Expert Systems with Applications 42 (2015) 1857–1863

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Chinese comments sentiment classification based on word2vec and SVM^{perf}

Dongwen Zhang [a,b,1], Hua Xu [a,*,1], Zengcai Su [a,b,1], Yunfeng Xu [a,b,1]

[a] State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

[b] School of Information Science and Engineering, Hebei University of Science and Technology, Hebei 050018, China

ARTICLE INFO

Article history:
Available online 22 October 2014

Keywords:
Sentiment classification
Word2vec
SVM^{perf}
Semantic features

ABSTRACT

Since the booming development of e-commerce in the last decade, the researchers have begun to pay more attention to extract the valuable information from consumers comments. Sentiment classification, which focuses on classify the comments into positive class and negative class according to the polarity of sentiment, is one of the studies. Machine learning-based method for sentiment classification becomes mainstream due to its outstanding performance. Most of the existing researches are centered on the extraction of lexical features and syntactic features, while the semantic relationships between words are ignored. In this paper, in order to get the semantic features, we propose a method for sentiment classification based on word2vec and SVM^{perf}. Our research consists of two parts of work. First of all, we use word2vec to cluster the similar features for purpose of showing the capability of word2vec to capture the semantic features in selected domain and Chinese language. And then, we train and classify the comment texts using word2vec again and SVM^{perf}. In the process, the lexicon-based and part-of-speech-based feature selection methods are respectively adopted to generate the training file. We conduct the experiments on the data set of Chinese comments on clothing products. The experimental results show the superior performance of our method in sentiment classification.

1. Introduction

With the rapid development of Web technology, compared with the traditional way of shopping, e-commerce is becoming increasingly important. In order to attract more consumers and improve the consumer's shopping experience, many e-commerce websites allow consumers to comment on a variety of products. Therefore, the number of product comments on the Internet are rapidly growing. These product comments contain a wealth of information about product evaluation from consumers. The huge number of comments, however, make it difficult for consumers to gain integrated and comprehensive understanding of products by reading comments before buying the product. It is also difficult for product manufacturers to further improve product design by tracking comments and gain a competitive advantage. Sentiment classification techniques can automatically divide product comments into positive class and negative class. It helps consumers and manufacturers get the integrated and comprehensive evaluation of product from a large number of product comments on the Internet (Pang & Lee, 2008). As a result, quite a number of manufacturers, companies and e-commerce websites have a strong demand on sentiment classification. The study in this field has also been the focus widespread concerned by many researchers.

The work of sentiment classification focuses on consumer's sentiment expressed in comment texts or overall opinion towards the subject matter – for example, whether a product comment is positive or negative (Pang, Lee, & Vaithyanathan, 2002). Different from the traditional topic-based classification, sentiment classification is more challenging. Topics often can be identified by keywords alone, while the identification of sentiment needs to extract more implicit information (Liu, 2010). Therefore, sentiment classification seems to require more understanding than usual topic-based classification (Pang et al., 2002).

In recent years, machine learning-based methods for sentiment classification have been widely adopted due to their excellent performance (Li, Wang, Zhou, & Lee, 2011, 2012; Xia, Wang, Hu, Li, & Zong, 2013; Ye, Zhang, & Law, 2009; Yin, Wang, & Zheng, 2012). How to extract complex features rather than simple features and figuring out which types of features are more valuable are two

* Corresponding author at: State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China.

E-mail addresses: xuhua@tsinghua.edu.cn (H. Xu), sorres@163.com (Z. Su).

[1] Indicates equal contributions from these authors.

http://dx.doi.org/10.1016/j.eswa.2014.09.011
0957-4174/

key issues in machine learning-based methods (Zhai, Xu, Bada, & Peifa, 2011b). Up to this day, a variety of feature extraction methods have been proposed, including single words (Tan & Zhang, 2008), single-character n-grams (Raaijmakers & Kraaij, 2008), multi-word n-grams (Li & Sun, 2007), lexical–syntactic patterns and many other novel models (Xia & Zong, 2010; Zhai, Xu, Jun, & Peifa, 2009; Zhang, Liu, Lim, & O'Brien-Strain, 2010). Nevertheless, semantic features have been rarely considered in sentiment classification; as a matter of fact, semantic features can reveal the deep and implicit semantic relationships between words, which is beneficial to sentiment classification. Hence, the classification based on semantic features should be able to obtain better results.

Word2vec[2] is a very hot open source tool based on deep learning in the past few months. It can learn the vector representations of words in the high-dimensional vector space and calculate the cosine distances between words. That is to say, the tool can find the semantic relationships between words in the document. So, we attempt to apply word2vec to sentiment classification. The SVM^{perf} package[3] is not commonly used in existing work of classification. But it trains faster and predicts more accurate than any other SVM packages in large-scale training set, due to its optimized kernel algorithm (Joachims & Yu, 2009). In this paper, we first use word2vec to cluster the synonyms referring to the same product feature in order to verify its capability to extract the semantic features. Then we use machine learning-based method to classify the comment texts. An approach of combining word2vec with SVM^{perf} is proposed in this paper. The experimental results show that our method can obtain satisfactory classification performance.

The rest of this paper is organized as follows. Section 2 reviews related work on similar features clustering and sentiment classification. Methodology and key techniques are described in detail in Section 3. The experimental results are illustrated and discussed in Section 4. Finally, this paper is summarized in Section 5.

2. Related work

In this section, we give an brief introduction to the previous work on the tasks of similar features clustering and machine learning-based methods for sentiment classification. The studies of the two tools, word2vec and SVM^{perf}, are also presented.

2.1. Similar features clustering

The purpose of similar features clustering is to group the synonyms which express the same product feature under the same feature group. Zhai, Liu, Xu, and Peifa (2010) proposed to use the semi-supervised EM algorithm to solve the problem, which is improved by considering two soft constraints based on sharing of words and the lexical similarity. And in the next year, Zhai, Liu, Xu, and Peifa (2011a) improved the accuracy further by allowing the labeled examples to switch classes because the constraints can make mistakes. Its accuracy, however, still can't reach the extent of the practical application. Although several methods have been proposed to extract product features from comment texts, limited work has been done on clustering of similar features (Zhai et al., 2011a), especially in the field of Chinese sentiment analysis.

2.2. Sentiment classification based on supervised machine learning

The supervised machine learning-based methods aim to train a sentiment classifier using labeled corpus. Pang et al. (2002) applied the method based on machine learning to sentiment classification task for the first time. They try to use the n-grams model and compare NB, ME and SVM classification models, find that choosing unigrams as feature set and SVM as classification model can obtain the best classification results. Various feature selection methods and classification models have been proposed for the past few years. Yessenalina, Yue, and Cardie (2010) proposed a joint two-level approach for document-level sentiment classification that simultaneously extracts subjective sentences and predicts document-level sentiment based on the extracted sentences. Zhai et al. (2011b) extracted sentiment-words, substrings, substring-groups and key-substring-groups as features. Wang, Li, Song, Wei, and Li (2011) proposed an effective feature selection method based on fisher's discriminant ratio for subjectivity text sentiment classification. Yao, Wang, and Yin (2011) used statistical-based machine learning methods to select features and reduce dimensionality for sentiment classification of Chinese online reviews based on hotels. Xia, Zong, and Li (2011) took advantage of ensemble frameworks for integrating different feature sets and classification algorithms to boost the overall performance. Abbasi, France, Zhang, and Hsinchun (2011) proposed a rule-based multivariate text feature selection method called Feature Relation Network (FRN) that considers semantic information and also leverages the syntactic relationships between n-grams features. Wang, Yin, Yao, and Liu (2013), Wang, Yin, Zheng, and Liu (2014) adopted document frequency (DF), information gain (IG), chi-squared statistic (CHI) and mutual information (MI) to select features and then apply Boolean weighting method to set feature weights and construct a vector space model. Moraes, Valiati, and Neto (2013) adopted a standard evaluation context with popular supervised methods for feature selection and weighting in a traditional bag-of-words model.

2.3. Word2vec and SVM^{perf}

Word2vec is a tool based on deep learning and released by Google in 2013. This tool adopts two main model architectures, continuous bag-of-words (CBOW) model and continuous skip-gram model, to learn the vector representations of words. The CBOW architecture predicts the current word based on the context, and the skip-gram predicts surrounding words given the current word (Mikolov, Chen, Corrado, & Dean, 2013a). The algorithms are described in detail in Mikolov et al. (2013a, 2013b, 2013c).

SVM^{perf} is an implementation of the Support Vector Machine (SVM) formulation for optimizing multivariate performance measures described in Joachims (2005). Furthermore, SVM^{perf} implements the alternative structural formulation of the SVM optimization problem for conventional binary classification with error rate and ordinal regression described in Joachims (2006). And most of all, SVM^{perf} trains sparse kernel SVMs via the cutting-plane subspace pursuit (CPSP) algorithm described in Joachims and Yu (2009), which enhances the speed and accuracy of prediction.

3. Methodology

3.1. Overview

As far as we know, word2vec shows superior performance in texts classification and clustering in English (Mikolov et al., 2013a, 2013b, 2013c). Yet for now, none of existing researches can prove that word2vec will also be of great value to Chinese texts clustering. Therefore, in order to test and verify that, we first utilize word2vec to group the synonyms under the same feature groups. After that, we adopt the method based on combining word2vec with SVM^{perf} to classify the comment texts into positive class and negative class. Fig. 1 illustrates the general framework of our work.

[2] <https://code.google.com/p/word2vec/>.
[3] <http://www.cs.cornell.edu/People/tj/svm_light/svm_perf.html>.

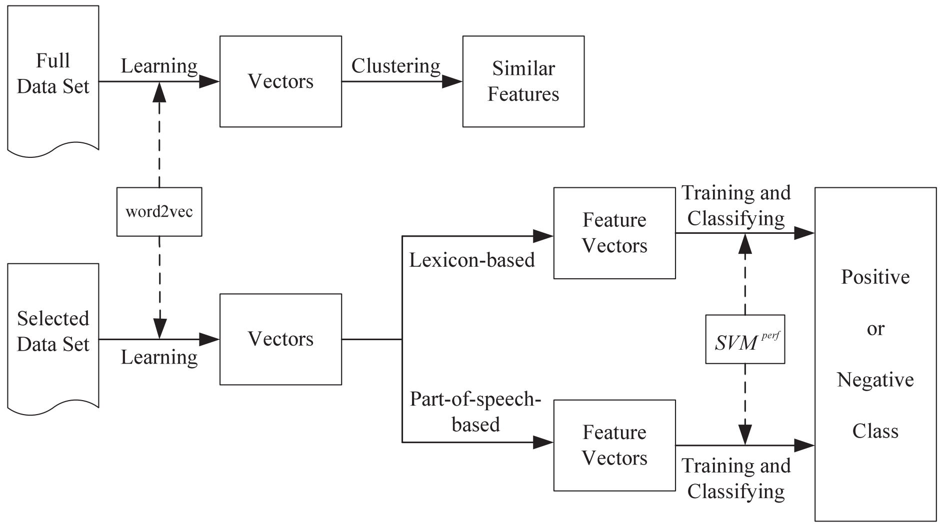

Fig. 1. The general framework of our work.

We expand upon these two steps in the following subsections.

3.2. Similar features clustering

For a product feature, consumers may express it with many different words. To produce a useful comment summary, these words which are domain synonyms need to be grouped under the same feature group (Zhai et al., 2010, 2011a). Here we elaborate on how we achieve the clustering of similar features using word2vec.

Step 1. Preprocessing We adopt ICTCLAS[4] system, which is developed by Institute of Computing Technology, Chinese Academy of Science, to segment the collected Chinese comment texts into words and tag them with proper part-of-speech (POS) tags. After deleting stop words and punctuation characters and other necessary processing, we obtain the training file.

Step 2. Training We use word2vec to train the training file with the purpose of getting the model file. Table 1 gives the explanations and default values of the key parameters in the training command.
The word2vec tool takes a training file as input and produces a model file as output. It first constructs a vocabulary from the training file and then learns high dimensional vector representations of words according to the parameters.

Step 3. Clustering If the model file is stored with the common format, each word in the document and its corresponding vector is visible. Word2vec provides a command called *distance* to make a comparison in semantic similarity between words to achieve the goal of synonyms clustering. It reads every words and their corresponding vectors in the model file and calculates the semantical distances between input word and other words using cosine similarity. The higher the cosine value, the closer the two words semantically. After sorting the values in descending order, we get the list of closest words for input word and their distances.

Table 1
The key parameters of training command.

Parameters	Explanations	Default values
-Train	Name of input file	Train.txt
-Output	Name of output file	Vectors.bin
-cbow	Choice of training model 0: Skip-gram model 1: CBOW model	0
-Size	Dimension of vectors	200
-Window	Size of training window	5
-Negative	Choice of training method 0: Hierarchical Softmax method 1: Negative Sampling method	0
-hs	Choice of training method 0: Negative Sampling method 1: Hierarchical Softmax method	1
-Sample	Threshold of sampling	$1e-3$
-Threads	Number of running threads	12
-Binary	Mode of storage 0: Common format 1: Binary format	1

3.3. Sentiment classification

Different from the traditional approaches for sentiment classification, we achieve the classification with the help of two tools, word2vec and SVM^{perf}. Firstly, word2vec prunes the words whose occurrence frequency in the corpus is less than five before training. The remaining words are considered to be the frequent words and we treat them as candidate features. And then word2vec trains the corpus and generates a model file which contains a list of frequent words and their corresponding vector representations with one dimension. Two feature selection methods, one is based on lexicon and the other is based on part-of-speech, are utilized to select valuable features from candidate set.

3.3.1. Feature selection method based on lexicon

This method needs a dictionary of sentiment words and phrases with their associated orientations and weight, and incorporates

[4] <http://www.ictclas.org/>.

intensifiers and negations (Liu, 2012). We choose HowNet sentiment word set[5] and IARDict as two original dictionaries. HowNet sentiment word set contains a list of opinion words extracted from HowNet on-line knowledge base, and IARDict is collected by our research team in previous work.

We first extract top ten opinion words with the highest weight from original dictionaries as input words and run *distance* command to get more opinion words. In this way, the original dictionaries are expanded.

Algorithm 1. Feature selection based on lexicon

1: *word_set* ← frequent words
2: *dic_set* ← opinion words in lexicon
3: **for** each *w* in *dic_set* **do**
4:   **if** *w* is in *word_set* **then**
5:     add *w* to *feature_set*
6:   **else**
7:     *continue*
8:   **end if**
9: **end for**

And then, we extract the words which appear both in candidate feature set and the expanded dictionaries as the final training features. Algorithm 1 gives the general selection process. Here, *feature_set* represents the final training features set.

3.3.2. Feature selection method based on part-of-speech

This method selects valuable features according to the part-of-speech of words. The different choices of part-of-speech can lead to different results (Liu & Zhang, 2012). For instance, if the adjectives alone are chosen as features, the classification result will not be better than the result of choosing adverbs, verbs and adjectives at the same time. This is because many words with different part-of-speech can be sentiment indicators.

In this method, after part-of-speech selection, we keep adjectives, adverbs, verbs and nouns, which are four of the most common words in the documents. We treat the different combinations of them as training features.

3.3.3. Training and classifying

In this step, the selected feature vectors are trained by a classifier to predict the sentiment polarity of the test documents.

Lots of previous researches prove that SVM shows substantial performance gains and is more robust in the work of classification, compared with other state-of-art models (Pang et al., 2002; Tang, Tan, & Cheng, 2009). Due to this, SVM is adopted in this paper.

SVM^{perf} is an optimized version of SVM^{light} developed by Joachims, who is also the author of SVM^{light}. Its overall architecture follows SVM^{light}, but the kernel algorithm is advanced. That make it obtain enhanced speed and accuracy of classification. Consequently, we utilize the SVM^{perf} package for training and testing.

4. Experiment and discussion

4.1. Data sets

Firstly, more than 100,000 Chinese comments on clothing products are crawled down from Amazon.[6] After the removal of duplicates and meaningless data, 92,220 comments remained.

In our research, the two experiments are conducted on data sets of different sizes. Similar features clustering based on word2vec does not need to identify the polarity of texts. Moreover, the larger the corpus, the better the performance of word2vec. So all the collected comments are used as data set of similar features clustering.

Table 2
A brief summary of data sets.

Our work	Positive	Negative	Total number
Similar features clustering	N/A	N/A	92,220
Sentiment classification	5000	5000	10,000

Our major work is sentiment classification based on word2vec and SVM^{perf}, which is a supervised machine learning method. All the collected comments are divided into five levels according to user star ratings. At first, we treat comments with 5-star as positive class and comments with 1-star as negative class. But the ratio of positive class to negative class is nine to one. The serious imbalanced data set has an adverse effect on classification results. So in order to keep the data set balanced, we adjust the strategy. All of the comments with 1-star are still considered the negative class, while the positive class is an equal number of randomly selected comments with 5-star. A brief summary of data sets is presented in Table 2. To carry out the experiments, data set is divided into two equal parts, one part with 2500 positive comments and 2500 negative ones is used for training and the remaining half is used for testing.

4.2. Evaluation criteria

Like previous classification tasks, we evaluate the experimental results with precision, recall and F1. These three classic values are utilized to measure the performance of positive and negative class respectively. Accuracy is another criterion, which is used to evaluate the overall performance of sentiment classification.

4.3. Experimental results

We conduct the experiments to evaluate our work of similar features clustering and sentiment classification. In this section, the experimental results are shown and discussed respectively.

4.3.1. Result of similar features clustering

In this subsection, the result of similar features clustering is presented and analyzed.

For the data set of Chinese clothing comments, we choose Jia4Ge2 (price), Mian4Liao4 (fabric), Chi3Ma3 (size) and Kuan3Shi4 (style) as representative features. These four features appear most frequently in Chinese clothing comments. After obtaining the lists of synonyms, we keep only the two-character nouns and then select the top five words as the final clustering results. We train word2vec at five different dimensions of vector.

Table 3 shows the results of similar features clustering based on word2vec. For each representative feature, its associated similar features have the same or similar Chinese meaning. Moreover, at different dimensions of vector, the lists of similar features change nothing but the sequence between several words, which has little influence on the accuracy of clustering. The nearly perfect results prove the strong capability of word2vec to capture the deep semantic relationships between words in Chinese texts clustering.

4.3.2. Performance of sentiment classification

The proposed approach for sentiment classification is based on word2vec and SVM^{perf}, which adopts two feature selection methods, lexicon-based method and part-of-speech-based method. In this subsection, their classification performance is shown and discussed respectively.

[5] <http://www.keenage.com/html/c_index.html>.
[6] <http://www.amazon.cn/>.

Table 3
The results of similar features clustering based on word2vec.

Representative features	Dimensions of vector	Similar features				
Jia4Ge2	200	Jia4Wei4	Jia4Qian2	Jia4Ma3	Jia4Zhi2	Tian1Jia4
	500	Jia4Wei4	Jia4Qian2	Jia4Ma3	Jia4Zhi2	Tian1Jia4
	1000	Jia4Wei4	Jia4Qian2	Jia4Ma3	Jia4Zhi2	Tian1Jia4
	5000	Jia4Wei4	Jia4Qian2	Jia4Ma3	Tian1Jia4	Jia4Zhi2
	10000	Jia4Wei4	Jia4Qian2	Jia4Ma3	Jia4Zhi2	Tian1Jia4
Mian4Liao4	200	Liao4Zi	Bu4Liao4	Zhi4Di4	Cai2Zhi4	Cai2Liao4
	500	Liao4Zi	Bu4Liao4	Cai2Liao4	Zhi4Di4	Cai2Zhi4
	1000	Liao4Zi	Bu4Liao4	Zhi4Di4	Cai2Zhi4	Yi1Liao4
	5000	Liao4Zi	Bu4Liao4	Zhi4Di4	Cai2Zhi4	Yi1Liao4
	10000	Liao4Zi	Bu4Liao4	Zhi4Di4	Cai2Zhi4	Cai2Liao4
Chi3Ma3	200	Chi3Cun4	Hao4Ma3	Xing2Hao4	Ma3Hao4	Ma3Zi
	500	Chi3Cun4	Hao4Ma3	Ma3Zi	Ma3Hao4	Xing2Hao4
	1000	Chi3Cun4	Hao4Ma3	Xing2Hao4	Ma3Hao4	Ma3Zi
	5000	Chi3Cun4	Hao4Ma3	Xing2Hao4	Ma3Hao4	Ma3Zi
	10000	Chi3Cun4	Hao4Ma3	Ma3Zi	Ma3Hao4	Xing2Hao4
Kuan3Shi4	200	Yang4Shi4	Kuan3Xing2	Wai4Guan1	Yang4Zi	Wai4Xing2
	500	Yang4Shi4	Kuan3Xing2	Yang4Zi	Wai4Xing2	Wai4Guan1
	1000	Yang4Shi4	Kuan3Xing2	Wai4Guan1	Yang4Zi	Wai4Xing2
	5000	Yang4Shi4	Kuan3Xing2	Yang4Zi	Wai4Guan1	Wai4Xing2
	10000	Yang4Shi4	Kuan3Xing2	Yang4Zi	Wai4Guan1	Shi4Yang4

Table 4
The performance of sentiment classification based on lexicon.

Features	Positive			Negative			Accuracy (%)
	Precision (%)	Recall (%)	F1 (%)	Precision (%)	Recall (%)	F1 (%)	
IARDict	91.36	86.70	88.97	87.35	91.80	89.52	89.25
HowNet	86.27	87.65	86.95	87.45	86.05	86.74	86.85
the Combination	91.14	88.50	89.80	88.82	91.40	90.09	89.95

Table 5
The performance of sentiment classification based on part-of-speech.

Features	Positive			Negative			Accuracy (%)
	Precision (%)	Recall (%)	F1 (%)	Precision (%)	Recall (%)	F1 (%)	
{a,d}	87.21	82.85	84.97	83.67	87.85	85.71	85.35
{a,d,v}	91.38	89.00	90.17	89.28	91.60	90.42	90.30
{a,d,n}	91.91	85.15	88.40	86.17	92.50	89.22	88.83
{a,d,v,n}	91.89	84.95	88.28	86.01	92.05	89.14	88.72
All	92.66	84.60	88.45	85.83	93.30	89.41	88.95

The performance of lexicon-based feature selection method is listed in Table 4. For the three kinds of features, the combination of IARDict and HowNet performs a little better than IARDict alone in total accuracy, while HowNet shows the worst performance, apparently. The reason why HowNet is inferior to the other two is probably that HowNet features do not contain negation words. Such words may change the sentiment orientations, which can cause the wrong classification results. The features of combining IARDict with HowNet consist of positive words, negative words, negation words and intensifiers. Not only the sentiment itself but also the sentiment shift are taken into account.

The performance of part-of-speech-based feature selection method is listed in Table 5. As illustrated by these data, it is obvious that selecting adjectives, adverbs and verbs as features outperforms the other selection strategies, both in F1 and accuracy. The performance of selecting adjectives and adverbs is disappointing, which produces the worst results in terms of precision, recall, F1 and accuracy. The strategy of selecting all frequent words as features has highest precision of positive class and recall of negative class, but the lower recall of positive class and precision of negative class pull down the F1 value and overall accuracy. The rest of two strategies obtain nearly the same results with very little difference.

Table 6
Accuracy for different combinations with two feature selection methods.

Method	Lexicon-based (%)	Part-of-speech-based (%)
TF-IDF + LibSVM	82.81	82.05
Word2vec + LibSVM	84.28	87.10
TF-IDF + SVM^{perf}	87.63	83.48
Word2vec + SVM^{perf}	89.95	90.30

We adopt the TF-IDF weighting scheme and LibSVM classification model to compare the performance of our approach. Table 6 shows the result of the comparison. As we expect, our approach is significantly better than other combinations.

To achieve the optimal accuracy of sentiment classification, we tune the performance of experiments by means of adjusting the regularization parameter C of SVM^{perf}. As the best strategy in respective feature selection method, the combination of IARDict and HowNet, and {adjectives, adverbs, verbs} are chosen to

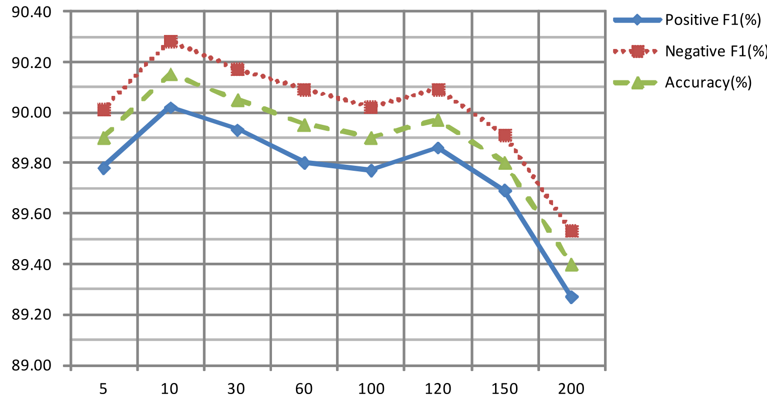

Fig. 2. The performance of lexicon-based with various *C*.

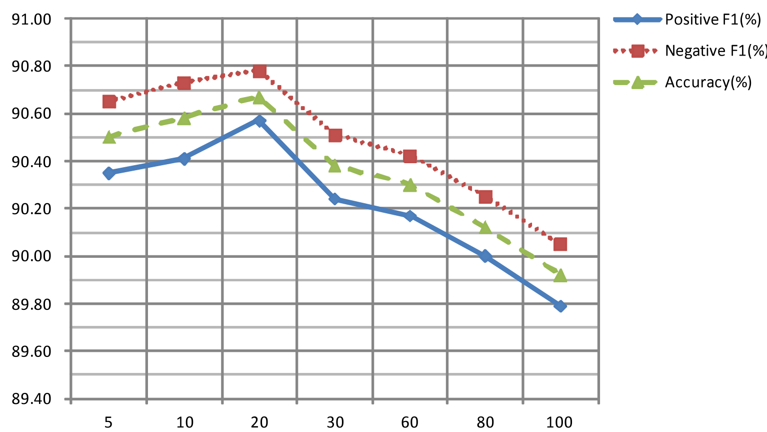

Fig. 3. The performance of part-of-speech-based with various *C*.

conduct the optimization experiments. Figs. 2 and 3 show the results of experiments with various *C*.

As the results show, the performance is enhanced slightly, both in F1 and accuracy. The two strategies obtain the best accuracy when $C = 10$ and $C = 20$, respectively.

4.4. Discussion

From the experiments above, it is evident that both lexicon-based method and part-of-speech-based method can get excellent results. The total classification accuracy reaches more than 90%. After carrying out a series of experiments, in this subsection, we discuss the reasons for the effectiveness of our method.

Firstly, the vector representations of words learned by word2-vec can extract the deep semantic relationships between words, which contribute more to sentiment classification. Secondly, as a new training algorithm for linear classification SVM based on SVM^{light}, SVM^{perf} is much more precise and faster than SVM^{light} for large-scale data sets. As a result, based on word2vec and SVM^{perf}, the proposed method for sentiment classification achieves encouraging performance.

5. Conclusion and future work

Different from most of the conventional methods for sentiment classification, our research focuses on the semantic features between words rather than the simple lexical or syntactic features. In this paper, we make use of two tools, word2vec and SVM^{perf}, to classify the Chinese comment texts. To conduct the experiments, tens of thousands of Chinese comments on clothing products are crawled as data set. We first cluster the similar features by word2-vec, and the results prove that word2vec is also suitable for Chinese texts clustering. The best experimental results of proposed word2vec and SVM^{perf} based sentiment classification reach over 90% accuracy, whether the lexicon-based feature selection method or the part-of-speech-based method is used. Our method proves to be effective for sentiment classification.

Even if the results are pretty good for sentiment classification, our research is far from perfect. We have a lot of work ahead of us. In order to conform to the format of SVM^{perf}'s training file, the dimension of vectors learned by word2vec is set to 1. How to apply the high-dimensional feature vectors to SVM^{perf} model is a challenging problem waiting to be explored. In addition, the two feature selection methods we used are not enough to find out all of the sentiment features in each sentence. So we need to extract the structured information and composition unit existing in every sentences in the future work.

Acknowledgments

This work is sponsored by National Natural Science Foundation of China (Grant No.: 61175110) and National Basic Research Program of China (973 Program, Grant No.: 2012CB316305).

References

Abbasi, A., France, S., Zhang, Z., & Hsinchun, C. (2011). Selecting attributes for sentiment classification using feature relation networks. *IEEE Transactions on Knowledge and Data Engineering, 23*, 447–462.

Joachims, T. (2005). A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on machine learning* (pp. 377–384). ACM.

Joachims, T. (2006). Training linear svms in linear time. In *Proceedings of the 12th ACM conference on knowledge discovery and data mining* (pp. 217–226). ACM.

Joachims, T., & Yu, C.- N. J. (2009). Sparse kernel svms via cutting-plane training. In *Proceedings of the European conference on machine learning* (pp. 179–193).

Li, J., & Sun, M. (2007). Experimental study on sentiment classification of chinese review using machine learning techniques. In *Proceedings of the international conference on natural language processing and knowledge engineering* (pp. 393–400). IEEE.

Li, S., Ju, S., Zhou, G., & Li, X. (2012). Active learning for imbalanced sentiment classification. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 139–148). Association for Computational Linguistics.

Li, S., Wang, Z., Zhou, G., & Lee, S. Y. M. (2011). Semi-supervised learning for imbalanced sentiment classification. In *Proceedings of the 22nd international joint conference on artificial intelligence* (pp. 1826–1831).

Liu, B. (2010). Sentiment analysis and subjectivity. *Handbook of natural language processing* (2nd ed.) (pp. 1–38).

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies, 5*, 1–167.

Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. *Mining text data* (pp. 415–463).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. In *Proceedings of workshop at ICLR*.

Mikolov, T., Sutskever, I., Chen, K., Greg, C., & Jeffrey, D. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of neural information processing systems*.

Mikolov, T., Yih, W.-t., & Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies*. Association for Computational Linguistics.

Moraes, R., Valiati, J. F., & Neto, W. P. G. (2013). Document-level sentiment classification: An empirical comparison between svm and ann. *Expert Systems with Applications, 40*, 621–633.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval, 2*, 1–135.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 79–86). Association for Computational Linguistics.

Raaijmakers, S., & Kraaij, W. (2008). A shallow approach to subjectivity classification. In *Proceedings of the second international conference on weblogs and social media* (pp. 216–217).

Tan, S., & Zhang, J. (2008). An empirical study of sentiment analysis for chinese documents. *Expert Systems with Applications, 34*, 2622–2629.

Tang, H., Tan, S., & Cheng, X. (2009). A survey on sentiment detection of reviews. *Expert Systems with Applications, 36*, 10760–10773.

Wang, H., Yin, P., Yao, J., & Liu, J. N. (2013). Text feature selection for sentiment classification of chinese online reviews. *Journal of Experimental & Theoretical Artificial Intelligence, 25*, 425–439.

Wang, H., Yin, P., Zheng, L., & Liu, J. N. (2014). Sentiment classification of online reviews: Using sentence-based language model. *Journal of Experimental & Theoretical Artificial Intelligence, 26*, 13–31.

Wang, S., Li, D., Song, X., Wei, Y., & Li, H. (2011). A feature selection method based on improved Fisher's discriminant ratio for text sentiment classification. *Expert Systems with Applications, 38*, 8696–8702.

Xia, R., Wang, T., Hu, X., Li, S., & Zong, C. (2013). Dual training and dual prediction for polarity classification. In *Proceedings of the 51st annual meeting of the association for computational linguistics* (pp. 521–525).

Xia, R., & Zong, C. (2010). Exploring the use of word relation features for sentiment classification. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 1336–1344). Association for Computational Linguistics.

Xia, R., Zong, C., & Li, S. (2011). Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences, 181*, 1138–1152.

Yao, J., Wang, H., & Yin, P. (2011). Sentiment feature identification from chinese online reviews. In *Proceedings of 2011 international conference on computer science and education: Communications in computer and information science* (pp. 315–322). Springer.

Ye, Q., Zhang, Z., & Law, R. (2009). Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Systems with Applications, 36*, 6527–6535.

Yessenalina, A., Yue, Y., & Cardie, C. (2010). Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 1046–1056). Association for Computational Linguistics.

Yin, P., Wang, H., & Zheng, L. (2012). Sentiment classification of chinese online reviews: Analysing and improving supervised machine learning. *International Journal of Web Engineering and Technology, 7*, 381–398.

Zhai, Z., Liu, B., Xu, H., & Peifa, J. (2010). Grouping product features using semi-supervised learning with soft-constraints. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 1272–1280). Association for Computational Linguistics.

Zhai, Z., Liu, B., Xu, H., & Peifa, J. (2011a). Clustering product features for opinion mining. In *Proceedings of the fourth ACM international conference on web search and data mining* (pp. 347–354). ACM.

Zhai, Z., Xu, H., Bada, K., & Peifa, J. (2011b). Exploiting effective features for chinese sentiment classification. *Expert Systems with Applications, 38*, 9139–9146.

Zhai, Z., Xu, H., Jun, L., & Peifa, J. (2009). Sentiment classification for chinese reviews based on key substring features. In *Proceedings of 2009 international conference on natural language processing and knowledge engineering* (pp. 1–8). IEEE.

Zhang, L., Liu, B., Lim, S. H., & O'Brien-Strain, E. (2010). Extracting and ranking product features in opinion documents. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 1462–1470). Association for Computational Linguistics.