Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 147 (2019) 361–368

Procedia Computer Science

www.elsevier.com/locate/procedia

2018 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI 2018

Sentiment Analysis of Weibo Comment Texts Based on Extended Vocabulary and Convolutional Neural Network

Xiaoyilei Yang[a] , Shuaijing Xu[b], Hao Wu[c]*, Rongfang Bie[d]

[a]*bnuyxle1@163.com, Beijing Normal University, Beijing 100875, China*
[c]*wuhao@bnu.edu.cn, Beijing Normal University, Beijing 100875, China*

Abstract

In the era of big data and Internet, social network platforms, blogs, and recommender systems generate thousands of subjective information every day. The emotional content of these information may be related to books, characters, commodities, activities and so on. Analyzing and mining subjective emotional information is conducive to personal decision making, enterprise reform, and government's public opinion regulation.

In this paper, based on Weibo comment texts, we use the network term and the wiki Chinese data set to expand the original vocabulary, train word embeddings and realize the sentence-level sentiment classification based on the convolution neural network. At the same time, an optimization method according to the statement length of pooling layer is put forward. The method is proved to be effective with high accuracy on our data set.

Keywords: Weibo sentiment analysis, convolution neural network, extended vocabulary, word2vec

* Corresponding author. Tel.: +86-010-5880-0446; fax: +86-010-5880-0446.
E-mail address: wuhao@bnu.edu.cn

10.1016/j.procs.2019.01.239

1. Introduction

Sentiment analysis is the practice of using natural language processing and text analysis and computational techniques to automatically identify, extract or classify subjective information from text. [1] Subjective information can be a film review, a book review, or even a public opinion on breaking news. Sentiment analysis mainly concentrates on opinions with sentimental polarities, negative or positive. [2] The task of sentiment analysis can be divided into three levels: document level, sentence level, and feature based approaches (aspect level). [3] In this paper, we use Weibo comment texts for sentence-level sentiment polarity classification.

Weibo has a wide range of user-base and real-time information that is intentionally or unintentionally, tangibly or intangibly affecting people's life, enterprise development, or even government actions. The virtuality of the Internet allows people to freely express their real opinions and emotions on Weibo. All these make the information flowing on Weibo become important and valuable for mining. But Weibo, as its large data base and its own style, needs to be created in a more targeted and efficient way.

2. Related work

Commonly used sentiment classification methods are lexicon-based methods and machine learning-based methods. Lexicon-based methods depend on emotional dictionary which is a list of words or phrases containing sentiment polarity information. [4] In 2016, Saif et al. [5] proposed the SentiCircles model, considering the co-occurrence patterns of words in different contexts to capture their semantics and update the strength and polarity of their pre-allocated semantic emotional vocabulary to get more appropriate emotional dictionary. This model performs better on Twitter text than SentiStrength model [6] which is the best before it.

Lexicon-based methods can achieve good effect by making fine-grained emotional judgment of the texts. However, when analyzing comment texts generated by the real-time web platform, due to the high popularity of network language, we cannot update the emotional dictionary in time to keep up with the popularity of the Internet, which makes it more difficult to recognize the emotion of new words. Jidong Li et al. [7] studied the neologism of Weibo and established an emotional dictionary applicable to Weibo. However, a sentence may contain both positive and negative words or it may not have a single emotive word but still express subjective feelings. In a certain context, people will use positive words to satirize or oppose. In these cases, the analysis process will be tedious while the accuracy is limited if only the lexicon-based methods are used.

Machine learning methods can be divided into supervised learning and unsupervised learning. [8] In the field of sentiment classification, supervised machine learning is mainly used. In the case of sufficient data quantity and diversity, this method can effectively avoid the problems mentioned above and the analysis steps are relatively simple compared to lexicon-based methods. Tingting Li et al. [9] improved the support vector machine method and conditional random field method, combined with a variety of feature combinations, making up for the lack of feature extraction in traditional machine learning methods.

In the age of big data, deep learning method, as a branch of machine learning method, is more convenient to operate under the support of massive data and powerful computing ability. In recent years, many researches related to natural language processing can be solved by using Recurrent Neural Network (RNN), but for the classification problem of emotion analysis, the analysis effect of Convolutional Neural Network (CNN) is not inferior to RNN. In addition, CNN has simpler structure and higher training efficiency. Yoon Kim [10] proposed Convolutional Neural Networks for Sentence Classification in 2014, which opens the door for using CNN in sentiment classification. The paper proves the rationality of the model by experimenting on English corpus. There are already some improved experiments based on CNN. Zhang et al. [11] did a study on CNN text classification at the character level in 2015. Huiping Cai et al. [12] combined word2vec with CNN and improved the accuracy rate by 5.04%, but the word vector model they trained was only based on the training text; Liang Bin et al. [13] introduced various attention mechanisms to CNN for emotional analysis. However, there is not much research on the improvement of CNN network structure, so the sentence classification model based on CNN still has room for improvement

In this paper, we try to improve the structure of CNN text classification model by using two kinds of pooling methods. In addition, lexicon is the smallest linguistic unit, even machine-learning based methods need to rely on sentiment lexicon in feature engineering. Proper use of well-designed lexicon will improve the performance of

sentiment analysis system. Thus, we attempt to expand the vocabulary to fit Weibo texts and treat it as the raw material of our network to help our model have a better comprehension of sentences.

3. Model

In this chapter, we introduce the model used for sentiment classification task, as shown in figure 1. The model mainly includes four parts: input layer, convolution layer, two kinds of pooling layer and their corresponding fully connected layers.

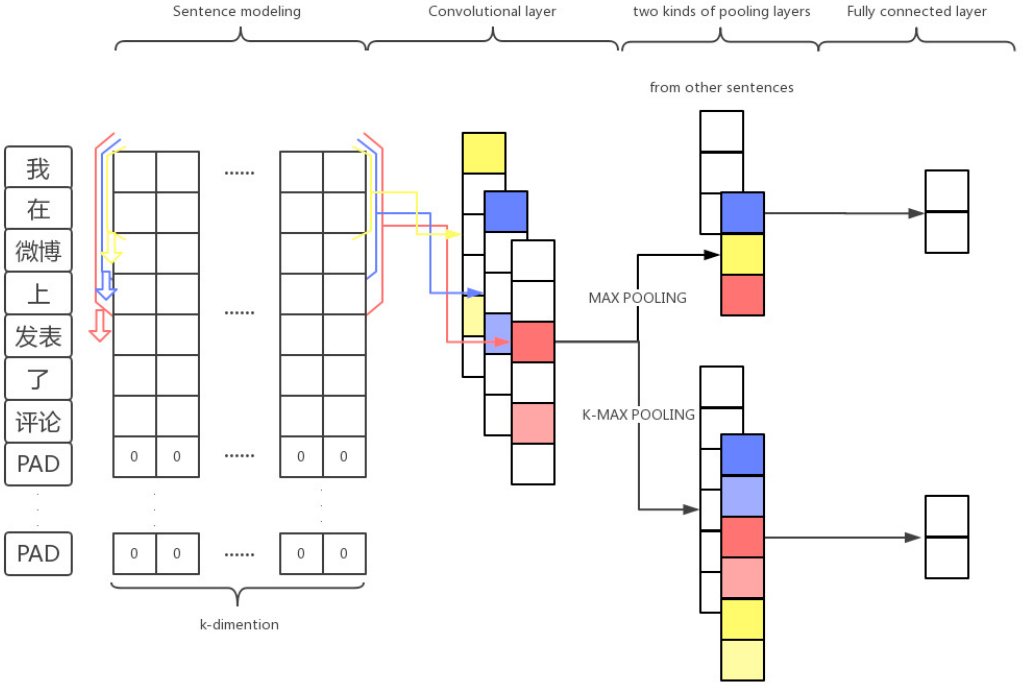

Fig. 1. Network structure of model

3.1. Model input

A computer's reading of text is not like reading of an image: the image itself is a multidimensional vector while the text is a string with different coding forms. The neural network requires input in the form of vector matrix, so it is very important to select the appropriate vector representation.

In this paper, we adopt word2vec as tools to generate word embeddings to represent words. Word2vec includes two models: CBOW and Skip-gram. [14] In this paper, Skip-gram model is adopted, because compared with CBOW model, Skip-gram model combined with negative sampling can achieve higher efficiency. From the paper [15], we can get this conclusion.

Corpus is composed of sentences, sentence is made up of words. Sentences as direct input to the network also need to be represented in vector form. Let S be the sentences set of the corpus, let $s_j \in S$ represent the j[th] sentence of the corpus. Let w_i represent the i[th] word's vector in s_j . Then we can define a sentence of length l as follows:

$$s_j = w_{1:l} = w_1 \oplus w_2 \ldots \oplus w_l, 0 < l \leq \max\left(length(s_j)\right), 0 < j \leq |S|$$

Where $\oplus$ is a joint mark.

The length of the above sentence model is not certain. In order to facilitate network input, we use the blank vector (n-dimensional zero vector) complementary strategy, and the sentence model after completion is described as follows:

$$s_j = w_{1:m} = w_1 \oplus w_2 \ldots \oplus w_m, 0 < l \leq m$$

Where $w_{l+1}, w_{l+2}, \ldots, w_m$ are n-dimensional zero vector. The value of m depends on the choice of pooling method.

3.2. CNN structure with improved pooling layer

The traditional convolutional neural network used for text classification includes convolutional layer, max pooling layer and fully connected layer.

A convolutional operation is applied to a window containing h words to generate a feature. The size of the convolutional window is h*n and the size of h may be different, while the size of n is fixed as the dimension of the word vector. Let filter $\omega \in \mathbb{R}^{h \times n}$, it applies to a convolutional window and generate a feature vector c_i. Thus, a convolutional operation can define as follows:

$$c_i = f(\omega \cdot w_{i:i+h-1} + b)$$

Where is the activation function, b is a bias.

We call a set of feature vectors feature map, which is represented as follows:

$$\boldsymbol{c} = [c_1, c_2, \dots, c_{n-h+1}]$$

Multiple features can be obtained by selecting different window sizes and using multiple convolution cores of different sizes.

The pooling layer is designed in two schemes and compared by experiment.

Max pooling is the way to extract a maximum value from each feature vector which is one-to-one corresponding to the filter. The point of this is to extract the most significant features and ignore the unnecessary features. On the other hand, max-pooling converts the array into a value, which reduces the number of parameters of the model to some extent, which is conducive to avoiding the problem of over-fitting.

When using this method, all sentences should be complemented by blank vectors (n-dimensional zero vectors) to the length of the longest sentences and then input into the network. At this point, the value of m is as follows:

$$\mathrm{m} = \max\left(\mathrm{length}\left(s_j\right)\right), 0 < j \le |S|$$

However, for longer texts, this method may ignore many features. Therefore, we design the second scheme, k-max pooling based on sentence length.

Different from max pooling which only takes one maximum value, k-max pooling takes top-k values of all feature values and keep them in their original order. The value of k is determined by the length of sentence. The highest frequency length is obtained by counting the sentence length (by words not by characters) of all texts.

$$\begin{cases} k = l/l_c \ , & l\%l_c = 0 \\ k = l/l_c + 1, & l\%l_c \neq 0 \end{cases}$$

When using this pooling method, all sentences should be complemented by blank vectors (n - dimensional zero vector) to k times l_c . At this point, the value of m is as follows:

$$\mathrm{m} = \mathrm{k} \cdot l_c$$

Features are summarized in fully connected layer. We obtain probability distribution of labels using Softmax function. In the end, we output a 2-dimensional vector for each sentence:

$$[\mathrm{pos}, \mathrm{neg}], pos, neg \in [0,1]$$

Where pos denotes the probability that the sentence is positive while neg denotes negative.

4. Experimental design and implementation

In this chapter, we mainly introduce the experiments we design for model implementation which include the data, the expansion of vocabulary, the establishment of word embedding and training details.

4.1. Data introduction

The data we used is from NLPCC2017 competition website. The training set contains 1,110,000 Weibo comment texts which have already been segmented and cleaned. The test set contains 5418 Weibo comment texts without segmenting and cleaning. The source data is categorized into six classes: others, likeness, sadness, disgust, anger and happiness corresponding to labels 0 to 5.

4.2. Expansion of vocabulary

In order to better understand the meaning of words and make up for the lack of contextual information caused by the short source data, we add the Chinese Wiki texts to Weibo texts. The processed wiki text and Weibo training data are putting together as a corpus to generate word embeddings.

Another advantage of doing this is that it increases the generalization ability of the model. The word list extracted from the Weibo training set is not enough to include all the words on Weibo platform. When the model is applied to other Weibo texts, the word embeddings can only be randomly initialized if there is no such word in our dictionary. Using wiki texts to expand the vocabulary can reduce the probability of this occurrence to use word embeddings which we have already trained and retain semantic information.

4.3. Generation of word embeddings

Firstly, we generate word embeddings by self-training so that we can convert words into vector form and map to high-dimensional space. Then we visualize the result through PCA dimensionality reduction. The visualization results reflect the aggregation of semantically similar words in vector space. According to this, we adjust the parameters to achieve the best aggregation effect, which means the word embeddings we generate can represent the correlation between words better. The four pictures in fig.2 show the aggregation of some words, and it can be found that these words cluster with apparently similar meanings. Table 1 shows the parameters applied to generate these vectors.

Self-trained Word2vec model has limit in text size and its training time is longer. We choose to use Gensim Word2vec tool to generate word embeddings during network training. The optimum parameters we get from self-training is applied in this step. We can use not only Weibo texts but also wiki texts as source training data. After the word embedding generation model is obtained, we generate the corresponding vector for each word in our dictionary by calling this model and store the vectors in a TXT file. At the start of our network training process, we upload this file along with the dictionary. By this way, we avoid calling word2vec model multiple times during network training, saving time and space.

Fig. 2. Words aggregation

Table 1. Training parameters of word embedding.

Parameter	Value
Model type	Skip-gram
Dimension of word embeddings	300
Size of window	5
Number of negative sampling noise words	20
Number of iterations	5

4.4. Training details

We adopt stochastic gradient descent (SGD) and Adam adaptive optimization algorithm during training. Dropout strategy and L2 regularization are also applied to improve the model generalization ability. We refer to the experimental conclusion of paper [17] in the process of parameter adjustment. Table 2 shows the setting of hyperparameters when the model performs best. Figure 3 shows the loss decrease situation during the training process displayed by TensorBoard.

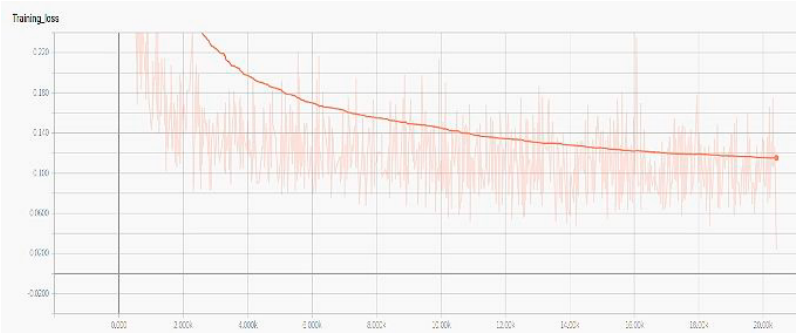

Fig. 3. Training loss

Table 2. Parameter settings of network.

Parameter	Column B (*t*)
Size of convolutional window	3,4,5
Number of convolutional windows	128
Learning rate	0.01
Dimension of word embeddings	300
Aviation function	tanh

5. Experimental results and discussion

In this chapter, the results of comparative experiments and integrated experiment are expounded and analyzed.

5.1. Max pooling vs k-max pooling based on sentence length

From Table 3, we can see that the accuracy improved by using k-max pooling method based on sentence length. The reason is when we use max-pooling method, on one hand there are too many blank vectors used to pad sentences causing imprecise feature capturing, on the other hand, it extracts less features from long sentences causing inaccuracy judgement.

Table 3. Experimental results of two pooling method

	Max pooling	k-max pooling
Accuracy	94.74%	95.09%
Precise	94.11%	94.27%
Recall	94.47%	94.66%
F1-measure	94.29%	94.46%
Accuracy on test data	96.90%	97.06%

5.2. Whether to use the custom vocabulary to re-segmentation

Table 4. Experimental results of different segment methods

	Using original segmented data	Using re-segmented data (define custom vocabulary)
Accuracy	94.36%	94.74%
Precise	94.07%	94.11%
Recall	93.58%	94.47%
F1-measure	93.82%	94.29%
Accuracy on test data	95.83%	96.90%

It can be seen that the result of using re-segmented data is better. The reason is when we do not add cyberwords to customize vocabulary, the segmenting tool will split some words into characters because it does not recognize them. Through this comparison experiment, we can see the importance of accurate word segmentation according to the characteristics of data sets.

5.3. Whether to expand vocabulary

Table 5. Experimental results of whether to expand vocabulary

	Only using Weibo texts	Using wiki data to expand Weibo data
Size of vocabulary	283633	537166
Accuracy	94.27%	94.74%
Precise	93.41%	94.11%
Recall	94.05%	94.47%
F1-measure	93.73%	94.29%
Accuracy on test data	95.37%	96.90%

We can see that the result is better when we expand the vocabulary. One reason is after expansion, we get more contextual information, which makes word's meaning more accurate, the other is we reduce the probability of confronting unknown words and improve the generalization ability. Through this comparison experiment, it is proved that the method of improving the model with extended vocabulary is effective.

5.4. Result of integrated experiment

Table 6. Experimental scheme and result

Scheme		Result	
Whether to re-segmented	YES	Accuracy	95.09%
Whether to expand vocabulary	YES	Precise	94.27%
Word embedding generating model	Skip-gram	Recall	94.66%
Pooling method	k-max pooling based on sentence length	F-measure	94.46%
Training parameters of word embedding	Shown in Table 1	Accuracy on test data	97.06%
Training parameters of network	Shown in Table 2		

The experimental results have high accuracy, which proves the validity of the model presented in this paper.

6. Conclusion

The sentiment analysis of Weibo comments is a meaningful and challenging task. Guided by the task, we introduce the model based on extended vocabulary and CNN and prove the validity of our model by a series of experiments. In

this paper, three improvements are made on the basis of the basic text classification model of CNN: By customizing cyberwords during segmentation, we maintain more information of Weibo texts; By expanding the vocabulary with wiki data, we increase the correlation between words; By using k-max pooling method based on length of sentence, we capture more features. All these improvements contribute to the classification accuracy.

In the future, efforts will be made to improve the model generalization ability and try to solve the problem of data skew.

Acknowledgement

This research is sponsored by National Natural Science Foundation of China (No.61371185, 61401029,61472044,61472403,61571049£¬61601033,11401028) and the Fundamental Research Funds for the Central Universities (No.2014KJJCB32, 2013NT57) and by SRF for ROCS, SEM. and China Postdoctoral Science Foundation Funded Project (No.2016M590337).

References

[1] Hussein E D M. A survey on sentiment analysis challenges[J]. Journal of King Saud University - Engineering Sciences, 2016.

[2] Liu B. Sentiment Analysis and Opinion Mining[C]// Morgan & Claypool, 2011:167.

[3] Liu B. Sentiment analysis and subjectivity[J]. 2010, 30(36):152-153.

[4] Bagheri H, Islam M J. Sentiment analysis of twitter data[J]. 2017.

[5] Saif H, He Y, Fernandez M, et al. Contextual semantics for sentiment analysis of Twitter[J]. Information Processing & Management, 2016, 52(1):5-19.

[6] Thelwall M, Buckley K, Paltoglou G. Sentiment strength detection for the social web[J]. Journal of the Association for Information Science & Technology, 2012, 63(1):163–173.

[7] Li Jidong, Wang Yizhi. An emotional analysis of Chinese microblogs based on extended dictionary and semantic rules [J]. Computer and modernization, 2018(2).

[8] Yang S C, Jia L X. Comparison Between Supervised Learning and Unsupervised Learning in Neural Networks[J]. Journal of Xuzhou Institute of Architectural Technology, 2006.

[9] Li Tingting, Ji Donghong. Emotional analysis of Weibo based on the combination of SVM and CRF [J]. Computer application research, 2015, 32(4):978-981.

[10] Kim Y. Convolutional Neural Networks for Sentence Classification[J]. Eprint Arxiv, 2014.

[11] Zhang X, Zhao J, Lecun Y. Character-level Convolutional Networks for Text Classification[J]. 2015:649-657.

[12] Cai Huiping, Wang Lidan, Duan Shukai. Emotion classification model based on word embedding and CNN [J]. Journal of research in computer application, 2016, 33(10):2902-2905.

[13] Liang Bin, Liu Quan, Xu Jin, et al. Objective affective analysis based on multi-attention convolutional neural network [J]. Computer research and development, 2017, 54(8):1724-1735.

[14] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.

[15] Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and their Compositionality[J]. 2013, 26:3111-3119.

[16] Metin S K, Karaoğlan B. STOP WORD DETECTION AS A BINARY CLASSIFICATION PROBLEM[J]. 2017, 18(2):1-1.

[17] Zhang Y, Wallace B. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification[J]. Computer Science, 2015.