Hindawi
Scientific Programming
Volume 2020, Article ID 8865983, 13 pages
https://doi.org/10.1155/2020/8865983

Research Article

Chinese Microblog Sentiment Detection Based on CNN-BiGRU and Multihead Attention Mechanism

Hong Qiu,[1] **Chongdi Fan**,[2] **Jie Yao**,[1] **and Xiaohan Ye**[2]

[1]*Chongqing Chuanyi Automation Co., Ltd., Chongqing 401120, China*
[2]*School of Mathematics and Computing Science, Xiangtan University, Xiangtan, Hunan 410011, China*

Correspondence should be addressed to Hong Qiu; 201610111092@smail.xtu.edu.cn

Received 26 August 2020; Accepted 30 September 2020; Published 15 October 2020

Academic Editor: Michele Risi

With the rapid development of the Internet, Weibo has gradually become one of the commonly used social tools in society at present. We can express our opinions on Weibo anytime and anywhere. Weibo is widely used and people can express themselves freely on it; thus, the amount of comments on Weibo has become extremely large. In order to count up the attitudes of users towards a certain event, Weibo managers often need to evaluate the position of a certain microblog in an appropriate way. In traditional position detection tasks, researchers mainly mine text semantic features through constructing feature engineering and sentiment dictionary, but it takes a large amount of manpower in feature selection and design. However, it is an effective method to analyze the sentiment state of microblog comments. Deep learning is developing in an increasingly mature direction, and the utilization of deep learning methods for sentiment detection has become increasingly popular. The application of convolutional neural networks (CNN), bidirectional GRU (BiGRU), and multihead attention mechanism- (multihead attention-) combined method CNN-BiGRU-MAttention (CBMA) to conduct Chinese microblog sentiment detection was proposed in this paper. Firstly, CNN were applied to extract local features of text vectors. Afterward, BiGRU networks were applied to extract the global features of the text to solve the problem that the single CNN cannot obtain global semantic information and the disappearance of the traditional recurrent neural network (RNN) gradient. At last, it was concluded that the CBMA algorithm is more accurate for Chinese microblog sentiment detection through a variety of algorithm experiments.

1. Introduction

Microblog refers to a broadcast social media and network platform based on user relationship information sharing, dissemination, and acquisition, which shares short real-time information through the following mechanism. Users are able to realize instant sharing and communication of information in multimedia forms such as text, pictures, and videos. The most famous microblog application in the world is Twitter [1] in the United States. In China, Sina Weibo, Tencent Weibo, and NetEase Weibo are the microblog applications possessed by tremendous users. With the popularization of the Internet, an increasing number of people begin to use Weibo, and the number of Weibo posts and comments posted by users is increasing as well, which also makes it increasingly difficult for Weibo managers to evaluate the sentiment of a certain microblog in the traditional way. The sentiment trend evaluation of microblog position refers to the judgment of sentiment trend through the analysis of microblog comments.

When we comment on a microblog, we can enter text or insert the emoji provided by Weibo official into the text. Chinese microblog comments are shown in Figure 1(a). Three microblog comments were selected in the figure, which are statements commented by users and statements received by the Weibo background. There will be obvious emoji in user comments, but it does not affect data storage. It is since that when storing these comments, all emoji will be converted into text, as shown in Figure 1(b). Emoji is a way for users to express their sentiment, and we hope to apply this important information when training data.

Deep learning and related technologies are an important research field of experts and scholars. We can train precise models through these techniques from tremendous features

Figure 1: Comparison of microblog comment data format, they should be listed as (a) the format of user comments; (b) the format of the data received by the Weibo server.

[2] and then apply the model to determine the classification result of a piece of data. These deep learning methods include convolutional neural networks, recurrent neural networks, and attention mechanisms. CNN [3] and RNN [4] are common deep learning methods in the field of natural language processing. CNN have the capacity to realize the learning and representation of data sample features well through “end-to-end” learning. However, the recurrent neural network is mostly used to process sequence data in accordance with actual application requirements. In order to be able to memorize longer data sequences, cyclic neural networks have gradually evolved into a long short-term memory (LSTM) networks [5] and a bidirectional long short-term memory (BiLSTM) network [6]. LSTM is a time cyclic neural network that is specially designed to solve the long-term dependency problem of general RNN. Compared with the unidirectional LSTM model, the BiLSTM has the capacity to analyze a large amount of contextual information from the context effectively. As the optimized structure of BiLSTM, BiGRU [7] remains the original effect while making the structure simpler. The application of CNN is able to encode the character information of each word into its character representation, which can extract the character features on the dataset effectively.

A CBMA algorithm with feature templates was proposed in this paper on the basis of the previous description. This algorithm extracts features with CNN firstly and then extracts global features of text with BiGRU, so as to solve the problems of single convolutional neural networks failing to obtain global semantic information and the disappearance of traditional circular neural network gradient. At last, the multihead attention mechanism is applied to improve the accuracy. The CBMA algorithm combines a deep learning algorithm with a manually selected feature template. It is able to achieve better performance than existing models while simplifying the structure simultaneously.

2. Related Operating

Traditional machine learning methods have been applied in classification problems. For example, Li et al. [8] used the XGBoost classification model to analyze the relationship between user characteristics and rumor refuting behavior from the five main rumor categories of economics, sociology, catastrophe, political science, and military science. At the same time, some researchers have improved traditional methods to better classify. For example, Han et al. [9] proposed a Fisher kernel function and FK-SVM method based on probabilistic latent semantic analysis. Fisher kernel function improves the kernel function of support vector machine. Compared with HIST-SVM and PLSA-SVM, the accuracy of FK-SVM method is improved.

With the rapid development of natural language processing, numerous deep learning methods have begun to be applied to classification problems. Kim [10] utilized CNN to perform sentence-level classification tasks on pretrained word vectors. A series of experiments based on word2vec convolutional neural networks presented that convolutional neural networks can be applied to sentence classification tasks well. Dai et al. [11] proposed a black-box backdoor attack to deal with the text classification system based on LSTM. Sentiment analysis experiments were applied to evaluate backdoor attacks. Experimental results presented that a small number of poisoned samples can obtain a higher attack success rate. Ye et al. [12] proposed a Web service classification method based on WiDE and BI-LSTM model. The wide area learning model was applied to realize the width prediction of the Web service category, which captured the interaction between feature vectors of Web service description documents.

In recent years, some researchers have begun to use deep learning methods for sentiment analysis. Li et al. [13] proposed a BiLSTM sentiment classification method based on the self-attention mechanism and multichannel features.

The model consists of two parts: self-attention mechanism and multichannel features. Fu et al. [14] utilized the attention-based CNN-LSTM network to learn general sentence representations in embedded systems and introduced an attention mechanism. Experimental results presented that the CNN encoder is small in size, suitable for small embedded systems, and possessed with excellent performance. Sun et al. [15] proposed a sentiment analysis method for product comment that combines semantic feature mining and dictionary-based technology, which has more advantages than traditional machine learning methods. Yu et al. [16] proposed a word vector refinement model that does not require an annotated corpus, which can be applied to any pretrained word vector. Experiments presented that this method is able to improve the word embedding and sentiment embedding of traditional binary, ternary, and fine-grained sentiment classification. In addition, the performance of various deep neural network models has also been improved. Xu et al. [17] proposed a Chinese sentiment analysis method based on an extended dictionary. The extended sentiment dictionary includes the basic sentiment dictionary, part of the field sentiment words, and polysemous field sentiment words. The naive Bayesian field classifier was applied to classify the text field where the polysemous sentiment word was located, so as to distinguish the sentiment polarity of the word. Chanlekha et al. [18] developed a semiautomatic sentiment dictionary construction tool for sentiment analysis in Thai. This method utilized sentiment cooccurrence and contextual consistency features to propagate sentiment polarity to unresolved sentiment feature pairs. Naseem and Musial [19] proposed a sentiment analysis method DICET based on a converter. This method improves the quality of tweets through encoding the representations in the converter and applying deep intelligent context embedding. At the same time, the emotional, polysemous, syntactic, and semantic knowledge of words are taken into consideration. Sailunaz and Alhajj [20] detect and analyze the sentiment people express in their Twitter posts and utilize them to generate recommendations.

From the above researches, it was found that popular machine learning methods such as logistic regression, SVM, and XGBoost have achieved certain results in the early stage of sentiment analysis research. However, due to its weak feature extraction capability and nonlinear fitting capability, it was difficult to adapt to the current sentiment analysis problems in the big data environment. Nevertheless, most sentiment analysis researchers based on deep learning often failed to take the context relation into consideration when using CNN for sentiment classification task, while the LSTM model can only consider the above relation with a slow convergence rate. The BiGRU model with bidirectional sequential structure happened to be able to solve this problem. However, the direct application of the BiGRU model may cause excessive computational overhead due to the excessively high input dimension. Therefore, the application of CNN was taken into consideration to reduce the dimension of the word vector matrix formed by the original data in this paper and then integrated the BiGRU model for sentiment analysis. At last, a multihead attention mechanism was introduced to further improve the operating efficiency and prediction accuracy of the model.

3. Materials and Methods

CBMA algorithm was applied in Chinese microblog sentiment detection in this paper. This algorithm model is composed of CNN, BiGRU, and attention mechanism. In order to elaborate the combined model in more detail, the aspects of word embedding, CNN, BiGRU, multihead attention mechanism, and CNMA algorithm model structure were introduced in this paper, respectively.

3.1. Word Embedding. Word embedding is a general term for language model and representation learning technology in natural language processing (NLP). It refers to embedding a high-dimensional space whose dimension is the number of all words into a continuous vector space with a much lower dimension [21], and each word or phrase is mapped to a vector on the real number field. Word embedding methods include artificial neural networks, dimension reduction of word cooccurrence matrix, probability model, and explicit representation of the word in context. In the underlying input, the method of applying word embedding to represent phrases has greatly improved the effect of the parser and text sentiment analysis on NLP [22].

Chinese microblog comments are usually a continuous Chinese sentence. In order to better train the deep learning model, it is necessary to decompose comments into multiple words and then train the words into word vectors. The data processing stage process of this paper is shown in Figure 2. The dataset utilized in this paper is weibo_senti_100 k, which is a CSV file containing two columns. The sentiment state of the comment is indicated with 1 or 0 in the first column, and the second column is the content of the comment. In order to facilitate the calculation, the data was converted into a TXT file. The first column and the second column were separated by the TAB key and then the word segmentation was performed. English sentences consist of multiple English words, and each word is separated by a space. Although Chinese sentences are also composed of multiple Chinese words, the adjacent words are closely connected and there is no separator. It is necessary to split the sentence into multiple words if you would like to train words as the most basic input in deep learning. A stammering tokenizer was applied commonly for word segmentation. After the word segmentation is completed, the stop terms need to be removed and then carry out the word vector training.

Word2Vec was applied to train Chinese word vectors commonly. Word2Vec is a word embedding model proposed by Mikolov in 2013. It can be utilized for word vector calculation and word vector generation. The algorithm used by Word2Vec is a shallow neural network with a layer number of 3. The word vector generated can be applied as input to other neural networks in numerous tasks. Word2Vec mainly includes two models: CBOW (continuous bag of words) and Skip-gram. CBOW generates the current headword from the context information of the word

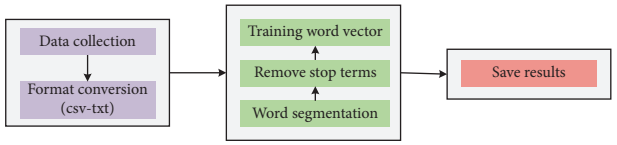

Figure 2: Data processing process of Chinese microblog comment.

[23], while Skip-gram generates its contextual words from the current headword. In this way, word vectors containing certain semantic parameters can be obtained. The word2Vec model utilized in this paper is the Skip-gram model.

Skip-gram model consists of a three-layer structure of input layer, mapping layer, and output layer. The content shown in Figure 3 is the architecture diagram of the model.

In terms of a known word w_t $(w_{t-n}, w_{t-n-1}, \ldots, w_{t-1}, w_t, w_{t+1}, \ldots, w_{t+n-1}, w_{t+n})$, there are $2n$ context words as target words, and the probability of achieving the target word is $p(\text{count}(w)|w)$. Its objective function is shown in formula (1):

$$F = \sum_{w \in N} \log_2^{p(\text{Count}(w)|w)}. \tag{1}$$

In terms of the Skip-gram model, the idea is to generate (input and output) datasets. First of all, a dataset is established for the known words and their context and the window size is set. Afterward, such a dataset is generated by combining the input words and the target words of window size.

3.2. CNN. CNN is one of the representative algorithms of deep learning algorithm [24]. It includes convolution calculation and is a feedforward neural network with deep structure. Scientists have been working on convolutional neural networks since the 1980s and 1990s. After entering the 21st century, CNN have developed rapidly with the introduction of deep learning theory and the improvement of computer equipment, and people have begun to apply CNN to computer vision and natural language processing.

CNN was constructed through imitating the biological visual perception mechanism, which is able to perform supervised learning and unsupervised learning. The convolution kernel parameter sharing in the hidden layer and the sparsity of the connections between layers enable the CNN to obtain lattice point features with a small amount of calculation.

The structure of the CNN is shown in Figure 4. The whole structure is composed of input layer, convolutional layer, pooling layer, and fully connected layer.

Each input in the input layer is a sentence [25]. However, this sentence is a sentence after word segmentation. In addition, the input is the word vector of each word in the sentence and one word vector corresponds to one row of the input layer in the above figure. Suppose that the comment text sentence is preprocessed into n words, each word is converted into a vector through Word2Vec word embedding, which is mapped into an m-dimensional vector, and the word sequence in the sentence is spliced and mapped into $n \times m$-dimensional matrix:

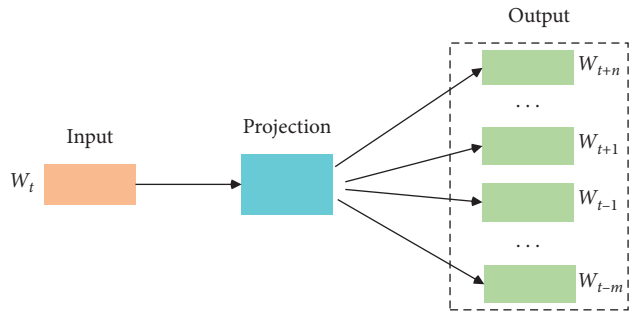

Figure 3: Skip-gram model.

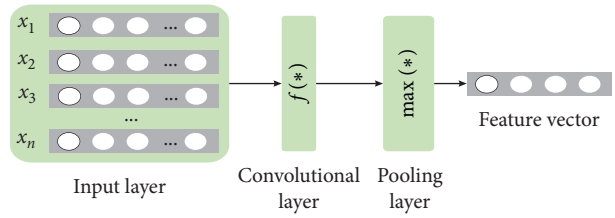

Figure 4: Structure diagram of CNN.

$$\begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1m} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ x_{n1} & x_{n2} & \ldots & x_{nm} \end{bmatrix}. \tag{2}$$

The convolutional layer performs convolution calculation on the input through the convolution kernel to obtain the feature map. One convolution kernel is one feature extractor, and a plurality of convolution kernels are a plurality of feature extractors. In order to better use convolution kernels to extract features, multiple convolution kernels are utilized to conduct feature extraction generally. $[P_1, P_2, P_3, \ldots, P_z]$ is used to represent the combination containing z convolution kernels, where P_z represents the size of the z-th convolution kernel, that is, the longitudinal dimension of the convolution kernel window. The horizontal size of the convolution kernel window is the vector dimension of the word vector. Through the calculation of z convolution kernels, z feature map vectors will be obtained. In terms of the sentence information as $n \times m$-dimensional matrix, assuming that the size of the convolution window is h, the size of the convolution kernel will be $k \times m$. Specifically, slide k words in accordance with the step length t, and apply the convolution kernel to perform the convolution operation to extract the local features of the text on the input word windows x_1^h, x_2^{h+1}, x_3^{h+2}, ..., x_{n-h+1}^n. Assuming that the input sentence d is composed of n word vectors $x_1, x_2, \ldots, x_n$, the operation of the convolutional layer can be expressed as

$$y_i = f\left(W \cdot x_{i:\, i+h-1} + b\right), \tag{3}$$

where $x_{i:\, i+h-1}$ refers to the combination of vector $x_i, x_{i+1}, \ldots, x_{i+h-1}$, h refers to the dimension of the convolution kernel, $f(*)$ refers to the nonlinear function, W refers to the weight matrix, and b refers to the bias vector.

The eigenvector y obtained after convolution kernel extraction is

$$y = \{y_1, y_2, y_3, \ldots, y_{n-h+1}\}. \tag{4}$$

After the convolution operation, the pooling layer performs pooling processing on each eigenvector, and a multidimensional vector is converted into a value after pooling processing, which is used as an element of the pooled vector. The pooling method used by the pooling layer is the maximum pooling method; that is, the sequence output from the convolutional layer is input to the pooling layer. The maximum pooling method will select the largest element in the sequence $y_1, y_2, y_3, \ldots, y_{n-h+1}$ and eventually obtain a new vector y:

$$y = \max(y_i). \tag{5}$$

3.3. BiGRU. Gated Recurrent Unit (GRU) was proposed by Cho et al. [26], which is a kind of RNN. Similar to LSTM, it is also proposed to solve problems such as long-term memory and gradients in backpropagation. RNN is a class of recurrent neural networks which performs recursion in the evolutionary direction of sequences with sequential data as input, and all neurons are connected in a chain. Due to the addition of cyclic factors in the hidden layer, neurons are able to receive information from their own historical moments as well as other neurons at the same time. Therefore, RNN has the characteristics of memory and parameter sharing. In addition, RNN is superior in the nonlinear feature learning of serial data [27]. In terms of the problem of RNN gradient disappearing and being unable to learn long-term historical load features, scholars proposed LSTM, which has the capacity to learn the correlation information between long short-term sequence data. In recent years, in response to the problem of LSTM with excessive parameters and slow convergence rate [28], GRU has been derived. GRU is a variant of LSTM, which has fewer parameters and has been possessed with faster convergence performance while maintaining good learning performance of LSTM. The GRU model is internally composed of updating gate and resetting gate. Different from LSTM, GRU replaces the input gate and forgetting gate of LSTM with updating gate, where the updating gate represents the influence of the output information of the hidden layer neurons at the previous moment on the hidden layer neurons at the current moment. When the updating gate value is larger, the influence degree is greater. The resetting gate represents the neglect degree of the hidden layer neuron output at the previous moment. When the value of the resetting gate is larger, the less information is ignored. The structure of GRU is shown in Figure 5.

The hidden layer unit A can be calculated by the following formula:

$$\begin{aligned} z_t &= \sigma(W_z \cdot [h_{t-1}, x_t]), \\ r_t &= \sigma(W_r \cdot [h_{t-1}, x_t]), \\ \widetilde{h}_t &= \tanh(W \cdot [r_t * h_{t-1}, x_t]), \\ h_t &= (1 - z_t) * h_{t-1} + z_t * \widetilde{h}_t, \end{aligned} \tag{6}$$

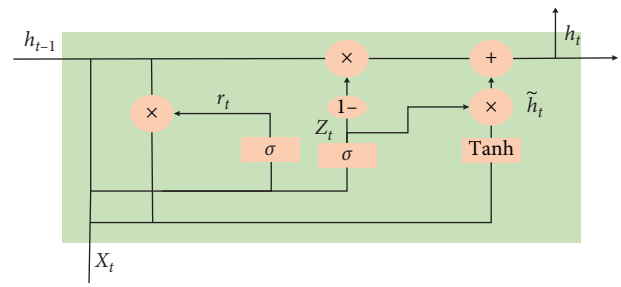

Figure 5: Structure diagram of GRU.

where z_t and r_t are the updating gate and resetting gate, respectively; σ is the Sigmoid function; tanh is the hyperbolic tangent function; W_r, U_r, W_z, U_z, and U are all training parameter matrices. The candidate activation state $\widetilde{h}_t$ at the current moment is jointly determined by the resetting gate r_t, the output h_{t-1} of the hidden layer neuron at the previous moment, the input x_t at the current moment, and the training parameter matrices W and U.

BiGRU network has the capacity to learn the relationship between past and future load influencing factors and current load, which is more conducive to extracting the deep features of load data [29]. The structure of BiGRU is shown in Figure 6.

It is calculated as

$$y_2 = g(VA_2 + V'A_2'), \tag{7}$$

and A_2' are calculated as

$$\begin{aligned} A_2 &= f(WA_1 + Ux_2), \\ A_2' &= f(W'A_3' + Ux_2). \end{aligned} \tag{8}$$

In the forward calculation, the hidden layer value s_t is related to s_{t-1}. In the reverse calculation, the hidden layer value s_t is related to s_{t-1}. The final output depends on the sum of the forward and reverse calculations. The calculation method of the bidirectional recurrent neural network is

$$\begin{aligned} o_t &= g(Vs_t + V's_t'), \\ s_t &= f(Ux_t + Ws_{t-1}), \\ s_t' &= f(U'x_t + W's_{t-1}'). \end{aligned} \tag{9}$$

3.4. Cross-Entropy Loss Function. The cross-entropy loss function is often applied for classification problems, especially for the classification problem in neural networks [30], and the cross entropy is used as the loss function frequently. In addition, since cross-entropy involves calculating the probability of each category, cross entropy appears with the Sigmoid (or softmax) function [31] almost every time. The expression of the Sigmoid function is as follows:

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \tag{10}$$

After deriving the Sigmoid function $\sigma(z)$, the following function will be obtained:

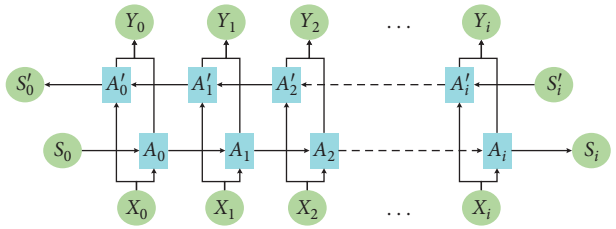

Figure 6: Structure diagram of BiGRU.

$$\sigma'(z) = \frac{e^{-z}}{(1+e^{-z})^2} = \delta(z)(1-\delta(z)). \tag{11}$$

When the value of x is larger or smaller, the curve of the Sigmoid function will be more smooth, which indicates that the derivative $\sigma'(x)$ is closer to zero. In the case of dichotomy, there are only two cases where the model needs to predict in the end [32]. The predicted probabilities are p and $1-p$ for each of these categories. At this time, the cross-entropy loss function can be expressed as

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -[y_i \cdot \log(p_i) + (1-y_i)\cdot \log(1-p_i)], \tag{12}$$

where y_i represents the label of sample i, positive class is 1, negative class is 0, and p_i represents the probability that sample i is predicted to be positive.

Learning tasks were divided into dichotomy and polychotomy cases. The learning processes of these two situations were discussed, respectively. Take a gradient descent of a single sample as an example:

$$\begin{aligned} z &= Wx + b, \\ \hat{y} &= a = \sigma(z), \\ L_1(y,a) &= \frac{1}{2}(y-a)^2, \\ L_2(y,a) &= -(y\log(a) + (1-y)\log(1-y)). \end{aligned} \tag{13}$$

The first two formulae are the linear and nonlinear parts of the forward propagation, respectively. The third formula is the mean square error loss function. The fourth formula is the cross-entropy loss function. The purpose of gradient descent, explicitly, is to reduce the distance between the true value and the predicted value. However, the loss function is applied to measure the distance between the true value and the predicted value. Therefore, the purpose of gradient descent is to reduce the value of the loss function. How to reduce the value of the loss function? The variables are only w and b; thus, what we have to do is to constantly modify the values of w and b to make the loss function increasingly smaller.

The renewal process of w and b is as follows:

$$\begin{aligned} w &= w - \alpha\frac{\partial L(y,a)}{\partial w}, \\ b &= b - \alpha\frac{\partial L(y,a)}{\partial w}, \end{aligned} \tag{14}$$

where α represents the learning rate, which is used to control the step length, that is, the length of one step down.

3.5. Multihead Attention Mechanism. The visual attention mechanism is a brain signal processing mechanism unique to human vision. Human vision scans the global image quickly to obtain the target area that needs to be focused on, which is commonly known as the focus of attention. Afterward, more attention resources are devoted to this area to obtain more detailed information about the target that needs to be paid attention, thus inhibiting other pieces of useless information [33]. This is a means for human beings to quickly screen out high-value information from a large amount of information with limited attention resources. It is a survival mechanism formed in the long-term evolution of human beings. The human visual attention mechanism greatly improves the efficiency and accuracy of visual information processing.

In recent years, the attention mechanism has been widely applied in various fields of deep learning. Attention mechanisms are commonly applied in different types of tasks, whether image processing, speech recognition, or natural language processing. Therefore, it is necessary to understand the working principle of the attention mechanism for technicians who are concerned about the development of deep learning technology.

In the task of Chinese text classification, it is necessary to pay attention to the word vector vectors of key Chinese words and ignore the word vectors which are not related to the context. Adopting the attention mechanism to the input text data enables the word vectors of key Chinese words to become the dominant information, thereby improving the efficiency and accuracy of the entire neural network model. The attention mechanism will also prompt the model to focus on the Chinese words when similar sentences appear again in the future and improve the learning and generalization capabilities of the model. The self-attention mechanism is a special case of the general attention mechanism. Q, K, and V are applied to represent the attention-related query matrix, key matrix, and value matrix, respectively. In the self-attention mechanism, $Q = K = V$. Its advantage lies in that it ignores the distance between words and directly calculates the dependency relationship. In addition, it has the capacity to learn the internal structure of a sentence [34] and pay attention to the connection between its internal words. Combined with RNN, the application of CNN model is conducive to improving model learning ability and enhancing the interpretability of the neural network.

The basic structure of multihead attention is shown in Figure 7. The scaled dot-product attention at the central position is a variant of the general attention. Given matrices $Q \in R^{n*d}$, $K \in R^{n*d}$, and $V \in R^{n*d}$, scaled dot-product attention can be calculated by the following formula:

$$\text{Attention}(Q,K,V) = \text{soft max}\left(\frac{QK^T}{\sqrt{d}}\right)V, \tag{15}$$

where d is the number of hidden units in the neural network. Multihead attention adopts the self-attention mechanism,

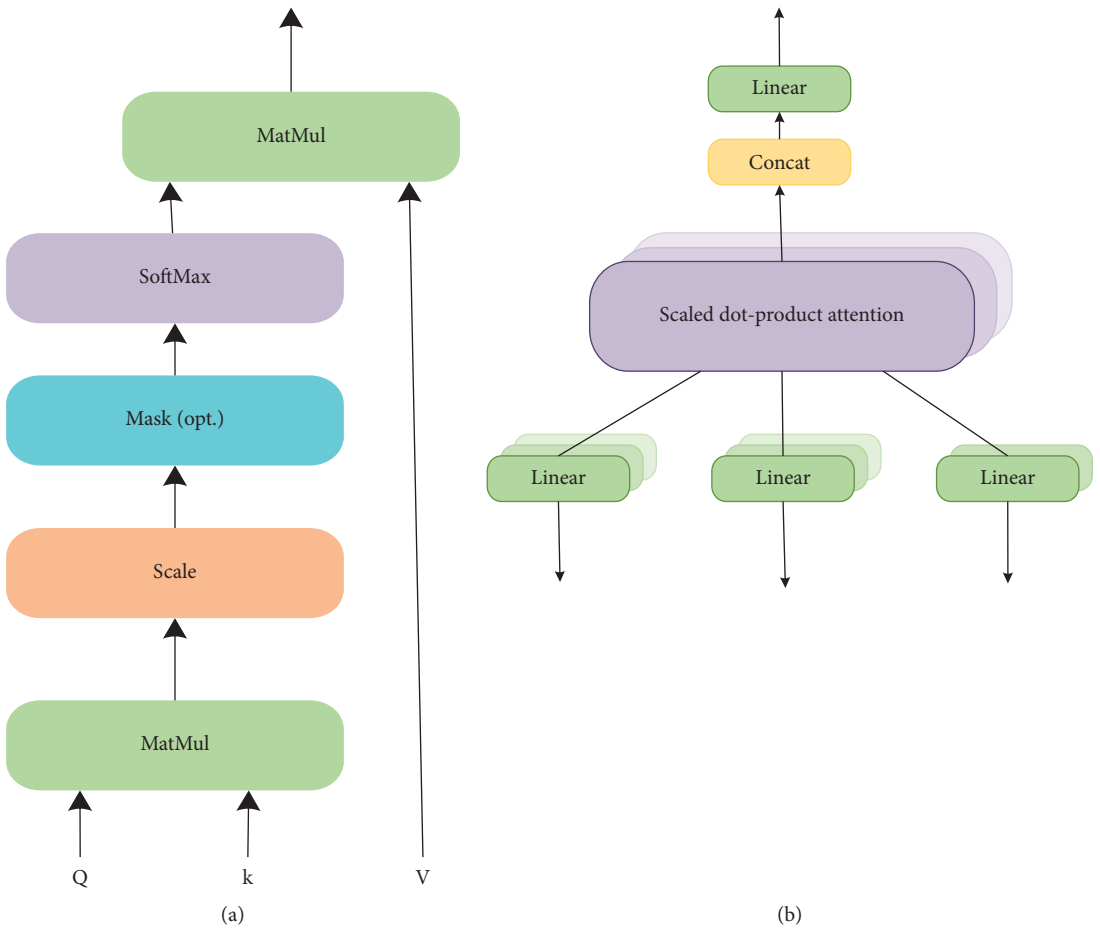

Figure 7: Multihead attention mechanism model. (a) Self-attention structure; (b) multihead attention structure.

which means $Q = K = V$ in the figure. The advantage of this is that the information of the current position and the information of all other positions can be calculated to capture the dependencies within the entire sequence. For example, if the input is a sentence, each word in it must be attention calculated with all words in the sentence.

In this model, multihead attention performs linear transformation on the inputs Q, K, and V, three vectors, before performing calculations. Since it is a "multihead attention" mechanism, the calculation of the scaled dot-product attention part needs to be performed for numerous times. The number of "heads" means the number of calculations, but the linear projections of Q, K, and V are different for each head calculation. Take the i-th head as an example:

$$\begin{aligned} Q' &= Q * W_i^Q, \\ K' &= K * W_i^K, \\ V' &= V * W_i^V. \end{aligned} \tag{16}$$

Since this layer receives the output of the BI-GRU layer, therefore

$$Q = K = V = y_t. \tag{17}$$

The final result of this head is

$$M_i = \text{soft max}\left(\frac{Q'K'^T}{\sqrt{d}}\right)V'. \tag{18}$$

3.6. CBMA Model. The CBMA model is shown in Figure 8. Before the model training, microblog comments were segmented into words. Afterward, microblog comments were converted into word vectors through word2Vec embedding, and the trained word vectors were taken as the input of the model. A convolutional network was applied firstly in this model to perform feature extraction on the input word vector. The output after convolution feature extraction was utilized as the input of BiGRU. The BiGRU was followed by an attention mechanism module, which then performed pooling processing through the maximum pooling layer and was connected to a fully connected layer. At last, the sigmoid classification function is applied for classification. In addition, the cross-entropy loss function is utilized to evaluate the model, and the category of the input sentence will be obtained in the end.

4. Results

4.1. Evaluation Index. Accuracy, precision, recall, and F-measure (F1) were applied as the evaluation indexes of the model in this paper. Accuracy is the score of sentiment

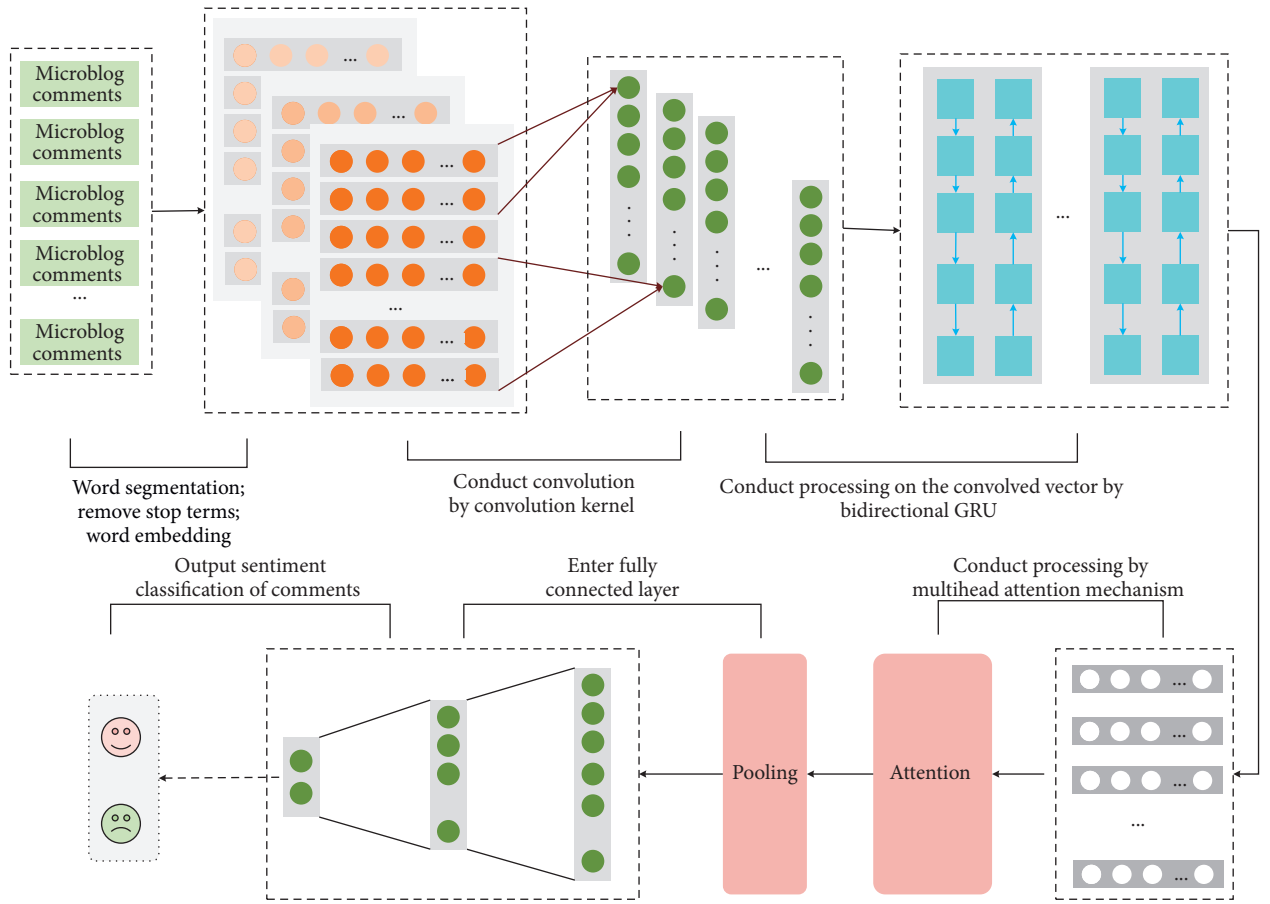

Figure 8: Structure diagram of CBMA algorithm model.

correctly predicted in all microblog comments [35], which is the percentage of examples that the classifier obtains from the total number of examples predicted by a given label. The precision is the fraction of relevant instances among all retrieved instances. The recall rate is the fraction of the total amount of relevant instances that are actually retrieved. F-measure (F1) is the harmonic average of accuracy and recall rate. Their calculation formulae are shown as follows:

$$\begin{aligned} \text{Accuracy} &= \frac{T_P + T_N}{T_P + F_N + F_P + T_N}, \\ \text{Precision} &= \frac{T_P}{T_P + F_P}, \\ \text{Recall} &= \frac{T_P}{T_P + F_N}, \\ F_1 &= \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}, \end{aligned} \tag{19}$$

where T_P represented the number of positive evaluation samples correctly predicted in positive evaluation samples, F_P represented the number of positive evaluation samples incorrectly predicted in negative evaluation samples, F_N represented the number of negative evaluation samples incorrectly predicted in positive evaluation samples, and T_N represented the number of negative evaluation samples correctly predicted in negative evaluation samples.

4.2. Dataset. The dataset utilized in this experiment is the microblog comment corpus weibo_senti_100 k from Sina Weibo, which contains sentiment annotation data. All the data performed sentiment annotation. The distribution of positive and negative data in the dataset is shown in Table 1.

The dataset has a total of 119988 data, including 59,994 pieces of positive comment data and 59,994 pieces of negative comment data. The datasets were segmented and the positive and negative data were divided into training data, test data, and verification data in a certain proportion. The specific ratio of the division is shown in Table 2.

4.3. Experimental Results

4.3.1. Experiment with Different Convolution Kernels. In the training model, different numbers and sizes of convolution kernels were utilized to test the model. The test results are shown in the following table. [1, 3, 5] in the following table indicated that the convolution kernel with a size of 3 and a step size of 1, 3, and 5 was utilized to test the model. Different sizes and different numbers of convolution kernels will affect the efficiency of model training and the accuracy of

Table 1: The positive and negative dataset distribution in the dataset.

	Dataset	Percentage
Positive	59994	0.5
Negative	59994	0.5

Table 2: The distribution of the training set, test set, and validation set of the dataset.

	Training dataset	Testing dataset	Val. dataset	Total number of SMS
Positive	10000	39994	10000	59994
Negative	10000	39994	10000	59994
Percentage	80%	10%	10%	100%

experimental results. In terms of this problem, 7 convolution kernels with different quantities and sizes were utilized for experimental testing in this paper. The accuracy, precision, recall rate, and F1 value of the test results are shown in Table 3.

From the experimental data in the above table, it can be seen that when four convolution kernels [1, 3, 5, 7] were utilized, the accuracy, precision, recall rate, and F1 value of the model were slightly higher than other convolution kernel combination methods. An analysis of the experimental process of these 7 types of convolution kernels is shown in Figure 9. First of all, the variation trend of the accuracy of verification set during training is presented.

The six figures above correspond to the experimental results of convolution kernel allocation method experiments in Table 4. These results mainly include the changes of the training set with the number of iterations and the change of accuracy of the validation set with the increase of iteration times.

4.3.2. Experiments with Different Layers of BiGRU. In the CBMA model utilized in this paper, the number of layers of the BiGRU can be one layer or multiple layers. The optimal number of layers was selected through experimental comparison. In order to verify the influence of the number of BiGRU layers on this model and to find an optimal number of BiGRU layers, multiple numbers of two-way GRU layers were applied to conduct experiments. The experimental results are shown in Table 5.

With the increase in the number of BiGRU, the training time also changed. The change in the training time of each round after changing the BiGRU in this experiment is shown in Figure 10.

4.3.3. Experiments with Different Learning Rates. As an important hyperparameter in supervised learning and deep learning, learning rate determines whether the objective function converges to the local minimum and when it converges to the minimum. A suitable learning rate can make the objective function converge to the local minimum in a suitable time. If the learning rate is too small, the convergence will be excessively slow. If the learning rate is too large, it will cause the cost function oscillation. In order to find an optimal learning rate, experiments with multiple learning rates were conducted. The accuracy, precision, recall, and F1 value after the experiment were recorded. The experimental results are shown in Table 6.

Table 3: Experimental results under different convolution kernels conditions.

	Accuracy	Precision	Recall	F1
[1, 3]	0.9750	0.9891	0.9605	0.9746
[1, 3, 5]	0.9745	0.9869	0.9617	0.9741
[1, 3, 5, 7]	0.9761	0.9894	0.9624	0.9757
[1, 3, 5, 7, 9]	0.9751	0.9880	0.9618	0.9747
[1, 3, 5, 7, 9, 11]	0.9753	0.9845	0.9658	0.9751
[1, 3, 5, 7, 9, 11, 13]	0.9756	0.9893	0.9616	0.9753

When the model conducted training under these five learning rates, as the number of iterations increased, the accuracy changes were shown in Figure 11. The change in the loss function is shown in Figure 12.

4.3.4. Experiments with Different Learning Rates. A variety of methods would be carried out to test. These methods included traditional machine learning algorithms, such as decision tree, KNN, Naive Bayes, random forest, GBDT, SVM, and logistic regression. The results of the experiment also recorded accuracy, precision, recall rate, and F1 value. These four evaluation indexes were applied to evaluate each model. The experimental results are shown in Table 4.

At the same time, some deep learning algorithms have also been tested, such as the combined model of GRU and multihead attention, BiGRU and multihead attention mechanism model, and convolution and GRU multihead attention model. The results of the experiment also recorded accuracy, precision, recall rate, and F1 value. These four evaluation indexes were applied to evaluate each model. The experimental results are shown in Table 7.

5. Discussion

5.1. Subsection

5.1.1. Experiment Analysis of Experiment 1. In order to verify the influence of the convolutional network layer on the CBMA algorithm model in the feature extraction process, the convolution kernel with the same number and size was applied to carry out experiments. The experimental results are shown in Table 3. It can be seen from the experimental results that the accuracy of the model increased

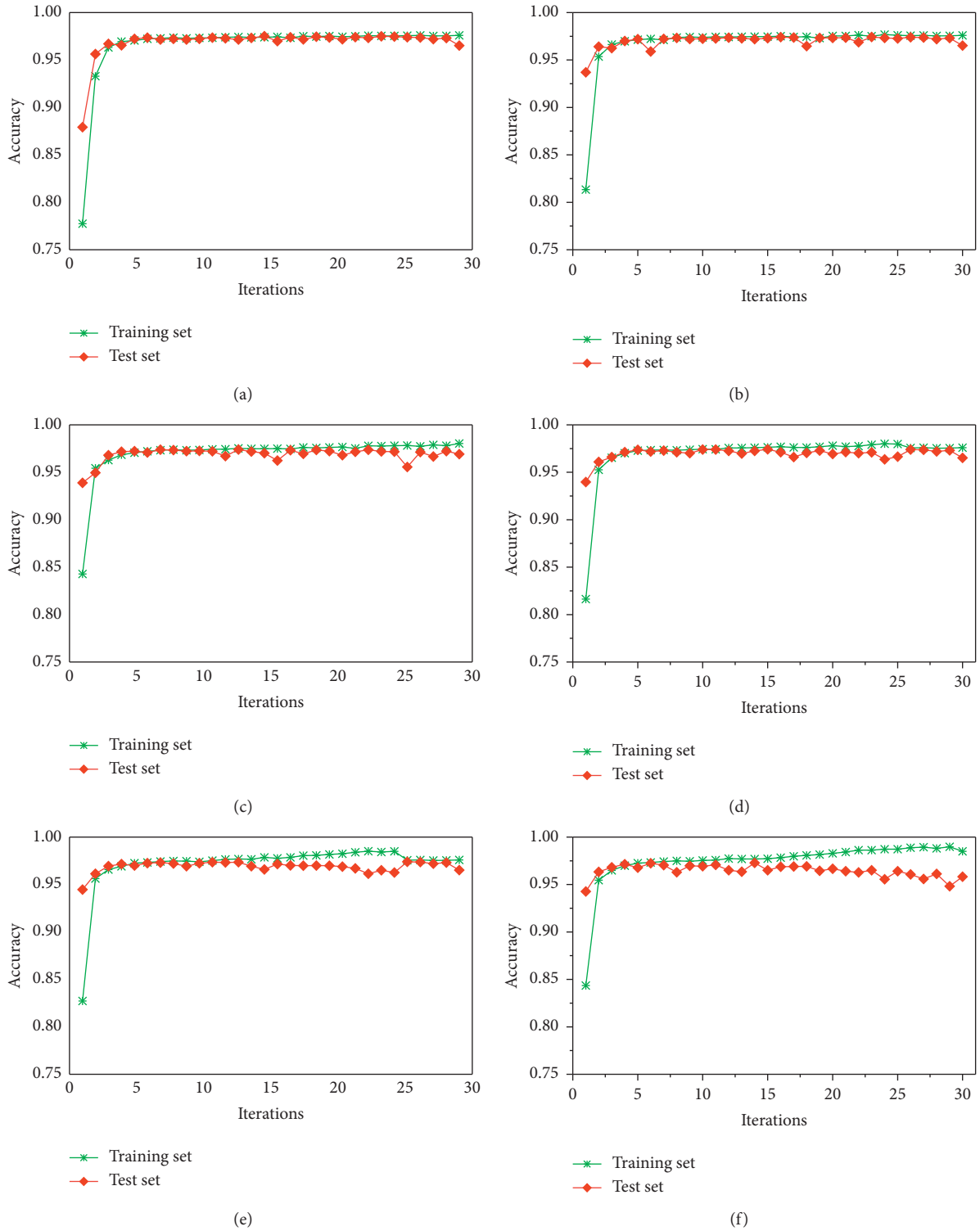

Figure 9: Experiments with different convolution kernels, the accuracy of training set, and validation set variations with the number of iterations. (a) Convolution kernel: [1, 3], (b) convolution kernel: [1, 3], (c) convolution kernel: [1, 3, 5], (d) convolution kernel: [1, 3, 5, 7], (e) convolution kernel: [1, 3, 5, 7, 9, 11], and (f) convolution kernel: [1, 3, 5, 7, 9, 11, 13].

firstly and then decreased with the increase of the number of convolution kernels. When the number and size of the convolution kernel was [1, 3, 5, 7], the accuracy of the model was the highest.

5.1.2. Contrast Test of Multilayer BiGRU. The multilayer BiGRU test results are shown in Table 5, and the model training time is shown in Figure 12. It can be seen from Table 5 that as the number of GRU layers increased, the

Table 4: Comparison of test results with various traditional machine learning methods.

	Accuracy	Precision	Recall	F1
Decision tree	0.7247	0.7277	0.7182	0.7229
KNN	0.8048	0.8290	0.7679	0.7973
Naive Bayes	0.8151	0.8252	0.7995	0.8122
Random forest	0.8455	0.8434	0.8485	0.8459
GBDT	0.8684	0.8640	0.8745	0.8692
SVM	0.8758	0.8682	0.8861	0.8771
Logistic regression	0.8890	0.8820	0.8981	0.8900
CNN-BiGRU-MAttention	0.9765	0.9901	0.9606	0.9751

Table 5: Experimental results under different BiGRU layers.

	Accuracy	Precision	Recall	F1
1	0.9746	0.9863	0.9626	0.9743
2	0.9751	0.9893	0.9606	0.9747
3	0.9751	0.9861	0.9637	0.9748
4	0.9759	0.9880	0.9634	0.9765

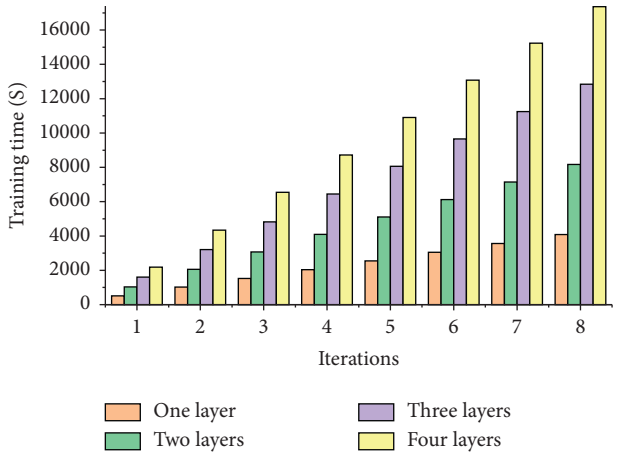

Figure 10: The changes in time consumption of multilayer BiGRU layer with the increase of the number of iterations.

Table 6: Experimental results of experiments conducted under 5 different learning rates.

	Accuracy	Precision	Recall	F1
0.1	0.5000	0.0000	0.0000	0.0000
0.01	0.9743	0.9834	0.9649	0.9741
0.001	0.9749	0.9856	0.9635	0.9744
0.0001	0.9745	0.9878	0.9609	0.9741
0.00001	0.9405	0.9361	0.9454	0.9407

accuracy rate increased slightly. In addition, the accuracy of the model was the same when the number of layers was 2 and 3. It can be seen from Figure 12 that as the number of BiGRU layer increased, the training time for each iteration cycle would also increase significantly. When the number of iterations of the model increased, the training time of the model would greatly increase. In view of this case, a BiGRU layer was selected for training in the CBMA algorithm model.

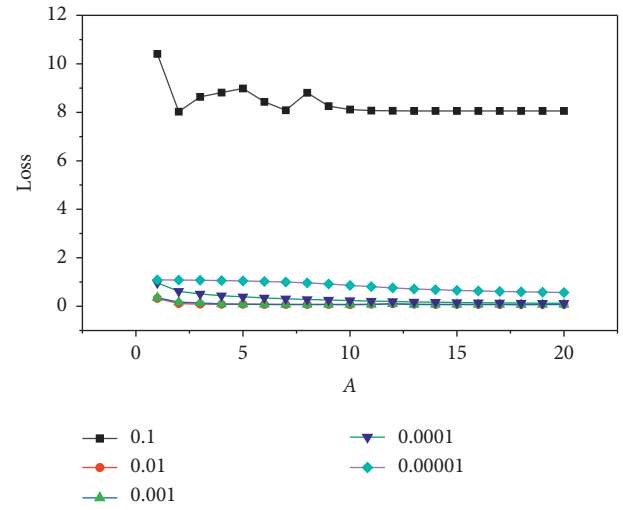

Figure 11: Experiments conducted under different learning rates and the change of the loss function value of the verification set as the number of iterations increased.

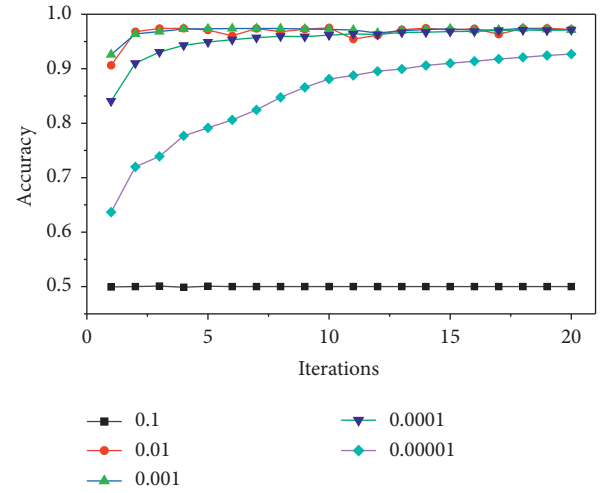

Figure 12: Experiments conducted under different learning rates and the change of the accuracy of the verification set as the number of iterations increased.

Table 7: Comparison of test results with various deep learning methods.

	Accuracy	Precision	Recall	F1
GRU-MAttention	0.9735	0.9833	0.9637	0.9734
BiGRU-MAttention	0.9746	0.9842	0.9648	0.9744
CNN-GRU-MAttention	0.9747	0.9854	0.9638	0.9745
CNN-BiGRU-MAttention	0.9765	0.9901	0.9606	0.9751

5.1.3. Contrast Test of Various Learning Rates. The contrast test results of various learning rates are shown in Table 1. A total of 5 learning rates were utilized in the experiment. It can be seen from the experimental results that as the learning rate decreased, the accuracy increased firstly and then decreased. When the learning rate was 0.001, the accuracy rate

was the highest, which was selected in the CBMA algorithm model for experiments.

5.2. Experiment Analysis of Experiment 2

5.2.1. Contrast Test of Traditional Machine Methods. In order to verify the feasibility and reliability of the algorithm proposed in this paper in Chinese microblog sentiment detection, the same embedded word vector was put into a variety of models for training and testing. The experimental results are shown in Table 4. The models utilized in the test included traditional machine learning algorithms. It can be seen from Table 4 that, among the traditional machine learning methods, the test accuracy of the decision tree was the lowest compared to other methods, which was only 72.47%. Logistic regression had the highest accuracy rate compared with other methods, which was 97.65%. Experiments were conducted through applying a variety of traditional machine learning methods. Experimental results presented that the CBMA model proposed by us had a great advantage in accuracy.

5.2.2. Comparison of Traditional Deep Learning Methods. In addition, four deep learning models were applied for testing as well. The test results are shown in Table 7. It can be seen from the test results that the accuracy of GRU-MAttention model was the lowest, which was 97.35%. The CBMA model had the highest accuracy of 97.65%, which presented that the CBMA model had better results than other deep learning models.

6. Conclusions

Aiming at the sentiment detection of Chinese microblog, a CBMA algorithm model that combines CNN and BiGRU networks and introduces multihead attention mechanism was proposed based on the respective characteristics of CNN, bidirectional long short-term memory networks, and multihead attention mechanism, which is applied to the sentiment detection field of Chinese microblog. The advantages of the CNN in extracting local features of the text and the BiGRU network in extracting the global features of the text were fully taken into consideration in this model, as well as the information in the context of the text, and the features of the text were extracted effectively. Experimental analysis in every small step has been carried out in this paper, such as testing various convolutions and various BiGRU. Moreover, various traditional machine learning methods were tested as well. Tremendous experiments presented that CBMA algorithm model has a better effect on the weibo_senti_100 k dataset of microblog comments. In addition, we hope that the study in this paper has the capacity to play a certain role in the field of microblog sentiment detection as well.

Data Availability

The datasets used in this paper to produce the experimental results are publicly available. Weibo_senti_100k can be downloaded from https://github.com/SophonPlus/ChineseNlpCorpus/blob/master/datasets/weibo_senti_100k/intro.ipynb.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the Demonstration Project of Technological Innovation and Application in Beibei District, Chongqing (Grant no. 2020-06).

References

[1] S. Kulkarni, N. Bhagat, M. Fu et al., "Twitter heron: stream processing at scale," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data-SIGMOD'15*, pp. 239–250, New York, NY, USA, May 2015.

[2] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[3] H. Wang, J. He, X. Zhang, and S. Liu, "A short text classification method based on N-gram and CNN," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 248–254, 2020.

[4] Z. Han, M. Shang, Z. Liu et al., "SeqViews2SeqLabels: learning 3D global features via aggregating sequential views by RNN with attention," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 658–672, 2019.

[5] A. Massaro, V. Maritati, D. Giannone, D. Convertini, and A. Galiano, "LSTM DSS automatism and dataset optimization for diabetes prediction," *Applied Sciences*, vol. 9, no. 17, p. 3532, 2019.

[6] Q. Chen, Q. Xie, Q. Yuan, H. Huang, and Y. Li, "Research on a real-time monitoring method for the wear state of a tool based on a convolutional bidirectional LSTM model," *Symmetry*, vol. 11, no. 10, p. 1233, 2019.

[7] C. Zhang, D. Wang, L. Wang et al., "Temporal data-driven failure prognostics using BiGRU for optical networks," *Journal of Optical Communications and Networking*, vol. 12, no. 8, pp. 277–287, 2020.

[8] Z. Li, Q. Zhang, Y. Wang, and S. Wang, "Social media rumor refuter feature analysis and crowd identification based on XGBoost and NLP," *Applied Sciences*, vol. 10, no. 14, p. 4711, 2020.

[9] K.-X. Han, W. Chien, C.-C. Chiu, and Y.-T. Cheng, "Application of support vector machine (SVM) in the sentiment analysis of twitter DataSet," *Applied Sciences*, vol. 10, no. 3, p. 1125, 2020.

[10] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751, Doha, Qatar, October 2014.

[11] J. Dai, C. Chen, and Y. Li, "A backdoor attack against LSTM-based text classification systems," *IEEE Access*, vol. 7, pp. 138872–138878, 2019.

[12] H. Ye, B. Cao, Z. Peng, T. Chen, Y. Wen, and J. Liu, "Web services classification based on wide & Bi-LSTM model," *IEEE Access*, vol. 7, pp. 43697–43706, 2019.

[13] W. Li, F. Qi, M. Tang, and Z. Yu, "Bidirectional LSTM with self-attention mechanism and multi-channel features for

sentiment classification," *Neurocomputing*, vol. 387, pp. 63–77, 2020.

[14] Q. Fu, C. Wang, X. Han et al., "A CNN-LSTM network with attention approach for learning universal sentence representation in embedded system," *Microprocessors and Microsystems*, vol. 74, 2020.

[15] Q. Sun, J. Niu, Z. Yao, and D. Qiu, "Research on semantic orientation classification of Chinese online product reviews based on multi-aspect sentiment analysis," in *Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies—BDCAT'16*, pp. 262–267, Shanghai, China, December 2016.

[16] L.-C. Yu, J. Wang, K. R. Lai, and X. Zhang, "Refining word embeddings using intensity scores for sentiment analysis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 3, pp. 671–681, 2018.

[17] G. Xu, Z. Yu, H. Yao, F. Li, Y. Meng, and X. Wu, "Chinese text sentiment analysis based on extended sentiment dictionary," *IEEE Access*, vol. 7, pp. 43749–43762, 2019.

[18] H. Chanlekha, W. Damdoung, and M. Suktarachan, "The development of semi-automatic sentiment lexicon construction tool for Thai sentiment analysis," in *Advances in Natural Language Processing, Intelligent Informatics and Smart Technology. SNLP 2016, Advances in Intelligent Systems and Computing*, T. Theeramunkong, R. Kongkachandra, and T. Supnithi, Eds., vol. 684, Berlin, Germany, Springer, 2018.

[19] U. Naseem and K. Musial, "DICE: deep intelligent contextual embedding for twitter sentiment analysis," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, pp. 953–958, Sydney, Australia, September 2019.

[20] K. Sailunaz and R. Alhajj, "Emotion and sentiment analysis from twitter text," *Journal of Computational Science*, vol. 36, 2019.

[21] B. Klein, G. Lev, G. Sadeh et al., "Associating neural word embeddings with deep image representations using Fisher Vectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4437–4446, Boston, MA, USA, June 2015.

[22] J. Lilleberg, Y. Zhu, Y. Zhang et al., "Support vector machines and Word2vec for text classification with semantic features," in *Proceedings of the IEEE International Conference on Cognitive Informatics and Cognitive Computing*, pp. 136–140, Beijing, China, July 2015.

[23] D. Zhang, H. Xu, Z. Su, and Y. Xu, "Chinese comments sentiment classification based on word2vec and SVMperf," *Expert Systems With Applications*, vol. 42, no. 4, pp. 1857–1863, 2015.

[24] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[25] F. Zhao, P. Li, Y. Li, J. Hou, and Y. Li, "Semi-supervised convolutional neural network for law advice online," *Applied Sciences*, vol. 9, no. 17, p. 3617, 2019.

[26] K. Cho, B. van Merrienboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[27] K. Cho, B. Van Merrienboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder—decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, Doha, Qatar, October 2014.

[28] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: a search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[29] C. Ma, C. Yang, F. Yang et al., "Trajectory factory: tracklet cleaving and Re-connection by deep siamese Bi-GRU for multiple object tracking," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, San Diego, CA, USA, April 2018.

[30] K. Hu, Z. Zhang, X. Niu et al., "Retinal vessel segmentation of color fundus images using multiscale convolutional neural network with an improved cross-entropy loss function," *Neurocomputing*, vol. 309, pp. 179–191, 2018.

[31] X. Zeng, Y. Zhang, X. Wang, K. Chen, D. Li, and W. Yang, "Fine-grained image retrieval via piecewise cross entropy loss," *Image and Vision Computing*, vol. 93, Article ID 103820.

[32] Y. Zhou, M. Wang, M. Zheng, J. Zhu, R. Zheng, and Q. Wu, "MPCE: a maximum probability based cross entropy loss function for neural network classification," *IEEE Access*, vol. 7, no. 99, pp. 146331–146341, 2019.

[33] E. Wu, D. Talbot, F. Moiseev et al., "Analyzing multi-head self-attention: specialized heads do the heavy lifting, the rest can Be pruned," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, Florence, Italy, July 2019.

[34] S. Gu and Y. Feng, "Improving multi-head attention with capsule networks," in *Proceedings of the Natural Language Processing and Chinese Computing, Lecture Notes in Computer Science*, pp. 314–326, Dunhuang, China, October 2019.

[35] T. Xia and X. Chen, "A discrete hidden Markov model for SMS spam detection," *Applied Sciences*, vol. 10, no. 14, p. 5011, 2020.