Chinese Sentiment Classification Using Extended Word2Vec

ZHANG Sheng (张 胜)*, ZHANG Xin (张 鑫), CHENG Jia-jun (程佳军), WANG Hui (王 晖)
College of Information System and Management, National University of Defense Technology, Changsha 410072, China

Abstract: Sentiment analysis is now more and more important in modern natural language processing, and the sentiment classification is the one of the most popular applications. The crucial part of sentiment classification is feature extraction. In this paper, two methods for feature extraction, feature selection and feature embedding, are compared. Then Word2Vec is used as an embedding method. In this experiment, Chinese document is used as the corpus, and tree methods are used to get the features of a document: average word vectors, Doc2Vec and weighted average word vectors. After that, these samples are fed to three machine learning algorithms to do the classification, and support vector machine (SVM) has the best result. Finally, the parameters of random forest are analyzed.

Key words*: *sentiment classification*; *Chinese documents*; *Word2Vec
CLC number: TH17 Document code: A
Article ID: 1672-5220(2016)05-0823-04

Introduction

Sentiment analysis of textual content is of great significance for many practical applications such as word-of-mouth tracking and public opinion monitoring. Typically, these tasks are fulfilled *via* sentiment classification followed by sentiment summarization. Namely, user-composed web documents are classified separately to be positive, negative, or neutral, or to be different levels of sentiment intensity, and then the classification results are aggregated along different dimensions (or aspects) to form opinion summaries. Providing the basis for both qualitative and quantitative summaries, sentiment classification is crucial for the aforementioned opinion mining processes.

To implement the accurate sentiment classification, a key step is to discover features that can effectively discriminate opinionated documents from non-opinionated ones and the different pre-defined classes of sentiments, *i. e.*, feature mining. Existing techniques for feature mining can be largely categorized into two groups, namely feature selection and feature embedding. The first group are often based on some sentiment lexicons, and exploit information gain, N-gram model or χ^2 test to select the most discriminative features. As natural languages are evolving constantly and novel words and creative spellings keep appearing, features defined upon fixed lexicons will inevitably become less effective over time. What is more, since the feature lengths are normally proportional to those of the lexicon, these techniques usually suffer from the curse of dimensionality.

In this paper, the latter class of techniques, feature embedding, is adopted which overcomes the two limitations of feature selection methods mentioned above by mapping texts into fixed-length vectors.

To summarize, the main contributions of this paper are three-pronged. The embedding method (Word2Vec) is introduced to extract the feature of Chinese documents. Also, the idea of transfer learning is drawn. Apart from that, Word2Vec is extended to get the vector of a document.

1 Related Works

Bengio *et al.* [1] proposed the neural network language model (NNLM) and obtained their word vectors. They successfully combined neural networks with NLP tasks. However, the NNLM proposed suffers too much from its daunting complexity. To simplify NNLM, Miklov *et al.* [2-3] proposed the Word2Vec model that could efficiently train word vectors. Nowadays, Word2Vec are widely used in feature extraction of texts.

Le and Mikolov [4] improved Word2Vec to Doc2Vec and used the similar method to get the vector of a document, and then they used these vectors to do the sentiment classification and got a great result in English document.

Maas *et al.* [5] proposed a new model, a mix of unsupervised and supervised techniques, to get word vectors, and they used probabilistic models to train the vectors. Mesnil *et al.* [6] combined many models into an ensemble model, and compared their method to other single methods. They achieved a conclusion that different model combinations were superior to individual models.

Nevertheless, those papers proposed mainly designed for English documents, while very few had been devoted to Chinese texts. This situation motivates us to explore Word2Vec for Chinese sentiment classification.

2 Methodology

2.1 Word2Vec models

The network in Ref. [1] is made up of three layers: input layer, projection layer and output layer. However, it is time-consuming. Therefore, Mikolov proposed an improved model, Word2Vec, which deleted the hidden layer and added a projection layer. Figures 1-2 illustrate the construction of the models.

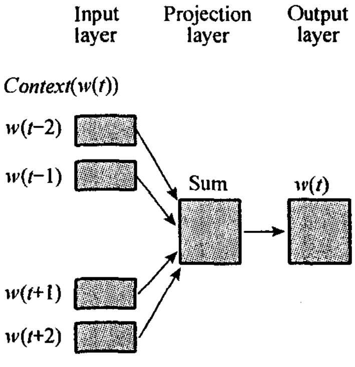

Fig. 1 Continuous bag-of-words model(CBOW)

Received date: 2016-01-10
Foundation item: National Natural Science Foundation of China (No. 71331008)

* Correspondence should be addressed to ZHANG Sheng, E-mail: zhangsheng@nudt.edu.cn

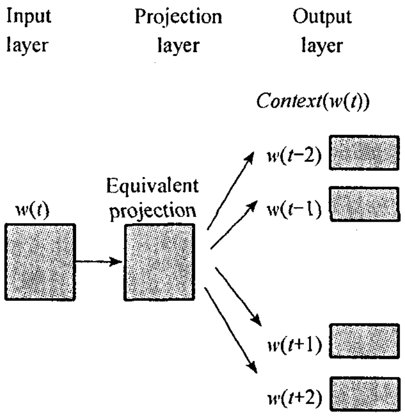

Fig. 2 Continuous skip-gram model

We can see that Word2Vec has two models. One is the CBOW, the other is the continuous skip-gram model. They are very similar. The former aims to get the possibility of a certain word $w(t)$ given a window of document. But the latter uses the specified word $w(t)$ to get other words in the window.

More specifically, given a certain word w, its neighbor words in the same window is *Context*(w) , so the purpose of CBOW is to get the log maximum likelihood function value L,

$$L = \sum_{w \in C} \log p(w \mid Context(w)), \qquad (1)$$

where C is the dictionary containing all words in corpus.

Similarly, the log maximum likelihood function value L is,

$$L = \sum_{w \in C} \log p(Context(w) \mid w), \qquad (2)$$

In practice, two methods are often used to implement these two models, one is Hierarchy Softmax, the other is Negative Sampling. These two methods aim to reduce the complexity of calculating matrix in NNLM and accelerate the calculation. More introductions can be found in Maklov' s paper [2-3].

2.2 Feature extraction for document

Sentiment feature extraction is one of the most important steps in sentiment classification. Three Word2Vec-based feature embedding methods are used to extract sentiment features from Chinese documents.

2.2.1 Average word vectors

Before doing the classification, we have to extract the feature of each document. An easy approach is average word vectors, which just gets the average of every word in the document. The vector of the document $V(D)$ can be described as

$$V(D) = \frac{1}{|w \in D|} \sum_{w \in D} v(w), \qquad (3)$$

where D is the document, $v(w)$ is the vector of word w.

2.2.2 Doc2Vec

Much similar to Word2Vec, Le and Mikolov[4] introduced Doc2Vec. He used an extra vector, called the document vector, as a unit in the input layer. So after the training, the Doc2Vec can acquire a document vector. He used this vector to represent the whole document.

2.2.3 Weighted average word vectors

It is well-known that the topic sentences always appear at the beginning or the end of the document. So the words in those areas are supposed to be more important than the words in other areas. Therefore, average word vectors method can be improved by weighting the words. The weighted document vector is calculated,

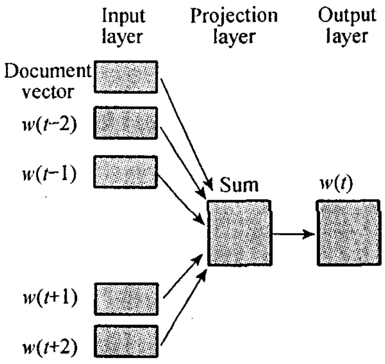

Fig. 3 Framework for learning paragraph vector

$$V(D) = \frac{1}{|w \in D|}\left\{1.5 \sum_{w_1 \in D_1} v(w_1) + 0.5 \sum_{w_2 \in D_2} v(w_2)\right\}, \qquad (4)$$

where w_1 is the word at the beginning or the end of the document D_1 , w_2 is the word at other area D_2 . A document D is divided into two parts D_1 and D_2 , making sure that they have the same number of words.

2.3 Machine learning algorithms

After the features of the documents are got, these datasets are fed to the machine learning algorithms to do the sentiment classification. Supervising learning algorithms are always used as the classification. Basically, the sentiment is usually divided to two classes, positive and negative. So this is a binary classification.

In this paper logistic classification (LC), support vector machine (SVM) and random forest (RF) are used to do the sentiment classification.

2.3.1 LC

LC function is shown,

$$g(x) = \frac{1}{1 + e^{-\theta^{T} x}} x, \qquad (5)$$

where $\theta^{T} x$ is a weighted linear function, and θ is the parameter. If a sample x satisfies $g(x) > 0.5$, then it is a positive sample, otherwise it is a negative sample.

2.3.2 SVM

SVM is a supervising learning algorithm, whose target is to find a hyper layer, dividing data from different classes apart. The closest data points from two classes to the hyper layer are called support vectors. The main idea of SVM is to find the hyper layer which has the max distance to support vectors.

The kernel function is widely used in SVM, which projects the linearly inseparable point into a higher dimension, so the SVM can solve non-linear classification problems.

2.3.3 RF

RF is an ensemble machine learning algorithm. It is made up of many decision trees. Though a decision tree may preform poor in classification, with an ensemble algorithm, RF is an effective way to do classification tasks. It is also a supervising learning algorithm and has the following advantages:

(1) it is easy to be parallelized, and the algorithms can be run in many nodes at the same time;

(2) the algorithm can be used to figure out which feature is more important in the dataset;

(3) the algorithm preforms great in many datasets, and it

is easy to handle high-dimension data.

3 Experiments and Results

After illustrating these models and methods, the Chinese documents on the Internet are used to do our experiment.

3.1 Data

In our experiment, Jingdong products reviews dataset [8] is used as a corpus to do sentiment classification tasks. Jingdong is a big e-commercial retailer. The dataset contains 1 914 negative documents and 1 829 positive documents. In our experiment, 1 200 positive reviews and 1 200 negative reviews are used as the training set, and the rest as the test set.

It should be noted that the volume of lots of corpus for sentiment classification is limited, so it is insufficient for Word2Vec to train the word vectors. So we draw on the idea of transfer learning, adding other different distributed corpuses as training corpus to get word vectors. Therefore, the data from Weibo, a famous microblog platform in China, are used as the extra corpus due to its short length and oral expression, which is similar to products reviews.

3.2 Data preprocessing

(1) Simplified Chinese conversion

In this dataset, many user would like to use traditional Chinese, so for our convinence, we converse all traditional Chinese into simplified Chinese.

(2) Word segmentation

Unlike English and other western languages, Chinese has no separator between words. So we should use Chinese segmentation utensils to do the segmentation. Therefore, Fudan NLP(FNLP)[9] is used to segment the Chinese document in the dataset.

(3) Removing stop words

Stop words refer to the most common words in the language, such as "a", "the", and "on" in English. However, most of them has no certain meaning. In order to reduce the influence of these words, we build a stop words list, and remove stop words in the document.

3.3 Feature extraction

In the experiment, three methods are used to extract features. A Python package genism[10] is used to train Word2Vec model and Doc2Vec model.

When training Word2Vec model, both CBOW model and skip-gram model are used, and the length of word vector in each model is 200. Then we concatenate two vectors into a 400-length vector. The window size is 10 and the minimum count is 1. Furthermore, we choose negative sampling as our method to implement Word2Vec.

After that, we use Eqs. (3)-(4) to get the feature of the document.

Similarly, both paragraph vectors with distributed memory model (PV-DM) and paragraph vectors with bag of words (PV-DBOW) are used, and the length of document vector in each model is 200. Then we concatenate two vectors into a 400-length vector. The window size is 10 and the minimum count is 1. Furthermore, we choose negative sampling as our method to implement Word2Vec.

Essentially, the document vector we get is the feature of that document.

3.4 Training algrothms

In the experiment we use a famous machine learning tool called scikit-learn [11] to train the classifiers.

We then feed the training dataset and their labels ("1" for positive, "0" for negative) to the algorithms, and then get the accuracy of each algorithm in the condition of three feature extraction methods, as what Table 1 shows.

In our experiment, LC is 0.5 as the threshold of positive and negative. SVM uses Gaussian Kernel, the error term C is 1, and the value of gamma is 1. RF uses Gini impurity as a criterion, and uses 50 trees as a forest. The definition of accuracy is shown as

$$\text{Accuracy} = \frac{\text{Number of rightly classified samples}}{\text{Number of total samples}}. \tag{6}$$

Table 1 Accuracy of each algorithm in the condition of three methods

Item	Accuracy/%		
	LC	SVM	RF
Doc2Vec	77.8	88.9	75.3
Average word vectors	84.6	90.1	83.7
Weighted average word vectors	85.4	90.3	87.6

Comparing features we got from different approaches, the Doc2Vec performs poor in Chinese sentiment classification, but average word vectors get a considerable performance and weighted average word vectors yield state-of-the-art results. Therefore, as for Chinese sentiment classification of products reviews datasets, weighted average of word vector is better than other two methods. Apart from that, the accuracy of SVM is superior to other two algorithms in all three methods.

RF preforms poorer than other two algorithms. Therefore, in order to find the influence of parameter of RF, we set different parameters and find their impacts on accuracy. The two main parameters in RF are the number of trees and classify criterion (Gini impurity or information gain).

So in our experiment, we range the number of tree from 3 to 100 and use two different criteria, using average word vectors because of RF's poor performance, and then draw the line graph shown in Fig. 4.

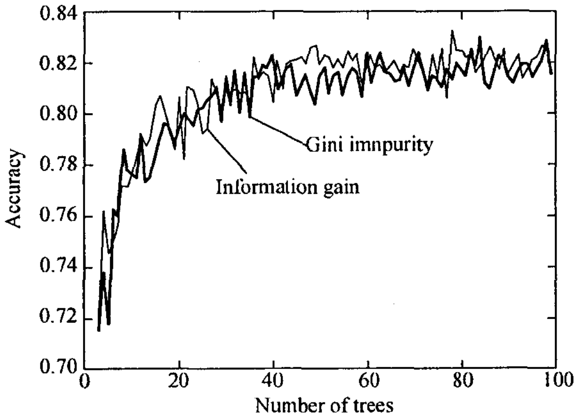

Fig. 4 Accuracy of RF

The trends of two lines are consist. If the number of trees is less than 20, the accuracy is 80% or lower. Otherwise, the accuracy keeps stable at 83%.

To sum up, the number of trees and classify criterion make little influences on our dataset, so in our experience, the parameters are chosen appropriately.

4 Conclusions

In this paper, two different ways of feature extraction were introduced. One is feature selection, which is simple but sometimes suffers from curse of dimensionality. The other is feature embedding, which maps text into a fixed length vector. We used the latter to do the feature extraction.

Doc2Vec performs poor in our Chinese sentiment classification, while our improved weighted average vector is better than other two methods. Also, SVM preforms better than other two algorithms in our experiment.

References

[1] Bengio Y, Ducharme R, Vincent P, *et al.* A Neural Probabilistic Language Model [J]. *Journal of Machine Learning Research*, 2003, **3**: 1137-1155.

[2] Mikolov T, Sutskever I, Chen K, *et al.* Distributed Representations of Words and Phrases and Their Compositionality [J]. Advances in Neural Information Processing Systems, 2013, **26**: 3111-3119.

[3] Mikolov T, Yih W T, Zweig G. Linguistic Regularities in Continuous Space Word Representations [C]. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013), Atlanta, USA, 2013: 746-751.

[4] Le Q V, Mikolov T. Distributed Representations of Sentences and Documents [J]. *arXiv e-print*, 2014: 1405.4053.

[5] Maas A L, Daly R E, Pham P T, *et al.* Learning Word Vectors for Sentiment Analysis [C]. ·Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Potland, Oregon, USA, 2011: 142-150.

[6] Mesnil G, Ranzato M A, Mikolov T, *et al.* Ensemble of Generative and Discriminative Techniques for Sentiment Analysis of Movie Reviews [J]. *arXiv e-print*, 2014: 1412.5335.

[7] garnettyige_1. Jingdong Products Reviews Dataset [OL]. [2016-01-10]. http://download. csdn. net/detail/linger2012liu/7758939,2014.

[8] Pak A, Paroubek P. Twitter as a Corpus for Sentiment Analysis and Opinion Mining [C]. Proceedings of the International Conference on Language Resources and Evaluation, Valletta, Malta, 2010: 1320-1326.

[9] QiuX P, Zhang Q, Huang X J. FudanNLP: a Toolkit for Chinese Natural Language Processing [C]. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Sofia, Bulgaria, 2013: 49-54.

[10] Řehůřek R, Sojka P. Software Framework for Topic Modelling with Large Corpora [C]. Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks, Valletta, Malta, 2010: 46-50.

[11] Pedregosa F, Varoquaux G, Gramfort A, *et al.* Scikit-Learn: Machine Learning in Python [J]. *Journal of Machine Learning Research*, 2011, **12**(10): 2825-2830.