Chinese Text Sentiment Classification Based on Extreme Learning Machine

Fangye Lin and Yuanlong Yu

Abstract With the rapid growth of the Web text data, mining and analyzing these text data, especially the online review data posted by the users, can greatly help better understand the usersconsuming habits and public opinions, it also plays an important role in decision-making for the enterprises and the government. But in the process of vectoring text, many current Chinese text sentiment classifications treat words as atomic units, there is no notion of similarity between words. In order to solve this problem, this paper imports word embedding to capturing both the semantic and syntactic information of words from a large unlabeled corpus. In the section of experiment, we toke the noun, verb, and adjectives as candidate set, used χ^2 statistic to reduce the number of dimensions. We mainly compared one-hot representation and word embedding as the expression of word to certain tasks, we also proposed the pooling method with word embedding to standardizing the vector, the ELM with kernels was adopted to analyze the text emotion tendentiousness. Finally the paper summarizes the current status, remaining challenges, and future directions in the field of sentiment classification.

Keywords Sentiment classification · Word embedding · Extreme learning machine

1 Introduction

With the rapid increase usage of the Internet, there are more and more subjective information appearing at the social medium, such as forum, community, blog and shopping websites. Both individual and organization became strongly relying on the review information obtained from the Internet to make their own decisions. However, due to the huge amount of information available on the Internet, one has to

F. Lin · Y. Yu (✉)
The College of Mathematics and Computer Science, Fuzhou University,
Fuzhou 350116, Fujian, China
e-mail: yu.yuanlong@fzu.edu.cn

F. Lin
e-mail: linfangye20@163.com

J. Cao et al. (eds.), *Proceedings of ELM-2016*, Proceedings in Adaptation,
Learning and Optimization 9, DOI 10.1007/978-3-319-57421-9_14

search, check and judge each review one by one before the person or organization can make the final decision. In this situation, it will be very useful to first summarize the relevant huge amount of information, this summary will be valuable for both the customer and manufacturer. This kind of work is called opinion-based multi-document summarization. Furthermore, it will greatly enhance the customers efficiency to obtain the information if there is an automatic analysis of the original information, for example, which is positive attitude, which is negative attitude, and to what extent. This is called sentiment classification, which is a very important research topic in the field of natural language processing. While sentiment classification for English text has made a great progress, current research work on sentiment classification for Chinese text is still in its infancy. Due to the huge difference between English and Chinese in syntax, semantics and pragmatics etc., we face more problems in the processing of Chinese text.

In recent years, domestic scholars have done the relevant research according to the characteristics of emotion classification problem. Xu jun [1] used Naive Bayes and Maximum Entropy classification for the sentiment classification of Chinese news and reviews, the experimental results show that the methods they employed perform well. Moreover, they found that selecting the words with polarity as features, negation tagging and representing test documents as feature presence vectors can improve the performance of sentiment classification. ZHOU Jie [2] summarized the characteristics of netnews comments firstly, and selected different sets of feature, different feature dimensions, different feature-weight methods and parts of speech to construct classifiers, then made the comparison and analysis to the experimental results. The results of comparison showed that the features combining sentiment words and argument words perform well to those only employing sentiment words.

In this paper, sentiment classification for Chinese text will be seen as two classification problems, namely positive and negative tendencies. We selected 2607 negative reviews and 5149 positive reviews as the experimental data, and the rate of training sample and testing sample is 2:1. Finally the model of ELM with kernels learned the training sample set, and gave out the result of sentiment classification on the testing sample set. Section 2 will describe some basic models used in sentiment classification and some novel technologies which are proposed in recent years. Section 3 will be detailedly introduce the data processing and feature extraction, we will show the comparison of experimental results. The last section we will have the conclusion of the present stage of the work.

2 System Architecture

The proposed method consists of two modules as shown in Fig. 1: training and testing.

In the part of training phase to need to training the sentiment lexicons from training set, the training set which consists of positive samples and negative samples. We also need to training the word embedding from corpus. In the stage of testing, the

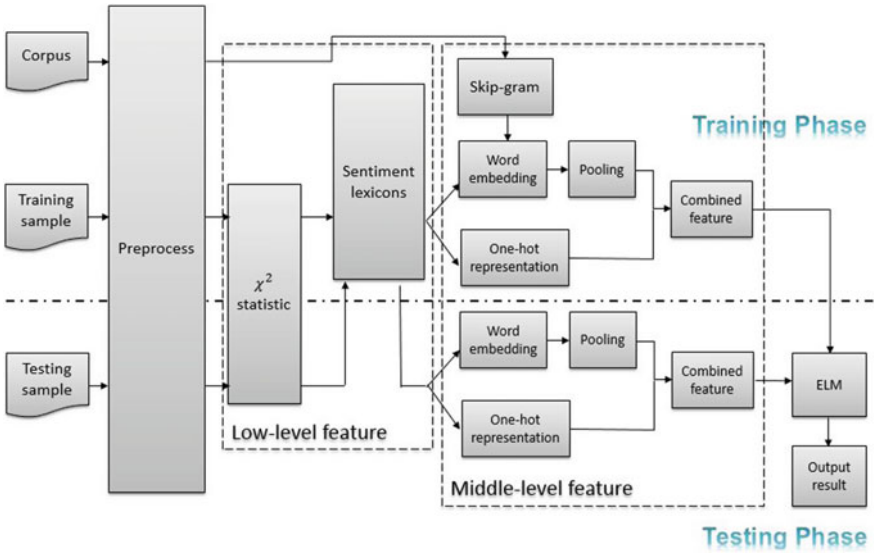

Fig. 1 Architecture of the proposed sentiment classification

testing samples are mapped to the sentiment lexicons after preprocess, all the data set are expressed as middle-level feature and put them into ELM for classifying.

2.1 Preprocess

Denoising: In the stage of experiment the tendency of the reviews should not be equivocal, so we check the whole corpus and delete some ambiguous content and repeated content.

Segmentation: In this paper, we use the Institute of Computing Technology, Chinese Lexical Analysis System (ICTCLAS) [3] as word segmentation tool. After this part we get a stream of words which bring the part-of-speech tagging (POS), we saved all these information by using hashmaps for the next step of study.

Filtering the Stop-Word: After comparison we choose the baidu stop-list to remove some words which have little contribution but occur frequently to the sentimental classification processing.

2.2 Low-Level Feature

Vector space model is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers. Each dimension corresponds to a

separate term. If a term occurs in the document, its value in the vector is non-zero. forms are as follows:

$$D = \left(W_{term1}, W_{term2}, ..., W_{termn}\right) \tag{1}$$

Each dimension in the vector means the weights of the term in this document, and it describes the influence of the words in the document, several different ways of computing these values, also known as (term) weights, have been developed. One of the best known schemes is term frequency-inverse document frequency (TF-IDF) weighting.

$$W_{ik} = \frac{tf_{ik} * idf_k}{\sqrt{\sum_{k=1}^{t} (tf_{ik})^2 \left(idf_k\right)^2}} \tag{2}$$

$$tf_{ik} = \frac{n_{ik}}{\sum_{i=1}^{n} n_{ik}} \tag{3}$$

$$idf_k = \log \frac{|D|}{m + \left|\left\{k : t_i \in d_k\right\}\right|} \tag{4}$$

The $|D|$ is the total number of documents in the document set; the $|\{k : t_i \in d_k\}|$ is the number of documents containing the term t_i. If the term is not appeared in this document, the divisor is zero. tf_{ik} means the frequency of the term appear in this document set and idf_k means the words on the distribution of the documents in the collection of quantitative.

Considering the traditional vector space model needs more time expense as its vector dimension turns greater, we use the χ^2 statistic (CHI) [4] as feature selection method to reduce the number of dimensions. The χ^2 statistic measures the lack of independence between t and c and can be compared to the χ^2 statistic distribution with one degree of freedom to judge extremeness. Using the two-way contingency table of a term t and a category c, where A is the number of times t and c co-occur, B is the number of time the t occurs without c, C is the number of times c occurs without t, D is the number of times neither c nor t occurs, and N is the total number of documents, the term-goodness measure is defined to be:

$$\chi^2(t, c) = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)} \tag{5}$$

2.3 Middle-Level Feature

Many current natural language processing (NLP) systems and techniques treat words as atomic units, there is no notion of similarity between words. The most common expression of word is one-hot representation, in this method every word is expressed as a long vector, the dimension of this vector is the vocabulary size, only one dimension of the value is 1 and others value is 0, this dimension expresses current word and we use sparse coding to storage it, this choice has several good reasons like simplicity, robustness and the observation that simple models trained on huge amounts of data outperform complex systems trained on less data.

However, the simple techniques have some drawbacks in many tasks. For example, the amount of relevant in-domain data for automatic speech recognition is limited, the performance is usually dominated by the size of high quality transcribed speech data. In machine translation, the existing corpora for many languages contain only a few billions of words or less. Thus, there are situations where simple scaling up of the basic techniques will not result in any significant progress, and we have to focus on more advanced techniques.

With progress of machine learning techniques in recent years, it has become possible to train more complex models on much larger data set, and they typically outperform the simple models. Probably the most successful concept is to use distributed representations of words [5] -word embedding. It can capture both the semantic and syntactic information of words from a large unlabeled corpus and has attracted considerable attention from many researchers.

Mikolov [6] and his team proposed two novel model architectures for computing continuous vector representations of words from very large data sets. The two architectures are continuous bag-of-words (CBOW) model and skip-gram model. the models are shown in Fig. 2. We choose the Skip-gram model to train word embedding, it tries to maximize classification of a word based on another word in the same sentence. More precisely, we use each current word as an input to a log-linear classifier with continuous projection layer, and predict words within a certain range before and after the current word. The training complexity of this architecture is proportional to:

$$Q = C \times (D + D \times \log_2(V)) \tag{6}$$

where C is the maximum distance of the words. Thus, if we choose $C = 5$, for each training word we will select randomly a number R in range $< 1; C >$, and then use R words from history and R words from the future of the current word as correct labels. This will require us to do $R \times 2$ word classifications, with the current word as input, and each of the $R + R$ words as output. In the following experiments, we use $C = 5$.

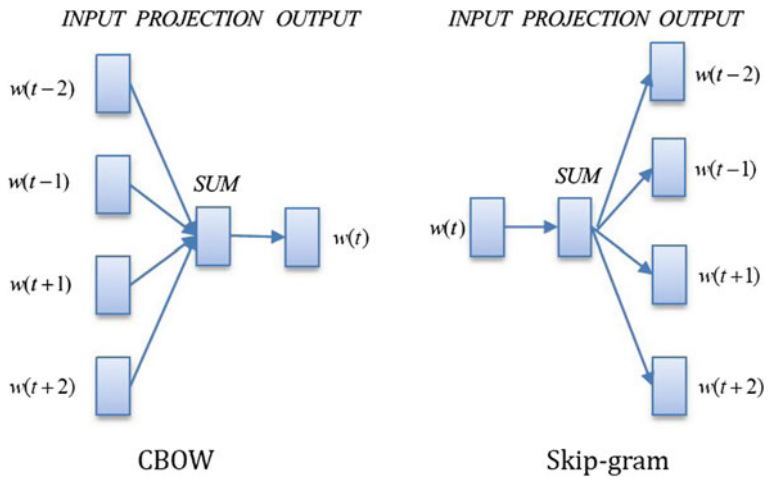

Fig. 2 The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word

2.4 Extreme Learning Machine

In the field of machine learning, a series of traditional machine learning algorithms were improved in order to satisfy the higher data processing needs. For instance, the model of Naive Bayes, Maximum entropy, Support vector machines (SVMs) [7] and so on which reduced the difficulty of solving a certain task. However, there are still some problems with those algorithms: (1) the speed of solution is slower than required for large data; (2) the model related to SVMs need to manual adjustment parameters (C, γ) frequently, they also repeat training in order to obtain the optimal solution with tedious time-consuming process and poor generalization ability.

Under the circumstances, extreme learning machine provides a new way to solve these problems. Extreme Learning Machine was first proposed by Huang [8] in 2006, ELM is generalized single-hidden layer feedforward networks (as illustrated in Fig. 3). 'Extreme' means it breaks limitations of traditional artificial learning methods and aims to work like the brain. Compare to the SVM algorithm, ELM may get better or similar predictive accuracy with less time.

The output function of ELM for generalized SLFNs (take one output node case as an example) is:

$$f(x) = \sum_{i=1}^{L} \beta_i G(a_i, b_i, x) = \beta \cdot h(x) \tag{7}$$

ELM can guarantee the regression prediction accuracy by minimizing the output error:

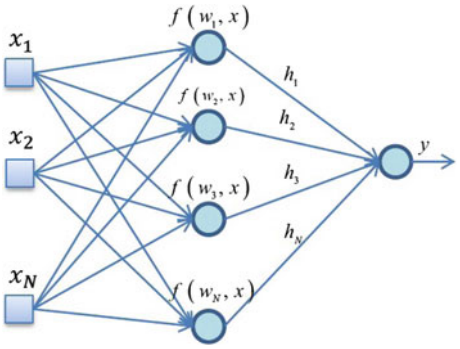

Fig. 3 Feedforward neural network with single hidden layer

$$\lim_{L\to\infty}(f(x)-f_o(x))=\lim_{L\to\infty}(\sum_{i=1}^{L}\beta_i h_i(x)-f_o(x))=0 \tag{8}$$

where L is the number of hidden neurons h_i and $f_o(x)$ is the goal of the forecast function value.

At the same time, ELM is guaranteeing the generalization ability of the network by minimizing the output weights. In general, β is calculating with the least squares, the formula is:

$$\beta = H^{\dagger}O = H^T\left(HH^T\right)^{-1}O = H^T(\frac{1}{C}+HH^T)^{-1}O \tag{9}$$

where H is the hidden-layer output matrix and $H^{\dagger}$ is the Moore-Penrose generalized inverse of matrix H. It can add a constant to get a better generalization ability according to ridge regression.

As for ELM with kernels, it obtains a better regression and classification accuracy by introducing kernel.

$$f(x)=h(x)\beta=\begin{pmatrix}K(x,x_1)\\ \ldots\\ K(x,x_N)\end{pmatrix}\left(\frac{1}{C}+\Omega_{ELM}\right)^{-1}O \tag{10}$$

$$\Omega_{(i,j)}=\exp(-\gamma(x_i-x_j)^2) \tag{11}$$

where Ω_{ELM} is the kernel function and N is the dimension of input.

3 Experiments

In the experimental part, we choose the hotel BBS reviews information which was collected by song-bo tan as our original corpus. After filtering some ambiguous content and repeated content. finally we select 2607 negative reviews and 5149 positive reviews as the experimental data, and the rate of training sample and testing sample is shown in Fig. 4.

In this part,we compare different parts of speech to build the suitable sentiment lexicon. Intuitively, we would think that the adjective directly determines the emotion of the text but we find only the adjective can not completely express the semantic information of the review.the result is shown in Table 1. If we only choose the adjective as candidate set, althrough the accuracy is 82.63% we only get 799 terms from training sample, too much information is lost. Considering there are not enough adverbs are trained in the word embedding,finally we choose the noun, verb, adjective to build the sentiment lexicon.

Meanwhile, we compare the different dimension of the vector and we find feature dimension has less influence on the accuracy of classification, The result is shown in Table 2. So we decide to choose the 3000 as the feature dimension, after this section every text is represented as a vector.

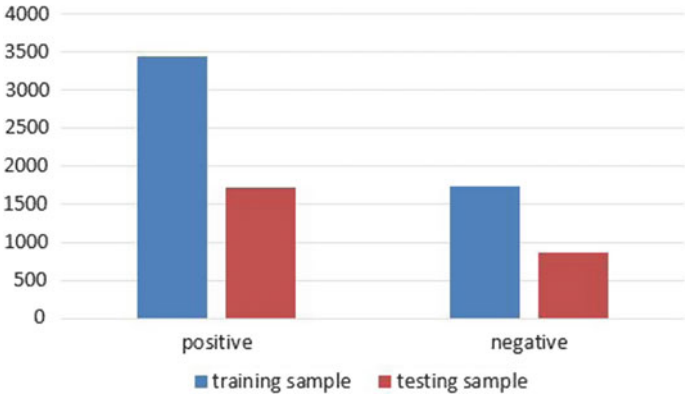

Fig. 4 The rate of training sample and testing sample

Table 1 Comparison between different parts of speech

Parts of speech	Accuracy (%)	TrainingTime (s)	TestingTime (s)
Noun	73.00	2.9317	1.6657
Noun, verb	75.42	3.3108	1.7063
Noun, verb, adjective	80.81	3.1306	1.9216
Noun, verb, adjective, adverb	80.97	2.8238	1.4792

Table 2 Comparison between different dimensions of vector

Dimension of vector	Accuracy (%)	TrainingTime (s)	TestingTime (s)
2000	80.78	2.3067	1.0994
3000	80.81	3.1306	1.9216
4000	80.70	3.1515	1.9125

Word embedding models capture useful information from unlabeled corpora, so we try to use word embeddings as the feature to improve the performance of certain tasks. The first task is the sentiment classification in which the word embedding is the only feature, the second task we use word embedding as an additional feature to achieve the sentiment classification.

According to experiment of Siwei Lai [9] which proves the simplest model, Skip-gram, is the best choice when using a 10M- or 100M- token corpus. We use the Skip-gram model to train the word embedding and the dimension of every term is 100, the word embeddings are used as feature to classify.

3.1 Pooling

In the section we use word embedding as feature, we meet a problem that we find it is hard to standardizing the vector, we hope to use the word embedding to reduce the dimension of vector and we can also remain the semantic information of every review at the same time. However, after preprocess every text are divided as diverse words and the number of words is different. At the first time we simply add the every word embedding which occurs in the text together Unfortunately we find the result is not good enough, the method of simple addition which loses the connection between words, so we propose the pooling method to standardizing the vector, we split the 3000 index of words into 10 parts and we add every word embedding which occurs in same part, if there is no words in this part, we use a zero vector of 100 dimension to express this part, finally we connect the vector according to the sequential order as the final expression of text. After comparison we find pooling method performs much better than simple addition method, the result is shown in Table 3.

Table 3 Comparison between pooling and no pooling

Method	Accuracy (%)	TrainingTime (s)	TestingTime (s)
Word embedding	78.51	1.5723	0.3145
Word embedding (pooling)	79.54	2.0491	0.7345

Table 4 Comparison between SVM and ELM with kernels

Classifier	Accuracy (%)	TrainingTime (s)	TestingTime (s)
SVM	79.5391	103.8211	69.9009
ELM with kernels	80.81	3.1306	1.9216

3.2 Classification Result

In this section we firstly compare SVM and ELM with kernels as the sentiment classifier in one-hot representation, the simulations for SVM and ELM with kernels algorithms are carried out in MATLAB environment running in a Core i7-4770, 3.40GHz CPU, 32G RAM. In Table 4. we find ELM with kernels performs better than SVM both in accuracy and saving times.

Then we compare the one-hot representation and word embedding with pooling, we try to connect the one-hot representation and word embedding as combined feature, it works but the promotion of accuracy is very tiny. The final result is shown in Table 5. The curve of training accuracy versus the kernel parameters is shown in Fig. 5.

Table 5 Comparison between one-hot representation and word embedding

	Accuracy (%)	TrainingTime (s)	TestingTime (s)
One-hot Representation	80.68	6.1624	3.8013
Word embedding (pooling)	79.86	3.4444	1.1522
One-hot + Word embedding (pooling)	80.89	5.5068	2.9478

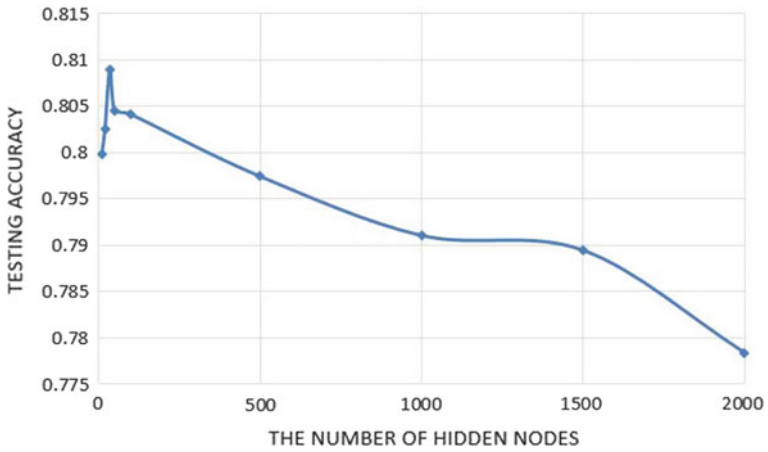

Fig. 5 The curve of testing accuracy versus the Kernel Parameters

4 Conclusion

This paper thought the analysis of the emotional polarity of text as two-classification problems. We used the VSM model to represents a document, and compared one-hot representation and word embedding in expressing words, ELM with kernel gave out the result of classification. Our main operation to the data set was cleaning, word segmentation, removing stop words, feature selection and classification. We found word embedding with pooling method has more advantages than one-hot representation in reducing the dimension of text vectoring, simultaneously it also captured both the semantic and syntactic information of words. In the part of classifier we found it took less time for ELM to training and testing the same data set than SVM. The further research we think is to design a better corpus for getting better word embeddings, we hope the word embedding can help to improve some certain tasks of sentiment classification.

References

1. Xu, J., Ding, Y.X., Wang, X.L.: Sentiment classification for Chinese news using machine learning methods. J. Chin. Inf. Process. **21**(6), 95–100 (2007)
2. Zhou, J., Lin, C., Bi-Cheng, L.I.: Research of sentiment classification for net news comments by machine learning. J. Comput. Appl. **30**(4), 1011–1014 (2010)
3. Zhang, H.P., Yu, H.K., Xiong, D.Y., Liu, Q.: HHMM-based Chinese lexical analyzer ICTCLAS. In: Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, 17, pp. 184–187 (2003)
4. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. ICML **97**, 412–420 (1997)
5. Hinton, G.E., McClelland, J.L., Rumelhart, D.E.: Distributed representations. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition. MIT Press, Cambridge, MA (1986)
6. Mikolov, T., Chen, K., Corrado, G.: Efficient estimation of word representations in vector space. Comput. Sci. (2013)
7. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: ACL-02 Conference on Empirical Methods in Natural Language Processing, pp. 79–86 (2002)
8. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: 2004 IEEE International Joint Conference on Neural Networks, 2004. Proceedings, vol. 2, pp. 985–990 (2004)
9. Lai, S., Liu, K., He, S., Zhao, J.: How to generate a good word embedding? Intell. Syst. IEEE (2015)