Text Sentiment Polarity Classification Method Based on Word Embedding

Xiaojie Sun
Guilin University Of Electronic Technology
China
+867732291451
390810552@qq.com

Menghao Du
Guilin University Of Electronic Technology
China
+867732291451
1006484960@qq.com

Hua Shi
Guilin University Of Electronic Technology
China
+867732291451
1670225528@qq.com

Wenming Huang
Guilin University Of Electronic Technology
China
+867732291451
995456524@qq.com

ABSTRACT

Most of the machine learning algorithms for text sentiment analysis use the word embedding obtained by word2vec training as their inputs. However, the word embedding of word2vec training contains only semantic information. An algorithm for text sentiment analysis is proposed solve the problem of text containing semantics, syntax, sentiment and other information. It begins with the learning of original text-multi word embedding in the semantic, syntactic, and sentiment information, followed by proceeding the word embedding fusion. The improved convolution neural network is applied for sentiment analysis. Thus, it solves the problem that the word embedding contains monotonous text information. K-means text clustering is applied by dividing similar text into the same cluster, thus improving the classification accuracy. The application of the Principal Component Analysis (PCA) dimensionality not only extracts the principal component information, but also solves the problem of redundancy embedding and improves the computational performance of classification model. The experiment results show that the presented method has a significant improvement in the accuracy, recall rate and F value of the sentiment polarity analysis of the critical text in comparison with other fusion algorithms.

CCS Concepts

• Computational methods→ Artificial intelligence → Natural language processing → Lexical semantics.

Keywords

Word Embedding; Convolution Neural Network; Sentiment Analysis; Auto Encoder.

1. INTRODUCTION

With the rapid development of communication internet technologies and the advent of the web 3.0 era, the internet users have been changed from passive acceptance of internet information into voluntarily accepting and disseminating the internet information. Weibo, forums and others generate a great deal of user-contributed subjective comments. Sentiment polarity classification identify the sentiment information of users in their comments, and the sentiment information is divided which into positive type and negative type. By analyzing the sentiment direction of these comments, the government can better understand the people's demands on policies, and the companies can better understand the market demand. So, it is of great importance to study an effective textual sentiment analysis method.

Sentiment polarity analysis is to identify the sentiment information of users in their comments. Taboada publish an article [1] about lexicon-based methods for sentiment analysis in 2011. Then, the sentiment information is divided into positive type and negative type. At present, Deep Learning [2,3] has been widely applied in topic classification and sentiment analysis. Kim [4] et al. proposed the modelling for sentences by applying the Convolutional Neural Network (CNN), and through the softmax classification layer, the texts are classified. Socher [5,6] successively proposed several recursive neural network models such as Recurrent Neural Network (RNN), Modified Recurrent Neural Network (MRNN) and Recursive Neural Tensor Network (RNTN). Tai et al. used a more complex Long-Short Term Memory (LSTM) model to solve problems related with sentiment analysis.

Sentiment analysis is currently done in the following ways. Du Hui [7] et al. based on the Continuous Bag of Words (CBOW) neural network model in Word2vec model to learn semantic and emotional word embedding, and finally applied CNN for classification. Liang Jun [8] et al. and others combined RNN and LSTM to capture semantic information and language structure-layer information for text sentiment analysis. These are the textual sentiment analysis methods that use deep learning methods to learn semantic information, sentiment information and syntactic information for partial integration.

2. THE MODEL

The comment text contains a wealth of sentiment, semantic and syntactic information. We can combine these messages to analyze better to the tendentiousness of comment texts. Currently, there are two methods for merge text information. One method is based on the CBOW model in Word2vec to learn the word embedding containing semantic and sentiment information through applying the sentiment polarity as a label. Another method is based on the RNN and CNN model to learn the semantic information and syntactic structure information. Therefore, we uses the following steps：First of all, separately obtaining the semantic embedding, the syntactic embedding and the sentiment embedding, then these vectors are fused by using the PCA method on the basis of k-

ICACS '18, July 27–29, 2018, Beijing, China
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6509-3/18/07…$15.00
https://doi.org/10.1145/3242840.3242850

means. Finally, the word embedding input the CNN, and the sentiment polarity classification of the text is performed. The process for text sentiment polarity analysis is shown in Figure 1.

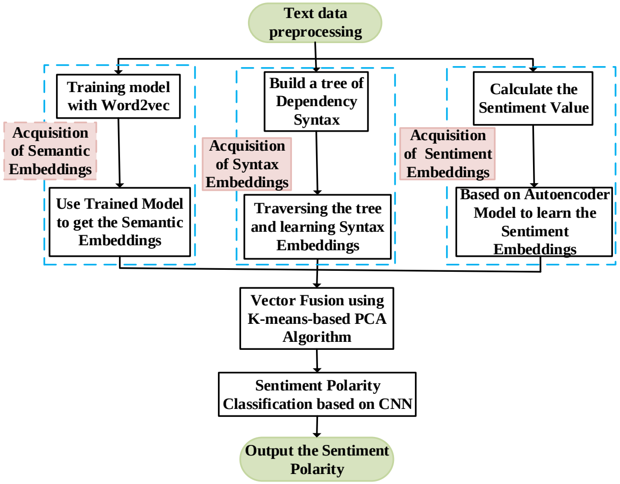
Text data preprocessing
Training model with Word2vec
Acquisition of Semantic Embeddings
Use Trained Model to get the Semantic Embeddings
Build a tree of Dependency Syntax
Acquisition of Syntax Embeddings
Traversing the tree and learning Syntax Embeddings
Calculate the Sentiment Value
Acquisition of Sentiment Embeddings
Based on Autoencoder Model to learn the Sentiment Embeddings
Vector Fusion using K-means-based PCA Algorithm
Sentiment Polarity Classification based on CNN
Output the Sentiment Polarity

Figure 1 Process of the sentiment analysis model

2.1 Acquisition of Semantic Embedding

The existing methods of converting Chinese text into word embedding include One-hot method and Google's Word2vec [9] method. But the One-hot method induces problems of "Semantic Gap " and " dimension disaster ". The word2vec by Google researcher Mikolov [10] and it reduces the complexity of model training. Specifically, there are two different models in the word2vec frame: the CBOW model and the Skip-gram model. The core idea of CBOW model is to predict the current word from its context. As for the skip-gram model, its core idea is to predict the context based on the current word. In this paper, word2vec's skip-gram model is used to train the comment texts and obtain semantic embedding. The main steps are shown as follows.

Firstly, we obtain a lot of comment corpus from the website. Then we pre-process the corpus use by removing non-text symbols, numbers, emoticons and letters. After that, we use the Hanlp segmentation tool to segment the processed corpus. Finally, using the skip-gram model of Word2vec trains the corpus text.

2.2 Acquisition of Syntax Embedding

There are two steps to get syntactic information: 1. Build a dependent syntactic tree for the commentary text; 2. Get a vector containing syntactic information, which by learning structural information of the dependent syntactic tree. To construct the tree, the comment text should be preprocessed first by removing the non-text symbol. Then, using the Hanlp to analyze the dependency syntax and construct a dependency syntax tree. For example, the syntax tree constructed by the phrase " 这家酒店环境优越，服务态度好，下次继续住。" is shown in Figure 2.

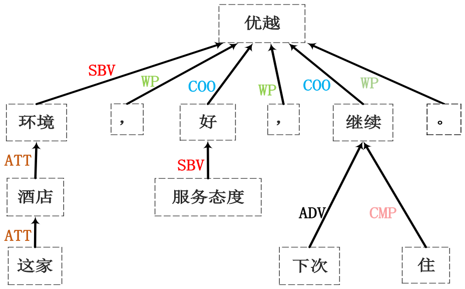

Figure 2 Dependency syntax tree

The leaf node, the root node and the parent node are words of the dependent syntactic analysis tree, and the dependent relationship exists among the nodes. After then, used the following method to learn syntactic embedding. Firstly, the tree node is coded by 0,1, as the input of the autoencoder model, and the encoding vector w is learned by using the unsupervised autoencoder. Then the syntactic structure information is fused to the encoding vector and the syntax embedding is learned. As follows.

①The breadth-first traverse (BFS) algorithm is used to traverse the nodes of the dependency syntactic structure tree.

② In the process of traversing, the depth of the current node and the offset position at the current depth are determined. If the current node is the root node, formula (1) is adopted; if the current node is a tree node or a leaf node, Using formula (2).

③ Query the current node's parent node and child nodes vector, and through the formula (3) to normalize the current node vector.

$$f(w,c,t) = \tanh(w \bullet c \bullet t) \qquad (1)$$

$$f(w,c,t) = \tanh(\alpha * f_p(w,c,t) + \beta * \sum' f_c(w,c,t) \ + \chi * \sum' f_s(w,c,t) \qquad (2)$$

$$code(w,c,t)_{1\times m} = \frac{f(w,c,t) - \mu}{\sigma} \qquad (3)$$

In the above formulas, t and c denote the offset positions of the current node at the depth of the tree and the current depth, respectively. $f_p(w \bullet c \bullet t)$,the syntax embedding which represents the parent node of the current node. $\sum' f_c(w,c,t)$ and $\sum' f_s(w,c,t)$ respectively represent the sum of the syntactic embedding of all child nodes and brother nodes of the current node. In formula (3), μ is the mean before encoding normalization, and σ is the squared-error before encoding is normalized. In formula (2), α,β,χ represents the weighting factor, and the correlation coefficient method is used to obtain these weighting factors. The correlation coefficient r_{ij} of the evaluation factor is calculated by the formula (4) and the formula (5). The formula (6) is then used to calculate the weight of the evaluation factor.

$$L_{ij} = \sum k_i k_j ((\sum k_i)(\sum k_j)/m) \qquad (4)$$

$$r_{ij} = L_{ij} \Big/ \sqrt{L_{ii} L_{jj}} \qquad (5)$$

$$w_i = \sum_{j=1}^{m} r_{ij} \Big/ \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \qquad (6)$$

The current node's father node, child node and brother node as the evaluation factor. Calculate the weight factor α through the parent node and the child node, the child node and the brother node to get β , the brother node and the father node to get χ . K in Equation (4) represents the syntax of the node. If the syntax embedding exists, the direct assignment to k, if not, using the formula (1) to get. In the above formula i and j are the number of evaluation factors. For example, when calculating the weighting factor α , i denotes the number of parent nodes of the current node and j denotes the number of child nodes of the current node.

2.3 Acquisition of Sentiment Embedding

Sentiment information in the comment text is reviewed in terms of sentiment polarity and sentiment intensity. The existing CBOW model and autoencoder all directly use sentiment polarity as a label, they ignore the information about text sentiment intensity. To solve this problem, In this paper, the improved autoencoder is used to learn sentiment embedding with sentiment information. The flow chart of sentiment embedding learning is shown in Figure 3.

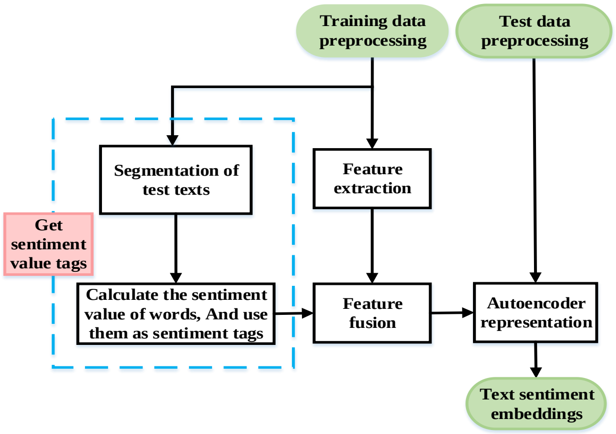
Training data preprocessing
Test data preprocessing
Segmentation of test texts
Feature extraction
Get sentiment value tags
Calculate the sentiment value of words, And use them as sentiment tags
Feature fusion
Autoencoder representation
Text sentiment embeddings

Figure 3 Sentiment embedding learning flow chart

Sentiment value tags calculation: The sentiment intensity value d is obtained from the sentiment intensity dictionary, and the sentiment polarity dictionary obtains the sentiment polarity value k (K is -1 if the sentiment polarity is negative and K is 1 if it is positive). Then the formula $emotion\_value = k\sum d$ is used to calculate the sentiment value, and then Z-score method is used to normalize the data.

Feature extraction: Firstly, the training data to determine the scope of a reasonable area r. In the observation space R^m , if there is a spherical centered on p, the radius of the sphere is, satisfy the formula $B_r(p) = \{x \in X \mid d(x,p) < r\}$, it is included in set U, the set U is the area of point p, the r is the distance in R^m .

$$r = \left\| \sum (x_i - x_i') \right\| \qquad （7）$$

In formula (7), x_i' is the coordinate of the i -th dimensions of the refactoring data in R^m , corresponding to the output of the i -th neuron in the output layer of AE (Auto Encoder) network, x_i is the i -dimensional coordinate of the input data. Then, according to r construct reconstruction error function $error = r^2/2$. Finally, through Random Gradient Descent (SGD) algorithm to backpropagation error, which to learning the optimal parameters. By learning features, a new autoencoder is obtained.

Feature fusion: In the stage of feature extraction, by using the calculated sentiment value label, A new acoustic emission (AE) network model at the feature extraction stage is reconstructed to supervise learning. It can enrich the text content. Finally, the network model is retrained by the SGD algorithm to obtain a new autoencoder representation model.

2.4 PCA fusion model based on k-means

PCA can not only reduce the dimension of high-dimensional data but also can denoise. Therefore, this paper use PCA method is used to reduce the dimension and fusion of word embedding. And the following we proposed a model of PCA based on K-means. There are two main parts: Firstly, the data are classified into different clusters based on k-means algorithm, and then the PCA algorithm is used to reduce the dimension and fusion of word embedding of different clusters respectively. This method not only retains the main component information, but also avoids the influence of word embedding fusion due to the distribution of data. The value of the parameter k determines the effect of the final classification. Therefore, Elbow algorithm is used to obtain the best value of k. The overall specific algorithm steps are as follows:

Input: Sample Set $X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ M & M & M \\ x_{n1} & x_{n2} & x_{n3} \end{pmatrix} = (x_i, L\ , x_n)$;

①using the Elbow algorithm to obtain through the formula (8) to calculate the value of the clustering algorithm parameter k.

$$J(c^1, K\ , c^n, \mu_1, K\ , \mu_m) = \frac{1}{n}\sum_{1}^{n}(\left\| x_i - \mu_{c^i} \right\|) \qquad （8）$$

In formula (8), c^i represents the cluster center subscript closest to x_i and μ_m represents the cluster center. The value of J represents the sum of the distance from each sample to the cluster center, and take the value of m corresponding to the minimum value of J as the value of parameter k in the clustering algorithm.

②Using the k-means algorithm to cluster the samples and get the centroids set after clustering. $\mathrm{centroid} = \{a_1, a_2, L\ a_n\}$

③The centroid set after clustering is used as the centroid of PCA algorithm.

④Calculation the covariance matrix of samples.

$$C_{n \times n} = (c_{i,j}, c_{i,j} = \mathrm{cov}(Dim_i, Dim_j))$$

Solving the characteristic solution of covariance matrix C. Take the feature vector corresponding to the largest the number k eigenvalues $w_1, w_2, K\ , w_k$. Enter the fused word embedding: $W = (w_1, w_2, K\ , w_k)$. If the data set has three dimensions like $\{x, y, z\}$, then the sample covariance matrix is:

$$c_{3\times 3} = \begin{pmatrix} \mathrm{cov}(x,x) & \mathrm{cov}(x,y) & \mathrm{cov}(x,z) \\ \mathrm{cov}(y,x) & \mathrm{cov}(y,y) & \mathrm{cov}(y,z) \\ \mathrm{cov}(z,x) & \mathrm{cov}(z,y) & \mathrm{cov}(z,z) \end{pmatrix}$$

2.5 Improved Neural Network Classification Model

The CNN only applies convolution and Down—pooling when applying text classification. However, a convolution operation can't extract text features effectively. Therefore, We add a dynamic convolution-pooling layer as shown in Figure.4.

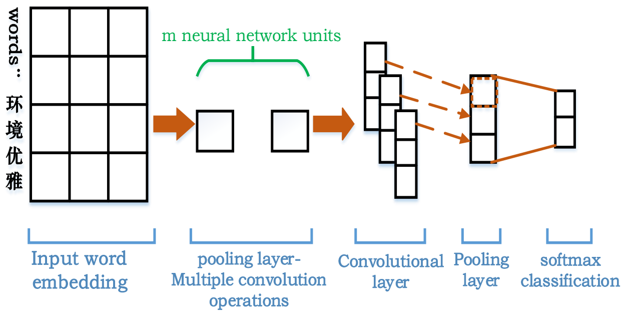

Figure 4 Improved Convolutional Neural Network Model

The improved convolutional neural network model takes as input the word embedding containing semantic, ,syntactic and affective information. Adopts the maximum pooling method. And the neural network unit shown in Figure 5.

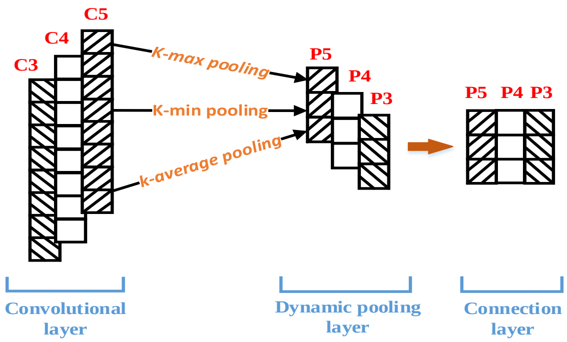

Figure 5 Neural network unit model diagram

The network unit consists of a convolution layer, a dynamic pooling layer and a connecting layer. The convolutional layer is convolved with different sliding windows of 3, 4, and 5 to obtain C3, C4 and C5. The dynamic pooling layer consists of three parts: K-max pooling, is to choose k largest eigenvalues; k-min pooling, is to choose k smallest eigenvalues; k-average pooling, is dividing the sequence into k segments, then take the average of each paragraph. The connecting layer flips the pooled features as input to the next network element model. A value of the pooling layer parameter k is calculated based on the length of the sentence and the depth of the network, and is calculated as formula (9).

$$k_i = \max(k_{top}, \mathrm{int}((l-i)*s/l)) \qquad (9)$$

In the above formula, k_{top} is the number of convolutional layers, l is the number of neural network units, s is the length of the input text, and i is the number of layers of the network where the current neural network unit is located. And $\mathrm{int}((l-i)*s/l)$ is rounded up. For example, when the m value is 3, the length of the input text is 24, the value of k_1 is 16, the value of k_2 is 8, and the value of k_3 is 4.

3. EXPERIMENT AND ANALYSIS

3.1 Experimental Data, Environment and Evaluation Index

The experimental data used in this article is the hotel review data collected from eLong.com. The experimental data contains 70,000 positive comments and 30,000 negative comments. Training data set and test data set have the same format. The label of positive comment is 1, and the label of negative comment is 0.

The experimental environment is listed as follows: the memory is 4G, the Central Processing Unit (CPU) is the Inter (R) Core 2 Duo E7500 processor, the operating system is Window7 and Ubuntu 14.04 dual systems, and the programming languages are the python language and the java language. In this paper, the accuracy rate, recall rate and F value are set as the main evaluation index of classification performance.

3.2 Experimental Parameters Set

The experimental parameters set in this paper mainly set the dimension of the word embedding input by the neural network model and the number of neural network units in the improved neural network model.

Setting of the word embedding dimension: Set the neural network unit in the neural network model as 1; set the word embedding dimension of the text as 100, 150, 200, 250, 300, 350, 400, 450, 500. The classification F value is shown in Figure 6.

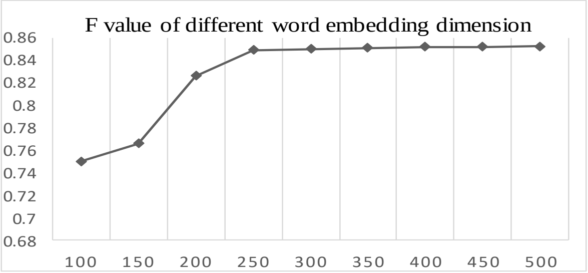

Fig.6 Word embedding text classification F values in different dimensions

As shown in Figure 6, when the vector dimension is low, the semantic information can't be fully expressed and some of the semantic features are missing. As the dimension increases, the semantic information is fully expressed and the semantic features are fully extracted. The accuracy of classification increases and tends to be stable. However, the higher the word embedding dimension is, the longer the training model is, and the higher the demand for computer resources is. Therefore, in the following comparative experiment, the semantic word embedding, syntactic word embedding and sentiment word embedding are all set as 300 dimensions. The word embedding obtained by PCA-fused word embedding based on k-means are also set as 300 dimensions. Setting the number of neural network units m: The dimension of word embedding is determined as 300. Then, different numbers of neural network units are subject to comparative experiments. Its classification is shown in Figure 7.

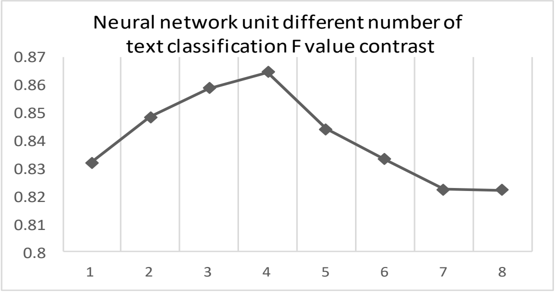

Figure 7 neural network unit different number of text classification F value comparison

As can be seen from the comparison of F values in Figure 7, with the increase of neural network elements, the improved convolutional neural network model can learn the deep features of texts and the F value of text classification is improved. However, when the number of neural network elements is increased to 5, the convolutional neural network model leads to the decrease of F value due to the gradient explosion. Therefore, the experiment in this paper sets the number of NE units to 4.

3.3 Analysis of Results

In this paper, we study the word embedding through several different word embedding models, and then use the CNN model to classify them, then compare and analyze the classification results. The comparison word embedding mainly has the following models.

CNN-W2v: Semantic word embedding retrieved by word2vec.

CNN-CBOW-S: By improving the CBOW model in word2vec, the semantic and sentiment word embedding is learned by using the sentiment polarity as a label.

CNN-S1-S2: Splicing the sentiment and semantic embedding which had learned. The two sets of word embedding are all 300-dimensional, and the splicing will result in a 600-dimensional word embedding containing sentiment and semantic information.

CNN-S1-S2-S3: Splicing the syntax, sentiment and semantic embedding which had learned. The three groups of word embedding are all 300-dimensional, and the splicing will obtain 900-dimensional word embedding containing sentiment, syntactic and semantic information.

CNN-PCA-S1-S2: The semantic and sentiment embedding which had learned through the k-means PCA algorithm based on the fusion of the word embedding.

CNN-PCA-All: The syntax, semantic and sentiment embedding which had learned through the k-means PCA algorithm fusion of the word embedding.

The experimental results are shown in Table 1.

Table 1 different model classification accuracy

Model	Accuracy	Recall rate	F value
CNN-W2v	0.8747	0.8563	0.8654
CNN-CBOW-S	0.9130	0.8945	0.9036
CNN-S1-S2	0.8936	0.8732	0.8832
CNN-S1-S2-S3	0.9142	0.9021	0.9081
CNN-PCA-S1-S2	0.9313	0.9125	0.9218
CNN-PCA-All	0.9411	0.9243	0.9326

As shown in Table 1, comparison of the overall experimental data shows that using google word2vec learning semantic embedding, sentiment polarity accuracy, recall rate and F value is the lowest. Comparing the models CNN-CBOW-S, CNN-S1-S2 and CNN-PCA-S1-S2, we can see that all indicators are lower than the CBOW model. In the PCA fusion method based on k-means, improving the classification The accuracy rate of word embedding containing semantic and sentiment is 0.9313, and the recall rate and F value are also improved. Comparisons of CNN-CBOW-S, CNN-PCA-S1-S2 and CNN-PCA-All show that after the syntactic information was fused, the accuracy, recall rate and F-value of affective classification were enhanced again.

CBOW-based model can only learn semantic and sentiment information. Based on CNN and recurrent neural network model can only learn semantic and syntax information. In this paper, the PCA fusion method based on k-means not only can fuse a variety of textual information, but also can control the growth of word embedding dimension, and improve the accuracy, recall rate and F value of text sentiment polarity analysis.

4. CONCLUSION

The word embedding fusion method mentioned in this paper can fuse many kinds of text information at the same time and also solve the problems of vector redundancy. By experiments, showing the accuracy of such method in classification results is significantly higher than that of other word embedding fusion methods. The results of this article can be applied to reviews in the shopping platform, allowing users to understand the cost-effectiveness of the product and the merchant's understanding of the advantages and disadvantages of the product. But, these are all non-negligible sentiment messages due to the subjective and objective elements of the commentary, as well as the non-textual expressions and characters. The other hand, negatives and conjunctions in contexts can lead polarity shifts. Therefore, the direction of further research should how to effectively extract the non-text sentiment features and solve the problem of sentiment polarity shift in contexts.

5. ACKNOWLEDGMENTS

The research is supported by technology research project of Guangxi Province(No.1598019-6), Key Laboratory of Intelligent Processing of Computer Images and Graphics (NoGIIP201707), the Special Program of GUET Graduate Education Innovation Project for Reform and Development of Degrees and Graduate Education(No2016XWYJ10) and Basic Ability Enhancement Project for Middle-aged and Young Teachers in Colleges and Universities in Guangxi (2017KY0200).

6. REFERENCES

[1] Taboada M, Brooke J, Tofiloski M, et al. Lexicon-based methods for sentiment analysis[J]. Computational Linguistics, 2011, 37(2):267-307.

[2] BENGIO Y, COURVILLE A, VINCENT P. Representation Learning: A Review and New Perspectives[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798–1828.

[3] TANG D, QIN B, LILT T. Document modeling with gated recurrent neural network for sentiment classification [C]Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015: 1422-1432.

[4] Kim Y.Convolutional neural networks for sentence classification[C]. EMNLP.2014：1746-1751

[5] Socher R, Manning CD, Ng A Y. Learning continuous phras representations and syntactic parsing with recursive neural networks[C]. Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop. 2010:1-9.

[6] Socher R, Perelygin A, Wu J Y, et al. Recursive deep models

for semantic compositionality over a sentiment treebank[C]. EMNLP.2013:746-751

[7] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[C]//Proceedings of Work-shop at ICLR.2013

[8] T. MIKOLOV, K. CHEN, G. CORRADO, et al. Efficient Estimation of Word Representations in Vector Space. 2013 Workshop at International Conference on Learning Representations, 2013

[9] T. MIKOLOV, I. SUTSKEVER, K. CHEN, et al. Distributed Representations of Words and Phrases and Their Compositionality. 2013 International Conference on Neural Information Processing Systems, 2013

[10] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[C]//Proceedings of Work-shop at ICLR.2013