IEEE Access Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

SPECIAL SECTION ON ARTIFICIAL INTELLIGENCE AND COGNITIVE COMPUTING FOR COMMUNICATION AND NETWORK

Received March 15, 2019, accepted March 31, 2019, date of publication April 9, 2019, date of current version April 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2909919

Sentiment Analysis of Comment Texts Based on BiLSTM

GUIXIAN XU[1], **YUETING MENG**[1], **XIAOYU QIU**[2], **ZIHENG YU**[1], **AND XU WU**[1]

[1]College of Information Engineering, Minzu University of China, Beijing 100081, China
[2]The Library, Shandong University of Traditional Chinese Medicine, Jinan 250355, China

Corresponding author: Guixian Xu (guixian_xu@muc.edu.cn)

This work was supported by Project of Humanities and Social Science, Ministry of Education of China, under Grant 18YJA740059.

ABSTRACT With the rapid development of Internet technology and social networks, a large number of comment texts are generated on the Web. In the era of big data, mining the emotional tendency of comments through artificial intelligence technology is helpful for the timely understanding of network public opinion. The technology of sentiment analysis is a part of artificial intelligence, and its research is very meaningful for obtaining the sentiment trend of the comments. The essence of sentiment analysis is the text classification task, and different words have different contributions to classification. In the current sentiment analysis studies, distributed word representation is mostly used. However, distributed word representation only considers the semantic information of word, but ignore the sentiment information of the word. In this paper, an improved word representation method is proposed, which integrates the contribution of sentiment information into the traditional TF-IDF algorithm and generates weighted word vectors. The weighted word vectors are input into bidirectional long short term memory (BiLSTM) to capture the context information effectively, and the comment vectors are better represented. The sentiment tendency of the comment is obtained by feedforward neural network classifier. Under the same conditions, the proposed sentiment analysis method is compared with the sentiment analysis methods of RNN, CNN, LSTM, and NB. The experimental results show that the proposed sentiment analysis method has higher precision, recall, and F_1 score. The method is proved to be effective with high accuracy on comments.

INDEX TERMS Sentiment analysis, artificial intelligence, social network, weighted word vectors, BiLSTM.

I. INTRODUCTION

In recent years, with the rapid development of the Internet and social networks, more and more users begin to freely express their opinions on web pages. Therefore, the big data of user comments is generated on the Internet. For example, the product comments are generated on E-commerce websites such as Jingdong and Taobao, and hotel comments are generated on travel websites such as Ctrip and ELong. With the explosive increasing of comments, it is difficult to analyze them manually. In the era of big data, mining the emotional tendencies of comment texts through artificial intelligence technology is helpful for timely understanding of network public opinion. The research of sentiment analysis is very meaningful for obtaining the sentiment trend of the comments.

Sentiment analysis is a kind of text classification, involving natural language processing, machine learning, data mining, information retrieval and other research fields [1]. Sentiment analysis of comments mainly focuses on the sentiment orientation analysis of comment corpus, which indicates that users express positive, negative or neutral sentiments towards products or events. In addition, sentiment analysis can be divided into news comment analysis [2], product comment analysis [3], film comment analysis [4] and other types. These comments convey the views of Internet users about products, hot events, etc. Merchants can master the user satisfaction with the relevant product comments. Potential users can evaluate products by viewing these product comments.

The essence of sentiment analysis is the text classification task, and the contribution of different words is different to classification. For sentiment classification tasks, learning a low-dimensional, non-sparse word vector representation for a word is a key step [5]. The widely used word representation is the distributed word vector obtained by Word2vec technology [6]. The word vector has a low dimension and contains the semantic information of the word. However, distributed

The associate editor coordinating the review of this manuscript and approving it for publication was Yin Zhang.

word vectors do not contain sentiment information about words. In this paper, the contribution of the word's sentiment information to text sentiment classification is embedded into the traditional TF-IDF algorithm, and the weighted word vector is generated.

In this paper, a sentiment analysis method of comments based on BiLSTM is proposed. The remainder of this article consists of four parts. Firstly, the research backgrounds of the text sentiment analysis method and the representation of the word vector are expounded. Secondly, the detail of the proposed sentiment analysis method of comments is described. Thirdly, the experiments are carried out and the experimental results are analyzed and discussed. Finally, the proposed method is summarized and the next research direction is introduced.

II. BACKGROUND

A. TEXT SENTIMENT ANALYSIS TECHNOLOGY

Text sentiment analysis technology mines text emotions through computer technology. According to the object of sentiment analysis, text sentiment analysis can be divided into three levels, respectively for words [7], sentences [8], chapters [9]. According to the classification method of sentiment orientation, it can be divided into binary sentiment classification [10], ternary sentiment classification [11] and multi-sentiment classification [12].

At present, text sentiment analysis methods are mainly divided into three categories: sentiment analysis method based on sentiment dictionary, sentiment analysis method based on machine learning, and sentiment analysis method based on deep learning.

The method based on sentiment dictionary uses the dictionary to identify sentiment words in the text and obtain sentiment values. Then, according to the sentiment calculation rules, the text sentiment tendency is obtained. The literatures [13], [14] introduced the representative research based on sentiment dictionary. Text sentiment analysis based on sentiment dictionary does not require manual labeling of samples and is easy to implement. However, the quality of the analysis is highly dependent on the sentiment dictionary. Most of the sentiment dictionaries have problems such as insufficient coverage of sentiment words and lack of domain words.

The earliest research on text sentiment analysis based on machine learning was Pang *et al.* [15]. They used naive Bayesian algorithm, maximum entropy algorithm and SVM algorithm to analyze the sentiment of film reviews. Finally, the experimental results showed that SVM algorithm worked best in dealing with the sentiment classification of movie reviews. Goldberg and Zhu [16] proposed a graph-based semi-supervised classification algorithm which scored 0-4 stars for the positive and negative comments. Wang *et al.* [17] studied the sentiment analysis of short texts. Based on multiple dimensions such as sentiment features, negative features and emoji, a high-dimensional mixed feature sentiment analysis model based on SVM was proposed. Sentiment analysis method based on machine learning tends to be more accurate, but it relies on the quality of the corpus labeled with polarity.

In recent years, many scholars have introduced the method of deep learning into sentiment analysis and achieved good results. The RNTN (Recursive Neural Tensor Network) model proposed by Socher *et al.* [18] introduced a sentiment tree library, which synthesized semantics on the syntactic tree of binary sentiment polarity and obtained good sentiment analysis results in the data set of movie reviews. The CharSCNN [19] (Character to Sentence Convolutional Neural Network) model used two convolutional layers to extract the features of related words and sentences, and mined semantic information to improve the sentiment analysis of short texts like Twitter. Irsoy and Cardie [20] used the Recurrent Neural Network based on time series information to obtain sentence representation, which further improved the accuracy of sentiment classification. Ta *et al.* [21] proposed a Tree-Structured Long Short-Term Memory Networks model, which had achieved good results in semantic association and sentiment classification. Baziotis *et al.* [22] introduced the attention mechanism into the LSTM, which achieved good results in the sentiment analysis of SemEval-2017 Task4 for Twitter.

Considering that the feature words of comments are sparse, and in order to better capture the context information, Bidirectional Long Short Term Memory [23] in deep learning is used to obtain the comment representation in this paper.

B. WORD REPRESENTATION

In natural language processing, words in sentences or documents are usually used as features [24]–[26]. Currently, there are two widely used word vector representations: one-hot representation and distributed representation.

The vector dimension of one-hot representation is decided by the words' number of the dictionary containing a large number of words and is the same as it. The vector of the word only has a dimension value of 1 corresponding to the position of the word in the dictionary, and the rest dimension values are 0. The method has the following problems: (1) The vector dimension will be too large if there are too many words contained in the dictionary; (2) The vector has too many 0 values, which causes the sparseness of the vector; (3) This method ignores the semantic association of the words.

Distributed representation was proposed by Hinton in the 1986 [27]. It maps each word into a low-dimensional real vector, which solves the problem that the dimension of the One-hot representation word vector is too large. All word representations constitute a word vector space, so the semantic similarity can be judged by calculating the distance between words.

Bengio *et al.* [28] first introduced word distributed representation into the language model of neural network, and proposed the Neural Network Language Model (NNLM). For the NNLM model, the context of the word w_t was represented as context = $\{\mathrm{w}_{(\mathrm{t-n})}, \mathrm{w}_{(\mathrm{t-(n-1)})}, \ldots, \mathrm{w}_{(\mathrm{t-1})}\}$. The NNLM model consisted of four layers which were the input layer, the mapping layer, the hidden layer, and the output layer. In order to obtain an efficient training model, Mikolov *et al.* [29] removed the nonlinear hidden layer of NNLM and proposed Word2vec technology. It contained two new log-linear models: Continuous Bag-of-Words (CBOW) and Skip-gram (SG). These models not only improved the accuracy of the word vector, but also greatly improved the training speed.

At present, word vectors have been applied into sentiment analysis. For example, Kim [30] took the lead in using word vectors as features and input word vectors into convolutional neural networks for sentiment classification. The model achieved good classification results. Tang *et al.* [31] introduced several neural networks to effectively encode context and sentiment information into word embeddings. The effectiveness of the word embedding learning algorithm is verified on Twitter dataset. Chen *et al.* [32] integrated user and product information into a hierarchical neural network to implement sentiment analysis of user product reviews. Liu and Zhang [33] used the news about food safety as the training corpus to obtain the word vectors. The trained word vectors were input into the underlying Recursive Neural Network to obtain the sentence representation. Then, sentiment analysis of the news was achieved through a high-level Recurrent Neural Network.

However, the word representations in the current researches of sentiment analysis do not comprehensively consider the sentiment information contained in the words and its contribution to the classification. In this paper, the sentiment information is integrated into the traditional TF-IDF algorithm to calculate the weight of the words. Thus, the word vector could be better represented.

III. RESEARCH METHODS

A. THE CONSTRUCTION OF THE WEIGHTED WORD VECTOR

In this paper, Word2vec model is used to obtain distributed representations of words. Word2vec technology includes CBOW model and Skip-gram model. Both CBOW model and Skip-gram model include input layer, projection layer and output layer. CBOW model predicts target words based on context distribution. For the word w_k, the context is expressed as follows:

$$context(w_k) = \left\{w_{k-t}, w_{k-(t-1)}, \ldots, w_{k+(t-1)}, w_{k+t}\right\} \quad (1)$$

On the contrary, the Skip-gram model predicts the context based on the target word w_k.

TF-IDF is a combination of TF and IDF weight calculation methods. It is the most commonly used weight calculation method in text categorization. The frequency of the word in a single document and the distribution of the word in the documents are considered in this method. It can better reflect the importance of a feature in classification. The formulas of TF-IDF are as follows:

$$\mathrm{w}\,(\mathrm{t}_i, \mathrm{d}) = \frac{\mathrm{tf}\,(\mathrm{t_i}, \mathrm{d}) \times idf\ (t_i)}{\sqrt{\sum_{t_i \in d} [tf\ (t_i, d) \times idf\ (t_i)]^2}} \quad (2)$$

$$\mathrm{idf}\,(t_i) = \log\,(N/n_{ti}) + 1 \quad (3)$$

$\mathrm{w(t_i,d)}$ denotes the weight of the word $\mathrm{t_i}$ in document d, $\mathrm{tf(t_i,d)}$ denotes the frequency of the word $\mathrm{t_i}$ in document d, N denotes the total number of documents and $\mathrm{n_{ti}}$ denotes the number of documents in which the word $\mathrm{t_i}$ appears.

In this paper, whether a word contains sentiment information is determined by matching sentiment dictionaries. At present, Hownet sentiment dictionary, National Taiwan University Sentiment Dictionary (NTUSD) and Li Jun's Chinese commendatory term and derogatory term Dictionary of Tsinghua University [34] are three commonly used sentiment dictionaries in Chinese sentiment analysis. If the number of words in sentiment dictionary is too large, it will contain a large number of words with low sentiment information. If the number of words in sentiment dictionary is too small, it will ignore a large number of sentiment words in the text and reduce the accuracy of sentiment classification. After the analysis and comparison, Li Jun's Chinese commendatory term and derogatory term Dictionary of Tsinghua University is used in this paper. The dictionary contains a moderate number of words with high sentiment information. The dictionary contains 10035 sentiment words. The related information is shown in TABLE 1 and TABLE 2.

TABLE 1. The related information of positive word.

Examples of positive word	Number
光荣(glory)、动听(enchanting)、 勇敢(brave)、祝福(bless)、谦虚(humble)......	5567

TABLE 2. The related information of negative word.

Examples of negative word	Number
下流(Squat)、伤悲(sadly)、谗言(calumny)、 焦虑(anxious)、怒斥(irritated)、出卖(betray).....	4468

The weight calculation method for word vectors is as follows:

$$w_i = tf - idf_i \cdot e \quad (4)$$

$$e = \begin{cases} \alpha, & t_i\ is\ a\ sentiment\ word \\ 1, & t_i\ is\ a\ non-sentiment\ word \end{cases} \quad (5)$$

where $\mathrm{t_i}$ is the word, $\mathrm{tf\text{-}idf_i}$ is the TF-IDF value of the feature word calculated by equation 2, $\mathrm{w_i}$ is the weight of the word, and e is the weight according to whether the word contains sentiment information. $\alpha > 1$.

$\vec{a}$ is defined as the distributed word vector trained by word2vec. The weighted word vector $\vec{v}$ constructed in this paper is defined as follows:

$$\vec{v} = w_i \cdot \vec{a} \tag{6}$$

B. RECURRENT NEURAL NETWORKS

The traditional neural network model is ineffective in dealing with the sequence learning because it is impossible to describe the correlation between the front and back of the sequence. RNN (Recurrent Neural Networks) is a sequence learning model that connects nodes between hidden layers and can learn sequence feature dynamically. RNN which is applied to Chinese text sentiment analysis is shown in FIGURE 1. In the figure, the input text is 酒店的环境不错 (The environment of the hotel is good). After word segmentation, it becomes 酒店/的/环境/不错. Each word is converted into the corresponding word vector (w_1, w_2,w_3,w_4), and then the corresponding word vector (w_t,w_t,w_t,w_t) is sequentially input into the RNN.

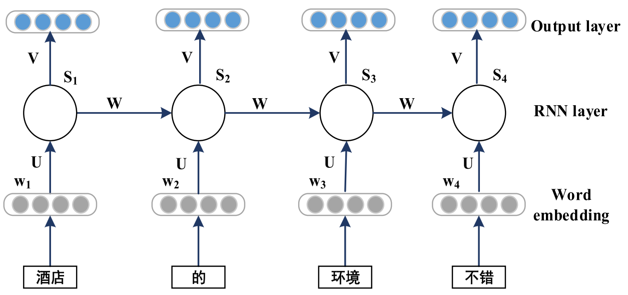

FIGURE 1. The sentiment analysis model of RNN.

The calculation process of RNN is as follows:

1) At the time t, w_t is input to the hidden layer.
2) s_t is the hidden layer's output of the step t. s_t is based on w_t and s_{t-1}. $s_t = f(U * w_t + W * s_{t-1})$, where f is usually the non-linear function, such as tanh or ReLU.
3) Finally, the output D is calculated according to $o_t = soft\max(V * s_t)_.$

C. LONG SHORT TERM MEMORY MODEL

The traditional recurrent neural network model cannot capture long-distance semantic connection, even if it can transfer semantic information between words. In the process of parameter training, the gradient decreases gradually until it disappears. As a result, the length of sequential data is limited. Long Short Term Memory (LSTM) overcomes the problem of gradient disappearance by introducing Input gate i, Output gate o, Forget gate f and Memorycell. The LSTM network structure [35] is shown in the following FIGURE 2.

Forget gate f determines the information to forget in Memorycell at the last moment. The input is h(t-1) and x(t). The output value is between 0 and 1. The calculation method is as follows:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + V_f c_{t-1} + b_f) \tag{7}$$

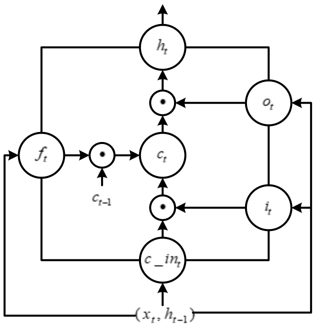

FIGURE 2. The diagram of LSTM network structure.

Among them, h_{t-1} and x_t are the inputs of LSTM unit. W_f is the connecting weight of x_t and forget gate f. U_f is the connecting weight of h_{t-1} and forget gate f. c_{t-1} is the state of Memorycell at the last moment. V_f is the connecting weight of c_{t-1} and forget gate f. b_f is the bias term. $\sigma(\cdot)$ is sigmoid activation function.

Input gate i determines the information to be updated in Memorycell at the current time. The calculation method is as follows:

$$\begin{cases} i_t = \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1} + b_i) \\ c\_in_t = \tanh(W_c x_t + U_c h_{t-1} + V_c c_{t-1} + b_c) \\ c_t = f_t \cdot c_{t-1} + i_t \cdot c_i n_t \end{cases} \tag{8}$$

Among them, W_i is the connecting weight of x_t and i_t. U_i is the connecting weight of h_{t-1} and i_t. V_i is the connecting weight of $c\_t - 1$ and i_t. W_c is the connecting weight of x_t and c_in_t. U_c is the connecting weight of c_in_t and h_{t-1}. tanh(.) is tanh activation function. f_t and i_t refer to weights of c_{t-1} and c_in_t. b_i and b_c are the bias terms.

Output gate o determines the output value of LSTM unit. The calculation method is as follows:

$$\begin{cases} o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o c_{t-1} + b_o) \\ h_t = o_t \cdot \tanh(c_t) \end{cases} \tag{9}$$

Among them, W_o is the connecting weight of x_t and o_t. U_o is the connecting weight of h_{t-1} and o_t. V_o is the connecting weight of c_{t-1} and o_t. b_o is the bias term.

D. SENTIMENT ANALYSIS OF COMMENTS BASED BILSTM

For overcoming the shortcomings in current comment sentiment analysis methods, a sentiment analysis method of comments based on BiLSTM is proposed in this paper.

In the traditional recurrent neural network model and LSTM model, information can only be propagated in forward, resulting in that the state of time t only depends on the information before time t. In order to make every moment contain the context information, BiLSTM which combines bidirectional recurrent neural network (BiRNN) models and LSTM units is used to capture the context information.

BiLSTM model treats all inputs equally. For the task of sentiment analysis, the sentiment polarity of the text largely depends on the words with sentiment information. In this

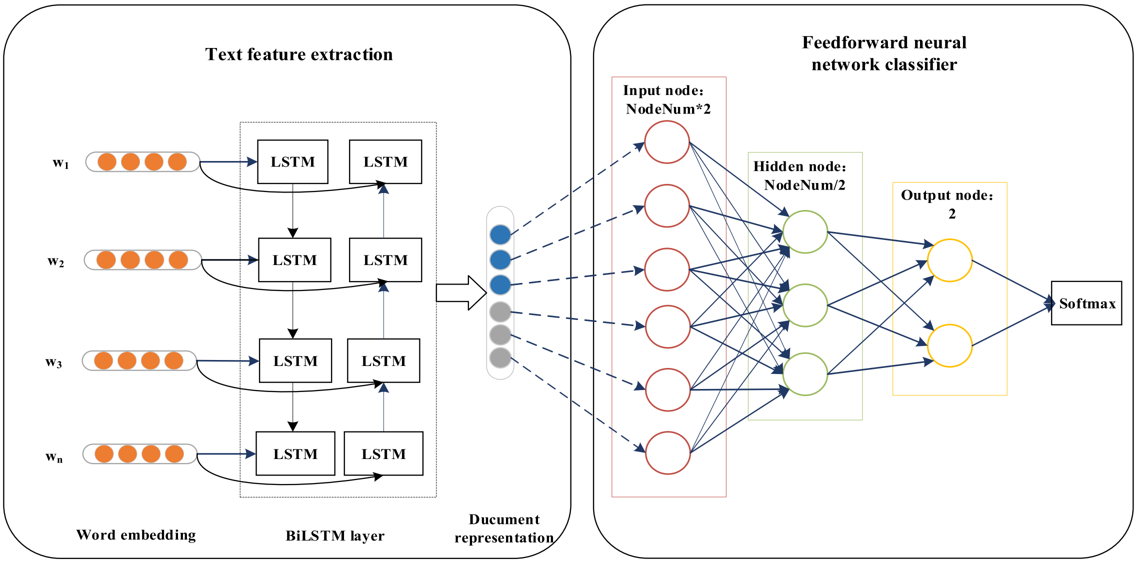

FIGURE 3. The comment sentiment analysis method proposed in this paper.

TABLE 3. Examples of data SET.

Positive	Negative
不错，下次还考虑入住。交通也方便，在餐厅吃的也不错。(I will consider living in this hotel next time. The traffic is convenient, and eating in the restaurant is also not bad.)	酒店比较旧，不符合四星标准，出行不是很方便。(The hotel is old and does not meet the four-star standard. Travel is not very convenient.)
挺好的住的是公寓房，环境不错哦我很喜欢!下次来还打算住那里。（I live in an apartment that is very nice. The environment is good. I like it very much and I plan to choose the hotel next time.）	停车要收费，房间没有窗户，贵，周边餐饮不方便，性价比差！(Parking is charged and the room has no windows. The hotel is too expensive. Meanwhile, there are not many dining nearby.The hotel is low cost performance)
住宿方便，房间清洁，价格实惠，服务尚可。（The accommodation is convenient. The room is clean. The price is affordable. The service is acceptable.）	房间小，进去还一股霉味，电视是老款，问题是还放不出来，这个房间真对不起这个价格。(The room is small and has a musty smell. The TV is old and broken. Therefore, the price of the hotel is expensive.)

paper, the sentiment reinforcement of sentiment word vector is realized. Sentiment analysis tasks are essentially text categorization tasks, and distributed word vectors do not take into account the contributions of different words to the categorization task. In Section A of Research Methods, the weighted word vectors containing sentiment information and classification contribution are constructed.

Firstly, the weighted word vectors are used as the inputs of BiLSTM model, and the outputs of BiLSTM model are used as the representations of the comment texts. Then, the comment text vectors are input into the feedforward neural network classifier. Finally, the sentiment tendency of the comments is obtained. The activation function of feedforward neural network is ReLU function. In order to prevent the over-fitting phenomenon in the training process, dropout mechanism was introduced, and dropout discarding rate was set to 0.5.

The schematic diagram of the sentiment method proposed in this paper is as FIGURE 3. The left subgraph is the process of comment text feature extraction. The right subgraph is the process of obtaining the sentiment polarity of the comment text. Among them, NodeNum refers to the number of nodes of LSTM hidden layer.

IV. EXPERIMENT

A. EXPERIMENTAL ENVIRONMENT

In this paper, the experimental hardware platform is Intel Xeon E5 (6 cores), 32G memory, GTX 1080 Ti. The experimental software platform is Ubuntu 16.04 operating system and development environment is Python3.5 programming language. The Tensorflow library and the Scikit-learn library of python are used to build the proposed sentiment analysis method and comparative experiments.

B. DATA SET

The experimental corpus which has equal number of positive and negative texts includes 15000 hotel comment texts (Data set) crawled from Ctrip (https://www.ctrip.com/). The polarities of the comment texts have been labeled on Ctrip website. Examples of Data set are shown in Table 3.

TABLE 4. Training parameters for word2VEC.

Window size	Dynamic window	Sub-sampling	Low-Frequency word	Iteration	Negative Sampling*
5	Yes	1e-5	10	5	5

TABLE 5. The relevant parameters.

	Labeled as the class	Labeled as other classes
The number of texts that the classifier recognizes as the class	a	b
The number of texts that the classifier recognizes as other classes	c	d

The distributed representations of words are the 300-dimensional word vectors trained by the Skip-Gram model provided by DataScience (https://mlln.cn). The parameters are shown in Table 4.

C. EVALUATION INDICATORS

The evaluation indicators in this paper are precision(P), recall(R) and F_1 score. The relevant parameters are shown in Table 5.

The formula for calculating precision(P), recall(R) and F_1 score is as follows:

$$P = \frac{a}{a+b} \tag{10}$$

$$R = \frac{a}{a+c} \tag{11}$$

$$F_1 = \frac{2 \times P \times R}{P+R} \tag{12}$$

D. HYPERPARAMETERS SETTING OF MODEL

The hyperparameters in the proposed sentiment model include the number of epochs, α value, learning rate, maxLen, nodeNum, and so on. The hyperparameters with the best classification effect of the model are studied. The Data set is randomly divided into a test set and a training set according to a ratio of 1:4. Other network parameters are unchanged, and the hyperparameters are changed to test.

1) EPOCHS

Epochs is the number of iterations of the training set. As Epochs increases, the generalization ability of the model enhances. However, if the number of epochs is too large, overfitting problem is easily generated, and the generalization ability of the model reduces. Therefore, it is important to choose the right Epochs. FIGURE 4 is the classification effect of the model at different Epochs.

It can be seen from FIGURE 4 that with the growth of Epochs, the classification performance F_1 score of the model gradually increases. It tends to be stable when Epochs is 70.

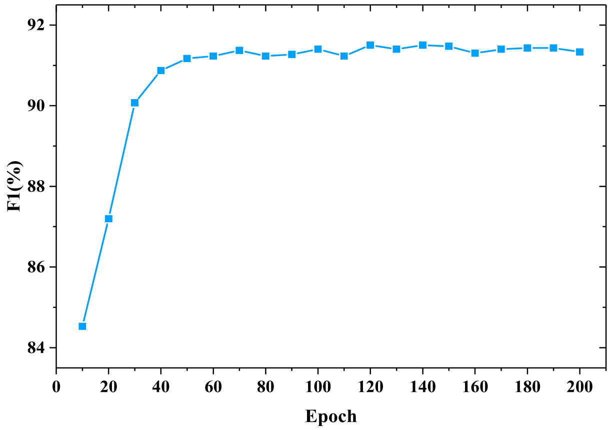

FIGURE 4. Relationship between Epochs and F_1 score.

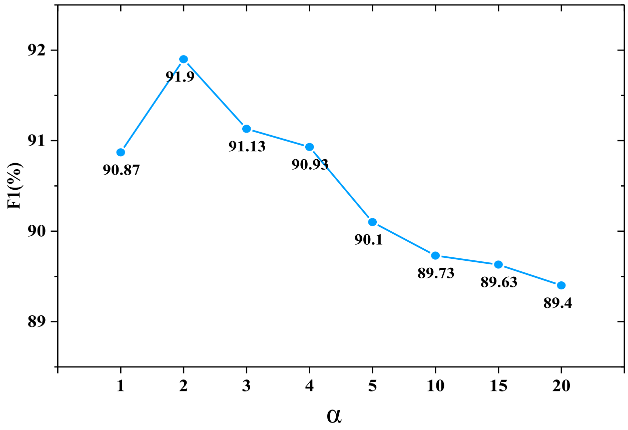

FIGURE 5. Relationship between α values and F_1 score.

2) α VALUE

In this paper, the weight of the word with sentiment information is α, which is the metric of the contribution of sentiment information to the sentiment classification task. If the α value is too small, it cannot fully reflect the difference between sentiment words and non-sentiment words, reducing the effect of sentiment classification; if the α value is too large, it will over-measure the contribution of sentiment information and reduce the accuracy of sentiment classification. FIGURE 5 is the classification effect of the model at different α value.

It can be seen from FIGURE 5 that when the α value is 1, F_1 score of the model is 90.87%; as the α value increases, F_1 score of the model increases first and then decreases. When α values are 2, 3, and 4, the contribution of the sentiment information to the model is integrated into the weight of the word vector, so that F_1 score of the model is higher than the base value of 90.87%; meanwhile, when α value is 2, F_1 score of the model is highest, reaching at 91.90%. When α

value is above 5, the weight difference between the sentiment word and the non-sentiment word is too large, resulting in F_1 score is lower than 90.87%.

3) LEARNING RATE

The appropriate choice of learning rate is important for the optimization of weights and offsets. If the learning rate is too large, it is easy to exceed the extreme point, making the system unstable. If the learning rate is too small, the training time is too long. FIGURE 6 is the classification effect of the model at different learning rates.

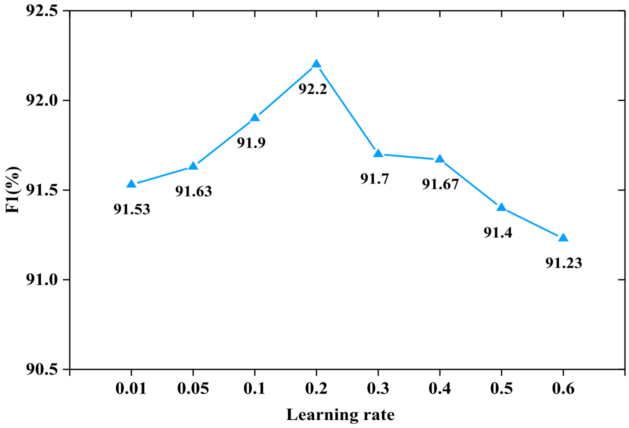

FIGURE 6. Relationship between learning rate and F_1 score.

It can be seen from FIGURE 6 that F_1 scores of the model are around 92%. Moreover, F_1 score reaches a maximum value of 92.2% when the learning rate is 0.2.

4) MAXLEN

In this paper, MaxLen is the number of word vectors input into BiLSTM. If the length of data is greater than MaxLen, the data will be truncated. If the length of data is less than MaxLen, a zero vector is added at the end of the data until the length reaches MaxLen.

The value of MaxLen is related to the input data of the model. If MaxLen is too large, the zero vector in the data is too much filled. If MaxLen is too small, the lost data information is too much. Thus, MaxLen has a great influence on the performance of the model. The distribution of the length of data is shown in FIGURE 7.

It can be seen from FIGURE 7 that data lengths are short and most data lengths are less than 200. Therefore, the maximum value of MaxLen is set 200. FIGURE 8 is the classification effect of the model at different learning rates.

It can be seen from FIGURE 8 that when MaxLen is 20, F_1 score is only 90.27% due to discard too much valid information about the data. When MaxLen increases, F_1 score shows an upward trend. When MaxLen = 100, F_1 score reaches a maximum value 92.20%. When MaxLen continues to increase, F_1 score shows a downward trend. This is because the zero vector is filled too much.

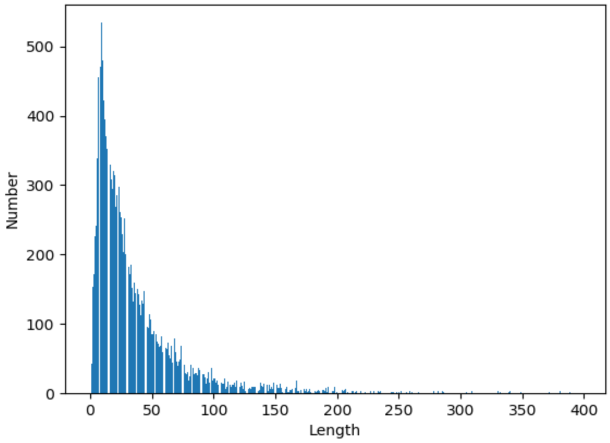

FIGURE 7. Text length distribution in the dataset.

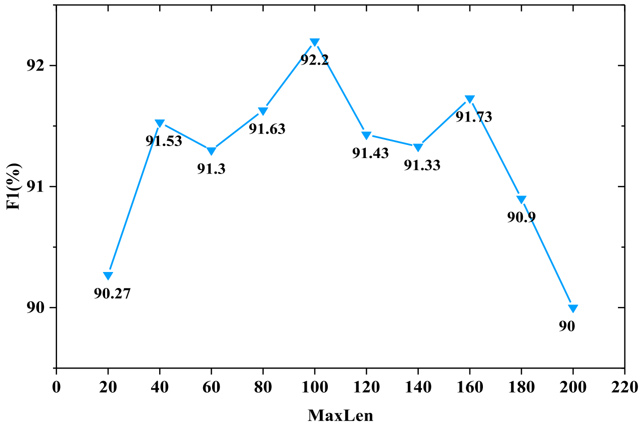

FIGURE 8. Relationship between Maxlen and F_1 score.

5) NODENUM

The number of hidden layer nodes influences on the complexity and effect of the model. If the number of nodes is too small, the network learning ability will be limited. If the number of nodes is too large, the complexity of the network structure is large. At the same time, it is easier to fall into local minimum points during the training process, and the network learning speed will decrease. FIGURE 9 is the classification effect of the model at different NodeNums.

It can be seen from FIGURE 9 that when the number of nodes increases from 32 to 128, F_1 score is improved slightly. When the number of nodes exceeds 128, F_1 score shows a downward trend. Therefore, the number of hidden layer nodes in the model is selected to be 128.

E. COMPARATIVE EXPERIMENTS

1) COMPARISON OF THE SENTIMENT ANALYSIS FOR DIFFERENT WORD REPRESENTATIONS

In order to verify the validity of the word representation proposed in this paper, different word representations are input into BiLSTM model. The effects of sentiment analysis

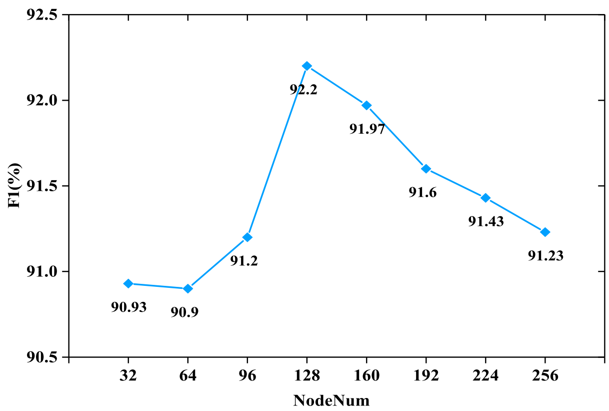

FIGURE 9. Relationship between NodeNum and F_1 score.

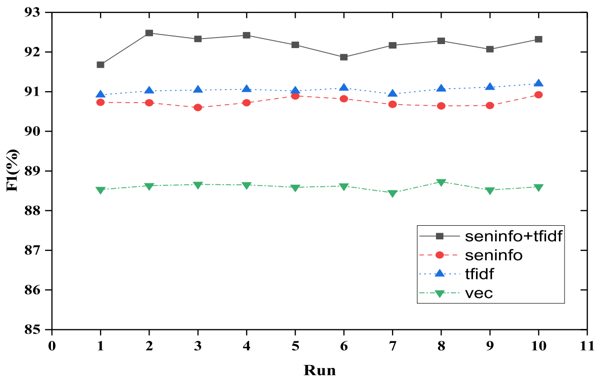

FIGURE 10. F_1 scores of ten experiments for different word representations.

TABLE 6. Hyperparameter list.

hyperparameters	value
Epochs	70
learning rate	0.2
Optimization function	Mini-Batch Gradient Descent
loss function	Cross Entropy Loss Function
MaxLen	100
Dropout	0.5
BatchSize	120
NodeNum	128
VectorSize	300
α value	2.0

TABLE 7. Comparison of various word vector representations.

Word representations	Precision	Recall	F_1 score
Seninfo+TF-IDF	91.50	92.87	92.18
Seninfo	90.85	90.63	90.74
TF-IDF	91.18	90.92	91.05
vec	89.06	88.14	88.60

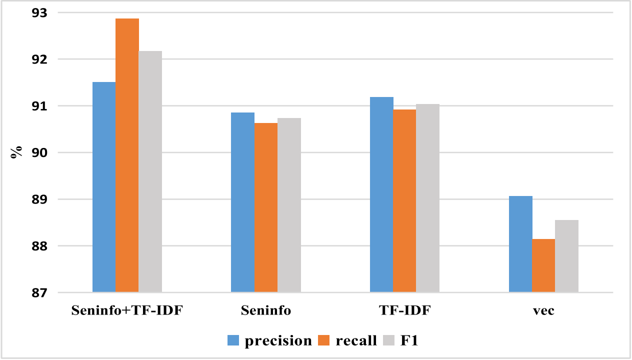

FIGURE 11. Comparison of the sentiment analysis of different word vector representations.

are compared through experiments. The specific parameters of the sentiment analysis model are shown in Table 6.

The vec in this paper refers to the distributed word representation generated by Word2vec containing the semantic information of the words. TF-IDF refers to the weighted distributed word vectors with TF-IDF, which embodies the contribution of different words to the classification task. Seninfo refers to the weighted distributed word vectors with sentiment information, which embodies the difference between sentiment words and other words. Seninfo+TF-IDF refers to the weighted distributed word vectors with TF-IDF and sentiment information.

The Data set is randomly divided into a test set and a training set according to a ratio of 1:4. F_1 scores of ten repeated experiments for different word vector representations are shown in FIGURE 10.

It can be seen from FIGURE 10 that F_1 score is only around 88.5% when seninfo or tfidf is not integrated to word representation; When the tfidf or seninfo is integrated to the weighted word vector, the sentiment analysis effect of comments is significantly improved, reaching around 91%, and integrating tfidf is slightly better than seninfo. After integrating tfidf and seninfo as the weight of word vector, the sentiment analysis has the best effect, and F_1 score is basically above 92%.

The average precision, recall and F_1 score of ten repeated experiments for different word vector representations are shown in Table 7 and FIGURE 11.

It can be seen from Table 7 and FIGURE 11 that the precision, recall and F_1 score of the word representation (Seninfo+TF-IDF) proposed in this paper are superior to other word representation methods. In particular, compared with the distributed word vector trained by Word2vec, the representation method proposed in this paper increases the precision by 2.44 percentage points, the recall increases by 4.73 percentage points, and F_1 score increases by 3.58 percentage points. The reason is that the distributed word vector trained by Wordvec mainly contains the semantic information of words, but cannot contain the sentiment information of words. At the same time, the distributed word vector trained by wordvec cannot reflect the different importance of different words to the classification task. When the word vectors are input into BiLSTM model for sentiment analysis, the degree of discrimination of the words is relatively weak, thus reducing the accuracy of sentiment analysis. The word representation method proposed in this paper takes into account the sentiment information contained in the words and the contribution to the classification task, which alleviates the above problems to some extent.

2) COMPARISON OF THE SENTIMENT ANALYSIS FOR DIFFERENT SENTIMENT ANALYSIS METHODS

In order to further prove the effectiveness of the sentiment analysis method proposed in this paper, the proposed method is compared with other traditional sentiment analysis methods (LSTM, RNN, CNN, Naive Bayesian). The inputs of models are the weighted word vectors proposed in this paper. The hyperparameters of RNN and LSTM are shown in Table 6. The CNN method uses a single channel, and the convolution filter size is set to 5. The Naive Bayesian method uses MultinomialNB with an alpha setting of 2.0.

The Data set is randomly divided into a test set and a training set according to a ratio of 1:4. F_1 scores of ten repeated experiments for different sentiment analysis methods are shown in FIGURE 12.

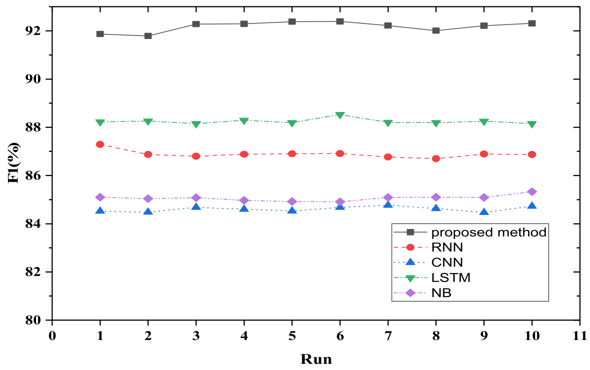

FIGURE 12. F_1 scores of ten experiments for different sentiment analysis methods.

It can be seen from FIGURE 12 that F_1 scores of the above methods are relatively stable in ten experiments. In above five methods, the sentiment classification effects of CNN and Naive Bayesian are poor, and F_1 score is only around 85%. When using RNN and LSTM suitable for sequence modeling, F_1 score can reach around 87%. The method proposed in this paper introduces BiLSTM structure, which can capture the semantic information of the context more effectively, so the sentiment analysis works best. F_1 score is significantly improved, reaching around 92%.

The average precision, recall and F_1 score of ten repeated experiments for different sentiment analysis methods are shown in Table 8 and FIGURE 13.

It can be seen from Table 8 and FIGURE 13 that precision of proposed method reaches 91.54%, recall reaches 92.82%, and F_1 score reaches 92.18%. F_1 scores of the other methods range from 84% to 89%, which are obviously lower than F_1 score of the proposed method. In addition, F_1 scores of RNN and LSTM suitable for sequential processing tasks are higher than those of CNN and Naive Bayesian. The reasons are:

TABLE 8. Experimental results of the proposed method and other traditional methods.

Method	precision	recall	F_1 score
proposed method	91.54	92.82	92.18
RNN	87.18	86.60	86.89
CNN	85.10	84.12	84.61
LSTM	88.46	88.03	88.24
Naive Bayesian	86.02	84.13	85.06

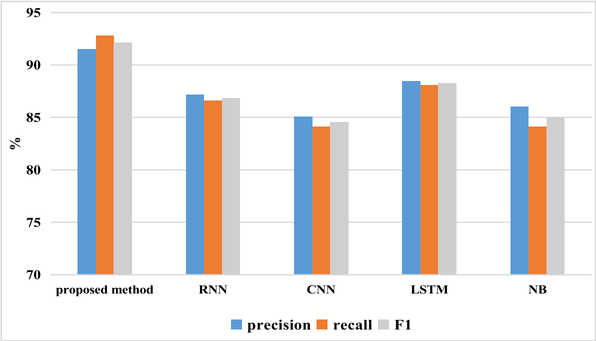

FIGURE 13. Comparison of experimental results between the proposed method and other traditional methods.

① RNN deep learning model can effectively transfer the semantics between words, but there is a gradient disappearance problem; ② CNN deep learning model can mine local information, but the semantic information passed by the sequences cannot be effectively modeled;③ LSTM deep learning model alleviates the gradient disappearance problem to some extent, but it is impossible to capture context semantic information because the information is only transmitted from front to back;④ Naive Bayesian machine learning model has a certain error rate because it determines the probability of posteriority through prior knowledge and data.

The proposed sentiment analysis method uses the improved word representation as input, so that the model can better learn the sentiment information contained in the words and the contribution to the classification task. The introduced BiLSTM model includes the LSTM unit. The gradient disappearance problem is solved to some extent by the gating mechanism. In addition, the forward sequence information and the reverse sequence information are considered, and the semantic information of the context is captured more effectively.

V. CONCLUSION

In the era of rapid development of Internet technology and social networks, it is very meaningful to explore the emotional tendency of comments through artificial intelligence technology. In this paper, a sentiment analysis method of comments based on BiLSTM is proposed and applied to the comment sentiment analysis task. According to the deficiency of the word representation method in the current researches, the sentiment information contribution degree is integrated into the TF-IDF algorithm of the term weight computation, and a new representation method of word vector based on the improved term weight computation is proposed. In addition, BiLSTM model fully considers the context information and can better obtain the text representation of the comments. Finally, through the feedforward neural network and softmax mapping, the sentiment tendency of the text is obtained. The experiments of different word representation methods prove the validity of the proposed word representation method in this paper. Through the comparison experiments with other traditional sentiment analysis methods, the accuracy of the proposed comment sentiment analysis method is improved. However, the sentiment analysis method of comments based on BiLSTM consumes a long time in the training model. In future work, the method to effectively accelerate the training process of the model will be studied.

REFERENCES

[1] L. Wang, D. Miao, and Z. Zhang, "Emotional analysis on text sentences based on topics," *Comput. Sci.*, vol. 41, no. 3, pp. 32–35, Mar. 2014.

[2] S. Krishnamoorthy, "Sentiment analysis of financial news articles using performance indicators," *Knowl. Inf. Syst.*, vol. 56, no. 2, pp. 373–394, Aug. 2018.

[3] N. Shelke, S. Deshpande, and V. Thakare, "Domain independent approach for aspect oriented sentiment analysis for product reviews," in *Proc. 5th Int. Conf. Frontiers Intell. Comput., Theory Appl.*, Singapore, Mar. 2017, pp. 651–659.

[4] P. Sharma and N. Mishra, "Feature level sentiment analysis on movie reviews," in *Proc. 2nd Int. Conf. Next Gener. Comput. Technol. (NGCT)*, Dehradun, India, Oct. 2016, pp. 306–311.

[5] Q. Zhang, S. Zhang, and Z. Lei, "Chinese text sentiment classification based on improved convolutional neural networks," *Comput. Eng. Appl.*, vol. 53, no. 22, pp. 111–115, Sep. 2017.

[6] D. Zhang *et al.*, "Chinese comments sentiment classification based on word2vec and SVMperf," *Comput. Sci.*, vol. 42, no. 4, pp. 1857–1836, Oct. 2016.

[7] L. Kang, L. Xu, and J. Zhao, "Co-extracting opinion targets and opinion words from Online reviews based on the word alignment model," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 636–650, Mar. 2015.

[8] Z. Hao, R. Cai, Y. Yang, W. Wen, and L. Liang, "A dynamic conditional random field based framework for sentence-level sentiment analysis of chinese microblog," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE)*, Guangzhou, China, Jul. 2017, pp. 135–142.

[9] Z. U. Rehman and I. S. Bajwa, "Lexicon-based sentiment analysis for urdu language," in *Proc. 6th Int. Conf. Innov. Comput. Technol. (INTECH)*, Dublin, Ireland, Aug. 2016, pp. 497–501.

[10] A. S. Manek, P. D. Shenoy, M. C. Mohan, and K. R. Venugopal, "Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier," *World Wide Web*, vol. 20, no. 2, pp. 135–154, Mar. 2017.

[11] M. Mubarok, S. Adiwijaya, and M. D. Aldhi, "Aspect-based sentiment analysis to review products using Naïve Bayes," in *Proc. AIP Conf.*, Budapest, Hungary, 2017, pp. 1–8.

[12] M. Bouazizi and T. Ohtsuki, "A pattern-based approach for multi-class sentiment analysis in twitter," *IEEE Access*, vol. 5, pp. 20617–20639, 2017.

[13] P. Turney and M. L. Littman, "Measuring praise and criticism: Inference of semantic orientation from association," *ACM Trans. Inf. Syst.*, vol. 21, no. 4, pp. 315–346, Oct. 2003.

[14] M. Taboada, "Lexicon-based methods for sentiment analysis," *Comput. Linguistics*, vol. 37, no. 2, pp. 267–307, Jun. 2011.

[15] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proc. Conf. Empirical Methods Natural Lang. Process. (ACL)*, Grenoble, France, Jul. 2002, pp. 79–86.

[16] A. B. Goldberg and X. Zhu, "Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization," in *Proc. Workshop Graph Based Methods Natural Lang. Process.*, Sydney, VIC, Australia, Jun. 2006, pp. 45–52.

[17] Y. Wang, X. Zheng, D. Hou, and W. Hu, "Short text sentiment classification of high dimensional hybrid feature based on SVM," *Comput. Technol. Develop.*, vol. 28, no. 2, pp. 88–93, Feb. 2018.

[18] R. Socher, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Seattle, WA, USA, 2013, pp. 1631–1642.

[19] C. D. Santos and M. Gattit, "Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts," in *Proc. 25th Int. Conf. Comput. Linguistics: Tech. Papers*, Dublin, Ireland, 2014, pp. 69–78.

[20] O. Irsoy and C. Cardie, "Opinion Mining with Deep Recurrent Neural Networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Doha, Qatar, 2014, pp. 720–728.

[21] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics*, China, Aug. 2015, pp. 1556–1566.

[22] C. Baziotis, N. Pelekis, and C. Doulkeridis, "Datastories at SemEval-2017 Task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, Vancouver, BC, Canada, Aug. 2017, pp. 747–754.

[23] F. Zhang, C. Hu, Q. Yin, W. Li, H.-C. Li, and W. Hong, "Multi-aspect-aware bidirectional LSTM networks for synthetic aperture radar target recognition," *IEEE Access*, vol. 5, pp. 26880–26891, 2017.

[24] Y. Liu, W. Song, L. Liu, and H. Wang, "Document representation based on semantic smoothed topic model," in *Proc. 17th IEEE/ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput.*, Beijing, China, May/Jun. 2016, pp. 65–69.

[25] L. Zhu, G. Wang, and X. Zou, "A study of chinese document representation and classification with word2vec," in *Proc. 9th Int. Symp. Comput. Intell. Des.*, Hangzhou, China, Dec. 2017, pp. 298–302.

[26] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep convolution neural networks for twitter sentiment analysis," *IEEE Access*, vol. 6, pp. 23253–23260, 2018.

[27] G. E. Hinton, "Learning distributed representations of concepts," in *Proc. 8th Annu. Conf. Cogn. Sci. Soc.*, vol. 1, Aug. 1986, p. 12.

[28] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, no. 6, pp. 932–938, Feb. 2003.

[29] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, Nevada, NV, USA, 2013, pp. 3111–3119.

[30] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Doha, Qatar, Aug. 2014, pp. 1746–1751.

[31] D. Tang, F. Wei, and B. Qin, N. Yang, T. Liu, and M. Zhou, "Sentiment embeddings with applications to sentiment analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 496–509, Feb. 2016.

[32] H. Chen, M. Sun, C. Tu, Y. Lin, and Z. Liu, "Neural sentiment classification with user and product attention," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Austin, TX, USA, 2016, pp. 1650–1659.

[33] J. Liu and Z. Zhang, "Sentiment analysis on food safety news using joint deep neural network model," *Comput. Sci.*, vol. 43, no. 12, pp. 277–280, Dec. 2016.

[34] J. Li. Chinese Sentiment Dictionary. Research Institute Information Technology, Tsinghua University, Beijing, China. Accessed: Oct. 4, 2018. [Online]. Available: http://nlp.csai.tsinghua.edu.cn/site2/index.php/zh/people?catid=13&id=13:v10

[35] W. A. Shu-heng, T. U. Ibrahim, and K. Abiderexiti, "Sentiment classfication of Uyghur text based on BLSTM," *Comput. Eng. Des.*, vol. 38, no. 10, pp. 2879–2886, 2017.

GUIXIAN XU was born in Changchun, Jilin, China, in 1974. She received the B.S. and M.S. degrees from the Changchun University of Technology, in 1998 and 2002, respectively, and the Ph.D. degree in computer software and theory from the Beijing Institute of Technology, in 2010.

Since 2002, she has been a Teacher with the Information Engineering College, Minzu University of China. She is currently an Associate Professor. Her research interests include data mining and machine learning.

YUETING MENG was born in Shijiazhuang, Hebei, China, in 1996. She received the B.S. degree in computer science and technology from the Hebei University of Science and Technology, in 2018. She is currently pursuing the master's degree in software engineering with the Minzu University of China. Her research interests include artificial intelligence, natural language processing, and data mining.

XIAOYU QIU received the M.S. degree in computer science from Shandong Normal University, in 2008. He is currently a Librarian with the Library of Shandong University of Traditional Chinese Medicine. His current research interests include different aspects of pattern recognition, artificial intelligence, and distributed systems.

ZIHENG YU was born in Taizhou, Zhejiang, China, in 1994. He received the B.S. degree in software engineering from Beijing Union University, in 2017. He is currently pursuing the master's degree in software engineering with the Minzu University of China. His research interests include data mining, natural language processing, and artificial intelligence.

XU WU was born in Fenghuang, Hunan, China, in 1993. He received the B.S. degree in software engineering from the Chongqing University of Posts and Telecommunications, in 2017. He is currently pursuing the master's degree in modern education technology with the Minzu University of China. His research interests include data mining, natural language processing, and artificial intelligence.