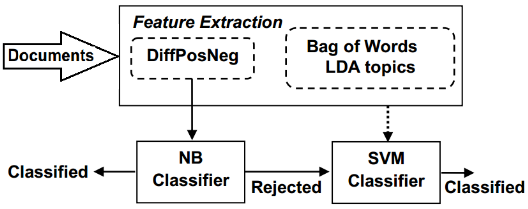

Figure 1: The architecture of our two-stage classifier.

In this positive (pos) and negative (neg) classification problem of sentiment polarity, we reject every sentiment document D satisfying the following rejection decision based on conditional probabilities:

$(\tau_1 > P(pos|D)$ **and** $P(pos|D) \geq P(neg|D))$
OR
$(\tau_2 > P(neg|D)$ **and** $P(neg|D) > P(pos|D))$

International Joint Conference on Natural Language Processing, pages 897–901,
Nagoya, Japan, 14-18 October 2013.

where thresholds $\tau_1, \tau_2 \in [0, 1]$. Otherwise, if document D does not satisfy the rejection decision, it is accepted to be classified by the NB.

A NB classifier at the first stage is to categorize accepted documents. Rejected sentiment documents, that are determined as *hard* to be correctly classified (most likely to be miss-classified) by the NB classifier in applying reject option, are processed at the second stage in a SVM classifier. In our system, the NB classifier categorizes document reviews based on a feature namely DiffPosNeg while the SVM one classifies document reviews with additional bag-of-words (BoW) and topic features.

DiffPosNeg feature

We exploit the opinion lexicons[1] of positive words and negative words (Hu and Liu, 2004) to detect the sentiment orientation of words in each document. We then employ basic rules presented in (Liu, 2010) to identify the sentiment orientation of phrases. The numerical distance between the numbers of positive and negative opinion phrases in a document D is referred to as its DiffPosNeg feature value.

BoW features

The BoW model is the most basic representation model used in sentiment classification, in which each document is represented as a collection of unique unigram words where each word is considered as an independent feature. We calculate the value of feature i in using *term frequency - inverse document frequency* weighting scheme for the document D as following:

$$BoW_{iD} = log(1 + tf_{iD}) * log\frac{|\{D\}|}{df_i}$$

where tf_{iD} is the occurrence frequency of word feature i in document D, $|\{D\}|$ is the total number of documents in the data corpus $\{D\}$, and df_i is the number of documents containing the feature i. We then normalize BoW feature vector of the document D as below:

$$\overrightarrow{\eta BoW_D} = \frac{\sum_{\delta \in \{D\}} \|\overrightarrow{BoW_\delta}\|}{|\{D\}| * \|\overrightarrow{BoW_D}\|} * \overrightarrow{BoW_D}$$

Topic features

Our system also treats each document review as a "bag-of-topics", and considers each topic as a feature. The topics are determined by using Latent Dirichlet Allocation (LDA) (Blei et al., 2003). LDA is a generative probabilistic model to discover topics for a corpus of documents. LDA represents each document as a probability distribution over latent topics, where each topic is modeled by a probability distribution over words. Using Bayesian inference methods, LDA computes posterior distribution for unseen documents. In our system, we refer to topic probabilities as topic feature values.

3 Experimental results

3.1 Experimental setup

We conducted experiments on the publicly available standard polarity dataset V2.0[2] of 2000 movie reviews constructed by Pang and Lee (2004).

We did not apply stop-word removal, stemming and lemmatization because such stop-words as negation words (e.g: no, not, isn't) were used in the basic rules to reverse the sentiment orientation of phrases, and as pointed out by Leopold and Kindermann (2002) stemming and lemmatization processes could be detrimental to accuracy. We kept 4000 most frequent words for each polarity class, after removing duplication, we had total 5043 BoW features.

For extracting LDA topic features, we used the JGibbLDA implementation[3] developed by Phan and Nguyen (2007), in which α is set to 0.5, β is set to 0.1 and the number of Gibbs sampling iterations is set to 3000. We exploited a corpus[4] of 50000 unlabeled movie reviews published by Maas et al. (2011) to build LDA topic models. We then applied these models to compute the posterior probability distribution over latent topics for each movie review in the experimented dataset of 2000 reviews.

In order to compare with other published results, we evaluate our classifier based on 10-fold cross-validation. We randomly separate the dataset into 10 folds; giving one fold size of 100 positive and 100 negative reviews. This evaluation procedure is repeated 10 times that each fold is used as the testing dataset, and 9 remaining folds are merged as the training dataset. All our performance results are reported as the average accuracy over the testing folds.

We utilized WEKA's implementations (Hall et al., 2009) of NB and SVM's fast training Sequential Minimal Optimization algorithm (Platt, 1999) for learning classification with the WEKA's default parameters (e.g: the linear kernel for SVM).

3.2 Results without reject option

Table 1 provides accuracies achieved by the single NB and SVM classifiers without the reject option: our NB and SVM classifiers were trained on the whole training dataset of 9 folds according to the above 10-fold cross-validation scheme. We consider BoW model as a baseline, similar to other approaches (Pang and Lee, 2004; Whitelaw et al., 2005; Tu et al., 2012).

In table 1, the accuracy results based on only *DiffPosNeg* feature are 70.00% for NB and 69.55% for SVM. The highest accuracies in utilizing LDA topics are 78.05% for NB classifier and 85.30% for SVM classifier due to 50 topic features. Besides, the accuracy accounted for SVM at 86.30% over the combination of

[1] http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar

[2] http://www.cs.cornell.edu/people/pabo/movie-review-data/

[3] http://jgibblda.sourceforge.net/

[4] http://ai.stanford.edu/~amaas/data/sentiment/

Table 2: Results in applying reject option (8 folds for training), and in other SVM-based methods

τ_1	τ_2	r_{Pos}	r_{Neg}	NB		SVM		Accuracy
0.79	0.81	0.764	0.987	236	13	1519	232 (tuned thresholds)	**87.75**
0.82	0.80	0.796	0.990	205	9	1554	232	**87.95**
1.0	1.0	1.0	1.0	0	0	1752	248	87.60

Pang and Lee (2004)	BoW	87.15
	BoW with minimum cuts	87.20
Whitelaw et al. (2005)	BoW (48314 features)	87.00
	BoW and appraisal groups (49911 features)	90.20
Kennedy and Inkpen (2006)	Contextual valence shifters with 34718 features	86.20
Martineau and Finin (2009)	BoW with smoothed delta IDF	88.10
Maas et al. (2011)	Full model and BoW	87.85
	Full model + additional unlabeled data + BoW	88.90
Tu et al. (2012)	BoW	87.05
	BoW & dependency trees with simple words	88.50
Wang and Manning (2012)	NBSVM-Unigram	87.80
	NBSVM-Bigram	89.45

Table 1: Results without reject option

Features	NB	SVM
BoW (**baseline**)	73.55	**86.05**
20 LDA topics	77.55	82.05
30 LDA topics	74.95	79.65
40 LDA topics	76.60	82.15
50 LDA topics	78.05	85.30
60 LDA topics	75.80	83.40
DiffPosNeg	70.00	69.55
DiffPosNeg & BoW	73.50	86.30
DiffPosNeg & 50-LDA	79.35	85.45
BoW & 50-LDA	73.60	87.70
DiffPosNeg & BoW & 50-LDA	73.85	87.70

DiffPosNeg and BoW features is greater than the baseline result of 86.05% with only BoW features. By exploiting a full combination of DiffPosNeg, BOW and 50 LDA topic features, the SVM classifier gains the exceeding accuracy to 87.70%.

3.3 Results in applying reject option

In terms of evaluating our two-stage approach, if the fold$_{i^{th}}$ is selected as the testing dataset, the fold$_{(i^{th}+1)\%10}$ will be selected as the development dataset to estimate reject thresholds while both NB and SVM classifiers will be learned from 8 remaining folds. By varying the thresholds' values, we have found the most suitable values τ_1 of 0.79 and τ_2 of 0.81 to gain the highest accuracy on the development dataset.

Table 2 presents performances of our sentiment classification system in employing reject option, where the NB classifier was learned based on the DiffPosNeg feature, and the SVM classifier was trained on the full combination of DiffPosNeg, BoW and 50 LDA topic features (total 5094 features). In the table 2, r_{Pos} and r_{Neg} are reject rates corresponding with positive label and negative label in the *testing* phase:

$$r_{Pos} = \frac{number\ of\ rejected\ positive\ reviews}{1000}$$

$$r_{Neg} = \frac{number\ of\ rejected\ negative\ reviews}{1000}$$

$$Overall\_reject\_rate = \frac{r_{Pos} + r_{Neg}}{2}$$

With the values τ_1 of 0.79 and τ_2 of 0.81, our two-stage classifier achieves the result of 87.75% on the testing dataset that as illustrated in table 2, it is comparable with other state-of-the-art SVM-based classification systems, many of which used deeper linguistic features. In total 10 times of cross fold-validation experiments for this accuracy, the NB accepted 249 documents to perform classification and rejected 1751 documents to forward to the SVM. Specifically, the NB correctly classified 236 documents whilst the SVM correctly categorized 1519 documents.

Additionally, in the setup of taking 8 folds for training NB and SVM, and not taking 1 fold of development into account, by directly varying values τ_1 and τ_2 on the testing dataset, our system can reach the highest result of 87.95% which is 1.9% and 0.35% higher than the SVM-based baseline result (86.05%) and the accuracy (87.60%) of the single SVM classifier without reject option, respectively.

3.4 Results in using less training data

To assess the combination of advantages of NB (requiring small amount of training data) and SVM (high performance in classification tasks), we also carried out experiments of using less training data. In this evaluation, if the fold$_i$ is selected as testing data, the fold$_{(i+1)\%10}$ will be selected as training dataset to build the NB classifier. Applying the rejection decision on 8 remaining folds with given reject thresholds, the dataset of rejected documents are used to learn the SVM classifier.

In experiments, the single NB classifier without reject option attains an averaged accuracy of 69.9% that is approximately equal to the accuracy on 9-fold training dataset at 70% as provided in the table 1. This comes from that our proposed *DiffPosNeg* feature is simple enough to obtain a good NB classifier from small training set. In these experiments, the given thresholds applied in the training phase to learn the SVMs are reused in the testing phase (i.e. the same thresholds for both training and testing phases).

Table 3: Reject option results using less training data

τ_1	τ_2	r^*_{Pos}	r^*_{Neg}	r_{Pos}	r_{Neg}	Acc_S	Accuracy
0.95	0.63	0.722	0.478	0.722	0.475	84.80	80.55
0.64	0.75	0.483	0.723	0.486	0.729	84.80	82.35
0.72	0.65	0.495	0.496	0.491	0.494	83.75	80.50
0.78	0.69	0.606	0.605	0.609	0.600	84.65	82.30
0.88	0.74	0.764	0.770	0.765	0.770	85.80	84.35
0.97	0.78	0.906	0.905	0.908	0.910	86.65	85.75

Table 3 summaries some reject option-based results taking less training data to learn the SVMs based on the full combination of 5094 features, where r^*_{Pos} and r^*_{Neg} are reject rates in the *training* phase, and Acc_S denotes the accuracy of the single SVM classifier without reject option. With the modest overall reject rate of 0.493 in testing phase, our classifier reached an accuracy of 80.50%, which it outperformed the single NB.

Table 4: Results with SVM trained on *DiffPosNeg* and BoW

τ_1	τ_2	r^*_{Pos}	r^*_{Neg}	r_{Pos}	r_{Neg}	Acc_S	Accuracy
0.95	0.63	0.722	0.478	0.722	0.475	84.50	80.55
0.64	0.75	0.483	0.723	0.486	0.729	84.00	81.60
0.80	0.68	0.618	0.591	0.622	0.585	83.90	81.05
0.85	0.73	0.726	0.745	0.732	0.753	84.65	83.65
0.97	0.78	0.906	0.905	0.908	0.910	85.70	84.85
0.92	0.80	0.854	0.941	0.861	0.945	85.70	85.35

In other experiments using less training data as presented in table 4, we trained the SVM classifier based on the combination of *DiffPosNeg* and BoW features. For the overall reject rate of 0.903 in testing phase, our system gained a result of 85.35% that is a bit of difference against the accuracy of the single SVM at 85.70%.

Table 3 and table 4 show that our classifier produced reasonable results in comparison with single NB and SVM classifiers without reject option.

3.5 Discussion

It is clearly that a different set of features could be used for learning the NB classifier at the first classification stage in our system. However, as mentioned in section 3.4, it is sufficient to have a good NB classifier learned from an unique *DiffPosNeg* feature. Furthermore, an obvious benefit of having the NB based on only one *easy-to-extract* feature is to enhance the efficiency in terms of time used in the document classification process. That is the reason why we applied only the DiffPosNeg feature at the first stage.

With regards to the processing time efficiency, it is because there are no recognition time evaluations associated to the other compared systems as well as it is not straightforward to re-implement those systems, hence, the comparison over processing time with the other systems is not crucial to our evaluation. Nevertheless, we believe that our classifier enables to get a fast complete recognition in which time spent to extract features is also taken into accounts, where the majority amount of the classification time is allocated to the feature extraction process.

Considering to feature extraction time, let Γ_1 be the time taken to extract DiffPosNeg feature and Γ_2 be the time spent for extracting other features (i.e. BoW and LDA topic features): our two-stage system then costs ($\Gamma_1+overall\_reject\_rate*\Gamma_2$) as opposed to ($\Gamma_1+\Gamma_2$) by the single SVM without reject option. Depending on the overall reject rate, our system could get a significant increase in the complete recognition time while the returned accuracy of our system is promising compared to that of the single SVM classifier.

4 Conclusion

In this paper, we described a study combining NB and SVM classifiers to construct a two-stage sentiment polarity system by applying reject option. At the first stage, a NB classifier processes easy-to-classify documents. Hard-to-classify documents, which are identified as most likely to be miss-classified by the first NB classifier in using rejection decision, are forwarded to be categorized in a SVM classifier at the second stage.

The obtained accuracies of our two-stage classifier are comparable with other state-of-the-art SVM-based results. In addition, our classifier outperformed a bag-of-words baseline classifier with a 1.9% absolute improvement in accuracy. Moreover, experiments also point out that our approach is suitable for under-resourced tasks as it takes less training data while still maintaining reasonable classification performance.

Acknowledgment

The authors would like to thank Prof. Atsuhiro Takasu at the National Institute of Informatics, Tokyo, Japan for his valuable comments and kind support.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 2200–2204.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March.

C. Chow. 1970. On optimum recognition error and reject tradeoff. *IEEE Trans. Inf. Theor.*, 16(1):41–46, September.

Giorgio Fumera, Fabio Roli, and Giorgio Giacinto. 2000. Reject option with multiple thresholds. *Pattern Recognition*, 33(12):2099–2101, December.

Giorgio Fumera, Ignazio Pillai, and Fabio Roli. 2004. A two-stage classifier with reject option for text categorisation. In *Proceedings of Joint IAPR International Workshops SSPR 2004 and SPR 2004*, volume 3138, pages 771–779.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Alistair Kennedy and Diana Inkpen. 2006. Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. *Computational Intelligence*, 22(2):110–125.

Edda Leopold and Jörg Kindermann. 2002. Text categorization with support vector machines. how to represent texts in input space? *Mach. Learn.*, 46(1-3):423–444, March.

Bing Liu. 2010. Sentiment analysis and subjectivity. In Nitin Indurkhya and Fred J Damerau, editors, *Handbook of Natural Language Processing, Second Edition*, pages 1–38.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 142–150.

Justin Martineau and Tim Finin. 2009. Delta tfidf: an improved feature space for sentiment analysis. In *Proceedings of the Third Annual Conference on Weblogs and Social Media*, pages 258–261.

Tony Mullen and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP '04*, pages 412–418.

Vincent Ng, Sajib Dasgupta, and S. M. Niaz Arifin. 2006. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 611–618.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 271–278.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 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*, pages 79–86.

Xuan-Hieu Phan and Cam-Tu Nguyen. 2007. GibbsLDA++: A C/C++ Implementation of Latent Dirichlet Allocation (LDA).

John C. Platt. 1999. Fast training of support vector machines using sequential minimal optimization. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in kernel methods*, pages 185–208.

P Pudil, J Novovicova, S Blaha, and J Kittler. 1992. Multistage pattern recognition with reject option. In *Proceedings 11th IAPR International Conference on Pattern Recognition (ICPR'92)*, pages 92–95.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Comput. Linguist.*, 37(2):267–307, June.

Zhaopeng Tu, Yifan He, Jennifer Foster, Josef van Genabith, Qun Liu, and Shouxun Lin. 2012. Identifying high-impact sub-structures for convolution kernels in document-level sentiment classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL '12, pages 338–343.

Peter D. Turney. 2002. 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, pages 417–424.

Sida Wang and Christopher D. Manning. 2012. Baselines and bigrams: simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL '12, pages 90–94.

Casey Whitelaw, Navendu Garg, and Shlomo Argamon. 2005. Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 625–631.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354.