Construction of Microblog-Specific Chinese Sentiment Lexicon Based on Representation Learning

Li Kong[1,2], Chuanyi Li[1,2(✉)], Jidong Ge[1,2(✉)], Yufan Yang[1,2], Feifei Zhang[1,2], and Bin Luo[1,2]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
kl_nju@126.com, yyfnju@163.com, 257561343@qq.com, luobin@nju.edu.cn

[2] Software Institute, Nanjing University, Nanjing, China
lcynju@126.com, gjdnju@163.com

Abstract. Sentiment analysis is a research hotspot in Nature Language Processing, and high-quality sentiment lexicon plays an important part in sentiment analysis. In this paper, we explore an approach to build a microblog-specific Chinese sentiment lexicon from massive microblog data. In feature learning, in order to enhance the quality of word embedding, we build a neural architecture to train a sentiment-aware word embedding by integrating three kinds of knowledge, including the context words and their composing characters, the polarity of sentences and the polarity of labeled words. Experiments conducted on several public datasets show that in both unsupervised and supervised microblog sentiment classification, the lexicon generated by our approach achieves the state-of-the-art performance compared to several existing Chinese sentiment lexicons and our feature learning method successfully catches both semantics and sentiment information.

Keywords: Sentiment lexicon · Representation learning · Microblog

1 Introduction

Sentiment analysis, also known as opinion mining, is a branch of Nature Language Processing (NLP). Its purpose is to help users acquire, organize and analyze relevant information, and analyze, process, induce and reason the subjective text [1]. Sentiment lexicon is an important component in sentiment analysis systems as it provides rich sentiment information. Sentiment lexicon consists of a list of subjective words or phrases, each of which is assigned with a sentiment polarity score. There have been some universal sentiment lexicons such as General Inquirer (GI), WordNet, HowNet, etc. However, compared to English sentiment lexicons, quality of existing Chinese sentiment lexicons is relatively low. That makes the automatic construction of domain-specific Chinese sentiment lexicon a meaningful and challenging task in the field of sentiment analysis.

Unlike English, texts of Chinese, Japanese and some other languages do not have obvious space between single words. Instead, a word of these languages consists of one

X. Geng and B.-H. Kang (Eds.): PRICAI 2018, LNAI 11012, pp. 204–216, 2018.
https://doi.org/10.1007/978-3-319-97304-3_16

or more characters and the semantics of each composing character is often related to the semantics of the word [2, 3]. For instance, '摇篮' (cradle) consists of '摇' (sway) and '篮' (basket). Considering the particularity of Chinese, researchers start from granularity finer than words and work on the level of characters, even structural parts to obtain better representation. Final representation for each word is the addition of word embedding and character embeddings, even structural part embeddings.

Microblog is a type of social media popular among young people where they can freely express emotional opinions and discuss various topics. Microblogs contain more free writing style which corresponds to people's expression custom, and richer emotions than traditional media. Various emoticons, such as and , are provided and widely used to directly convey positive or negative emotion, which can be seen as the indicator or label of the polarity of a microblog [4–6, 19]. With the help of microblog API, it is very easy to collect millions of posts.

In this paper, we utilize the Sina microblog (Weibo[1]) data as our corpus and attempt to learn a sentiment lexicon from massive collection of microblogs with emoticons indicating the polarity of sentences. We use the framework of [7] and make some modifications to realize a representation learning approach, as is shown in Fig. 1. Since a Chinese word is composed of one or more characters, it is helpful to take its composing characters into consideration when learning the semantic of the word. In addition, since we focus on sentiment polarity, known polarity of sentences and expanded seed words (labeled words) are used to optimize the feature learning of words. Based on these, we learn the continuous representation of Chinese words from three aspects: the context words and their composing characters, the polarity of sentences and the polarity of labeled words. The representation of each word in the corpus is then used as the features in a classifier to predict the sentiment polarity.

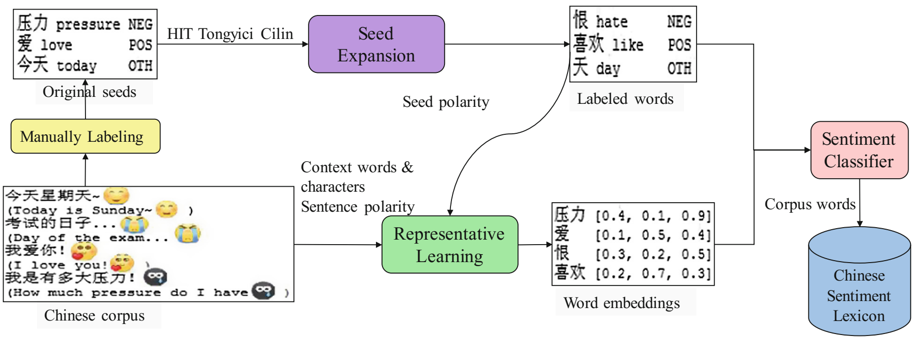


Fig. 1. The representation learning approach for building microblog-specific Chinese sentiment lexicon.

[1] https://weibo.com/.

The main contributions of our work are as follows:

- We combine context information and sentiment information to learn the embedding of both Chinese words and composing characters.
- We construct a microblog-specific Chinese sentiment lexicon using a sentiment classification framework.
- Our lexicon obtains the state-of-the-art performance compared to several existing sentiment lexicons, when used for microblog sentiment classification.

The rest of the paper is organized as follows. Section 2 introduces the related work on sentiment lexicon building. Our approach for microblog Chinese sentiment lexicon building is presented in Sect. 3. In Sect. 4 we provide the experimental results. Section 5 concludes the paper and presents future work.

2 Related Work

In this section, we will introduce two main approaches for sentiment lexicons construction: dictionary-based methods and corpus-based methods.

Dictionary-based methods rely on a seed sentiment dictionary which is usually small and a semantic knowledge base, such as WordNet[2] and Urban Dictionary[3], to explore the relationship between words, including synonymy, antonymy, upper and lower relationship etc. Hu and Liu accept a word as positive if it is a synonym of a positive seed word or an antonym of a negative seed word with the help of WordNet. The expansion of negative words is similar [8]. Heerschop et al. realize the emotion propagation of the seed word set utilizing the semantic relationship in WordNet [9]. A random walk-based algorithm is proposed in [10] to rank the word polarities in WordNet. They assume that the occurrence of the words in the glosses can express polarity properties.

Compared to semantic knowledge base, corpora are easier to get. Manually-labeled seed words and patterns in unlabeled corpora are used to induce domain-specific lexicons. Corpus-based methods can be further classified into two categories: conjunction methods and co-occurrence methods. Conjunction methods utilize conjunctions to judge the relationship of words around. For example, words around 'and' are probable to have the same polarity while words around 'but' may convey opposite polarities [11, 12]. A double propagation (DP) method is proposed in [13], which uses both sentiment and target relation and various connectives to extract sentiment words. Co-occurrence methods are based on the assumption that if two words often appear in the same sentence or document, they tend to have the same polarity or if a word often appear in positive (negative) sentences, it is more likely to express positive (negative) emotion. The degree of co-occurrence is usually measured by Point Mutual Information (PMI) [14]. Some researchers put forward several variants of PMI, such as positive PMI (PPMI) [15] and second order co-occurrence PMI (SOC-PMI) [16].

[2] https://wordnet.princeton.edu/.

[3] https://www.urbandictionary.com/.

The latest corpus-based approaches normally utilize the up-to-date machine learning models, such as neural networks, to learn a sentiment-aware distributed representation of words. In [17] sentiment attribute scores of each word are obtained through a neural network model based on tweets emoticons. Tang et al. modify the skip-gram model [18] and integrate the sentiment information of text into its hybrid loss function to learn sentiment-specific embedding. Then a softmax classifier is trained to predict the polarity according to a word embedding [7]. Wang et al. consider information on both word-level and document-level to learn word embedding and a logistic regression model is used later as the classifier [19].

Apart from techniques above, construction of Chinese sentiment lexicon also utilizes particularity of Chinese words and considers information of characters and structural parts to obtain better representation. Convolutional auto-encoder (convAE) is utilized in [20] to learn bitmap of characters to obtain semantics information. Since one character may have several meanings, Chen et al. learn one representation for each semantics [2]. Final representation for each word here is the addition of word embedding and character embeddings. It is worth noting that different characters make different contributions to the semantics of the word [21]. Considering from the finer granularity, radical is found to play an important part in the semantics of a character, e.g. '吃' (meal) has close connection with its radical '口' (mouth) [22]. Some researchers also make the assumption that characters with the same radicals have similar semantics and usage [23]. Yu et al. extend the work of [2] and jointly learn the embeddings of words, characters, and subcharacter components [3].

3 The Proposed Approach

In this section, we will introduce our approach for building a microblog-specific Chinese sentiment lexicon in detail. We follow [7] to form a classification framework. The major difference is the way we learn the representations of words. When modifying the skip-gram model to predict word embeddings according to the context, we also utilize information of characters of the context words. Besides considering sentiment polarity of each sentence in the corpus, we also utilize sentiment polarity of labeled words to guide the learning of word embeddings.

3.1 Microblog-Specific Word Embedding

Our feature learning method (*NJUWE*) is comprised of three parts: the context words and their composing characters, the polarity of sentences and the polarity of labeled words, as is shown in Fig. 2. Given a target word w_i, we will utilize its embedding x_i to predict its context words w_{i-1}, w_{i+1} and their composing characters c_{i-1}, c_{i+1}. Obviously the window size is 3 in this example and we can set it to 5 or others. As w_i is contained in the sentence s_j whose representation is se_j, x_i is also used to predict $spol_j$, the polarity of s_j. If w_i is a seed word, its polarity pol_i will also be predicted.

Context Information: We follow [2] to build a character-enhanced word embedding model as the first part of our feature learning method. We will realize this part based on

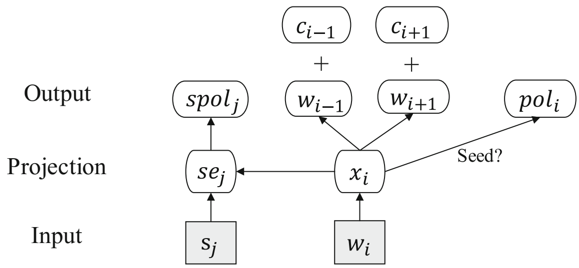


Fig. 2. Model of *NJUWE*.

skip-gram model to better integrate with the other parts. Skip-gram aims at predicting the context words given the target word. The optimization target is to maximize the average log probability below:

$$f_1 = \sum\nolimits_{w \in T} \log p(context(w)|w) \tag{1}$$

Where T is the occurrence of each word in the corpus, w is the target word and *context(w)* is the corresponding context words in the window. In order to accelerate the training procedure, we adopt Negative Sampling [24]. The objective is to distinguish the target word from the noise distribution using logistic regression. In the ideal situation, the probability of predicting the target word as w should be maximized and the probability of predicting the target word as other words should be minimized. We represent a context word x_j with both character embeddings and word embedding to get its embedding $\boldsymbol{x_j}$. The formula can be then written as:

$$\begin{aligned} f_1 = \sum_{w \in T} \sum_{w' \in context(w)} \sum_{u \in \{w\} \cup NEG\{w\}} & [L_{(u)}^{w'} * log(\sigma(\boldsymbol{x}_{\boldsymbol{w}}^T * \theta^u)) \\ & + \left(1 - L_{(u)}^{w'}\right) * log\left(1 - \sigma\left(\boldsymbol{x}_{\boldsymbol{w}}^{\boldsymbol{T}} * \boldsymbol{\theta}^u\right)\right)] \end{aligned} \tag{2}$$

$$\boldsymbol{x_w} = \frac{1}{2} * \left(\boldsymbol{e_w} + \frac{1}{m} \sum\nolimits_{k=1}^{m} \boldsymbol{c_k}\right) \tag{3}$$

Where $\boldsymbol{e_w}$ is the word embedding of w, $\boldsymbol{c_k}$ is the embedding of the k-th character in w, m is the number of characters in w and $L_{(u)}^{w'}$ is an indicator function which equals 1 if the sample u is predicted as w', otherwise 0. $\boldsymbol{\theta}^u$ is the auxiliary vector of u.

Sentence Sentiment Information: In this part, each sentence s is represented as a vector $\boldsymbol{s}$ which is the average of addition of embeddings of words in it. Since we deal with microblog sentences with emoticons indicating their sentiment polarities, we can utilize a sentence representation to predict the sentence polarity, which is actually a golden standard. The objective of the second part is to maximize the log sentiment probability below:

$$f_2 = \sum\nolimits_{s \in S} logp(spol \mid s) \tag{4}$$

Where S is the occurrence of each sentence in the corpus and $spol$ is the polarity of a sentence. An optimization borrowing Negative Sample is used here. We treat the predicted polarity that equals the golden standard as the positive sample and otherwise negative sample. Then the formula can be written as:

$$\begin{aligned} f_2 = \sum_{s \in S} \sum_{p\_s \in \{spol\} \cup NEG\{spol\}} & [L_{(p\_s)}^{spol} * log\left(\sigma\left(\boldsymbol{s}^{\boldsymbol{T}} * \boldsymbol{\theta}^{p\_s}\right)\right) + \\ & \left(1 - L_{(p\_s)}^{spol}\right) * log\left(1 - \sigma\left(\boldsymbol{s}^{\boldsymbol{T}} * \boldsymbol{\theta}^{p\_s}\right)\right)] \end{aligned} \tag{5}$$

$$\boldsymbol{s} = \frac{1}{n} * \sum\nolimits_{w \in s} \boldsymbol{w} \tag{6}$$

Where n is the number of words in a sentence and p_s is the predicted polarity of the sentence. Other parameters are similar to the description above.

Word Sentiment Information: Inspired by the discovery that sentence sentiment information greatly improves the quality of representation learning [7], we try to utilize sentiment information of labeled words to get better performance. We hope to predict the polarity of a word, which is actually known after seed expansion, with the corresponding word representation. Optimization similar to Negative Sampling is also used here. The objective of this part is to maximize the log sentiment probability:

$$\begin{aligned} f_3 = \sum_{w \in D} logp(pol \mid x_w) = \sum_{w \in D} \sum_{pol \in \{p\_w\} \cup NEG\{p\_w\}} & [L_{(p\_w)}^{pol} * \\ log\left(\sigma\left(\boldsymbol{x}_{\boldsymbol{w}}^{\boldsymbol{T}} * \boldsymbol{\theta}^{p\_w}\right)\right) + \left(1 - L_{(p\_w)}^{pol}\right) * & log\left(1 - \sigma\left(\boldsymbol{x}_{\boldsymbol{w}}^{\boldsymbol{T}} * \boldsymbol{\theta}^{p\_w}\right)\right)] \end{aligned} \tag{7}$$

Where D is the occurrence of each labeled word, pol is the polarity of the word, p_w is the golden standard and other parameters are similar to parameters in Eq. (5).

As a whole, we try to maximize the linear combination of the three parts:

$$f = \alpha_1 * f_1 + \alpha_2 * f_2 + \alpha_3 * f_3 \tag{8}$$

Where the parameters α_1, α_2 and α_3 weigh the different parts and the value is limited to 0 or 1. We train our neural model with stochastic gradient ascent. The window size is 5, the embedding length is 200 and the initial learning rate is 0.025 in our experiment.

3.2 Lexicon Construction

In this section, we follow the method introduced in [7] to build a machine learning based classifier to predict the sentiment polarity of a word in the corpus according to the word representation learned in Sect. 3.1. The feature vector of the word is the input of the classifier and word sentiment polarity is the output. Words of the same polarity are grouped together and a sentiment lexicon is formed.

Firstly, we collect and sort all the words in the Sina microblog corpus by Jieba[4] tokenizer. We select the most frequent 700 words and arrange 5 native speakers to annotate the sentiment polarity (positive, negative or others). Majority voting method is used when they have disagreement about the annotation. We get 135 positive words, 66 negative words and 299 most frequent other words as seeds.

Then we expand seed words in the same way as [7] utilizing HIT IR-Lab Tongyici Cilin[5]. After this step, we obtain 1175 positive words, 897 negative words and 1176 other words. These serve as labeled data.

A Random Forest (RF) classifier is trained in the next step. The input is the embeddings of each labeled word and the output is the sentiment score. The score of positive words, negative words and other words is 1, 2 and 3 respectively.

Finally, each of the words sorted in the first step except the labeled words, is mapped into a vector through *NJUWE* and the vector is fed to the RF classifier to predict the sentiment polarity. Positive and negative words obtained in this step are added to the sentiment lexicon together with labeled words.

4 Experimental Evaluation

We will evaluate the quality of our approach in this section. Since sentiment lexicon is built with the purpose of guiding sentiment classification, we will first evaluate the result of sentence-level classification utilizing lexicon information. The better the effect of the classification is, the higher the quality of the lexicon is. Next, we will evaluate the quality of our feature learning approach. Specifically, we will carry on word similarity computation and word-level sentiment classification which is based on word embeddings obtained from different feature learning approaches.

4.1 Corpus and Settings

We use Weibo corpus offered by [4] to train our model and collect our lexicon. This corpus contains about 5.88 million Chinese microblogs with emoticons from October 1, 2011 to July 31, 2012. We make the assumption that if a microblog only contains positive emoticon(s), it conveys positive emotion, so does a microblog with only negative emoticon(s). So we respectively pick one million sentences with only positive emoticons and another one million with only negative emoticons. Word segmentation using Jieba tokenizer and stop words removing are done in advance. Sentences with only stop words or emoticons or without Chinese words are ignored.

4 https://www.python.org/pypi/jieba/.

5 https://pypi.ltp-cloud.com.

4.2 Evaluation of Lexicon

We will evaluate the effect of sentiment lexicon when it is used to guide sentence-level sentiment classification, by both supervised and unsupervised methods. Accuracy and Macro-averaged F-measure (Macro-F1) are calculated to show the result of classification and measure the usability of sentiment lexicon in classification.

Unsupervised Sentiment Classification. For unsupervised sentiment classification, we calculate the sum of scores of all sentiment words in a sentence with the help of the sentiment lexicon. If the sum score is greater than 0, then the sentence will be judged as positive, otherwise negative.

Supervised Sentiment Classification. To evaluate the effect of the sentiment lexicon in supervised sentiment classification, besides counting the sum of scores of sentiment words, we also combine some pre-defined features. Considering characteristics of Sina microblogs, we follow [25] to extract some features as follows: (1) negative words; (2) degree words; (3) punctuation; (4) emoticon. Each sentence is represented as a feature vector made up of the extracted features. Details about the features are present in [25]. Then the feature vector is put into a traditional machine learning based classifier (here we use a RF model) to predict the sentiment polarity of the sentence.

Table 1. Statistic of sentence-level classification datasets.

Dataset	Training set		Test set	
	Positive	Negative	Positive	Negative
SC2012	407	1757	41	229
SC2013	4951	4991	1883	2660
SC2014	2078	1943	1037	892

Setups. For supervised classification, we utilize data offered by microblog sentence-level sentiment classification task in NLPCC2012–2014 (*SC2012–2014*) to train a RF classifier respectively. For simplicity, we group fine-grained sentiment labels (e.g., "happiness", "like", "disgust", "anger") into two classes: positive and negative. Since training data offered by the tasks is too little compared to the respective test data, we use test data in NLPCC tasks as our training data to train the RF model and check the classification performance on the given training data, which is our test data. We apply the RF model to predict our test data in which polarities are offered as the golden standard. For unsupervised classification, we use sentences in the original training data in *SC2012–2014*. The distribution of positive and negative sentences in the three data sets is present in Table 1. Notice that for each data set, the training set and test set have been exchanged here.

We compare the performance of our lexicon (*NJUSD*) with several existing Chinese sentiment lexicon[6], including *NTUSD* from National Taiwan University, the

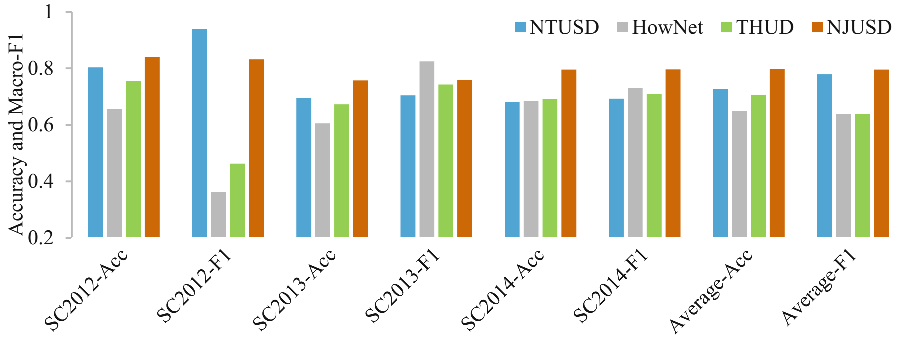
6 https://download.csdn.net/download/yunliangshen/9945179.

Table 2. Statistic of sentiment lexicons.

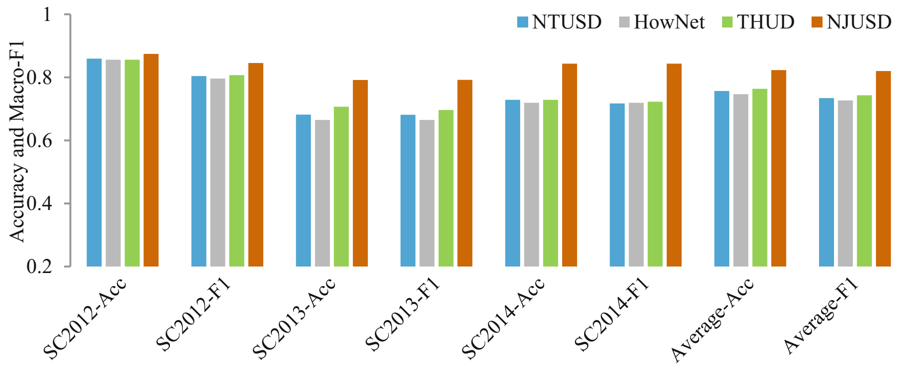
Lexicon	NJUSD	NTUSD	HowNet	THUD
Positive	4183	2812	836	5568
Negative	5087	8278	1254	4469

Chinese part of *HowNet* and Chinese Derogatory Dictionary from Tsinghua University (*THUD*). Statistics of these lexicons are shown in Table 2.

Results. Accuracy and Macro-F1 of classifying microblog sentences using different lexicons are present in Fig. 3. As we can see, both unsupervised and supervised classification tasks guided by *NJUSD* achieve the best performance. Only considering sentiment words, our lexicon yields 7.14% improvement on Accuracy and 1.02% improvement on Macro-F1 on average. By utilizing other features, our lexicon yields 7.69% improvement on Accuracy and 5.11% improvement on Macro-F1 on average.



(a) Unsupervised method



(b) Supervised method

Fig. 3. Accuracy and Macro-F1 on sentiment classification datasets using different lexicons.

That proves the availability and superiority of *NJUSD* in sentence-level sentiment classification. Performances of most lexicons are improved by adding pre-defined features, which agrees with the results in [7, 19].

4.3 Evaluation of Feature Learning

In order to demonstrate the effectiveness and advantage of our feature learning approach, we test the performances of our method and some existing methods when: (1) they are used in words semantic similarity computation; (2) they are used in word-level sentiment classification. In addition, we will see the influence of each composing part in Eq. (8), by ignoring one part at a time and testing the performance of the corresponding performance in word-level sentiment classification.

We use several state-of-the-art word embedding algorithms for comparison, including *CBOW*, *skip-gram* [18], *CWE* [2] and *JWE* [3]. For justice, we implement the last two algorithms based on skip-gram model and accelerate with Negative Sampling, the same as our approach.

Semantic Similarity. In this task, words are represented as vectors through a word embedding method. For a word pair with similar semantics, the similarity score, which is computed as the cosine similarity of word embeddings of the two words, should be high and vice versa. We compute the Spearman's rank correlation coefficient ρ_s [26] between the human-labeled scores and similarity scores computed by embeddings. If ρ_s is close to 1, it means there is a positive correlation the two compared data set. That is, the word embedding properly expresses the word's semantics.

The widely used data sets WordSim-240 and WordSim-297 are selected for word semantics similarity evaluation, which contain 240 and 297 Chinese word pairs respectively. A human-labeled similarity score, ranging from 1 to 10, is assigned to each pair. We remove a word pair if not both words are in the corpus or if the two words are the same. After that, we get 221 word pairs from WordSim-240 and 228 word pairs from WordSim-297.

ρ_s of each data set between given similarity scores and calculated scores are shown in Table 3. From the results, we can see that ρ_s between the human-labeled similarity score and the score got by computing the embeddings trained by *NJUWE* is the highest compared to the state-of-the-art methods. It yields 1.32% improvement on WordSim-240 and 2.33% improvement on WordSim-297. That means that our word embedding method can catch a word's semantics. Our Spearman correlation is smaller than that in [3] when the same word embedding method is applied with the same settings. It arises from our informal corpus and the different word segmentation tools. However, it is still passable.

Table 3. Results of word similarity evaluation.

Dataset	CBOW	Skip-gram	CWE	JWE	NJUWE
WordSim-240	0.4224	0.4443	0.4607	0.4276	**0.4739**
WordSim-297	0.3721	0.4194	0.4099	0.4015	**0.4427**

Sentiment Classification. In this task, words embeddings obtained from different feature learning methods are fed to a RF sentiment classifier to predict the polarity of the word. That requires the feature learning algorithm should not only catch the semantics, but also the sentiment information of a word.

We make the assumption that words appear in all three referential lexicons mentioned in Sect. 4.2 and have the same polarity are correctly labeled with sentiment polarity. We extract 900 such words, in which 450 are positive and 450 are negative. We randomly choose 300 positive and 300 negative words to serve as the training data. The rest of the words are the test data.

Since polarities of these words are known to us, accuracy and Macro-F1 can be calculated to present the performance, as is shown in Table 4. The results show that given a word, *NJUWE* can judge whether it is positive or negative more accurately than other methods. It yields 6.90% improvement on Accuracy and 7.71% improvement on Macro-F1 compared to *CWE*, which achieves the best performance among other methods. It is because our method considers not only the words' context, but also the sentiment of the sentences the words make up.

Table 4. Results of word sentiment classification.

Metric	CBOW	Skip-gram	CWE	JWE	NJUWE
Accuracy	0.5898	0.6289	0.6351	0.5959	**0.7041**
Macro-F1	0.5817	0.6263	0.6332	0.5619	**0.7043**

Influence of Composing Parts. In this part, we will see how the three composing parts in Eq. (8) influence the performance in word-level sentiment classification. Each time a composing part is ignored in training the word embeddings. Then the obtained embeddings are used as features to conduct word-level sentiment classification. Data set and setting are the same as above. Accuracy and Macro-F1 are present in Table 5. *NJUWE-P$_i$* means the *i*-th part is not considered in *NJUWE*. We can see that we get the best result if we consider all the three aspects, which proves the meaning of combination of different information in feature learning. However, our experiment shows that if we do not consider the context of words, the classification result will become much worse and if we do not consider the sentence polarity, the classification performance will not decrease that much, which is discordant with the conclusion in [7] that even if the context information is ignored, the performance is still remarkable only utilizing sentiment information. We think it may be because the language we deal with is different and Chinese character information greatly improves the positive influence of word context in feature learning.

Table 5. Results of word sentiment classification ignoring each part.

Metric	NJUWE	NJUWE-P$_1$	NJUWE-P$_2$	NJUWE-P$_3$
Accuracy	**0.7041**	0.5395	0.6562	0.6820
Macro-F1	**0.7043**	0.4978	0.6463	0.6785

5 Conclusion

In this paper, we propose the construction of a microblog-specific Chinese sentiment lexicon. We combine context information, sentence-level sentiment polarity and word-level sentiment polarity to learn features of words in the corpus and classify the words with a RF classifier. Experimental results demonstrate the worth of our sentiment lexicon in helping predict sentiment polarity and the benefits of our feature learning algorithm in learning semantic representations.

In the future, we will try more flexible weights to each of the three aspects in feature learning to optimize the result. We also plan to take more aspects, like user information, time information and domain information into consideration to build a more accurate and flexible sentiment lexicon. We hope to realize customized sentiment analysis in our work afterwards.

Acknowledgment. This work was supported by the National Key R&D Program of China (2016YFC0800803).

References

1. Liu, B.: Sentiment Analysis and Opinion Mining. University of Illinois at Chicago, Chicago (2012)
2. Chen, X., Xu, L., Liu, Z.: Joint learning of character and word embeddings. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, pp. 1236–1242 (2015)
3. Yu, J., Jian, X., Xin, H.: Joint embeddings of chinese words, characters, and fine-grained subcharacter components. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 286–291 (2017)
4. Feng, S., Song, K., Wang, D.: A word-emoticon mutual reinforcement ranking model for building sentiment lexicon from massive collection of microblogs. World Wide Web **18**, 949–967 (2015)
5. Wu, F., Huang, Y., Song, Y.: Towards building a high-quality microblog-specific Chinese sentiment lexicon. Decis. Support Syst. **87**, 39–49 (2016)
6. Tan, J., Xu, M., Shang, L., Jia, X.: Sentiment analysis for images on microblogging by integrating textual information with multiple kernel learning. In: Booth, R., Zhang, M.-L. (eds.) PRICAI 2016. LNCS, vol. 9810, pp. 496–506. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42911-3_41
7. Tang, D., Wei, F., Qin, B.: Building large-scale Twitter-specific sentiment lexicon: a representation learning approach. In: Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers, pp. 172–182 (2014)
8. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177 (2004)
9. Heerschop, B., Hogenboom, A., Frasincar, F.: Sentiment lexicon creation from lexical resources. In: Abramowicz, W. (ed.) BIS 2011. LNBIP, vol. 87, pp. 185–196. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21863-7_16

10. Esuli, A., Sebastiani, F.: PageRanking WordNet synsets: an application to opinion mining. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, ACL, pp. 424–431 (2010)
11. Hatzivassiloglou, V., McKeown, K.: Predicting the semantic orientation of adjectives. In: Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics, pp. 174–181 (1997)
12. Kanayama, H., Nasukawa, T.: Fully automatic lexicon expansion for domain-oriented sentiment analysis. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 355–363 (2006)
13. Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. Comput. Linguist. **37**(1), 9–27 (2011)
14. Turney, P.: Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 417–424 (2002)
15. Hamilton, W., Clark, K., Leskovec, J.: Inducing domain-specific sentiment lexicons from unlabeled corpora. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 595–605 (2016)
16. Islam, M., Inkpen, D.: Second order co-occurrence PMI for determining the semantic similarity of words. In: Language Resources and Evaluation, pp. 1033–1038 (2006)
17. Vo, D., Zhang, Y.: Don't count, predict! An automatic approach to learning sentiment lexicons for short text. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 219–224 (2016)
18. Mikolov, T., Chen, K., Corrado, G.: Efficient estimation of word representations in vector space. http://arxiv.org/abs/1309.4168 (2013)
19. Wang, L., Xia, R.: Sentiment lexicon construction with representation learning based on hierarchical sentiment supervision. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 513–521 (2017)
20. Su, T., Lee, H.: Learning Chinese word representations from glyphs of characters. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 264–273 (2017)
21. Xu, J., Liu, J., Zhang, L.: Improve Chinese word embeddings by exploiting internal structure. In: Proceedings of NAACL-HLT, pp. 1041–1050 (2016)
22. Yin, R., Wang, Q., Liu, R.: Multi-granularity Chinese word embedding. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 981–986 (2016)
23. Sun, Y., Lin, L., Yang, N., Ji, Z., Wang, X.: Radical-enhanced Chinese character embedding. In: Loo, C.K., Yap, K.S., Wong, K.W., Teoh, A., Huang, K. (eds.) ICONIP 2014. LNCS, vol. 8835, pp. 279–286. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12640-1_34
24. Mikolov, T., Sutskever, L., Chen, K.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
25. Mohammad, S.: Building the state-of-the-art in sentiment analysis of tweets. In: Proceedings of the Seventh International Workshop on Semantic Evaluation Exercises, SemEval 2013, pp. 321–327 (2013)
26. Myers, J., Well, A., Lorch, R.: Research Design and Statistical Analysis, 2nd edn. Routledge, London (2010)