Polarity Classification of Short Product Reviews via Multiple Cluster-based SVM Classifiers

Jiaying Song, Yu He, Guohong Fu
School of Computer Science and Technology, Heilongjiang University
Harbin 150080, China
jy_song@outlook.com, heyucs@yahoo.com, ghfu@hlju.edu.cn

Abstract

While substantial studies have been achieved on sentiment analysis to date, it is still challenging to explore enough contextual information or specific cues for polarity classification of short text like online product reviews. In this work we explore review clustering and opinion paraphrasing to build multiple cluster-based classifiers for polarity classification of Chinese product reviews under the framework of support vector machines. We apply our approach to two corpora of product reviews in car and mobilephone domains. Our experimental results demonstrate that opinion clustering and paraphrasing are of great value to polarity classification.

1 Introduction

With the rapid development of social networks over the past years, sentiment analysis of short social media texts has been attracting an ever-increasing amount of attention from the natural language processing community (Hu *et al.*, 2004; Fu *et al.*, 2014; Santos and Gatti, 2014). While substantial studies have been achieved on sentiment analysis to date (Pang *et al.*, 2002; Hu *et al.*, 2004; Wang and Manning, 2012; Kim *et al.*, 2013; Liu *et al.*, 2014; He *et al.*, 2015), it is still challenging to explore enough contextual information or specific cues for polarity classification of short text like online product reviews (Fu *et al.*, 2014; Santos and Gatti, 2014). On the one hand, online product reviews are short and thus contain a limited amount of contextual information for sentiment analysis. On the other hand, online product reviews actually consist of opinions about a special product attributes. It is thus very difficult to capture a variety of attribute-specific cues in different product reviews for polarity classification using a single general classifier. Furthermore, lacking large annotated corpora is still a fundamental issue for statistical sentiment analysis.

To address the above problems, in this work we explore review clustering and opinion paraphrasing to build multiple cluster-based classifiers for polarity classification of Chinese product reviews. To this end, we first explore a two-stage hierarchical clustering with multilevel similarity to cluster the training data into a set of opinion clustering and then building a polarity classifier for each review cluster via supported vector machines (SVMs). In addition, we also exploit paraphrase generation to expand product reviews in each cluster to achieve reliable training for the corresponding polarity classifier.

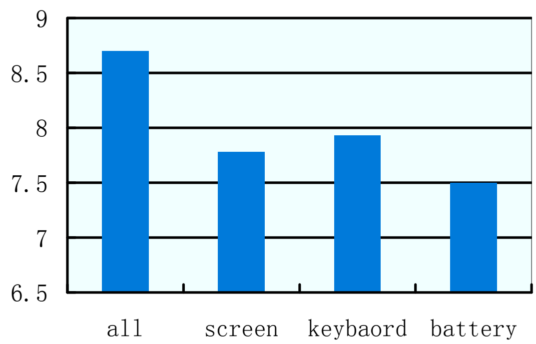

Figure 1. The entropy for product reviews in mobilephone domain before and after clustering.

Unlike most previous work with one classifier for polarity classification, our method uses multiple cluster-based classifiers to perform polarity classification, in which each classifier is tailored for a specific group of product reviews. At this point, our method actually provides a framework for attribute-based polarity classification and thus facilitate a feasible way to handle more attribute-specific cues for polarity classification. Therefore, we believe that cluster-based classification would be more precise in theory than most previous polarity

29th Pacific Asia Conference on Language, Information and Computation: Posters, pages 267 - 274
Shanghai, China, October 30 - November 1, 2015

classification methods with a separate generic classifier. This hypothesis can be further demonstrated by Figure 1, which presents the entropy of the training data in mobilephone domain before and after clustering.

The rests of the paper proceed as follows. Section 2 provides a brief review of the literature on sentiment classification. Section 3 describes in details the proposed multiple cluster-based SVM classifiers for polarity classification of Chinese product reviews. Section 4 reports our experimental results on two sets of product reviews. Finally, section 5 concludes our work and discusses some possible directions for future research.

2 Related Work

Polarity classification is usually formulated as a binary classification problem (Turney, 2002; Pang and Lee, 2008). Most previous studies employ supervised machine learning methods to perform polarity classification on different linguistic levels such words, phrases, sentences and documents, including naïve Bayes model, support vector machines (SVMs), maximum entropy models (MEMs), conditional random fields (CRFs), fuzzy sets, and so forth (Pang *et al*., 2002; Pang and Lee, 2008; Fu and Wang, 2010).

How to explore enough contextual information or specific cues is one important challenge for polarity classification of online product reviews (Fu *et al*., 2014; Santos and Gatti, 2014). Actually, online product reviews are short text with a limited amount of contextual information for sentiment analysis. Furthermore, online product reviews actually consist of opinions about a special product attributes. It is thus very difficult to capture a variety of attribute-specific cues in different product reviews for polarity classification using a single general classifier.

Lacking large manually-annotated corpora is one of the major bottlenecks that supervised machine learning methods must face. To avoid this problem, some recent studies exploit bootstrapping or unsupervised techniques (Turney, 2002; Mihalcea *et al.*, 2007; Wilson *et al*., 2009, Speriosu *et al*. 2011, Mehrotra *et al.* 2012; Volkova *et al.,* 2013). Unfortunately, unsupervised sentiment classifiers usually yield worse performance compared to the supervised counterparts.

Unlike most existing studies, in this study we attempt to build multiple cluster-based classifiers for polarity classification of Chinese product reviews by exploring review clustering and opinion paraphrasing. We believe that our method can facilitate a feasible way to handle more attribute-specific cues for polarity classification of short product reviews on the web. Furthermore, to alleviate the problem of data sparseness, we further exploit paraphrase generation to expand training corpora for each review cluster. As such, our current study is also relevant to paraphrase recognition and generation. Although a variety of methods, from dictionary-based methods to data-driven methods (Madnani and Dorr, 2010; Zhao *et al*., 2009), have been proposed for paraphrasing, here we do not want to look into paraphrasing issues. Instead, here we just employ the opinion element substitution based opinion paraphrase generation method (Fu *et al*., 2014) to achieve enough data for training the proposed cluster-based polarity classifiers.

3 Our Method

In this section, we develop cluster based techniques to explore attribute-specific features for polarity classification of short product reviews.

3.1 Overview

As shown in Figure 2, our method involves two major processes, namely the SVM modeling process based on review clusters and the polarity classification process with the cluster-based SVM classifiers.

Cluster-based SVM Modeling. As can be seen in Figure 2, we divide the training process into three main steps: (1) In the review clustering step, we first cluster reviews in the training corpus into a set of clusters $C=\{C_1, C_2, \ldots, C_n\}$ in terms of product attributes; (2) In order to achieve enough data for reliable modeling for each cluster, in the second step we further expand the training set for each cluster C_i ($1 \le i \le n$) via opinion paraphrase generation and thus obtain sets of expanded training data EC_1, EC_2, …, EC_n for opinion clusters C_1, C_2, …, C_n, respectively; (3) We finally employ SVMs to build a classification model M_i for each cluster $C_i \in C$ from the relevant expanded training data set EC_i. It should be noted that we have a special cluster Cx for all reviews that are out of any cluster in C during review clustering. For convenient, we refer to Cx as miscellaneous cluster and the relevant classification model (viz. Mx) as miscellaneous model.

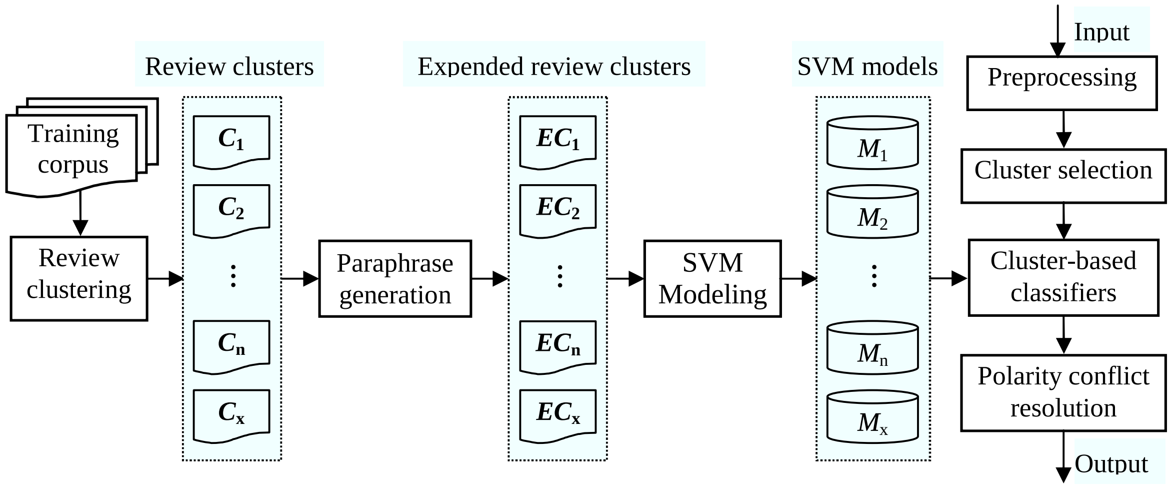

Figure 2. Overview of the cluster-based sentiment polarity classification system.

Cluster-based polarity classification. Given a input product review or opinionated sentence, we take four steps to determine its polarity category: To acquire linguistic information for subsequent polarity classification, in the preprocessing module we apply the morpheme-based lexical analyzer (Fu *et al.*, 2008) and the CRFs labeling technique to perform lexical analysis (viz. word segmentation and part-of-speech tagging) and opinion element recognition over the input, respectively. Then, we determine what clusters that the input should belong to in terms of the product attributes it contains. Thirdly, we employ the relevant cluster-based SVM classifiers to perform polarity classification. However, this step may yield different polarity classes for the input with multiple product attributes. So we finally use a polarity conflict resolution module to choose a final polarity for the input via a rule-based voting method.

3.2 Product Review Clustering

We cluster product reviews in the training data in terms of product attributes they contain. So the key to this task is how to resolve co-referred product attributes and implicit attributes in product reviews. To approach this, in this work we employ a two-stage hierarchically clustering algorithm with multilevel similarity.

2.2.1 Similarity for explicit attribute clustering

In order to handle different levels of connections between explicit attributes in real product reviews, we consider two similarities, namely the literal similarity based on Jaccard coefficient, the word embedding based semantic similarity.

Literal Similarity. Literal similarity is used to handle the literal linking between co-referred product attributes. Considering that edit distance cannot objectively reflect the real similarity for some co-referred feature expressions like 油耗 you-hao ‘fuel consumption’ and 耗油 hao-you ‘fuel consumption’, we exploit Jaccard coefficient in Equation (1) to calculate the literal similarity of two attributes a_1 and a_2.

$$S_L(a_1, a_2) = \frac{|set(a_1) \cap set(a_2)|}{|set(a_1) \cup set(a_2)|} \quad (1)$$

Where, $set(a_1)$ and $set(a_2)$ denote the set of characters within a_1 and a_2, respectively.

Semantic Similarity. In addition literal similarity, we also compute semantic similarity for some co-referred attributes without explicit literal connections, such as 像素 xiang-su ‘pixel’ and 分辨率 fen-bian-lv ‘resolution’. In order to avoid data sparseness, we use word embeddings (Mikolov, et al., 2013) to represent the semantics of product attributes. Given a pair of product attributes a_1 and a_2, let $vec(a_1)$ and $vec(a_2)$ be their respective word embeddings. In order to map the cosine value to [0, 1], then their similarity based on word embeddings, denoted by $S_S(a_1, a_2)$, can be defined by Equation (2).

$$S_C(a_1,a_2) = 0.5 + 0.5 \times \frac{vec(a_1) \bullet vec(a_2)}{|vec(a_1)| \times |vec(a_2)|} \quad (2)$$

Some complicated co-referred attributes may have both literal and semantic connections. To handle this problem, we further combine the above two similarity via linear interpolation and obtain the total similarity of a given explicit attribute pair, as shown in Equation (3).

$$S_{EA}(a_1,a_2) = \alpha \times S_L(a_1,a_2) + (1-\alpha) \times S_S(a_1,a_2) \quad (3)$$

Where, α is the interpolation coefficient.

2.2.2 Similarity for implicit attribute clustering

On the basis of the hypothesis that co-referred attributes tend to be collocated with similar evaluations, we thus exploit evaluation similarity to cluster reviews with implicit attributes. In particular, we consider explanatory evaluations as the context for implicit attributes because compared to non-explanatory evaluations, explanatory evaluations are feature-specific indicators for product attribute clustering (Kim *et al*., 2013; He *et al.*, 2015), as illustrated by Table 1. To extract explanatory evaluations for implicit attribute clustering, we use the explanatory segment labeling technique by (He *et al.*, 2015).

Definitions	Examples
A non-explanatory evaluation only presents the sentiment orientation on a given target without any explanations for the reasons of the sentiment.	这个手机的屏幕还不错。 'The screen of this handphone is good.'
An explanatory evaluation not only presents the sentiment orientation on a given target but also explains the reasons of the sentiment.	这个手机的屏幕分辨率很高。 'The screen resolution of this handphone is very high.'

Table 1. Explanatory vs. non-explanatory evaluations in Chinese product reviews.

Let e_1 and e_2 be the respective explanatory evaluations for two implicit product attributes a_1 and a_2, $Set(e_1)$ and $Set(e_2)$ be the respective synsets of the explanatory keywords within e_1 and e_2, we can then compute their evaluation similarity S_{IA} with Equation (4).

$$S_{IA}(a_1,a_2)=|Set(e_1)\cap Set(e_2)|/|Set(e_1)\cup Set(e_2)| \quad (4)$$

Here, we employ tf-itf to extract the explanatory keywords from the explanatory evaluations e_1 and e_2, and then obtain their respective synsets from the training data for word embeddings via semantic paraphrasing (Bhagat and Hovy, 2013).

2.2.3 The two-stage clustering algorithm

In this work we use a two-stage hierarchical clustering algorithm to perform review clustering, as shown in Figure 3. Where, *ClusterSimE(*C_i, C_j*)* is the average similarity between each pair of explicit attributes from C_i and C_j, respectively, and *ClusterSimI*(r_i, C_j) is the average evaluation similarity between the evaluation in r_i and the evaluation within reviews from C_j.

Input: A set of product reviews $R=\{r_1, r_2, \ldots, r_n\}$
Output: A set of review clusters $C=\{C_1, C_2, \ldots, C_k\}$.
1. Initialization: Separate R into two groups, namely the group R_E with explicit attributes and the group R_I with implicit attributes.

Stage 1: clustering reviews with explicit attributes
2. Let each review $r_i \in R_E$ be a cluster C_i ($1 \le i \le |R_E|$), and add it to C.
3. For $C_i \in C$, if $\exists C_j$ that makes *ClusterSimE*(C_i, C_j) be the maximum, and *ClusterSimE*(C_i, C_j) > θ,
4. then merge clusters C_i and C_j, and update C.
5. Repeat 2-4 until the number of clusters in C remains unchanged.

Stage 2: clustering reviews with implicit attributes
6. For each review $r_i \in R_I$
7. if $\exists C_j \in C$ that makes *ClusterSimI*(r_i, C_j) be the maximum,
8. then add r_i into C_j.
9. Output C as the review clusters.

Figure 3. The two-stage algorithm for Chinese product review clustering.

3.3 Opinion Paraphrase Generation

As we have mentioned above, the original training corpus will be separated into review clusters during review clustering. Each review cluster contains a group of reviews about a specific product attribute and are further used to training the specific classifier for the corresponding cluster. As a consequence, the dataset for some clusters may be too small for reliable training. To avoid this problem, we expand the review cluster via by paraphrasing each review via opinion element substitution (Fu *et al.*, 2014), which takes the following two main steps to generate all proper paraphrases for a given review *R*.

Items	Examples
Attribute	价格 'price'
Attribute co-references	价|价格|价钱|价位|…
Positive evaluations	Low:合适|适中|实惠|优惠|不高|公道|比较便宜|有优势|值|…
Negative evaluations	High:高|太高|真高|偏高|有点高|贵|太贵|偏贵|有点贵|不合理|有点无语|…

Table 2. An example of equivalent attribute -evaluation pairs from the training data.

(1) **Opinion element substitution.** We first generate a set of potential paraphrases for *R* by substituting opinion elements, viz. the attribution and its evaluation in *R* with their equivalent counterparts extracted from the training corpus (as shown in Table 2), and then store them with

word lattice. For convenience, here we refer this word lattice as paraphrase word lattice.

(2) **n-best paraphrase decoding.** The generated paraphrase word lattice actually contains all potential paraphrases, including both proper and improper paraphrases for the input review *R*. To exclude the improper paraphrase candidates, we further employ bigram language models to decode *n*-best paths from the paraphrase word lattice, where each path forms a probable paraphrase for *R*.

3.4 Polarity Conflict Resolution

Polarity conflict will arise if the input review sentence receives multiple but different polarity classes after polarity classification. The reason may be due to the fact that an opinionated sentence in product reviews may have more than one attribution. In this case, the system will assign more than one cluster to the input during cluster selection, and further exploit multiple different classifiers to perform polarity classification. As a consequence, an input opinionated sentence may get different polarity categories after polarity classification. In this case, polarity conflicts will arise.

In order to avoid the potential polarity conflicts, we further employ a simple rule-based voting mechanism. Given a review sentence, let K_{POS} and K_{NEG} be the respective total number of positive classes and negative classes produced by the system. Thus, we can determine its final sentiment polarity using the following three rules.

- Rule 1. if $K_{POS}>K_{NEG}$, then the final polarity is positive.
- Rule 2. if $K_{POS}<K_{NEG}$, then the final polarity is negative.
- Rule 3. if $K_{POS}=K_{NEG}$, then the final polarity is the same as the one yielded by the miscellaneous classification model M_x.

4 Experimental Results and Analysis

To assess our approach, we have conducted experiments over two corpora of product reviews from car and mobilephone domains, respectively. This section reports our experimental results.

4.1 Experiment Setup

Corpora. We use two corpora of product reviews in car and mobilephone domains that are manually annotated with multiple levels of linguistic and sentiment information, including word segmentation, part-of-speech tags, opinion elements and polarity classes. We further separate them into training and test sets, respectively. Table 3 presents the basic statistics of the experimental datasets.

Datasets	Car			Mobilephone		
	Total	Pos	Neg	Total	Pos	Neg
Training	1424	712	712	1266	633	633
Test	714	454	260	630	402	228

Table 3. Basic statistics of the experimental data.

Sentiment Lexicon. We use a sentiment lexicon in our system that contains a total of about 18K sentiment words built from the CUHK and NTU sentiment lexica [1] and HowNet[2].

Evaluation Metrics. We employ macro average precision/recall/F-score (denoted by P_{macro}, R_{macro} and F_{macro}, respectively) and micro average F-score (denoted by F_{micro}) to evaluate polarity classification performance.

LibSVM & Features. Considering the focus of our current work, we employ the LibSVM toolkit (Chang and Lin, 2011) with a linear kernel and the traditional one-hot feature representation to build our system.

Word embeddings learning. To achieve word embeddings based semantic similarity for review clustering, the Google open source tool[3], viz. word2vec, is used here to learn word embeddings from two larger corpora of car reviews (about 250K reviews) and mobilephone reviews (about 250K reviews). The dimension size is set to 100.

4.2 Effects of Different Parameters

As we have mentioned above, our clustering algorithm involves two parameters, viz α and θ for optimization. Where, α determines the importance of the two similarity, namely the literal similarity and the semantic similarity, while θ determines whether the clustering criteria is lenient or strict. In this work we employ the grid search (Bergstra and Bengio, 2012) to perform parameter optimization. Thus, we have α=0.8 andθ=0.15 for the mobilephone dataset, and α=0.6 andθ=0.3 for the car domain.

[1] http://www.datatang.com/data/43460
[2] http://www.keenage.com/
[3] http://code.google.com/p/word2vec/

θ	α	Mobilephone				Car			
		P_{macro}	R_{macro}	F_{macro}	F_{micro}	P_{macro}	R_{macro}	F_{macro}	F_{micro}
0.15	0.6	0.811	0.818	0.814	0.828	0.730	0.749	0.739	0.744
	0.7	0.845	0.846	0.846	0.859	0.782	0.776	0.779	0.800
	0.8	**0.856**	**0.874**	**0.865**	**0.871**	0.787	0.781	0.784	0.804
	0.9	0.849	0.859	0.854	0.864	0.790	0.796	0.793	0.809
0.20	0.6	0.830	0.841	0.835	0.846	0.770	0.750	0.760	0.785
	0.7	0.851	0.857	0.854	0.866	0.720	0.739	0.729	0.734
	0.8	0.840	0.851	0.846	0.856	0.797	0.787	0.792	0.812
	0.9	0.836	0.852	0.844	0.852	0.804	0.810	0.807	0.822
0.25	0.6	0.827	0.841	0.834	0.843	0.716	0.733	0.724	0.731
	0.7	0.837	0.854	0.845	0.853	0.802	0.812	0.807	0.820
	0.8	0.840	0.852	0.846	0.856	0.803	0.815	0.809	0.822
	0.9	0.831	0.850	0.840	0.847	0.787	0.794	0.790	0.806
0.30	0.6	0.830	0.841	0.835	0.846	**0.827**	**0.813**	**0.820**	**0.838**
	0.7	0.843	0.859	0.851	0.859	0.804	0.814	0.809	0.822
	0.8	0.836	0.854	0.845	0.852	0.797	0.805	0.801	0.816
	0.9	0.839	0.858	0.848	0.854	0.797	0.803	0.800	0.816

Table 4. Effects of the clustering parameters α and θ on polarity classification.

To verify the theoretical parameter optimization, we conducted an experiment to examine the effects of α and θ on polarity classification. The results are listed in Table 4.

As can be seen from Table 4, the experimental results conform to the theoretical optimization. The F-score reach the largest for mobilephone domain when θ=0.15 and α=0.8, while the corresponding real best values of θ and α are 0.3 and 0.6 for the car domain. Furthermore, it is also observed that larger value of θ and smaller value of α is beneficial to polarity classification for mobilephone domain while the trend is reversed for car domain. The reason may be that mobilephone products have less attributes than car products, suggesting a looser clustering standard for mobilephone domain. Moreover, looser standard will result in less number of clusters after review clustering, and in this case literal similarity will contribute more to review clustering. That is why mobilephone review clustering has a larger interpolation coefficient than car review clustering.

In addition to the above two clustering parameters, we have also conducted an experiment to examine the effect of the number of generated paraphrases on polarity classification. The results are shown in Figure 3.

Figure 3 reveals that the influence of paraphrase generation on polarity classification is changing with the number of generated paraphrases. When the number of generated paraphrases is less than 10, the F-score for polarity classification fluctuates with the number of generated paraphrases. However, when the number exceeds 100, the F-score will consistently rise with the number of generated paraphrases. The reason might be due to the fact that the noise introduced by paraphrase generation may have a relatively greater negative impact on polarity classification in case of the small size of paraphrase generation.

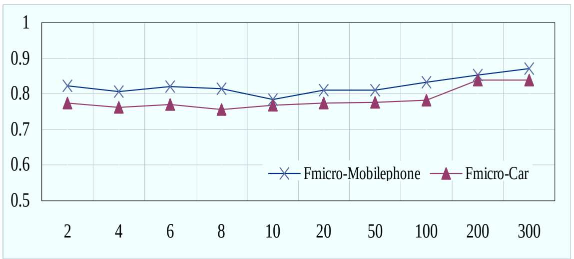

Figure 3. Effects of the number of generated paraphrases on polarity classification.

4.3 Experimental Results

In order to evaluate the effectiveness of the cluster-based method with multi-classifiers from the expanded review clusters (viz. M_SVM+Para), our experiment also involves three baselines for comparison, namely the traditional separate SVM classifier from the original training corpora in Table 1 (viz. S-SVM) or from the expanded original corpora via paraphrase generation (viz. S_SVM+Para), the cluster-based method with multiple SVM classifiers built from the review clusters without paraphrasing (viz. M-SVM). The experimental results are listed in Table 5 and Table 6.

Methods	P_{macro}	R_{macro}	F_{macro}	F_{micro}
S_SVM	0.831	0.855	0.843	0.840
M_SVM	0.815	0.828	0.822	0.832
S_SVMs + Para	0.847	0.870	0.858	0.859
M_SVMs + Para	**0.856**	**0.874**	**0.865**	**0.871**

Table 5. Results for the mobilephone domain data.

Methods	P_{macro}	R_{macro}	F_{macro}	F_{micro}
S_SVM	0.775	0.764	0.769	0.781
M_SVM	0.760	0.748	0.754	0.779
S_SVMs+Para	0.788	0.791	0.789	0.804
M_SVMs + Para	**0.827**	**0.813**	**0.820**	**0.838**

Table 6. Results for the car domain data.

From these results, we have several observations. First, the cluster-based system with paraphrasing yields the best performance for both domains, illustrating the benefits of opinion clustering and paraphrasing to polarity classification. Second, we can observe that the performance degrades when applying the clustering-based method to polarity classification without paraphrase generation. The reason may be due to the fact that the training data become too small for some clusters after review clustering. Finally, using opinion paraphrase generation results in consistent increasing of performance for the two datasets in use, showing in a sense opinion paraphrasing facilitates a effective way to expand training corpora for sentiment analysis.

5 Conclusions and Future Work

In this paper we present a new opinion cluster based framework that uses multiple cluster-based SVM classifiers to perform polarity classification of short product reviews. The main contributions of this paper are: (1) the idea of jointly using opinion clusters and paraphrases to explore richer contextual information or specific cues in short text for sentiment analysis; (2) the demonstration that opinion clustering and paraphrasing are of great value to polarity classification of short text like online product reviews.

For future work, we intend to exploit a more tailored method to achieve high-quality opinion clustering and paraphrase generation for polarity classification. Furthermore, we also plan to extend our current method to other feature representations like the emerging distributed vector representations or apply our system to other languages like English.

Acknowledgments

This study was supported by National Natural Science Foundation of China under Grant No. 61170148 and the Returned Scholar Foundation of Heilongjiang Province.

References

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of ACL'13*, pages 1608-1618.

Bo Pang, and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2): 1-135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of EMNLP'02*, pages 79-86.

Chih-Chung Chang, and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(27): 1-27.

Cícero Nogueira dos Santos, and Maíra Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING'14*, pages 69-78.

Guohong Fu, and Xin Wang. 2010. Chinese sentence-level sentiment classification based on fuzzy sets. In *Proceedings of COLING'10*, pages 312-319.

Guohong Fu, Chunyu Kit, and Jonathan J. Webster. 2008. Chinese word segmentation as morpheme-based lexical chunking. *Information Sciences*, 178(9): 2282-2296.

Guohong Fu, Yu He, Jiaying Song, and Chaoyue Wang. 2014. Improving Chinese polarity classification via opinion paraphrasing. In *Proceedings of CLP'14*, pages 35-42.

Hyun Duk Kim, Malú G. Castellanos, Meichun Hsu, ChengXiang Zhai, Umeshwar Dayal, and Riddhiman Ghosh. 2013. Ranking explanatory sentences for opinion summarization. In *Proceedings of SIGIR'13*, pages 1069-1072

James Bergstra, and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *The Journal of Machinese Research*, 13(1): 281-305.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP'14*, pages 1532-1543.

Kang Liu, Liheng Xu, and Jun Zhao. 2014. Extracting opinion targets and opinion words from online reviews with graph co-ranking. In *Proceedings of ACL'14*, pages 314-324.

Michael Heilman, Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Proceedings of NAACL'10*, pages 1011-1019.

Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. 2011. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 53-63.

Mining Hu, and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of SIGKDD'04*, pages 168-177.

Nitin Madnani, and Bonnie J. Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3): 342-387.

Peter D. Turney. 2002. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL'02*, pages 417-424.

Rada Mihalcea, Carmen Banea, Janyce Wiebe. 2007. Learning multilingual subjective language via cross-lingual projections. In *Proceedings of ACL'07*, pages 976-983.

Rahul Bhagat, and Eduard Hovy. 2013. What is a paraphrase? *Computational Linguistics*, 39(3): 463-472.

Rishabh Mehrotra, Rushabh Agrawal, and Syed Aqueel Haider. 2012. Dictionary based sparse representation for domain adaptation. In *Proceedings of CIKM'12*, pages 2395-2398.

Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. 2009. Application-driven statistical paraphrase generation. In *Proceedings of the ACL-IJCNL'09*, pages 834-842.

Sida Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of ACL'12*, pages 90-94.

Svitlana Volkova, Theresa Wilson, David Yarowsky. 2013. Exploring sentiment in social media: Bootstrapping subjectivity clues from multilingual twitter streams. In *Proceedings of ACL'13*, pages 505-510.

Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. 2010. Dependency tree-based sentiment classification using CRFs with hidden variables. In *Proceedings of HLT-NAACL'10*, pages 786-794.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):99-433

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Yu He, Da Pan, and Guohong Fu. 2015. Chinese explanatory segment recognition as sequence labeling. *Communications in Computer and Information Science*, 503: 159-168.