A New Method for Sentiment Classification on Weibo

Xianyun Tian, Guang Yu, Yongtian Yu, Pengyu Li, Jiayin Pei
Harbin Institute of Technology
China
{uncertainorcertain@gmail.com} {yug@hit.edu.cn} {P_B_M3511@163.com}
{lipengyuer@126.com} {frounaxp@hotmail.com}

ABSTRACT: *Sentiment classification of posts on Weibo is a key to analyse people's opinions and attitudes toward products, services, and social events. It is also at the core of many other natural language processing tasks. In this paper, we apply a new feature representation technique to building a sentiment classifier and classifying the posts. Firstly, we crawled a set of posts from Weibo and labelled them. Then, we cleaned the data to remove the noisy information and transformed the varying-length posts into fixed-sized input vectors based on five different feature engineering techniques. Finally, we evaluated the performances of the five different feature representation techniques on a same data set with the use of support vector machines, naive Bayes and classification and regression tree. Experimental results demonstrate that our new method is efficient and outperforms the other ones.*

Keywords: Sina Weibo, Sentiment Classification, Microblogging classification, Distributed representation of words

Received: 10 September 2015, Revised 8 October 2015, Accepted 14 October 2015

1. Introduction

The popularity of social media has greatly changed the way people communicate with each other. We can directly browse anyone's homepage to know what they have done, what they like and dislike, how they feel about this world and the changes of their emotions. The most famous social media sites are Twitter and Facebook, but they cannot be accessed in China. Instead, the most popular social media in China is Weibo, which was launched in 2009, and has attracted about 500 million users in five years. More than 100 million posts are posted every day. This large-scale user generated content provides us an opportunity to understand the emotions and behaviors of the public[1]. People use Weibo to express and exchange their opinions toward celebrities, products, companies, politicians and social events[2,3]. These data can help companies improve the qualities of their service or create new products to meet the needs of their customers. These data can also help the government understand the public deeply and provide better service for the society. People often post posts to express their anger, happiness, and opinions toward the social events that happen every day. Therefore, many researchers have done a large amount of work based on these data. For example, some researchers attempted to predict depression via the social media[4]. The better understanding of people's opinions and attitudes requires a sentiment classifier with higher accuracy. So we focus on improving the accuracy of the sentiment classification in this paper.

The feature engineering is a key to build a good sentiment classifier. This is typically where most of efforts go in a machine learning project. Previous research in sentiment classification on Weibo mainly uses lexicon-based methods[5] or just analyse the posts that contain emoticons[1,6]. However, the lexicon-based methods treat one word as an index in the vocabulary which ignore much information of the word. Since not all posts contain emoticons, the second method cannot analyse those posts which do not contain emoticons. By contrast, the newly emerged distributed representations of words can efficiently capture much of the semantic and syntactic information of the words, and it has been successfully applied into many natural language processing tasks. Previous research has shown that this new form of word representation can improve the performance of subjectivity classification and sentiment classification[7–9]. However, to the best knowledge of us, there is no research verifying the performance of it on the Chinese social media. Thusly, we decide to verify the performance of this new form of word representation by building a sentiment classifier whose feature engineering is based on the distributed representations of words.

Four steps were taken to perform a sentiment classification task. Firstly, we crawled a set of posts from Weibo and labelled them. The posts were categorized into two groups: positive and negative. Secondly, we preprocessed the data sets to remove the noisy information, such as the location information, URLs, nicknames and so on. The noisy information were removed because of making no contribution to the task. Thirdly, we used five different methods to transform the varying-length posts into fixed-sized input vectors which can be processed by the machine learning algorithms. Lastly, based on the five different kinds of input vectors, we evaluated the performances of the sentiment classifiers with the use of support vector machine, naïve Bayes and classification and regression tree. The framework of our work is described in Figure .1.

The rest of the paper is organized as follows. Closely related literature, such as the introduction of sentiment analysis and different methods of feature representation, is reviewed in section 2. In section 3, we describe the data sets and methods employed in this paper. In section 4, the experimental results and the analyses are given. Finally, we conclude the paper briefly and discuss future work.

2. Related Works

2.1 Sentiment Classification

In general, sentiment analysis is consisted of three stages: topic-based information retrieval, subjectivity classification and sentiment classification[10]. Our work, like most of previous research, focus on the last stage—sentiment classification.

Depending on the length of the text, the sentiment classification can be divided into document-level and sentence-level. Compared with the document-level task, the sentence-level one is more difficult because of the sparsity and ambiguity of the short text. Depending on the number of output classes, the sentiment classifier can be categorized into binary sentiment classifier[11] and multi-class classifier[12]. For example, some researchers focused on classifying the emotions into multiple types, including happiness, love, sadness, horror and anger[13]. Generally speaking, the more classes the classifier needs to distinguish, the more difficult for the classifier to correctly classify the texts.

Pang et al. are the first to introduce sentiment analysis and explored it[11]. Their results demonstrated that a combination of bag-of-words model with machine learning algorithms, such as the naïve Bayes, support vector machines and MaxEnt, cannot achieve an acceptable accuracy. The accuracy cannot be improved even when more features, like the part of speech, are imported. This implies that we should try some new features or feature representations which are quite different from previous ones. Other researchers followed and extended their work[14,15]. But the features used by these work did not make a big difference. Being China's most popular social media, Weibo has attracted many researchers to study sentiment classification on it[1,6,16]. These works also mainly rely on traditional methods to classify the posts.

2.2 Sentiment Classification Features

In most cases, the performances of machine learning projects are largely determined by the feature engineering. Hence, in this paper, we attempt to apply new feature representations to improve the performance of the sentiment classifier. The features used by previous research can be divided into three groups.

(1) Message-based Features: Message-based features mainly refer to the features extracted from the message, including the length of message, the number of digit characters, word frequency and punctuation[17]. There are also some other features

which are not directly extracted from the message, but by using some models, such as the part-of-speech tags and N-gram[11,18].

(2) Weibo-based Features: Weibo-based features mainly refer to the emoticons. Weibo supplies the users with a series of emoticons through which users can conveniently express their feelings. The emoticons play an important role in classifying messages, because they are very much like real person's expressions and can efficiently reflect individual's emotions. Go et al. created an English data set by extracting emotions from posts[19]. Pak et al. analysed the emoticons and divided them into two groups: the positive emoticons and negative ones[20]. However, as far as we know, there is no formal public available emoticon data set for Weibo, so we crawled the posts from Weibo to extracted the emoticons from them and built one.

(3) Sentiment Lexicon Features: Like the emoticons, the sentiment lexicons are also always treated as important indicators of the sentiment of a message. The sentiment lexicons are usually labelled by engineers[21] or collected through heuristic rules[22,23]. They are seen as import indicators because most of them have strong sentiment polarity. Therefore, many models are built based on them[5,24,25]. However, according to our observation, the words in Chinese sentiment lexical tools are very formal, but people often use informal words in Weibo which may results in feature sparsity when sentiment lexicons are used to build feature vectors.

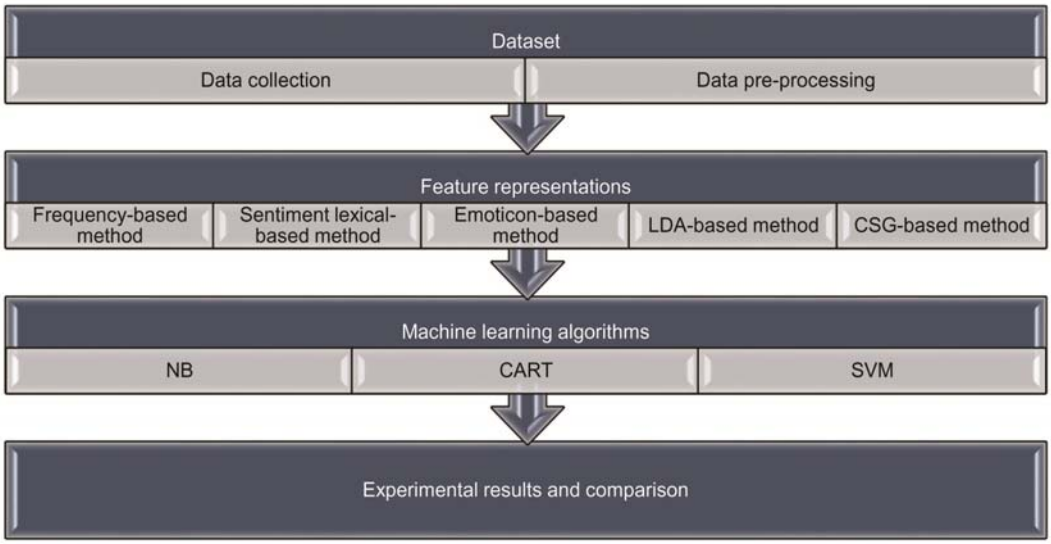


Figure 1. The framework of our paper

2.3 Representation of Words

The word representation is the core of many natural language processing tasks. The form of representing words can largely determine the performance of upper level tasks. We divide the representations of words into two groups.

(1) One-hot vector: Many machine learning algorithms require that the input must be fixed-sized vectors, resulting in many measures are taken to transform the varying-length text to meet the requirements. For example, a word can be represented as a one-hot vector. A one-hot vector may look like this, [0,0,0,0,…,1,…,0]. With a "1" in the 5ØVÜth bit of the vector, it means that 5ØVÜth word in a vocabulary occurs in the processed text. The vocabulary is generated either by doing statistical analysis on the data set or just by using a sentiment dictionary[5,24,25]. Thus, a one-hot vector in combination with a bag-of-words model can represent a varying-length text as a fixed-sized multi-hot input vector.

(2) Distributed representations: Distributed representations of words is in fact a by-product of statistical language model. But it is considered as a breakthrough in the natural language processing area. Because it captures more syntactic and semantic information of the words than the previous feature representations and improves the performances of many models that based on it. For example, based on the distributed representations of words, neural language model significantly outperforms N-gram model[26-28]. Some other studies have also shown that this new type of representation of words works well in many natural language processing tasks[29-31]. As far as we know, there is no work concentrates on applying distributed representations of words to sentiment classification on Weibo, so we decide to test whether it will perform well in Chinese social media.

3. Data and Method

3.1 Data

The data used in this paper were crawled from Weibo, a Chinese micro-blog service, like Twitter. It is the most popular social media in China and has more than 500 million users. However, as far as we know, there is no a public available data set for sentiment classification, so we created one. We randomly crawled two data sets from the Weibo through the API. One of them has more than 70 thousand posts and another one has about 20 million ones. The former one was selected and labelled to create a training data set that can be used to train and test the sentiment classifier. The information of this data set is listed in Table 1. The latter one was used to get the distributed representations of words.

Number of original posts	Number of labelled posts	Number of positive/negative posts
70,000	5,000	2,500/2,500

Table 1. Labelled data set information

3.2 Data Preprocessing

Some necessary measures were taken to preprocess the data sets before feature engineering, because the posts generated by users often contain some noisy information or some of them are written in traditional Chinese. We took four steps to clean the data:

(1) The posts that were written in traditional Chinese were converted into simplified Chinese ones.

(2) Some posts contain location information, like "I am at…", which is not helpful, so we removed the location information from posts.

(3) It is very common for users to mention others in the posts and the nicknames in them are not helpful in building models. Hence, we removed the nicknames.

(4) The URLs that can lead users to other websites are not helpful too. Hence, we also removed them from the posts.

For convenience of feature engineering, we segmented every piece of post into a sequence of words with the mmseg4j, a word-segmentation tool distributed by Google. This tool allows us to define our own dictionary which helps to improve the performance of the segmentation.

3.3 Feature Engineering

(1) Frequency-based Method: We did a statistical analysis on the words occurring in the posts and sorted them into a decreasing order by their counts. We selected the top-N words to construct a vocabulary. For the word sequence of a post , we checked whether a word in the sequence occurred in the vocabulary, if yes then the corresponding position is 1 otherwise 0. Thus, a post can be transformed into a multi-hot input vector that can be accepted by machine learning algorithms. This method is simple but effective. We guess the reason is that in comparison with other methods, such as sentiment lexicon-based method or emoticon-based method, it could avoid the sparsity of features in some degree.

(2) Sentiment lexicon-based Method: Some words, also called as sentiment lexicons, dominate a sentence or a document's sentiment polarity. Thusly, they are often selected to construct a vocabulary to help us to do sentiment analysis. Here, we used the National Taiwan University Sentiment Dictionary as our tool. The information of the sentiment dictionary is listed in Table 2. After being segmented into a word sequence, any post can be converted into a binary vector by checking out whether a word in the sequence occurs in the sentiment dictionary. Sentiment lexicon, in combination with bag-of-words model, can be used to transform the posts into fixed-sized input vectors.

(3) Emoticon-based Method: According to our observation, emoticons are frequently used by users on Weibo. For example, we randomly selected about 29 million posts from Weibo and found that 64.3% of the posts contain emotions. Like sentiment lexicons, emoticons are also very powerful signals to discriminate positive and negative posts. So we crawled a large amount

of posts from Weibo and constructed an "*emoticon vocabulary*" by extracting emoticons from them. Then, a post can be transformed into a multi-hot binary vector by checking out whether the emoticon in the post occurs in the "emoticon vocabulary". The vocabulary contains 234 emoticons (the emoticons we crawled through API were transformed into Chinese characters by Weibo), some of them are listed in Table 3.

(4) Latent Dirichlet Allocation-based method: Blei and Ng proposed the Latent Dirichlet Allocation (LDA) in 2001[32]. They assumed that a document is a mixture of k topics and each word in the document can be associated with one or more topics. LDA performs well in text categorization and some researchers used the extended version of it to learn the vectors of words and applied it to sentiment analysis.

A word's association with each topic can be obtained by training the LDA model. Based on this, every word can be represented as a k-dimensional vector and every element of it can be seen as the association strength with each topic. Then, a post can be converted into a fixed-sized vector by a linear combination of the vector of every word. For example, given a post which has been segmented into word sequence, [w_1, w_2 , ... w_n] and for a word w_i, its word vector is [w_{i1}, w_{i2} , ... w_{ik}], where k is the number of topics. So the vector of a post can be calculated with the Equation 1,

$$Tweet_{vector} = \sum_{i=1}^{T} w_i \oslash T \qquad (1)$$

where the w_i is the vector representation of a word, and T is the total number of words in a Post.

(5) Continuous skip-gram model. Different from the previous methods, the distributed representations of words allows us to make calculations between words. Then, many upper-level tasks can be done base on it, such as clustering. There are many models that can be chosen to learn the distributed representation of words. The continuous skip-gram model was chosen, because it performs well in many natural language processing tasks at a low computation cost. The continuous skip-gram (CSG) model[8] is similar to feedforward Neural network language model (NNLM) [26]. But the non-linear hidden layer in NNLM is removed and the projection layer is shared by all words. It is consisted of three layers: input, projection and output layer. The goal of the CSG model is to optimize free parameters of the model to maximize the probability of sentence. For instance, given a sentence, which has been segmented into words, [w_1, w_2 , ... w_T], the objective function of the model is,

$$\frac{1}{T}\sum_{i=1}^{T}\sum_{-c\le j\le c, j\neq 0} \log p(w_t|w_{t+j}) \qquad (2)$$

where c is the size of the window, w_t is the tth word of the sentence and T is the number of words. In our work, we used a tool distributed by Google to learn the distributed vector of words. When obtaining a word's vector, we can convert every post into a fixed-sized input vector with the help of Equation 1.

Name	Number of positive/negative entries
National Taiwan University Sentiment Dictionary(NTUSD)	2810/8276

Table 2. Sentiment lexicons in experiment

Emoticon	Meaning	Sentiment polarity
[哈哈]	Smile	Positive
[嘿嘿]	Smile	Positive
[嘻嘻]	Smile	Positive
[发怒]	Angry	Negative
[抓狂]	Angry	Negative
[爱你]	Love	Positive

Table 3. Example of emoticons

3.4 Machine Learning Algorithms

To build a sentiment classifier, we also need to choose the suitable algorithm, because the choice of the algorithm will greatly determine the performance of the classifier. In order to avoid the randomness and test the robustness of the feature representation, we chose three different and commonly used algorithms: support vector machine, naive Bayes and classification and regression tree.

(1) Support vector machine: Support vector machine[33] was once considered as the best classifier and performed very well in many different areas. The goal of the support vector machine is to find a hyperplane which can separate the data points in a d-dimensional space into two sides of the hyperplane in a maximum margin. It is briefly described below.

Given a training set (x_i, y_i), $i = 1, 2, \ldots, m$, where $x_i \in R^n$, $y_i \in \{-1, +1\}$, the data points can be classified by the below equations,

$$\text{for } y_i = +1,\ x_i \cdot w + b \geq +1 \qquad (3)$$

$$\text{for } y_i = -1,\ x_i \cdot w + b \leq -1 \qquad (4)$$

The objective function of the support vector machine is,

$$\text{Minimize } \frac{1}{2} w^T \cdot w \qquad (5)$$

$$\text{subject to } y_i(x_i \cdot w + b) \geq 1 \qquad (6)$$

(2) Naive Bayes: Naive Bayes model is a model based on Bayes theorem, following the independence assumption. Obviously, this assumption fails in many real applications, however, the naive Bayes model performs well in many tasks. In 1960s, it was introduced into information retrieval and obtained good performance in text categorization. Since text categorization is similar to sentiment classification, we decided to use the naive Bayes model to build a sentiment classifier. It is briefly described as below.

Given an example which has m features, it can be correctly classified by the following Maximum A Posteriori decision equation,

$$c^* = \arg max_c\, p(c) \prod_{j=1}^{m} p(f_j|c)$$

$$= \arg max_c\ \log p(c) + \sum_{j=1}^{m} \log p(f_j|c) \qquad (7)$$

(3) Classification and Regression Tree: Classification and regression tree was put forward by Leo Breiman[34]. It is consisted of two parts: classification tree and regression tree. This model has many advantages, its results are easy to understand and explain. Besides, the robustness of this model is very good. Most importantly, this model performs well in many tasks and has high computation efficiency. Hence, we also used it to build a sentiment classifier to test the performance of the sentiment classifier when the feature representation is different.

4. Experimental

4.1 Parameter Setup of Classifier

In our work, we used support vector machine, naive Bayes and classification and regression tree to test the performances of different feature representations. The configurations of each classifier are described as followed.

(1) Support Vector Machine: The choice of kernel function is very important for support vector machine. There are many kinds of kernel functions, such as the Radial Basis Function, Linear Function, Polynomial Function and Fourier function. Previous research have shown that the Radial Basis Function always obtains good performances, so we chose it. The libsvm is a very popular library for support vector machine, so we adopted it in our Matlab. Besides the kernel function, the gamma parameter was set as 0.01 and C as 0.7.

(2) Naive Bayes. We used the toolbox in Matlab: The prior distribution was set as 'kernel'.

(3) Classification and Regression Tree: We also used the toolbox provided by Matlab. There are two parameters required to be configured before building the model. The parameter 'method' was set as 'classification', the parameter 'prune' was set as 'on'.

4.2 Experimental Results

(1) Frequency-based Method: We did a statistical analysis on the labelled data set and sorted the words by their counts. Then, we selected the top 1000 words as the features. To evaluate the performance of this representation when the dimensionality is different, we also randomly selected 100, 200, 500 features from this feature set and built the sentiment classifier respectively. The result is shown in Fig.2. For the support vector machine and classification and regression tree, we see that the increasing dimensionality of input vector increases the accuracy of the sentiment classifier on the data set. However, for the naïve Bayes, the accuracy nearly makes no difference with the change of the dimensionality. The best result is achieved when the dimensionality is 1000 by using the support vector machine and the accuracy is 91.6%.

(2) Sentiment Lexicon-based Method: Most lexicons in NTUSD are very formal. By contrast, the words used in social media are quite informal. Therefore, it was very reasonable for us to infer that many of lexicons in NTUSD would not occur in the labelled data set. Thusly, we did a statistical analysis on the labelled data set and found that there were only 1363 sentiment lexicons occurring in the dataset, accounting for 13% of the NTUSD. These lexicons were selected as features to transform the posts into input vectors. We also evaluated the performances when choosing the subset of this feature set. The size of the subset is 100, 200, 500 and 1000. The result is shown in Fig.3. From Fig.3, we can see that naïve Bayes still performs poorly no matter what size of the input vector is. For the support vector machine and classification and regression tree, with the increase of dimensionality, the accuracy improves. The best result was achieved by using support vector machine when the dimensionality was 1000 and the accuracy is 0.734.

(3) Emoticon-based Method: Like the sentiment lexicons, emoticons play an important role in determining a post's sentiment polarity. In this paper, we collected 234 emoticons from a randomly selected dataset. These emoticons were used as features to transform the posts into input vectors. Like the previous methods, we also tested the performance of this feature representation when the dimensionality was different. The size of the subset is 13, 39, 117 and 234. Fig.4 illustrates that the naive Bayes performs badly on the data set. For support vector machine and classification and regression tree, the accuracies range from about 50% to 76% depending on the size of the dimensionality. The best result is achieved on support vector machine, but it just made little difference with classification and regression tree. The highest accuracy is 76.5%.

(4) LDA-based Method: Unlike the previous methods, the LDA-based method requires us firstly train the LDA model to get the distributed representation of words. We used the data sets of different sizes to train the LDA model. The size of the subset is 5,000,000, 1,000,000, 200,000, 40,000 and 8,000. Besides, we evaluated the performances of the sentiment classifier when the dimensionality of the word vector was different. The sizes of the dimensionality of the word vector were 500, 300, 200, 50, 10, 5 and 2. Hence, there were totally $5\times7 = 35$ kinds of combinations. The posts were transformed into input vectors based on every kind of combination and the Equation 1. Based on these combinations, the sentiment classifiers were built too. The results of each classifier are shown in Fig.5. Fig.5.a illustrates that the accuracies of the naive Bayes are about 50% or a little higher. The size of dataset from DS1 to DS5 is 8,000, 40,000, 200,000, 1,000,000 and 5,000,000. From Fig.5.b, we can see that all the accuracies of the classifier are the same, it means that the feature vector makes no contribution to the model building. The Figure.5.c illustrates that accuracies are in the range of about 60%-70%, the result suggests that this classifier performs better. However, the classification and regression tree always tends to be overfitting. In summary, the word vectors learned from LDA model cannot capture syntactic and semantic information of words precisely.

(5) CSG-based Method: Like the LDA-based method, we also trained the CSG model to get the distributed representations of words. Specifically, we used the same data set and dimensionality size as the LDA-based method for comparison. After the training, we also got $5\times7=35$ kinds of combinations of different training data sets and dimensionalities. Based on these combinations and the Equation 1, the posts were transformed into input vectors. With the use of support vector machine, naive Bayes and classification and regression tree, the sentiment classifiers were built too. The results of each classifier are shown in Fig.6. From Fig.6.a we can see that different from the previous four methods, the naive Bayes classifier performs much better here. It means the distributed representations of words learned from CSG model captures syntactic and semantic

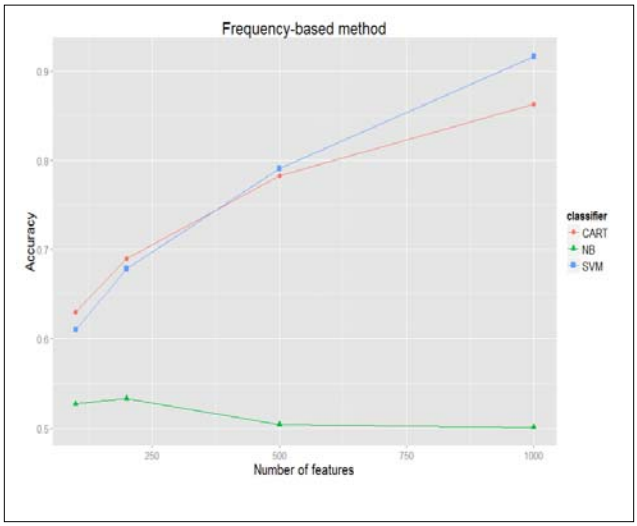


Figure 2. Frequency-based method experimental result

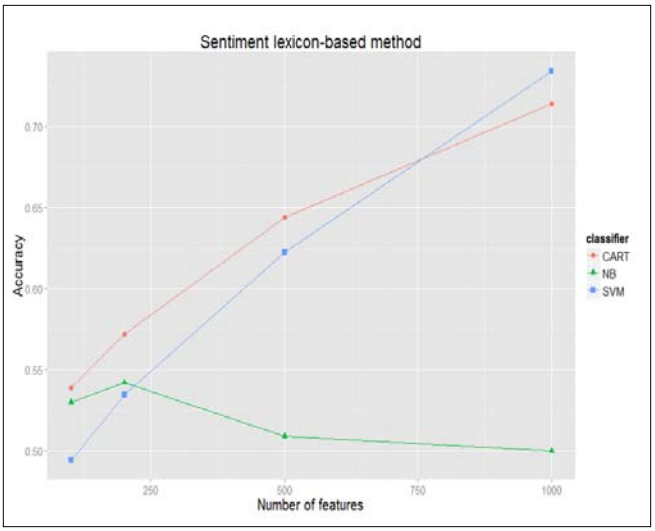


Figure 3. Sentiment lexicon-based method experimental result

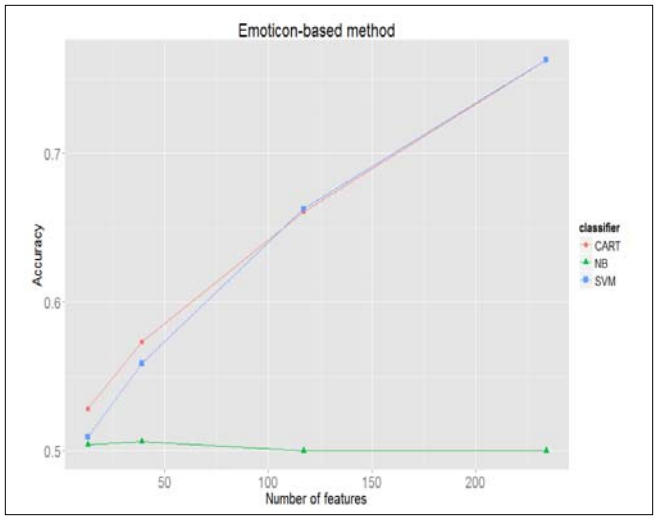
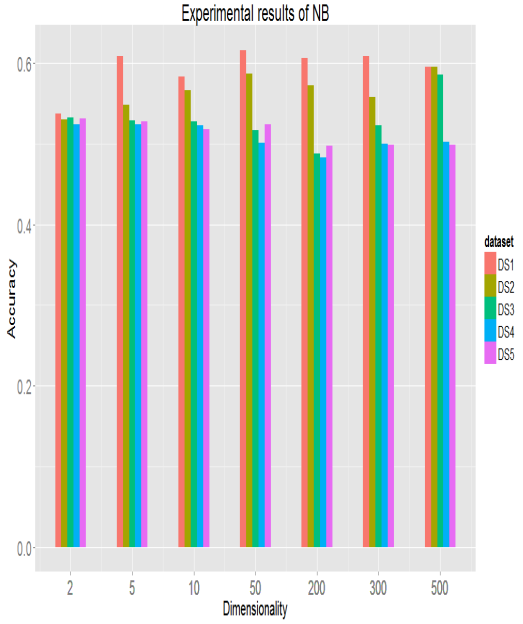
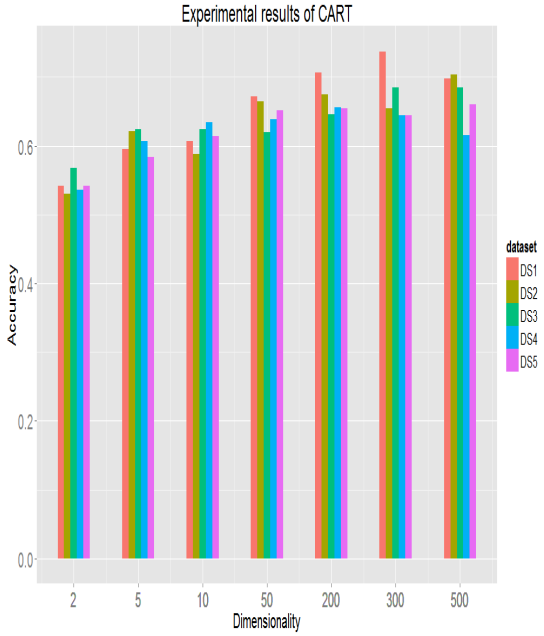


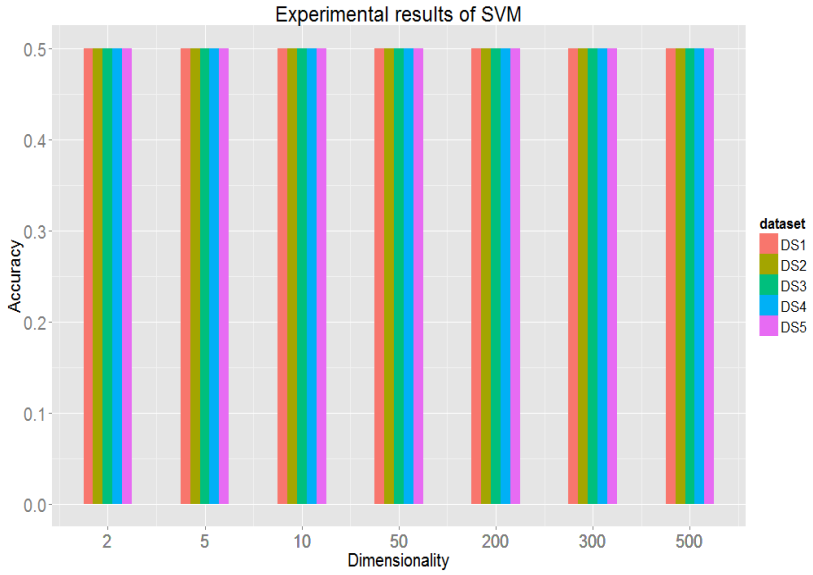
Figure 4. Emoticon-based method experimental result



a. Naïve bayes



b. CART



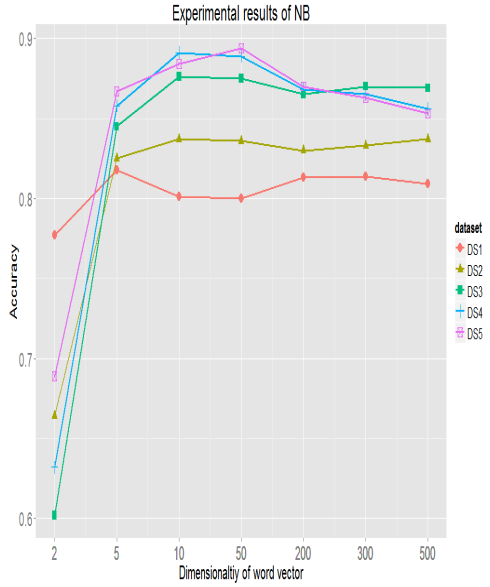
c. SVM

Figure 5. LDA-based method experimental results

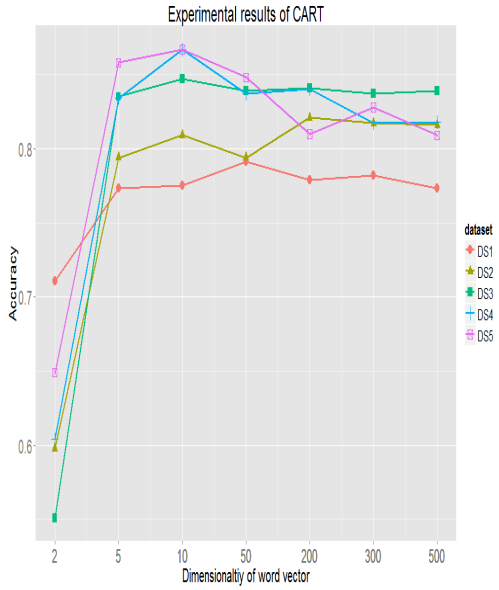
information of words precisely and has better robustness. The Fig.6.b and Fig.6.c show that for support vector machine and classification and regression tree, accuracies range from about 50% to about 90%, depending on the sizes of data set and dimensionality. The results also show that the accuracy tends to be higher when the size of the training data set is larger. However, there is no obvious pattern found when observing the accuracies with the change of the dimensionality size. The best result was achieved by support vector machine where the size of the word vector was 500 and the training data size had 5 million posts. The highest accuracy is 93.32.

Method	Classifier	Accuracy
Frequency-based method	SVM	91.6%
Sentiment lexicon-bsed method	SVM	73.4%
Emoticon-bsed method	SVM	76.52%
LDA-based method	CART	73.7%
CSG-based method	SVM	93.32%

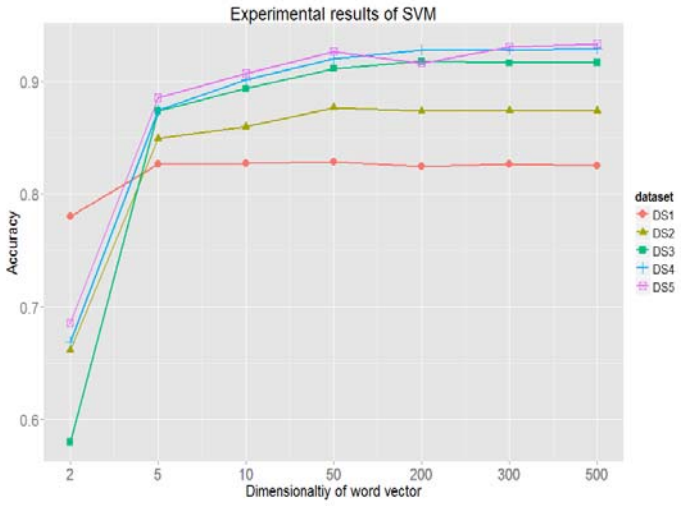
Table 4. Highest Accuracy of every method



a. Naïve bayes



b. CART



c. SVM

Figure 6. CSG-based method experimental results

4.3 Comparsion and Discussion

In this paper, we used five different methods to convert the posts into feature vectors and evaluated their performances on three different classifiers. In Table 4, we list the highest accuracy of the every method and the classifier where the accuracy was achieved.

From Table 4, we can see that most of the highest accuracies are achieved by support vector machine and the CSG-based method performs best. The frequency-based method is just a little worse than the CSG-based method, but much better than the sentiment lexicon method and emoticon-based method. We guessed the reason was that some sentiment lexicons or emoticons did not occur in some posts of the data set. Hence, we did a statistical analysis on the feature vectors transformed by sentiment lexicon-based method and emoticon-based method respectively. For the sentiment lexicon-based method, there are 1791 input vectors whose every element is zero. These input vectors come from two classes, and the ration is 1.29:1. It means 35% of the input vectors are identical with each other and nearly make no contribution to build the sentiment classifier. For the emoticon-based method, there are 2307 input vectors whose every element is zero. And, these input vectors come from two classes, the ration is 0.82:1. It means 46.14% of the input vectors are identical with each other and make no contribution to the model building. By contrast, for frequency-based method, there are only 64 input vectors whose every element is zero. For the LDA-based method, there is no such a problem, because all the words in a post are used. Therefore, we attribute the poor performance of the LDA-based method to the word vectors learned from LDA model cannot capture the syntactic and semantic information of the words precisely. The comparison of experimental results show that the degree of sparsity of the features can largely determine the performance of a feature representation. Fortunately, the CSG-based method avoids the feature sparsity and the word vectors learned from CSG model capture syntactic and semantic information precisely, so it obtained the best result in the task.

5. Conclusion and Future Work

In this study, we proposed a new method to classify posts on Weibo. We compared five different types of feature engineering techniques and obtained two important conclusions.

First of all, for the sentiment classification of posts on Weibo, the degree of feature sparsity largely determines the performance of the sentiment classifier. The experimental results show that the frequency-based method performs better than the sentiment lexicon-based method and emoticon-based method. However, in our intuition, the latter two ones were supposed to outperform the former one, because the sentiment lexicons and emoticons capture more information about emotions than the words selected by frequency.

Second, the distributed representation of words performs well in sentiment classification task on Chinese social media—Weibo. However, the LDA model and CSG were both used to learn words' vectors, but the performances of these two methods are quite different. The reason behind this is the word vectors learned from CSG capture information which can be used to compute the similarities between words and this helps to overcome the curse of dimensionality in some degree. But for the LDA model, it puts emphasis on the association between words and topics which fails to fight the curse the curse of dimensionality.

In terms of future work, the next step is to develop new techniques to transform the words' vectors into a post's vector. The method we used in this paper was quite simple and it ignored the order of the words. We believe that a new method which can capture the information of the sentence structure in the post will perform better and this will improve the accuracy of the sentiment classifier.

Acknowledgement

The authors acknowledge the support by the National Natural Science Foundation of China (grant no. 71171068).

References

[1] Zhao, J., Dong, L., Wu, J., Xu, K. (2012). Moodlens: an emoticon-based sentiment analysis system for chinese posts, *In*: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data mining. *ACM*, 1528-31.

[2] Hu, Y., John, A., Seligmann, DD., Wang, F. (2012). What Were the Posts About? Topical Associations between Public Events and Twitter Feeds. ICWSM.

[3] O'Connor, B., Balasubramanyan, R., Routledge, BR., Smith, NA. (2010). From posts to polls: Linking text sentiment to public opinion time series. *ICWSM*, 11: 122-9.

[4] De Choudhury, M., Gamon, M., Counts, S., Horvitz, E. (2013). Predicting Depression via Social Media. ICWSM. .

[5] Yuan, B., Liu, Y., Li, H., et al. (2013). Sentiment Classification in Chinese Microblogs: Lexicon-based and Learning-based Approaches, *In:* International Proceedings of Economics Development & Research, 68.

[6] Fan, R., Zhao, J., Chen, Y., Xu, K. (2013). Anger is more influential than joy: sentiment correlation in Weibo. arXiv preprint arXiv:13092402.

[7] Maas, AL., Daly, RE., Pham, PT., Huang, D., Ng, AY., Potts, C. (2011). Learning word vectors for sentiment analysis. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Vol. 1. *Association for Computational Linguistics*, 142-50.

[8] Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:13013781.

[9] Mikolov, T., Sutskever, I., Chen, K., Corrado, GS., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 3111-9.

[10] Shamma, DA., Kennedy, L., Churchill, EF. (2009). Post the debates: understanding community annotation of uncollected sources. *In*: Proceedings of the first SIGMM workshop on Social media. ACM, 3-10.

[11] Pang, B., Lee, L., Vaithyanathan, S. (2202). Thumbs up?: sentiment classification using machine learning techniques. Proceedings of the ACL-02 conference on Empirical methods in natural language processing- 10. *Association for Computational Linguistics*, 79-86.

[12] Pang, B., Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales, *In:* Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. *Association for Computational Linguistics*, 115-24.

[13] Yang, M., Kiang, M., Chen, H., Li, Y. (2012). Artificial immune system for illicit content identification in social media. *Journal of the American Society for Information Science and Technology.* 2012, 63, 256-69.

[14] Blitzer, J., Dredze, M., Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. ACL., p. 440-7.

[15] Matsumoto, S., Takamura, H., Okumura, M. (2005). Sentiment classification using word sub-sequences and dependency sub-trees. Advances in Knowledge Discovery and Data Mining. *Springer*, 301-11.

[16] Yuan, Z., Purver, M. (2012). Predicting emotion labels for chinese microblog texts. SDAD 2012 The 1st International Workshop on Sentiment Discovery from Affective Data.

[17] Baayen, H., van Halteren, H., Neijt, A., Tweedie, F. (2002). An experiment in authorship attribution. 6th JADT. Citeseer, , 29-37.

[18] Thet, TT., Na, J-C., Khoo, CS. (2010). Aspect-based sentiment analysis of movie reviews on discussion boards. Journal of Information Science..

[19] Go, A., Huang, L., Bhayani, R. (2009). Twitter sentiment analysis. Entropy, 17.

[20] Pak, A., Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. LREC. p. 1320-6.

[21] Das, S., Chen, M. (2001). Yahoo! for Amazon: Extracting market sentiment from stock message boards. *In* : Proceedings of the Asia Pacific Finance Association Annual Conference (APFA). Bangkok, Thailand, 43.

[22] Kim, S-M., Hovy, EH. (2007). Crystal: Analyzing Predictive Opinions on the Web. EMNLP-CoNLL. Citeseer, p. 1056-64.

[23] Esuli, A., Sebastiani, F. (2007). Pageranking wordnet synsets: An application to opinion mining. ACL. Citeseer, p. 442-31.

[24] Polanyi, L., Zaenen, A. (2006). Contextual valence shifters. Computing attitude and affect in text: Theory and applica

tions. Springer, 1-10.

[25] Choi, Y., Cardie, C. (2008). Learning with compositional semantics as structural inference for subsentential sentiment analysis. *In*: Proceedings of the Conference on Empirical Methods in Natural Language Processing. *Association for Computational Linguistics*, 793-801.

[26] Bengio, Y., Schwenk, H., Senécal, J-S., Morin, F., Gauvain, J-L. (2006). Neural probabilistic language models. Innovations in Machine Learning. Springer, 137-86.

[27] Mikolov, T., Deoras, A., Kombrink, S., Burget, L., Cernocký, J. (2011). Empirical Evaluation and Combination of Advanced Language Modeling Techniques. INTERSPEECH., 605-8.

[28] Schwenk, H. (2007). Continuous space language models. *Computer Speech & Language*. 21: 492-518.

[29] Collobert, R., Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning, *In:* Proceedings of the 25th International Conference on Machine learning. ACM, 160-7.

[30] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P. (2011).Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12: 2493-537.

[31] Turian, J., Ratinov, L., Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning, *In*: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. *Association for Computational Linguistics*, 384-94.

[32] Blei, DM., Ng, AY., Jordan, MI. (2001). Latent dirichlet allocation. Advances in Neural Information Processing Systems. 2001, p. 601-8.

[33] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features.Springer.

[34] Breiman, L., Friedman, J., Stone, CJ., Olshen, RA. (1984). Classification and regression trees.CRC press.