Word Vector Modeling for Sentiment Analysis of Product Reviews

Yuan Wang[1], Zhaohui Li[1] ⋆, Jie Liu[1,2], Zhicheng He[1], and Yalou Huang[3] and Dong Li[4]

[1] College of Computer and Control Engineering, Nankai University, Tianjin, China
[2] Information Technology Research Base of Civil Aviation Administration of China, Civil Aviation University of China, Tianjin, China
[3] College of Software, Nankai University, Tianjin, China
[4] College of Economic and Social Development, Nankai University, Tianjin, China
{yayaniuzi23,295583194,hezhicheng}@mail.nankai.edu.cn
{jliu,huangyl,lidongnk}@nankai.edu.cn

Abstract. Recent years, an amount of product reviews on the internet have become an important source of information for potential customers. These reviews do help to research products or services before making purchase decisions. Thus, sentiment analysis of product reviews has become a hot issue in the field of natural language processing and text mining. Considering good performances of unsupervised neural network language models in a wide range of natural language processing tasks, a semi-supervised deep learning model has been proposed for sentiment analysis. The model introduces supervised sentiment labels into traditional neural network language models. It enhances expression ability of sentiment information as well as semantic information in word vectors. Experiments on NLPCC2014 product review datasets demonstrate that our method outperforms the traditional methods and methods of other teams.

Keywords: Sentiment Analysis, Product Review, Neural Network Language Model, Semi-supervised Learning

1 Introduction

With the development of the internet environment, more and more people start shopping online, interactions between customers and shopping websites are also more frequent. According to the 33^{rd} China internet network development state report, the user scale of online shopping is 302 million. Customers often write down product reviews on the internet. The reviews contain opinions with different emotional colors and personal semantic information. Thus, mining these product reviews has vital benefit for both consumers and businesses. It helps the consumers to choose the right products and also helps the businesses to improve their products.

⋆ Corresponding author

However, faced with the huge amounts of product reviews on the internet, it is difficult to grasp the panorama of the product reviews accurately and quickly. Thus, mining and analyzing the huge amount of product reviews becomes a hot issue in recent years. An important research direction is sentiment analysis of product reviews, which can be classified into sentiment polarity classification[1, 2] and subjectivity classification[3]. Sentiment polarity classification includes binary classification[1](positive and negative) and multivariate classification[2, 4]. This paper mainly researches the binary classification, which makes a judgment of these product reviews by sentiment orientation (positive emotion and negative emotion). There are two main kinds of classification method, the methods based on sentiment knowledge and the methods based on machine learning techniques. The former ones judge texts' sentiment mainly relying on sentiment word dictionary and language rules, while the latter ones adopt machine learning classification techniques with feature selection.

With the prevalence of online language, review texts become more abundant, and emotional expressions also become more unconstrained. Besides the rewrite of homophonic, wrong characters, pictograms in the texts and tones, there are more irony and parody in tone. Due to these informal and diverse texts, traditional methods based on sentiment dictionary now face difficulties, such as frequent update and inaccurate human judgment. Sentiment analysis methods based on feature classification have attracted increasing attention of many researchers in recent years, as they can return feedback quickly and learn from data adaptively. Pang[1] firstly applied machine learning techniques to sentiment analysis, and demonstrated unigram's effectiveness on textual modeling. But unigram models treat words as independent indices in dictionary, which leads any two words have the same semantic relativity. However, it's far from the truth. For example, any two words out of "good", "well" and "China" have the same semantic distance via the unigram model. But the semantic distance between "good" and "well" is closer than that between "good" and "China". This example shows the method's flaws in text's semantic expressions. In order to solve the problem, the word expression method based on continuous vectors can model semantic information effectively[5–7]. Even so, the word vector method still can't distinguish the important sentiment information explicitly. For example, the model can learn out the information that "good" and "well" have the similar meaning, but it can't grasp the information that the both of the two words have strong positive affectivity. Thus, this work proposes a word vector neural network model, which takes both sentiment and semantic information into account. This word vector expression model not only fuses unsupervised contextual information and sentence level supervised sentiment labels, but also learns words' semantic information and sentiment at the same time. So that it can distinguish sentiments and finish the product reviews sentiment analysis task.

The content of this paper can be summarized as follows: Section 2 introduces the related works of sentiment analysis and the modeling of word vectors. We discuss how to introduce word vectors to help to supervise word modeling in Section 3. Section 4 introduces the method of model acceleration. Section 5 proposes the word vector method has the ability of sentiment analysis by the experiments. Conclusions are given in Section 6.

2 Related Work

Related researches of this paper mainly include sentiment classification and word semantic modeling.

The mainstream methods of sentiment classification can be divided into methods based on sentiment knowledge and methods based on machine learning techniques. The former ones discriminate sentiment orientation of texts by using universial[8] or domain specific[9] sentiment word dictionaries. Kim et al.[8] used WordNet and HowNet to classify sentiment orientation. Tong[9] manually chose sentiment phrases and developed a domain specific sentiment vocabulary for movie reviews. Besides choosing sentiment words manually, Turney[10] used point-wise mutual information to identify emotional words and their polarities. The later ones mainly use machine learning techniques to learn features of review texts, and then classify sentiment orientation of reviews by using discriminant methods, such as SVM, Naive Bayes, Maximum Entropy and etc. Pang[1] firstly modeled the sentiment polarity discrimination as a binary classification problem, using "n-gram" as features. On this basis, the following researches treat sentiment classification as feature selection tasks. Kim[11] put forward position features in addition to "n-gram" features. What's more, Pang[4] defined sentiment classification tasks at different fine-grained levels, using multivariate classification and regression classification to complete multi-classification task of sentiment.

For word semantic modeling, recent researches[5–7] show that word representation based on continuous vectors can model semantic information effectively, and it has already made great progress[5, 6] in different tasks for natural language processing. The classic methods include the language model of three-layer neural networks based on "n-gram" assumption[5], language model based on neural networks by scoring a word string of consecutive n words[6], and recurrent neural network language model[7]. In addition to neural network language models, probabilistic topic models, such as PLSA[12] and LDA[13], learn probability distribution of words under different topics by introducing latent variables, i.e. topics, where the topic distributions of words indicate words' locations in the topic space.

Taking advantage of continuous word vectors, Maas[14] merged it with supervised emotional information to learn the word vector semi-supervisedly based on probabilistic topic models. Inspired by neural network language models[5–7] and Maas[14], this work considers supervised sentiment information in the learning process of neural network language models, builds a semi-supervised model, and learns the word vectors with both abilities of sentiment orientation expression and semantics expression.

3 Word Vector Model for Sentiment Analysis

3.1 Neural Network Language Model

In neural network language model, every word w represents a vector $C(w)$, and the current word is predicted by words around it. The neural network model here is inspired by three layered neural network language model (NNLM) proposed by Bengio[5] in 2003. NNLM follows the n-gram assumptions, where current word is only related to $n-1$ words before it. So the generation probability of word is $P(w_t = i) = P(w_t = i|w_{t-n+1}^{t-1})$. And the generation probability of sentence with N_d words is:

$$P(w_1^{N_d}) = \prod_{t=1}^{N_d} P(w_t|w_1^{t-1}) \propto \prod_{t=1}^{N_d} P(w_t|w_{t-n+1}^{t-1}), \tag{1}$$

where $w_i^j = (w_i, w_{i+1}, \cdots, w_{j-1}, w_j)$ is a word sequence from w_i to w_j. The likelihood optimization of the model is shown in Formula 2.

$$max \sum_d \sum_{t=1}^{N_d} logP\left(w_t|w_{t-n+1}^{t-1}\right) \tag{2}$$

Illustrated in figure 1, the model has three layers. At the bottom is the input layer, including $n-1$ word vectors. The hidden layer in the middle contains hidden nodes with a linear projection input and a nonlinear projection output. The upper output layer is a predict layer with the same size as the dictionary.

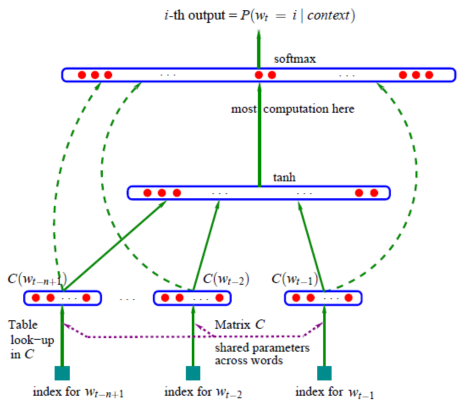


Fig. 1. Neural Network Language Model[5]

NNLM predicts word w_t in two main steps:

1 Words are mapped to m dimension vectors $C(w)$, C is a $|V| \times m$ word vector matrix. $|V|$ is the size of dictionary. The transform from word w to $C(w)$ is to extract the line indexed by w from the C.

2 The $n-1$ word vectors before w_t are connected to form the $(n-1) \times m$ dimension input, denoted as x. The output is a $|V|$-dimension vector, where element i is estimation probability of $P(w_t = i)$, denoted as y. The whole process can be expressed as:

$$f(i, w_{t-n+1}, \cdots, w_{t-1}) = g(i, C(w_{t-n+1}), \cdots, C(w_{t-+1})) \tag{3}$$

For the hidden layer, the input is $d + Hx$ instead of x in traditional neural network, and the output is a $|V|$ dimension vector y activated by $tanh$ function, y_i is the value of the node i. The input of the output layer is the log prediction probability of the next word, the output is the normalization probability through a $softmax$ activation function. The calculation of y is:

$$y = b + Wx + U * tanh(d + Hx), \tag{4}$$

where $x = (C(w_{t-n+1}), \cdots, C(w_{t-1}))$. The probability of the next word is:

$$P(w_t|w_{t-n+1}^{t-1}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_{w_i}}} \tag{5}$$

We estimate parameters with the gradient descent method. C is randomly initialized, (b, W, U, d, H) are all initialized with 0.

This paper adopts the NNLM structure in semantic learning, but semantic modeling of word vectors uses Skip-gram model[15]. The main differences between are, 1) Skip-gram model uses current word to predict other words in a certain range window around it, 2) mapping from the input layer to the hidden layer is simplified, where vector of w_t is the output of the hidden layers, i.e. $x_t = C(w_t)$. Given the current word w_t and the window size c, the maximization goal of the Skip-gram model is $\sum_d \sum_{t=1}^{N_d} \sum_{-c \le j \le c, j \ne 0} P(w_{t+j}|w_t)$. When window size is 5, the model makes likelihood probability largest when predicting the ten words around the current word, and smallest for other words. After simplification, the transformation from input to output turns into $y = b + Wx$.

3.2 Learning Word Sentiment

NNLM can grasp the semantic information of words, but can't consider supervised sentiment information for sentiment analysis. Thus, we use sentiment orientation information of reviews to help in the learning of words' sentiment polarity.

Since the prediction relationship between words is linear, we adopt a logistic regression linear model to model sentiment orientation of each word. First, we map sentiment labels into $[0, 1]$, denoted as s. For binary sentiment discrimination, $s = 1$ means positive and $s = 0$ means negative. Assume that every word has the same capability to express sentiment independently, the generation probability of a sentence sentiment label is:

$$P(s|d; C, a, b_s) = \frac{1}{N_d} \sum_{t=1}^{N_d} P(s|w_t; C, a, b_s), \tag{6}$$

where a is a m-dimension logistic regression parameter vector, b_s is bias of sentiment orientation. This logistic regression model defines sentiment orientation of the words as a probabilistic classification problem of a hyperplane:

$$P(s|w_t; C, a, b_s) = \sigma(aC(w_t) + b_s) \tag{7}$$

where the mapping method of w_t is the same as the Skip-gram model. $\sigma(x)$ is sigmoid activation function. The objective function is:

$$max \sum_d \frac{1}{N_d} \sum_{t=1}^{N_d} P(s|w_t; C, a, b_s). \tag{8}$$

This model makes words with similar sentiment orientation closer in the semantic space, and can be separated by a hyper plane.

3.3 Word Vector Model with Emotional Supervision

We combine the original semantic learning part (in Section 2.1) and the supervised sentiment learning part (in Section 2.2) to learn semantic information and sentiment polarity characteristics of words simultaneously. And the combined objective function is:

$$L = \sum_d \sum_{t=1}^{N_d} \sum_{-c \le j \le c, j \ne 0} logP(w_{t+j}|w_t; C, W, b) + \beta \sum_d \frac{1}{N_d} \sum_{t=1}^{N_d} logP(s|w_t; C, a, b_s) \tag{9}$$

where β is a parameter to balance two parts. The parameters need to learn is:

$$\theta = (C, W, b, a, b_s) \tag{10}$$

The parameters are $|V| \times m$ word vector matrix C showing semantic information with sentiment orientation, neural network's parameters W (a $m \times |V|$ matrix) and output biases b (a $|V|$-dimension vector), regression parameters a (a m-dimension vector) and bias b_s (a value) in the sentiment supervised learning part.

We learn parameters with stochastic gradient ascent methods. The update law is:

$$\theta \leftarrow \theta + \varepsilon \frac{\partial L}{\partial \theta}, \tag{11}$$

where ε is a learning rate. Moreover, multi-thread technology can be used to share model parameters and finish learning rapidly, because every update is independent for different reviews.

4 The Optimization of the Model

As we can see from Figure 1, we need to do an iterative calculation of all output nodes because of the softmax method used by output layer in the prediction of NNLM, which lowers the performance of model a lot. At the same time, the complexity of our model increases exponentially with $|V|$ increases. To reduce the computational complexity of output layer prediction, we use a hierarchical softmax method.

Hierarchical softmax, using Huffman coding to code output layer as a node tree, makes coding of high frequency words shorter and low frequency ones longer. The maximum depth of the node tree is $log|V|$. When predicting output and back-propagation gradient, we just need to calculate node error on the route. Prediction to every word just calculates values of nodes and residuals at most, which can improve the performance a lot. Details are in [16]. In addition, the tree encoding can adopt any hierarchical agglomerative clustering method to lessen the number of output nodes and improve performance.

This strategy mainly ignores normalizing term in softmax without normalized operation. It sees the output of the last layer as a classification task, which aims to make the model classify correctly and automatically. Concretely, it classifies the input x correctly to the category containing y. Hierarchical softmax expresses the category containing y paired by x as a tree structure, the training processing is doing classification layer by layer until reach the leaf nodes and simplifies the number of nodes needed to predict in the output layer. Benefiting from hierarchical softmax, the performance of the model promotes a lot.

5 Experiments

5.1 Dataset

This paper adopts the product review dataset in NLPCC2014 "Sentiment Classification with Deep Learning Technology" task to validate our model. The dataset is divided into training and testing sets. The training dataset contains 20,000 product reviews from product review websites, including 10,000 Chinese reviews and 10,000 English reviews. The datasets in different language are balanced, and each contains 5,000 positive samples and 5,000 negative ones. The testing dataset includes 5,000 reviews, including 2,500 product reviews for each language.

As our model analyzes sentiment orientation of product reviews by word vectors, we takes some necessary preprocessing for these informally written product reviews. 1) Remove all hyperlinks, such as `http://music.jschina.com.cn/adsu.asp?id=385&userid=62039`. 2) Remove the special characters in html, such as &lt, &gt and etc. 3) Use the Yebol Chinese word segmentation platform[5] to segment the sentences in Chinese. After that, words and punctuation characters with definite emotion, like exclamatory marks, question marks, apostrophes and tildes, in reviews are left.

The format of the raw data is XML, taking an English negative sample as an example: *<review id="9214">The Rice cooker works great. Missing the Steaming Basket. Please send it asap. That's the reason I purchased because it said "Rice Cooker/Steamer" Been waiting too long</review>*. This example reflects the complexity of sentiment analysis, as well as the shortages of traditional method based on sentiment words and grammatical rules. If we take the

5 http://ics.swjtu.edu.cn/

sentiment dictionary methods, “great” is an obvious positive word, and is not modified by negative words or adversative conjunctions. So this sentence is likely to be judged as positive, which leads to a mistake.

5.2 Quantitative Evaluation

In training phase of the sentiment classification task, we use classification accuracy (Acc) as our metric for guiding parameter selection quickly. In testing phase, metrics are based on precision (P), recall (R) and F1. Let TP be the number of the positive samples which are correctly classified, FP be the number of the negative samples which are falsely classified, TN be the number of the negative samples which are correctly classified, FN be the number of the positive samples which are falsely classified. The evaluation metrics are as follows.

$$\begin{aligned}
&P_{pos} = \frac{TP}{TP+FP}, R_{pos} = \frac{TP}{TP+FN}, F1_{pos} = \frac{2P_{pos}R_{pos}}{P_{pos}+R_{pos}} \\
&P_{neg} = \frac{TN}{TN+FN}, R_{pos} = \frac{TN}{TN+FP}, F1_{neg} = \frac{2P_{neg}R_{neg}}{P_{neg}+R_{neg}} \\
&Acc = \frac{TP+TN}{TP+FP+TN+FN}
\end{aligned} \tag{12}$$

5.3 Experiment Results

To verify our model's effectiveness in sentiment analysis, we compare different features as text representation, and then use a linear support vector machine[6] with 5-fold cross validation to classify sentiment orientation of product reviews. After learning word vectors, we take the same way as[14] to represent reviews. In training phase, 1) we compare our features with term frequency features[1] (denoted as TF) and sentiment knowledge features (denoted as *Senti*), 2) we combine the best two sets of features in experiment 1) and do parameter selection. In testing phase, the parameters that achieve the best performance in training phase are picked out to predict sentiment orientation of the test data set.

Among those features, *Senti* are composed of 17 features based on sentiment knowledge. They are 1-2) the number of positive/negative words, 3-9) the number of verbs/nouns/adjectives/exclamations/ellipses/tildes/question marks, 10-11) the number of positive/negative phrases, 12) emotional index of the whole review, 13-14) emotion index of the first/last sentence, and 15-17) the number of positive/negative/neutral sentences. Here we use HowNet as the sentiment dictionary. The 10-12 dimensions are obtained by a modified factor method[17], and the 13-17 dimensions are obtained by a rule method[18]. For *Senti*, we find classification accuracy all falls about 2%-6% after removing some of the dimensions. Due to space limitations, these experiments are not listed specifically. So, we use all 17-dimensional features as *Senti* in the experiments below.

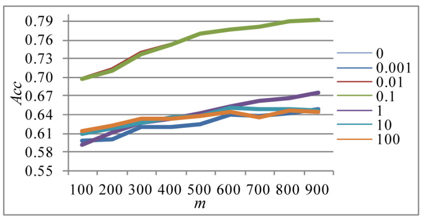
[6] http://www.csie.ntu.edu.tw/ cjlin/liblinear/

Experimental Results in Training Phase Table 1 shows the results on only one kind of features, which are word frequency features, sentiment knowledge features and features we proposed. From Table 1, we have following conclusions. (1) Each model's classification accuracy in English reviews are higher than those in Chinese ones by about 9%, that shows the complexity of Chinese and difficulties in sentiment analysis of Chinese reviews. (2) For English reviews, TF has achieved the best results, with classification accuracy at 81.48%, which is 2.26% higher than our model. (3) For Chinese reviews, our model has achieved the best, with classification accuracy at 71.45%, which is 1.39% higher than TF. Thus, our model is more adaptable to the complex Chinese texts than other models, and the sentiment orientation expression and semantics expression of the word vectors learned by our model are more powerful. Meanwhile, the results based on the sentiment knowledge features are the worst. The main reason is that online language is not standardized, full of rhetoric, strongly arbitrary and changes quickly, which also makes approaches based on manually constructed sentiment knowledge no longer applicable and unable to meet the sentiment characteristics of review texts.

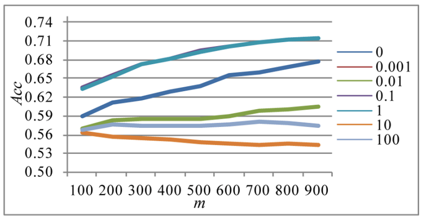
Table 1. Classification accuracy with different features

Dataset	Senti	TF	Our Model
English	75.68%	**81.48%**	79.22%
Chinese	66.37%	70.06%	**71.45%**

In our model, there are two important parameters, the dimension m of word vector $C(w)$ and the balance factor β between the language model part and sentiment discrimination part. Figure 2 shows the classification accuracy when parameters change. Figure 2 shows that, classification accuracy on the English



(a)

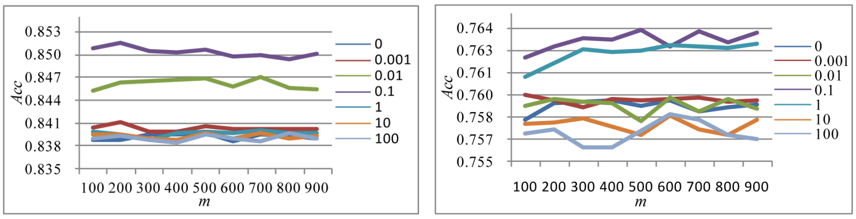


(b)

Fig. 2. Classification accuracy with different parameters on (a) English data and (b) Chinese data

dataset grows with the increase of the vector dimension, and reaches a maximum 0.7922 when $m = 900$ and $\beta = 0.01$. Performances are similar when β equals to 0.01 and 0.1. However, when β equals to 0.01, accuracy is 0.04% higher by average. In the Chinese dataset, the highest accuracy is 0.7145 when $m = 900$ and $\beta = 1$. Performances are similar when the balance factor β equals to 0.1 and 1. However when β equals to 0.1, classification accuracy is 0.05% higher by average. Compared to performance on the English dataset, (1) classification accuracy decreases with the increase of the vector dimension except when β equals to 10, (2) the performance of the model fluctuates with the increase of the vector dimension when β equals to 100. The other results are all similar to that of the English dataset.

The experiments above tell that sentiment knowledge features performed poorly, while word frequency features and our model show their own strengths on data sets in different language. So we consider a combination of these two features to accomplish the task. After parameter selection, the accuracy does have a great improvement. Among them, classification accuracy on English data reaches 85.16% when $m = 200$ and $\beta = 0.1$, 4.28% higher than the single feature method; that on Chinese data reaches 77.40% when $m = 500$ and $\beta = 0.1$, 5.5% higher than the single feature method. See Figure 3 for details.



(a) (b)

Fig. 3. Classification accuracy by using combination features on (a) English data and (b) Chinese data

The results show that TF increases the dimension of reviews' features, which makes accuracy less sensitive to vector dimension. When β is fixed, accuracy fluctuates, especially in the Chinese data set. In both data sets, accuracy performs the best when $\beta = 0.1$. When β equals to 0, the model degenerates into the original Skip-gram model, and the emotional discrimination performs worst. It verifies the effectiveness of the semi-supervised word vector learning model with emotional discriminant information proposed in this paper.

Experimental Results in Testing Phase Known from the experimental results in training phase, the classification accuracy is higher when using both word frequency features and features proposed in this paper. Thus, we combine the two kinds of features, and select two sets of parameters that achieve the best results on training dataset to predict labels for the testing datasets. Table 2 gives the evaluation results on testing datasets.

We know from the table that both results of the English dataset in the testing phase are slightly better than those in the training phase, while both results of the Chinese dataset in the testing phrase are worse than those in the training phase. However, the overall levels are almost the same, which verifies the effectiveness of our model.

Table 2. Classification results on test data and parameter selection

Dataset	Parameters	Negative			Positive		
		P	R	F1	P	R	F1
English	$m = 200, \beta = 0.1$	0.864	0.855	0.860	0.856	0.866	0.861
	$m = 100, \beta = 0.1$	0.865	0.851	0.858	0.853	0.867	0.860
Chinese	$m = 500, \beta = 0.1$	0.780	0.748	0.764	0.758	0.789	0.773
	$m = 700, \beta = 0.1$	0.678	0.452	0.545	0.591	0.794	0.678

6 Conclusion

This paper proposes a word vector learning method aiming at sentiment analysis and explores the performance of deep learning models on the task of sentiment classification for product reviews. The model solves the problem that the traditional unsupervised word model can't catch sentiment orientation information of words. The method introduces label information into the unsupervised neural network language model, so that the sentiment orientation and topical semantic information are both learned. We also accelerate our neural network language model. The experiments on the NLPCC2014 emotional evaluation dataset show the effectiveness of our model.

In the future, we intend to explore this model's performance under lots of unsupervised data when a little supervised data exists. Besides, our model is a general semi-supervised learning framework modeling sentiment label information as well as unsupervised semantic information. We want to explore the learning ability of the model on different semantic aspects of words when treating different label information as different guide information.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (No. 61105049 and No. 61300166), the Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China (No. CAAC-ITRB-201303), the Natural Science Foundation of Tianjin (No. 14JC-QNJC00600) and the Science and Technology Planning Project of Tianjin (No. 13ZCZDGX01098).

References

1. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: Sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10. EMNLP '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 79–86
2. Linhong, X., Hongfei, L., Jing, Z.: Construction and analysis of emotional corpus. Journal of Chinese Information Processing **22** (2008) 116–122
3. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics. ACL '04, Stroudsburg, PA, USA, Association for Computational Linguistics (2004) 271–278

4. Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. ACL '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 115–124
5. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of Machine Learning Research **3** (2003) 1137–1155
6. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning. ICML '08, New York, NY, USA, ACM (2008) 160–167
7. Toms, M., Martin, K., Luks, B., Jan, C., Sanjeev, K.: Recurrent neural network based language model. In: Proceedings of the 11th Annual Conference of the International Speech Communication Association. INTERSPEECH '10, Makuhari, Chiba, Japan (2010) 1045–1048
8. Kim, S.M., Hovy, E.: Automatic detection of opinion bearing words and sentences. In: Companion Volume to the Proceedings of IJCNLP-05, the Second International Joint Conference on Natural Language Processing, Jeju Island, KR (2005) 61–66
9. Tong, R.M.: An operational system for detecting and tracking opinions in on-line discussion (September 2001)
10. Turney, P.D.: 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. ACL '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 417–424
11. Kim, S.M., Hovy, E.: Automatic identification of pro and con reasons in online reviews. In: Proceedings of the COLING/ACL on Main Conference Poster Sessions. COLING-ACL '06, Stroudsburg, PA, USA, Association for Computational Linguistics (2006) 483–490
12. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '99, New York, NY, USA, ACM (1999) 50–57
13. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3** (March 2003) 993–1022
14. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1. HLT '11, Stroudsburg, PA, USA, Association for Computational Linguistics (2011) 142–150
15. Mikolov, T., tau Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies(NAACL-HLT-2013), Association for Computational Linguistics (May 2013) 746–751
16. Morin, F., Bengio, Y.: Hierarchical probabilistic neural network language model. In: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, Citeseer (2005) 246–252
17. Jinggang, W., Xiao, Y., Dandan, S., Lejian, L., Wei, Z.: A chinese bag-of-opinions method for weibo sentiment analysis. In: Proceedings of the First CCF Conference on Natural Language Processing and Chinese Computing. (2012) 1–6
18. Lixing, X., Ming, Z., Maosong, S.: Hierarchical structure based hybrid approach to sentiment analysis of chinese micro blog and its feature extraction. Journal of Chinese Information Processing **26** (2012) 73–83