

Research Article

Deep Learning Structure for Cross-Domain Sentiment Classification Based on Improved Cross Entropy and Weight

Rong Fei ¹, Quanzhu Yao ¹, Yuanbo Zhu,² Qingzheng Xu,³ Aimin Li,¹ Haozheng Wu,¹ and Bo Hu⁴

¹College of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China

²China Railway First Survey and Design Institute, Abu Dhabi 710043, China

³College of Information and Communication, National University of Defense Technology, Changsha, Hunan 710106, China

⁴Beijing Huadian Youkong Technology Co., Ltd., Beijing 100091, China

Correspondence should be addressed to Rong Fei; annyfei@xaut.edu.cn and Quanzhu Yao; qzyao@xaut.edu.cn

Received 10 January 2020; Revised 22 May 2020; Accepted 3 June 2020; Published 29 June 2020

Academic Editor: Rahman Ali

Copyright © 2020 Rong Fei et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Within the sentiment classification field, the convolutional neural network (CNN) and long short-term memory (LSTM) are praised for their classification and prediction performance, but their accuracy, loss rate, and time are not ideal. To this purpose, a deep learning structure combining the improved cross entropy and weight for word is proposed for solving cross-domain sentiment classification, which focuses on achieving better text sentiment classification by optimizing and improving recurrent neural network (RNN) and CNN. Firstly, we use the idea of hinge loss function (hinge loss) and the triplet loss function (triplet loss) to improve the cross entropy loss. The improved cross entropy loss function is combined with the CNN model and LSTM network which are tested in the two classification problems. Then, the LSTM binary-optimize (LSTM-BO) model and CNN binary-optimize (CNN-BO) model are proposed, which are more effective in fitting the predicted errors and preventing overfitting. Finally, considering the characteristics of the processing text of the recurrent neural network, the influence of input words for the final classification is analysed, which can obtain the importance of each word to the classification results. The experiment results show that within the same time, the proposed weight-recurrent neural network (W-RNN) model gives higher weight to words with stronger emotional tendency to reduce the loss of emotional information, which improves the accuracy of classification.

1. Introduction

Analysis of text emotional tendency, as an important research focus in the analysis of Internet public opinion, is mainly used to analyse and process subjective information, such as attitude, emotion, viewpoint, and tendency, in text. Sentiment analysis was first proposed by Pang et al. [1] for the positive or negative classification of movie reviews and Turney [2] for the positive or negative classification of cars and movies in 2002. Subsequent studies on sentiment analysis have been widely carried out for hotels, restaurants, product reviews, Weibo tweets, and other fields. Additional developments include positive or negative polarized classification methods [3], five classifications including ratings [4], and eight classifications including specific emotions [5].

Traditional sentiment analysis algorithms are mostly based on shallow machine learning, such as the maximum entropy model [6], conditional random field [7], support vector machine [8], and so on. With the increasing popularity of artificial intelligence, data-driven models have gradually become a focus on research of sentiment analysis models.

Deep learning algorithms have been widely used in the fields of speech, image, and natural language processing with their strong feature extraction and excellent information expression capabilities and have achieved better results than traditional models. In 1988, Rumelhart proposed the backpropagation neural network (BPNN) [6], which is a multilayer feedforward neural network (FNN) that uses the error backpropagation algorithm to adjust weights. It is the

most widely used NN model. LeCun et al. [9] used various deep neural networks to train language models at large-scale corpus level and constructed a probabilistic language model based on deep neural network, which solves common natural language processing tasks such as sentiment classification and part-of-speech tagging. Chen et al. proposed a deep learning method of learning potentially complex and irregular probability distributions, which can accurately estimate the values of cumulative distribution function (CDF) and probability density function (PDF) [10].

In the sentiment analysis task, the deep learning algorithm has also been widely used. At the same time, some people use convolutional networks to solve problems with the field of natural language processing and have achieved excellent results in tasks such as semantic analysis, query retrieval, and text classification. Since text is sequence data, there is a close relationship between words and characters as well. In 2006, Hinton [11] proposed a method for extracting features to the maximum extent and efficient learning, which has become a hotspot in deep learning research. Due to the excellent performance of deep learning in many fields, many researchers have begun to use deep learning for text sentiment analysis. Due to the long-term dependence of the cyclic neural network on the processing of long text tasks and the temporal information about words with the text, the LSTM is used [12, 13] for text emotion classification. Kennedy and Inkpen [14] considered the polarity transfer relationship of words in the text and determined the affective tendency by word counting based on the seed word set. Kim compared multiple deep learning models on multiple datasets and found that the experimental results of CNN were better than those of other methods [15]. Tang [16] considered the importance of user information and product information for sentiment classification, combined word vectors, user vectors, and product vectors at the input layer, and then used CNN for modelling and softmax for classification; the results were higher than those of the benchmark system at the sentence level and phrase level.

The difference between the predicted value and the real value of the model is usually evaluated by loss function, which generally tends to be the objective function in the classification or regression algorithms [17]. The smaller the loss function, the more its model can reflect the real data [18]. The closeness between the actual and the expected output is determined by cross entropy, which is essentially a measure of the difference between two codes [19]. Cross entropy is often the final loss function of machine learning or deep learning [20]. The closer the predicted distribution is to the real distribution, the smaller the value will be. With the wide application of cross entropy, in 2020, Cui et al. [21] applied a new loss function to optimize the end-to-end network for the first time, which is composed of binary cross entropy and dice coefficient; the best performance indexes can be achieved, thus verifying the validity of the model.

Deep learning methods have been applied to cross-domain sentiment mining tasks successfully with excellent representation learning and high efficiency classification abilities. Zhao et al. [22] presented a two-stage bidirectional LSTM (Bi-LSTM) and parameter transfer framework for

short text cross-domain sentiment classification tasks. In 2019, Dey et al. [23] explored a three-step methodology, in which distinct balanced training, text preprocessing, and machine learning methods were tested, using two languages: English and Italian. In [24], cross-domain-labeled Web sources (Amazon and Tripadvisor) are used to train supervised learning models (including two deep learning algorithms) that are tested for typically unlabelled social media reviews (Facebook and Twitter), whose train model is tested on Facebook data for both English and Italian. In weight computing, Dey et al. [25] calculated the sentiment score of the n-grams by using the individual sentiment scores of the unigrams and precalculated values of intensifiers and negations attached with it. These scores are multiplied with the corresponding feature-importance value to generate the final score of SEND features of each review.

In the deep learning network model, the CNN has made great achievements in the field of image processing, whose convolution and pooling structure can extract image information very well. Therefore, RNN is widely used as a neural network for processing sequence data in the field of text analysis. Because of its memory function, it is better at processing sequence-changing data, among which, the LSTM recurrent neural network solves the problem of gradient disappearance and gradient explosion in the recurrent network, which makes the analysis and modelling of long sequence data successful. This study focuses on the optimization and improvement of RNN and CNN to achieve better text sentiment classification. According to the characteristics and shortcomings of each deep neural network, the following three text sentiment classification models are proposed.

Based on the CNN model and LSTM network, the ideas of the hinge loss and the triplet loss are used to improve the cross entropy loss used in the two classifications problems. The LSTM binary-optimize (LSTM-BO) model and CNN binary-optimize (CNN-BO) model are proposed which are more effective in fitting the predicted errors and preventing overfitting.

Considering the characteristics of the processing text of the recurrent neural network, the influence of input words of the final classification is analysed, which can obtain the importance of each word with the classification results. The proposed weight-recurrent neural network (W-RNN) model gives higher weight to words with stronger emotional tendency to reduce the loss of emotional information, which improves the accuracy of classification.

The rest of the paper is organized as follows. Section 2 is about the deep learning structure of cross-domain sentiment classification. Section 3 elaborates our numerical example. Results and discussion are presented in Section 4, and Section 5 is the summary of our research work.

2. Deep Learning Structure for Cross-Domain Sentiment Classification

2.1. The Improved LSTM-BO and CNN-BO Models

2.1.1. Hinge Loss Function and Triplet Loss Function. The hinge loss function is a loss function in the machine learning field and can be used for the “max-margin” classification,

often used to be the objective function of the SVM. Triplet loss is a loss function in the deep learning, which was originally proposed by Schroff et al. [26] to train less sensitive samples, such as face similarity measurement. The input of triplet loss is a triple a, p, n : a (anchor); p (positive) referring to a sample of the same category as a ; n (negative) referring to a sample of a different category from a . The calculation of the sample similarity is achieved by optimizing the distance between a and p to be less than the distance between a and n . The formula as follows:

$$L = \max(d(a, p) - d(a, n) + \text{margin}, 0). \quad (1)$$

So, the ultimate optimization goal is to shorten the distance between a and p and extend the distance between a and n . The results are divided into three cases:

- (i) Easy triplets: $L = 0$, that is, $d(a, p) + \text{margin} < d(a, n)$; this situation does not need to be optimized and meets the requirement that the distance of a and p is close, and the distance of a and n is far.
- (ii) Hard triplets: $d(a, n) > d(a, p)$, that is, the distance between a and p is far.
- (iii) Semihard triplets: $d(a, p) < d(a, n) < d(a, p) + \text{margin}$, that is, the distance between a and n is very close but has a boundary value margin.

2.1.2. The Improved Cross Entropy Loss Function. The text sentiment analysis task is fundamentally a classification problem. For the classification model, there will exist a problem that the optimization goal and the evaluation index are inconsistent. In the two-category task, the model uses cross entropy as the loss function, whose source is the maximum likelihood estimate. However, the final evaluation goal of the sentiment classification task is the accuracy of the model, instead of the size of the cross entropy. Usually, the cross entropy is small and the classification accuracy is high, but this relationship is not necessarily true.

In the two-category task, due to problems such as model fitting ability and data category imbalance, it is difficult for the model to achieve the positive sample output to be 1 and the negative sample output to be 0. In the actual prediction, the model considers that when the classification result is greater than 0.5, it is a positive sample, and when it is less than 0.5, it is a negative sample. This means that the model can be selectively updated. Therefore, an improved model is proposed in this paper: we set a threshold M , where M belongs to $(0, 1)$. When the model's output of a positive sample is higher than M , or the output of a negative sample is lower than $1 - M$, the model will not be updated; the model will be updated only when the output of a sample is between M and $1 - M$, which can ensure that the model focuses on those samples that are not predictive. This can prevent the model from reducing the loss function and selecting those easy-to-fit samples to overtrain, making the model more effectively fit the samples whose prediction is wrong, thus improving the classification effect.

Based on the above model ideas, this passage references the thought of hinge loss and triplet loss to improve the loss function in the two-category model. The commonly used cross entropy loss function is formulated as follows:

$$L_{\text{old}} = - \sum_y y_{\text{true}} \log y_{\text{pred}}, \quad (2)$$

where y_{true} is the actual output result and y_{pred} is the expected value.

Select a threshold M and introduce the unit step function $\theta(x)$:

$$\theta(x) = \begin{cases} 1, & x > 0, \\ \frac{1}{2}, & x = 0, \\ 0, & x < 0. \end{cases} \quad (3)$$

Then, the new loss function is considered:

$$L_{\text{new}} = - \sum_y \lambda(y_{\text{true}}, y_{\text{pred}}) y_{\text{true}} \log y_{\text{pred}}, \quad (4)$$

where

$$\lambda(y_{\text{true}}, y_{\text{pred}}) = 1 - \theta y_{\text{true}} - m\theta(y_{\text{pred}} - m) - \theta(1 - m - y_{\text{true}})\theta(1 - m - y_{\text{pred}}), \quad (5)$$

where L_{new} adds corrections $\lambda(y_{\text{true}}, y_{\text{pred}})$ to the cross entropy, which means that while entering a positive sample, $y_{\text{true}} = 1$. Apparently, $\lambda(1, y_{\text{pred}}) = 1 - \theta(y_{\text{pred}} - m)$ at this time; if $y_{\text{pred}} > m$, $\lambda(1, y_{\text{pred}}) = 0$ will be established, and the cross entropy will automatically be 0 (reaching the minimum). On the contrary, if $y_{\text{pred}} < m$, then $\lambda(1, y_{\text{pred}}) = 1$; at this time, the cross entropy is maintained. That is to say, if the positive sample is higher than m , then it will not be updated. If it is less than m , it will continue to update; similarly, the negative sample can be analysed. As a result, the conclusion is that if the output is already lower than $1 - m$, then it will not be updated, and if it is higher than $1 - m$, it will continue to update.

2.1.3. The LSTM-BO and CNN-BO Models. The LSTM-BO and CNN-BO models are based on the LSTM and CNN in keras which are combined with improved cross entropy loss function described in Section 2.1.2.

2.2. Weight-Recurrent Neural Network (W-RNN) for Cross-Domain Sentiment Classification

2.2.1. Basic Structure of Recurrent Neural Network. Recurrent neural network (RNN) refers to the network structure that processes input data sequences in the same structure over time [27]. The proposed RNN effectively solves the problem of processing sequence information. In traditional neural networks, nodes inside the hidden layer are connectionless and each output is independent of one another. However, in RNN, the hidden layer nodes are

connected to each other in the time dimension, and the input of each node includes not only the input currently input by the input layer but also the output information of the previous state of the hidden layer, that is, the network can remember the previous information, used to calculate the current output, as shown in Figure 1.

The characteristic that the RNN can recall the previous information is based on the hidden layer, which is constantly repeated as a memory unit and saves the information from the previous state. As a logical structure, the internal structure of the memory unit is shown in Figure 2. At the time t , weight W of the input x_t and old information h_{t-1} from the time $t-1$ are processed by the self-joining matrix U through the hidden layer, whose sum adds the offset b together to obtain the output h_t of the current state hidden layer via an activation function (such as, Tanh).

The time t 's and previous information continues to propagate through h_t until the end which is the true output of the hidden layer. The above calculation process is formulated as

$$h_t = \text{Tanh}(Wx_t + Uh_{t-1} + b). \quad (6)$$

2.2.2. W-RNN Model for Cross-Domain Sentiment Classification. It can be acknowledged that in the standard recurrent neural network, the calculation result of the recurrent unit at each moment is not utilized, but only passed to the recurrent unit at the next moment for another round of calculation until the last moment, whose output is used as the output of the recurrent layer. The calculation of the state of each moment depends on the previous moment, and the temporal order information is continuously retained by such dependence. In this section, combined with the characteristics of the recurrent neural network, the importance of each word will be obtained by analysing the influence of the input words on the final classification. Based on this thought, the W-RNN model is constructed, which provides higher weight for words with stronger emotional tendency and reduces the loss of text emotional information, thus improving the accuracy of text sentiment classification.

Recurrent neural networks are one of the most important models of many sequence tasks. The common methods for text classification tasks are shown in Figure 3.

How to measure the importance of the impact of input w_1, \dots, w_n on the final classification results? Assuming that it is an emotional classification task, firstly, the words which have a more important impact on the final classification need to be found.

Because the state vector of the last step of the RNN (the vector represented by the orange shade in Figure 3) is passed to the subsequent classifier for classification, the state vector h_n of the last step is a target vector. RNN is a recursive process, and h_0, h_1, \dots, h_{n-1} is gradually approaching h_n .

So, the distance $\|h_n - h_0\|, \|h_n - h_1\|, \dots, \|h_n - h_{n-1}\|$, 0 of the intermediate vector to the target vector can be considered in turn. From h_i to h_{i+1} , because of the excessive consideration of the word w_{i+1} , it can be deduced that the distance between h_i and the target vector should be $\|h_n - h_i\|$,

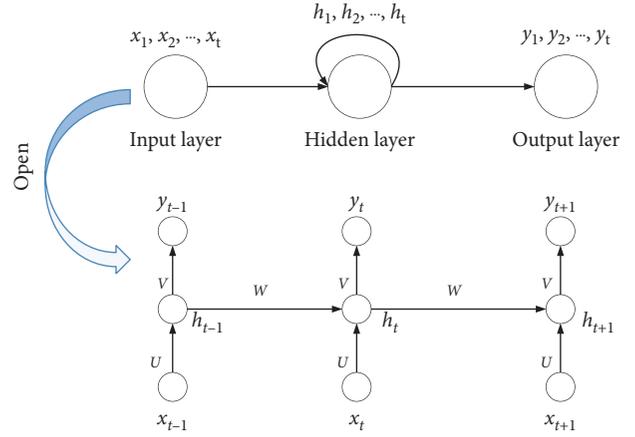


FIGURE 1: The structure diagram of recurrent neural network model.

but now the distance becomes $\|h_n - h_{i+1}\|$, so we can use the difference $\|h_n - h_i\| - \|h_n - h_{i+1}\|$ to measure the impact of the word w_{i+1} on the final classification. If the difference is positive, it means that the introduction of w_{i+1} narrows the distance from the target, so it promotes the correct classification; otherwise, if the difference is negative, it means that it has a reaction to the classification; the larger the value, the greater the degree of effect. So, this indicator can be used to sort in descending order and get the importance of each word. This article excludes the effects of dimension by dividing by the norm of the target vector:

$$\frac{\|h_n - h_i\|}{\|h_n\|} - \frac{\|h_n - h_{i+1}\|}{\|h_n\|}. \quad (7)$$

2.3. Algorithm Flow. According to the above model structure, we can get the algorithm flow of emotion classification by using W-RNN model (Algorithm 1).

Batch_size is the scale, used to group C_{TR} for training, and the left samples with less than Batch_size are grouped together; epochs represent the number of training iterations.

In this algorithm, firstly, distances are ranked according to h' . Secondly, every secondary feature c is calculated from the word vector. Then, the output z is obtained by c , and the z value is used to update the weight. The W-RNN gives high weight to words with strong emotional tendency and reduces the function of words with weak emotion in sentences, which decreases the loss of text emotional information.

3. Experiments and Setting

3.1. Experimental Environment. The specific experimental environment configuration of this model is shown in Table 1.

3.2. Dataset. Since the classification model may possess different adaptabilities of different languages and texts of different lengths, in order to verify the performance of the model, the experiment was tested with different types of datasets under several famous corpora. This

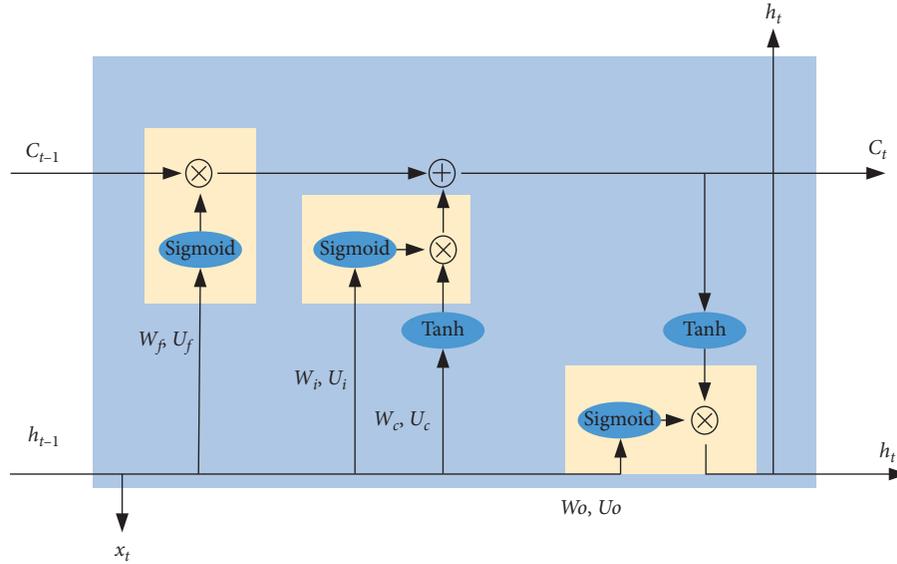


FIGURE 2: Structure of a standard RNN unit.

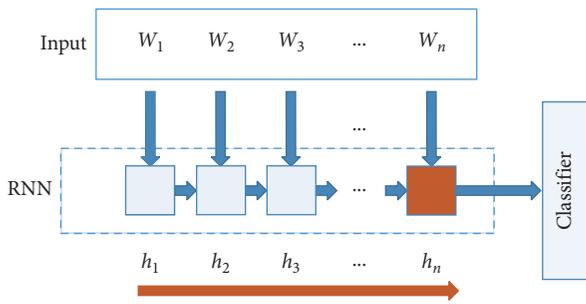


FIGURE 3: Text classification model based on recurrent neural network.

experiment uses IMDB [28] English film review data and Netease news classified text data, covering different languages, different lengths, and different types of text classification tasks.

The following two datasets are specifically described.

The IMDB English film review dataset is data from Amazon’s Internet Movie Database (IMDB), which includes a lot of information about the film, such as actors, film length, content introductions, ratings, and reviews. For the text classification task, the film review data used in this experiment distinguish positive review from negative review, that is, they contain two categories, which belong to the two-category sentiment analysis classification problem. The dataset contains a total of 50,000 comment text data, and its label distribution is balanced, that is, there are 25,000 positive reviews and 25,000 negative reviews. In addition, the dataset provides 50,000 unlabelled data for unsupervised learning.

The Chinese dataset is a collection of commodity reviews provided by Data Hall, which contains six aspects of comment data (books, hotels, computers, milk, mobile phones, and water heaters), a total of 21,107 text data, in which there are 10,428 negative data and 10,679 positive data. Figure 4 describes the above Chinese dataset.

In Table 2, the number of positive emotion samples and negative emotion samples and sum of samples for two datasets are given.

3.3. Data Preprocessing. Text datasets used in the experiment are both in Chinese and in English. The Chinese dataset is a collection of commodity review corpus provided by Data Hall, which contains six aspects of comment data, a total of 21,107 text data. The English dataset is the IMDB film review dataset, with a total of 50,000 comment text data. In the sentiment classification task, the training data and test data of the two datasets are randomly generated in a ratio of 80 : 20.

The preprocessing work on the data mainly includes the cleaning of invalid special characters and punctuation, the cleaning of common pause words in the language, and the segmentation of Chinese language using the jieba word segmentation tool based on python. This experiment introduces Word2Vec as pretraining, aiming to construct the word vector. The appropriate word vector can improve the performance and calculation speed of the model. Every word vector dimension pretrained by Word2Vec is set to 50, with the window size set to 10 and trained by the skip-gram model. The training parameter settings of the Word2vec model are described in Table 3.

3.4. Superparameter Setting. In the neural network model training process, the superparameter is a parameter that sets the value before the model training. Generally, the superparameter needs to be optimized, and a set of optimal superparameter is selected for the model to improve the performance and effect of the learning. The superparameter configuration of the model is shown in Table 4.

Table 5 shows the superparameter settings of each model, among which the selection of superparameter is obtained by the optimization experiment of the bold font parameters of Table 6.

Input:
 CWE-word vector
 CTR-training corpus
 CTE-test corpus

Output: Prediction results of test samples.

```

(1) pro_processing (CWE)
(2) Dict = word2vec (CWE)//create the word vector dictionary Dict
(3) batches [] ← Divide (CTR)//divide CTR into several batches
(4) for i ← 0 to epochs do
(5)   for j ← 0 to length (batches) do
(6)     for k ← 0 to length (batches[j]) do
(7)       v ← FindWord (batches [j][k])//find the words vector v in batches[j][k] from Dict
(8)       h ← v//the feature vector h is extracted from v
(9)       h' ← Measure (h)//measure the impact of h
(10)      v ← Sort (v, h')//sort words vector in descending order according to h'
(11)      c ← ExtractFeature (v)//extract secondary feature c from the word vector
(12)      z ← Softmax (c)//Get the prediction results of samples by Softmax classifier
(13)      end for
(14)      Update (z, w, (b))//update parameters w and b of the model by backpropagation
(15)    end for
(16)  end for
(17) for i ← 0 to length (CTE) do
(18)   v ← FindWord (CTE [i])
(19)   h ← v
(20)   h' ← Measure (h)
(21)   v ← Sort (v, h')
(22)   c ← ExtractFeature (v)
(23)   output ← Softmax (c)
(24) end for

```

ALGORITHM 1: Text sentiment analysis algorithm based on W-RNN.

TABLE 1: Experimental environment and configuration.

Experimental environment	Specific configuration
Operating system	Windows 10 × 64
CPU	Intel (R) Core (TM) i7-4790, 3.6 GHz
Memory	8G
Hardware	1T
Programming language	Python 3.5
Deep learning framework	keras2.0
Word vector training tool	Word2vec
Word segmentation tool	Jieba

4. Result Analysis

4.1. Analysis of Experimental Results Based on LSTM-BO and CNN-BO Models. The loss function of the traditional LSTM model and the CNN model improved, and the LSTM-BO and CNN-BO models constructed to perform the text sentiment analysis task, the parameter optimization experiment, and the benchmark model comparison experiment are used to verify the effectiveness of the new network model in the emotion classification task.

4.1.1. Parameter Optimization Experiment. In order to investigate the influence of each parameter on the model effect, in this paper, four sets of parameter optimization

experiments are designed and compared on the IMDB public dataset.

(i) Discussion of experimental results of threshold M selection

In this experiment, based on the LSTM-BO and CNN-BO models, the threshold M is selected from 0.5 to 1.0, increasing by 0.1 in turn. The experimental results are shown in Table 6.

It can be seen from Table 6 that the LSTM-BO and CNN-BO models have the highest accuracy at the threshold $M = 0.6$, which are 82.14% and 88.74%, respectively; the loss rates are the same when $M = 0.6$ and $M = 0.7$, which are, respectively, 0.2267 and 0.1637. It can be seen from Figures 5 and 6 that with the increase of the M value, the accuracy of the two models generally shows a trend of increasing first and then decreasing, and the overall loss rate tends to decrease first and then rise. When the threshold M is 0.6, the accuracy of the LSTM-BO model reaches the peak value, and the loss rate reaches the minimum value; when the M value changes from 0.5 to 0.6, the accuracy rate is greatly improved, which is 7.96% higher; when the M value is changed from 0.9 to 1.0, the loss rate changes greatly, increasing by 0.4435. The accuracy and loss rate of the CNN-BO model are similar to those of the

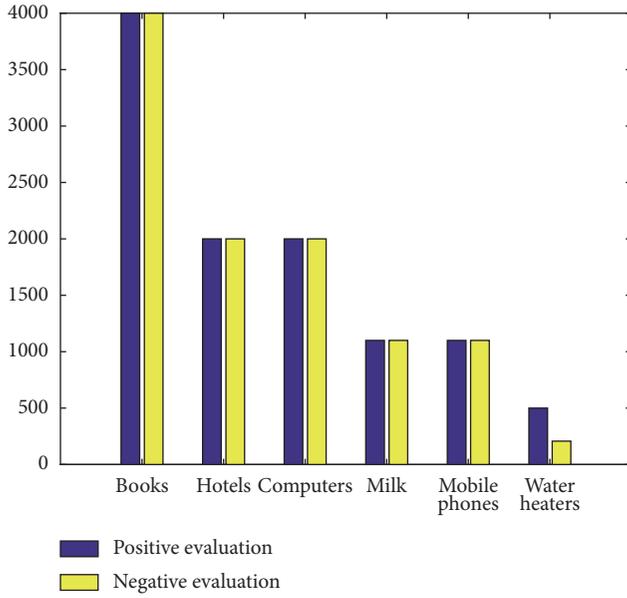


FIGURE 4: Chinese dataset category.

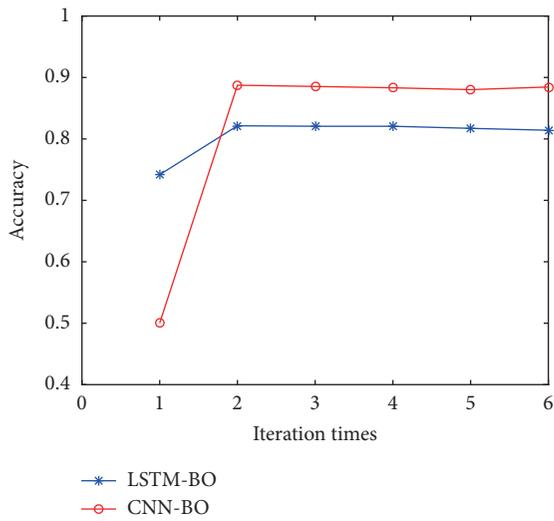


FIGURE 5: Variation rate of each model classification under different thresholds.

LSTM-BO model, but the overall effect is better. When $M = 0.6$, the accuracy reaches the peak and the loss rate is the smallest. When the M value is changed from 0.5 to 0.6, the accuracy rate is increased by 38.71%, and the loss rate is reduced by 0.1827. Based on the above analysis, the threshold M of this paper is 0.6.

- (ii) Influence of different loss functions on the model
 In Table 7, `binary_crossentropy` is a standard cross entropy loss function; `binary-optimize` is the loss function proposed in this paper; `hinge` is a hinge loss function, commonly used in SVM classifiers; `mean_absolute_percentage` (MAPE) is the average absolute percentage error loss

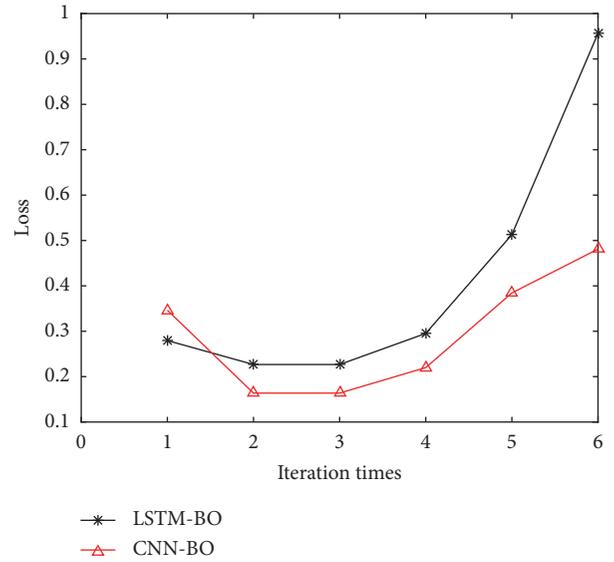


FIGURE 6: Change rate of each model under different thresholds.

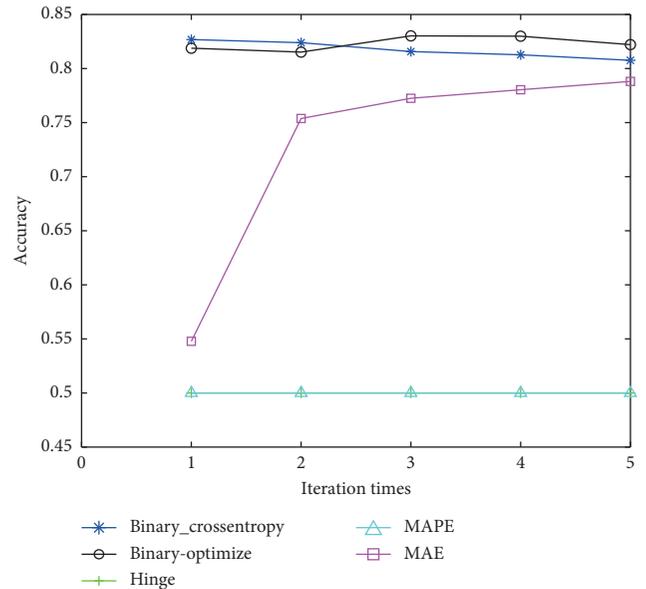


FIGURE 7: The variation diagram of accuracy rate with the number of iterations under different loss functions.

function; `mean_absolute_error` (MAE) is the absolute value variance loss function. Figure 7 shows the variation of the accuracy of the LSTM model with the number of iterations under different loss functions of the English dataset. It can be seen from the figure that the LSTM-BO model with improved loss function has the highest accuracy in the sentiment classification task, which is 82.21% and has been leading after the second iteration; the LSTM model using the standard cross entropy loss function with the hinge, MAPE, and MAE loss functions has high

TABLE 2: Dataset table.

Data	Positive emotion sample	Negative emotion sample	Sum of samples
English	25000	25000	50000
Chinese	10679	10428	21107

TABLE 3: Word2vec model training parameter settings.

Parameter	Value
Vector dimension	50
Context window	10
Sampling value	$1e-3$
Iteration times	4
Minimum number of occurrences	5
Algorithm	Skip-gram
Worker number	4

accuracy. The accuracy of the LSTM model using the hinge and MAPE loss functions is kept at 50% and the number of iterations is more than 5 times. Based on the above experimental results, the effectiveness of the improved loss function is proved.

(iii) The selection experiment of word vector dimension

In this experiment, the vector dimension of the selected words is 50, 100, 150, 200, 250, and 300, respectively. It can be seen from Table 8 that the LSTM-BO model has the highest accuracy of 82.48%, when the word vector dimension is 100 dimensions, and the loss rate is at least 0.2234. When the word vector dimension is 50 dimensions, the CNN-BO model has a maximum accuracy of 88.74% and a loss rate of 0.1637.

The dropout technique weakens the coadapting property of adjacent elements in the same layer by randomly discarding certain elements in the previous layer during the training process. By using dropout, the overfitting phenomenon is significantly reduced, and thus it is widely used in the training process of deep learning. In order to study the impact of dropout on the training process, this experiment sets the value of dropout to a series of different values during each training process with other parameters fixed. The results are shown in Table 9.

It can be seen from Table 9 that when dropout was set to 0.2, the LSTM-BO model achieved the highest accuracy of 82.14%, the minimum loss rate of 0.2247, and the shortest time consumption. When dropout was set to 0.2, the LSTM-BO model has a maximum accuracy of 88.74% and the shortest time consumption. When dropout is 0.3, the lowest loss rate is 0.1590.

4.1.2. Comparison of Experimental Results. In order to verify the validity of the LSTM-BO and CNN-BO models, based on the Chinese and English datasets, the results were compared with the emotional classification results of the benchmark

models LSTM and CNN. The experimental results are shown in Table 10.

(1) *Accuracy Analysis.* Figures 8 and 9 show the accuracy of each model in the training set and test set sentiment classification tasks under different datasets. The horizontal axis represents the number of iterations and the vertical axis represents the accuracy. The blue curve represents the change in the accuracy of the training set, and the orange curve represents the change in the accuracy of the test set. Figure 10 shows a bar graph of the accuracy of the final test set for each model on different datasets. Figure 11 shows a plot of the change in the accuracy of each model in the 10 iterations of the English dataset. From the above experimental results, the following conclusions can be drawn:

- (i) Table 10 shows that there exist some differences in the experimental results of different datasets. The accuracy of LSTM and LSTM-BO models in Chinese datasets is 5.94% and 5.37% higher than that of English datasets, respectively. The accuracy of CNN and BO models on the Chinese dataset is 0.5% and 0.17% higher than the English datasets, respectively.
- (ii) It can be seen from Table 10 and Figure 10 that the CNN-BO model has the highest accuracy of emotional classification in the Chinese and English datasets compared with the other three models, which are 88.91% and 88.74%, respectively. The accuracy of the LSTM-BO model is 1.16% and 0.59% higher than that of the benchmark LSTM model in the English dataset and the Chinese dataset, respectively. The accuracy of the CNN-BO model is higher than that of the benchmark CNN model in the English dataset and the Chinese dataset which is increased by 0.5% and 0.07%.
- (iii) As can be seen from Figures 8 and 9, the accuracy of the four models on the training set increases slowly with the number of iterations, and there is a significant change between the first and second iterations, and eventually it stabilizes. However, the increase in the number of iterations cannot increase the accuracy in the test set significantly. The fluctuation exists in the test, especially the LSTM and CNN models fluctuate greatly in the test. It can be seen from Figure 11 that in the English dataset, the LSTM-BO model has high accuracy after the second iteration, and the CNN-BO model has high accuracy after the fourth iteration. The analysis results in this paper are that LSTM-BO and CNN-BO models can more effectively fit the samples that predict errors, help

TABLE 4: Model parameter configuration.

Name	Description
Batch_size	Number of batch training samples
Hidden_dam	Hidden layer nodes
Embedding_dam	Word vector dimension
Filters	Filter size
Kernel_size	Number of convolution kernels
Max_features	The maximum characteristic number of word bag model
Epoch	Times of model iterations
Min_count	Minimum word frequency

TABLE 5: Model superparameter settings.

	LSTM	LSTM-BO	CNN	CNN-BO	W-RNN
Hidden_dam	256	256	256	256	256
Batch_size	32	32	32	32	32
Embedding_dam	50	50	50	50	50
Filters	—	—	250	250	—
Kernel_size	—	—	3	3	—
Max_features	5000	5000	5000	5000	5000
Epoch	10	10	10	10	10
Min_count	10	10	10	10	10
Dropout	0.2	0.2	0.2	0.2	0.5
Activation function	Sigmoid	Sigmoid	Relu	Relu	Sigmoid

TABLE 6: Accuracy and loss function of each model under different thresholds (M).

	LSTM-BO		CNN-BO	
	Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)
0.5	74.18	0.2795	50.03	0.3464
0.6	82.14	0.2267	88.74	0.1637
0.7	82.06	0.2267	88.54	0.1637
0.8	82.06	0.2952	88.32	0.2199
0.9	81.72	0.5129	88.02	0.3844
1.0	81.41	0.9564	88.45	0.4811

TABLE 7: The variation of accuracy rate with the number of iterations under different loss functions.

Times loss function	Iteration				
	1	2	3	4	5
Binary_crossentropy	82.68	82.39	81.57	81.27	80.76
Binary-optimize	81.88	81.52	83.02	82.99	82.21
Hinge	50.00	50.00	50.00	50.00	50.00
MAPE	50.00	50.00	50.00	50.00	50.00
MAE	54.78	75.38	77.25	78.04	78.81

prevent overfitting, and improve the accuracy of sentiment classification tasks.

(2) *Loss Rate Analysis.* Figures 12 and 13 show the variation curves of the loss rates of the models in the training set and test set sentiment classification tasks in different datasets, respectively. The horizontal axis represents the number of

iterations and the vertical axis represents the accuracy. The black curve represents the change in the training set loss rate, and the yellow curve represents the test set loss rate change. Figure 13 shows a plot of the change in loss rate for each model over 10 iterations in the English dataset. From the above experimental results, the following conclusions can be drawn:

- (i) It can be seen from Figures 12 and 13 that the loss rate of the four models in the Chinese and English training sets decreases with the increase of the number of iterations, but the change of the model on the test set is more obvious. The loss rate of the LSTM model and the CNN model in the test set fluctuated greatly and showed an upward trend. The loss rate of the LSTM-BO and CNN-BO models on the test set slowly decreased and finally tended to be flat.
- (ii) It can be seen from Figure 14 that after the second iteration of the English dataset, the loss rate of each model has a large trend, and the loss rate of LSTM and CNN models increases with the number of iterations and finally ends after the 10th iteration; they were 0.8170 and 0.5588, respectively. The LSTM-BO and CNN-BO models slowly decrease as the number of iterations increases, eventually reaching a steady state. The loss ratios of the LSTM-BO and CNN-BO models were reduced by 0.5903 and 0.3951, respectively, over the benchmark LSTM and CNN models. The analysis results in this paper are that the improved model has better generalization ability, which can converge after multiple iterations and achieve lower loss rate.

(3) *Time Performance Analysis.* Figure 15 shows the time consumption of each model in the Chinese and English datasets. As can be seen from the figure, the LSTM-BO is reduced by 162 seconds and 51 seconds, respectively, in the Chinese and English datasets compared to the LSTM model; CNN-BO is reduced by 128 seconds and 202 seconds, respectively, compared to the CNN model. The analysis results in this paper are that for the LSTM-BO and CNN-BO models, the samples will not be updated when the predicted value of the positive sample is higher than M or the predicted value of the negative sample is lower than M , focusing on those whose predictions are not accurate, thus reducing the

TABLE 8: Experimental results under different word vector dimensions.

Word vector dimension		50	100	150	200	250
LSTM-BO	Accuracy (%)	82.14	82.48	81.69	81.76	82.04
	Loss (%)	0.2267	0.2234	0.2372	0.2577	0.2302
CNN-BO	Accuracy (%)	88.74	87.66	88.56	88.34	87.46
	Loss (%)	0.1637	0.1845	0.1652	0.1732	0.1832

TABLE 9: Results of different dropout values.

	Dropout	Accuracy (%)	Loss (%)	Time (s)
LSTM-BO	0.1	81.62	0.2348	1265
	0.2	82.14	0.2267	1136
	0.3	82.06	0.2285	1148
	0.4	82.05	0.2274	1143
	0.5	81.98	0.2354	1254
CNN-BO	0.1	87.72	0.1833	1249
	0.2	88.74	0.1637	1172
	0.3	88.72	0.1590	1184
	0.4	88.39	0.1684	1227
	0.5	87.43	0.1842	1265

TABLE 10: Comparison of classification results on different network models.

Dataset	Network model	Accuracy (%)	Loss (%)	Time (s)
English dataset	LSTM	80.98	0.8170	1187
	LSTM-BO	82.14	0.2267	1136
	CNN	88.34	0.5588	1374
	CNN-BO	88.74	0.1637	1172
Chinese dataset	LSTM	86.92	0.6820	1425
	LSTM-BO	87.51	0.1705	1263
	CNN	88.84	0.5874	1523
	CNN-BO	88.91	0.2231	1395

consumption of time. The calculation amount is decreased with the preprocessing of Chinese data by removing the useless words and punctuations, but the same operation has not done on the English data.

4.2. Analysis of Results Based on W-RNN Model. To qualitatively and quantitatively evaluate the W-RNN model proposed in this section, this experiment compares the effects of different models in the emotional analysis task under the Chinese and English datasets. The specific method is as follows: for the quantitative evaluation experiment, some data are selected from the Chinese and English datasets as the training set, the classification model is trained, and finally the emotional classification task is completed in the test set to measure the accuracy; for the qualitative evaluation experiment, the emotional weight calculated by the analysis model is to verify the validity of the model.

4.2.1. Results and Discussion for Qualitative Experiments. In the qualitative analysis experiment, firstly, the classification model is trained in the training set. Then, we randomly select 3 comment corpora under the English and Chinese datasets and

process them by the trained W-RNN model and generate the results in the table. In the result, the first component in the brackets indicates the position of the word after the word segmentation, the second component indicates the word segmentation, and the third component indicates the weight of the word. For example, the first component in the first row of the table: (7, “poor”, 0.36663848) is the result of word “poor”, in which 7 means “poor” is the seventh position after sentence participle, “poor” means the word itself, 0.36663848 indicates the weighting factor of “poor”.

The following results can be obtained from the above experimental results:

- (i) It can be seen from Table 11 that the W-RNN sentiment classification model proposed in this paper ranks the words with strong emotional tendencies in the front and gives higher weights. For example, in the second sentence, the word “good” is given a weight of 0.2046071, and the word “very” in the fourth sentence is given a weight of 0.17955697.
- (ii) The evaluation scheme of such importance will automatically consider the influence of the position of the words. If an emotional word is repeated in the sentence, the words appearing later are generally weighted lower.

4.2.2. Results and Discussion for Quantitative Experiments.

In order to verify the effectiveness of the W-RNN model, based on the Chinese and English datasets, the results are compared with the emotional classification results of the benchmark model RNN. The experimental results are shown in Table 12.

(i) Accuracy analysis

Figure 16 shows the performance of each model in the training set and test set sentiment classification tasks in different datasets. The horizontal axis represents the number of iterations and the vertical axis represents the accuracy. The blue curve represents the change in the accuracy of the training set, and the orange curve represents the change in the accuracy of the test set. Figure 17 shows a bar graph of the accuracy of the final test set for each model on different datasets. The following results can be obtained from the above experimental results:

- (i) Table 12 shows that the experimental results of different datasets have certain differences. The accuracy of RNN and W-RNN models in Chinese datasets is 6.53% and 8.12% higher than that of English datasets, respectively.

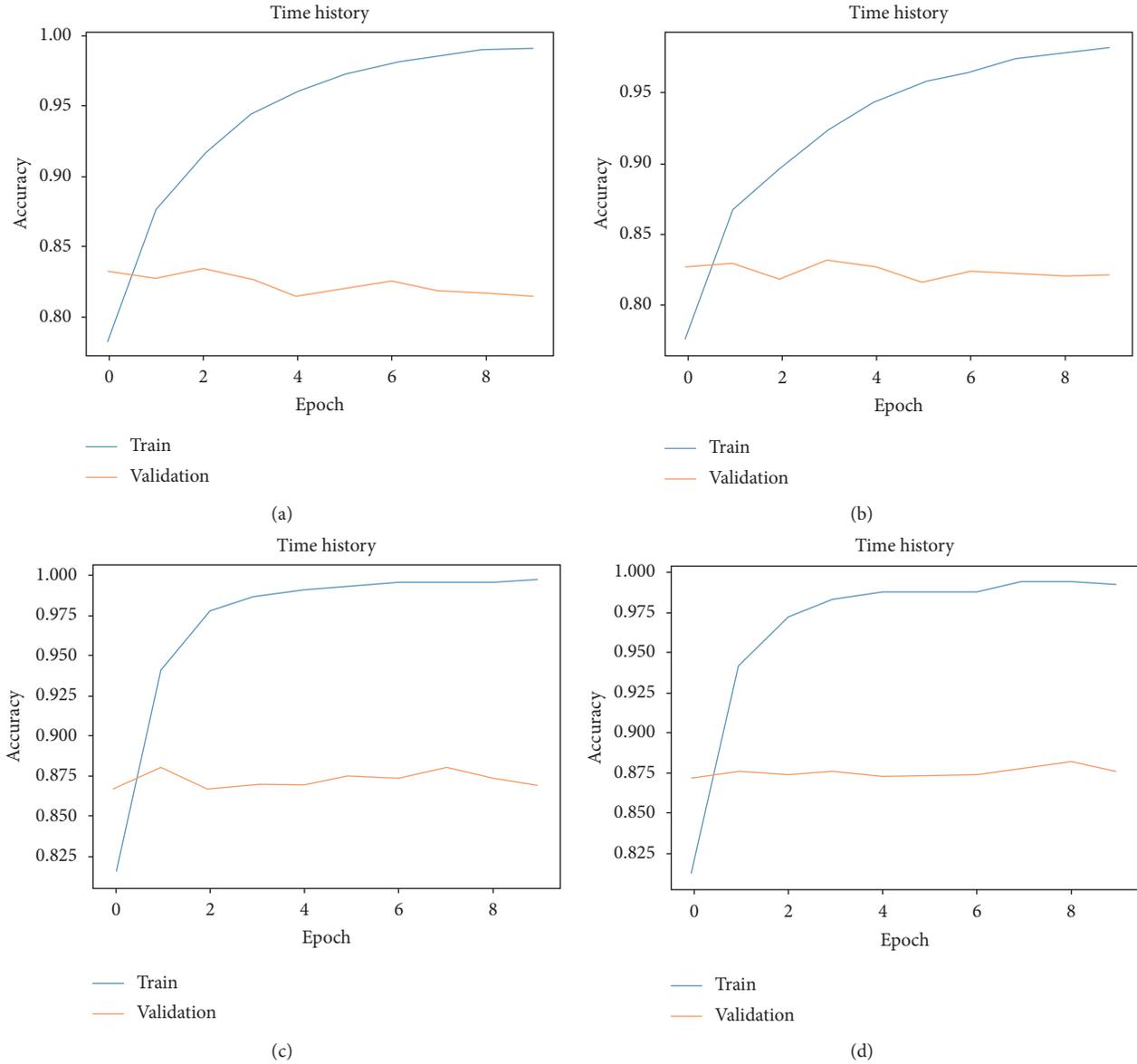


FIGURE 8: Accuracy change of LSTM model and LSTM-BO model in Chinese and English datasets. (a) LSTM (English dataset). (b) LSTM-BO (English dataset). (c) LSTM (Chinese dataset). (d) LSTM-BO (Chinese dataset).

(ii) Figure 16 shows that the accuracy of the two models in the Chinese and English training sets increases slowly with the number of iterations and eventually stabilizes. The accuracy rate is flat on the test set. After 10 iterations, the accuracy of the W-RNN model over the benchmark RNN model in the English dataset and the Chinese dataset increased by 1.56% and 3.19%, respectively. The analysis result of this paper is that W-RNN model can analyse the influence of input words on the final classification, assign higher weight to words with stronger emotional tendency, and reduce the loss of emotional information, thus improving the accuracy of text sentiment classification.

(ii) Loss analysis

Figure 18 shows the variation of the loss of the RNN model and the W-RNN model in different iterations of the Chinese and English datasets. The black curve represents the training set and the yellow curve represents the test set. It can be seen from the figure that the loss of the two models in the Chinese and English training sets shows a decreasing trend with the increase of the number of iterations and finally reaches a lower value and becomes stable. However, the difference between the two models in the test set is that the loss of the W-RNN model after 10 iterations in the English dataset is 0.1500, which is 0.3670 lower than the RNN model; the loss of the W-RNN model after 10

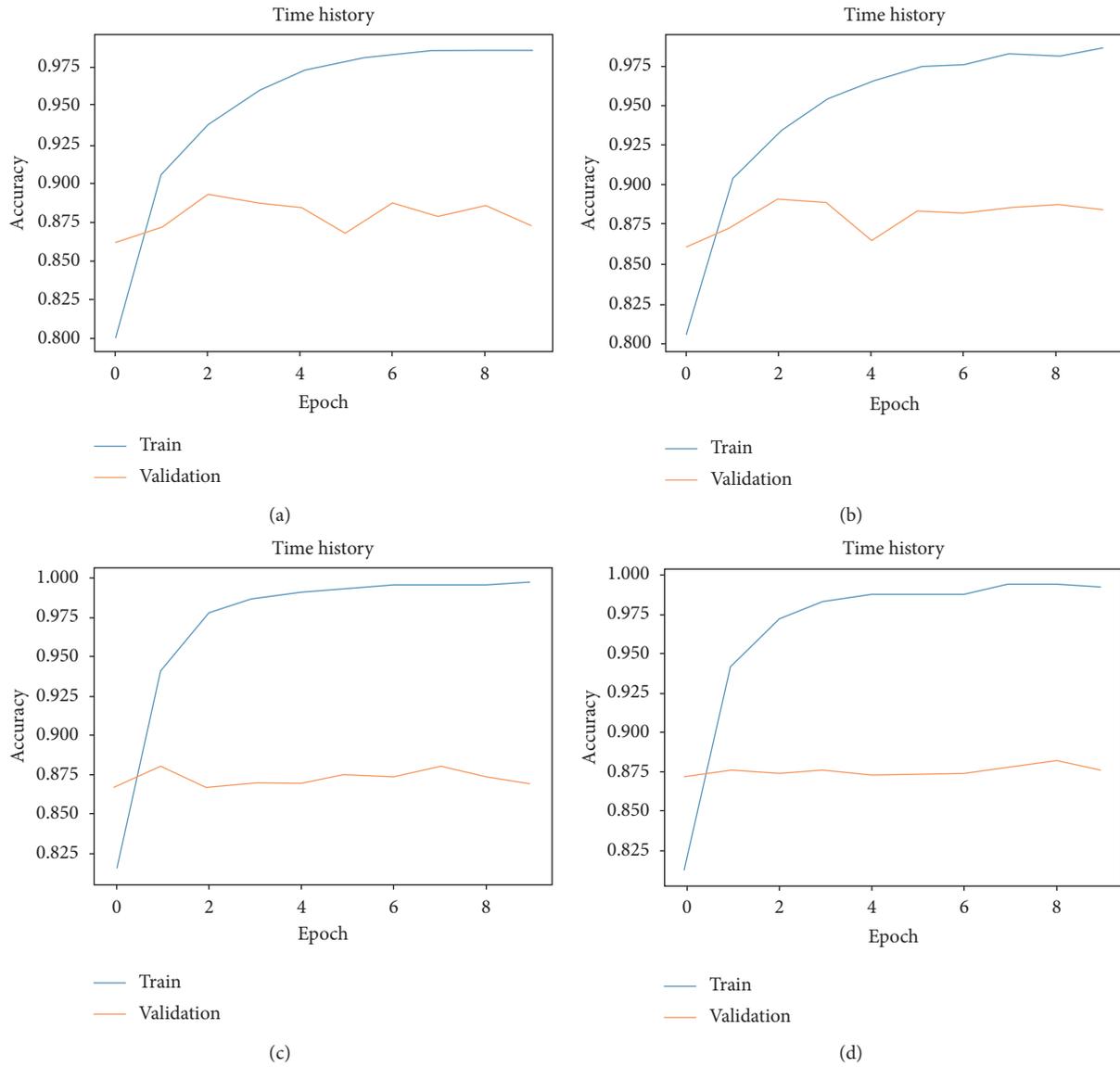


FIGURE 9: Emotional classification accuracy of CNN model and CNN-BO model in Chinese and English datasets. (a) CNN (English dataset). (b) CNN-BO (English dataset). (c) CNN (Chinese dataset). (d) CNN-BO (Chinese dataset).

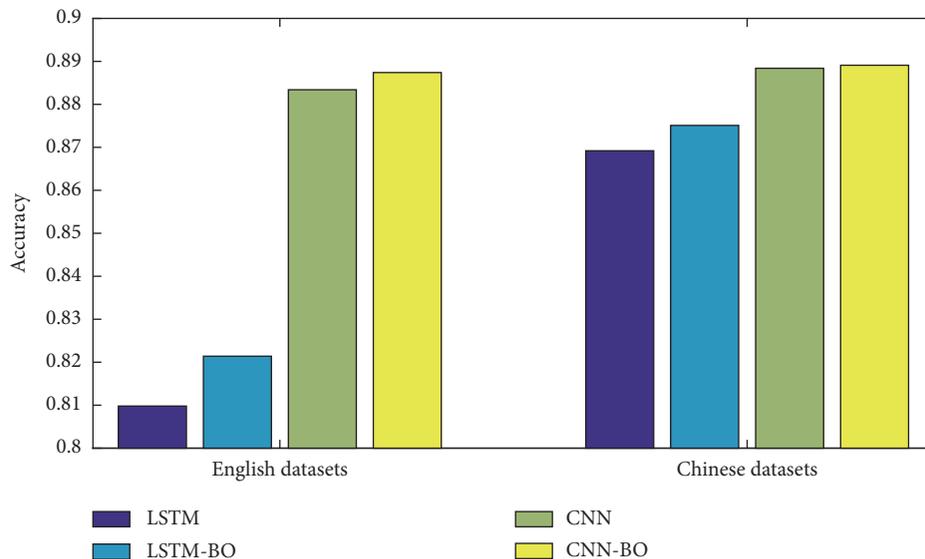


FIGURE 10: Analysis of the accuracy of the model in Chinese and English datasets.

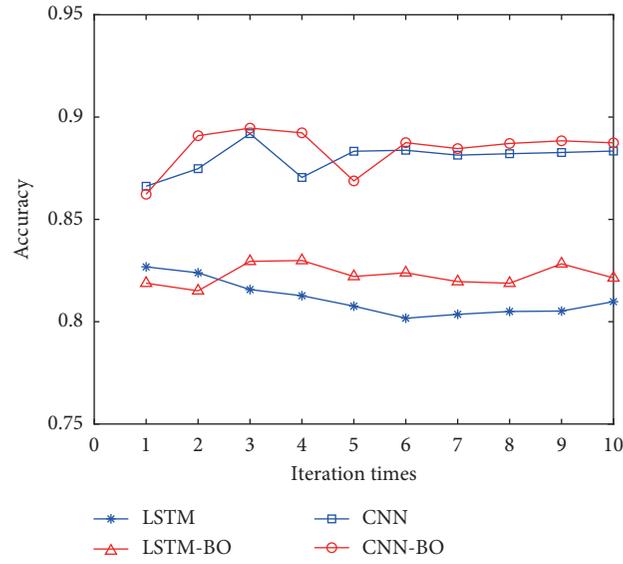


FIGURE 11: The accuracy of each model with the number of iterations under the English dataset.

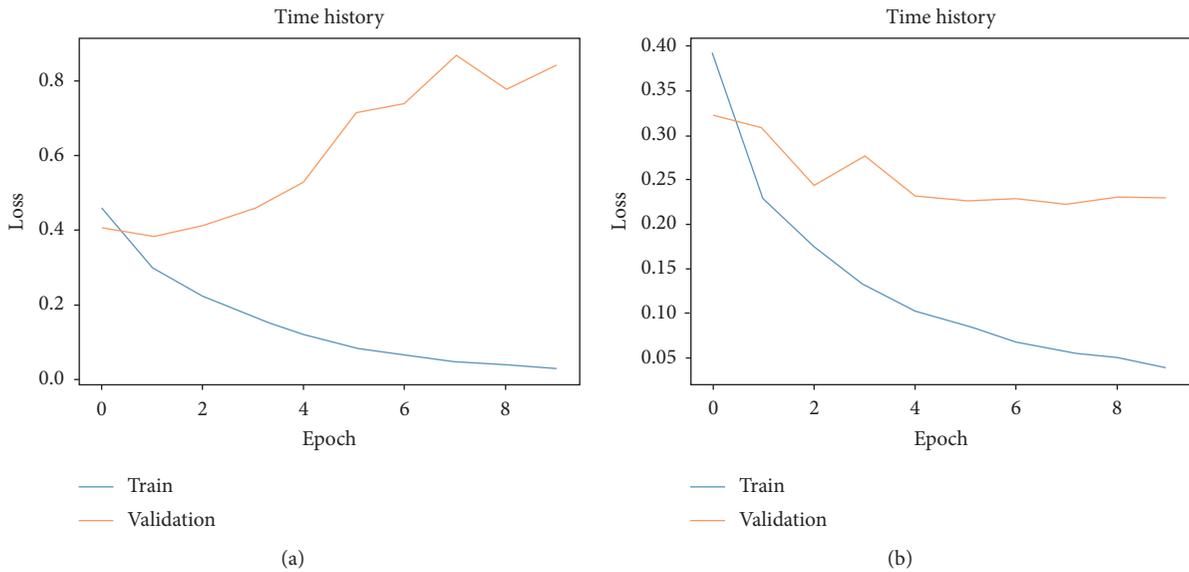


FIGURE 12: Continued.

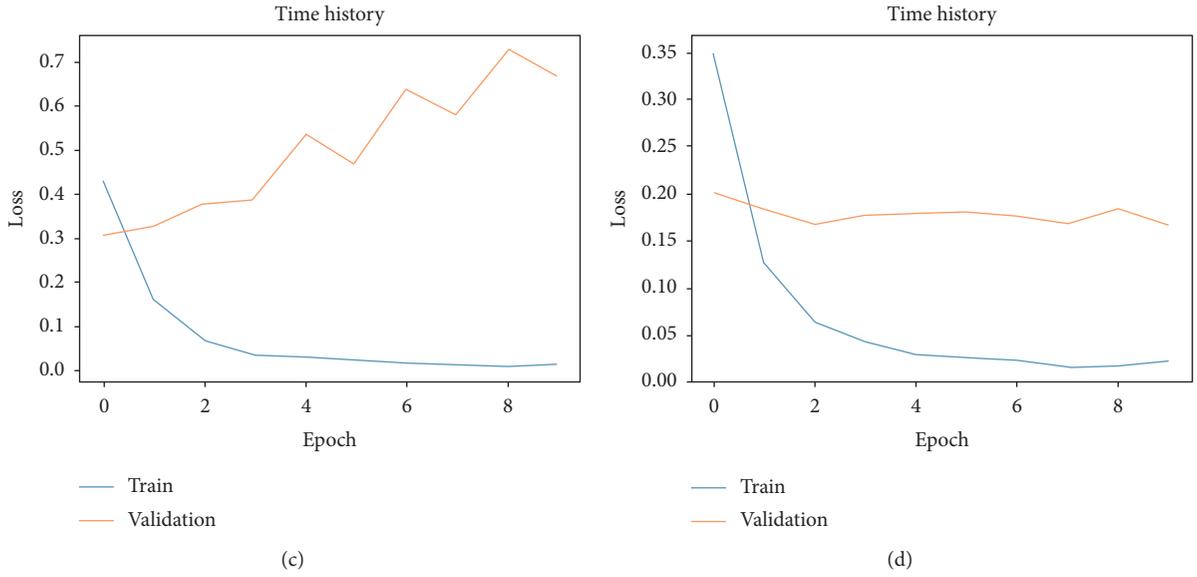


FIGURE 12: Loss rate change of LSTM model and LSTM-BO model in Chinese and English datasets. (a) Loss rate change of LSTM in English dataset. (b) Loss rate change of LSTM-BO in English dataset. (c) Loss rate change of LSTM in Chinese dataset. (d) Loss rate change of LSTM-BO in Chinese dataset.

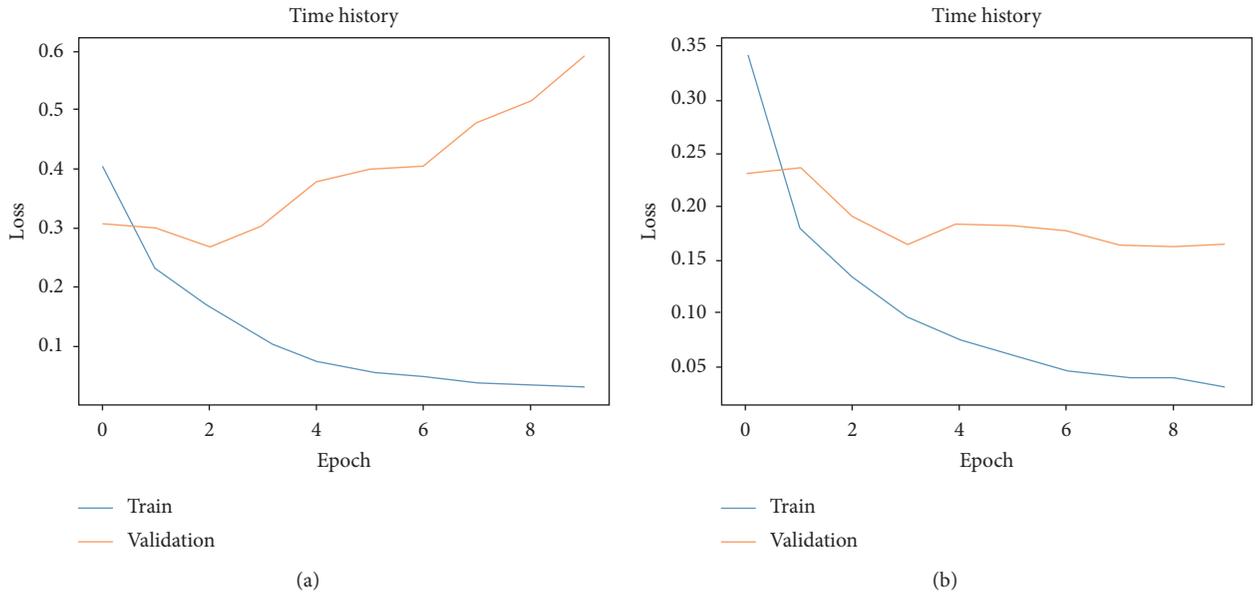


FIGURE 13: Continued.

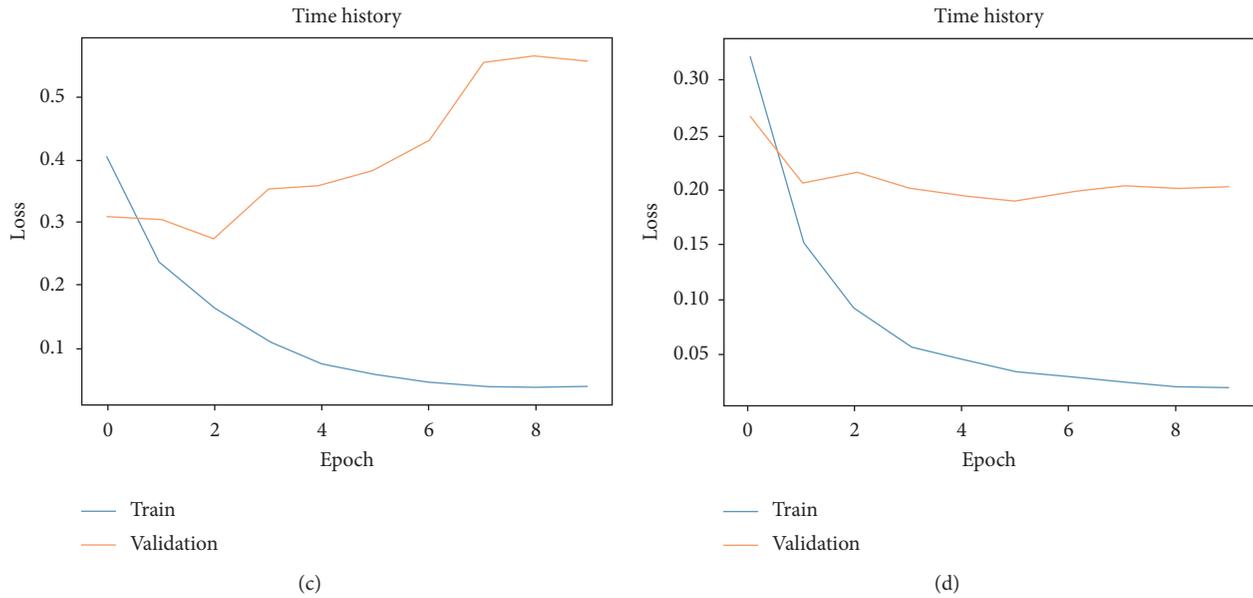


FIGURE 13: Changes in loss rate of CNN model and CNN-BO model in Chinese and English datasets. (a) Change in loss rate of CNN in English dataset. (b) Change in loss rate of CNN-BO in English dataset. (c) Change in loss rate of CNN in Chinese dataset. (d) Change in loss rate of CNN-BO in Chinese dataset.

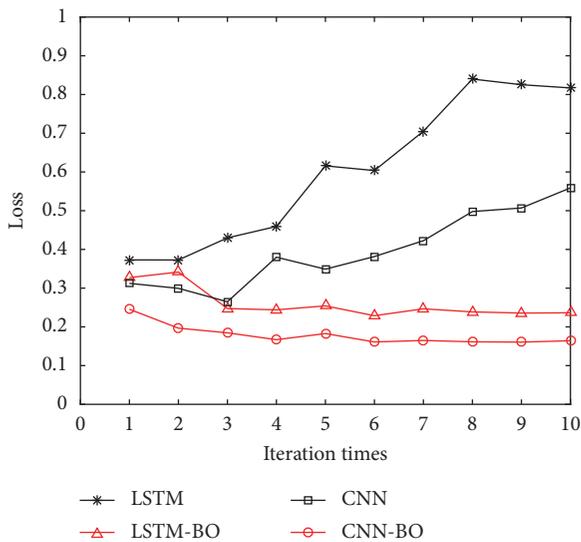


FIGURE 14: Changes in the loss rate of each model with the number of iterations under the English dataset.

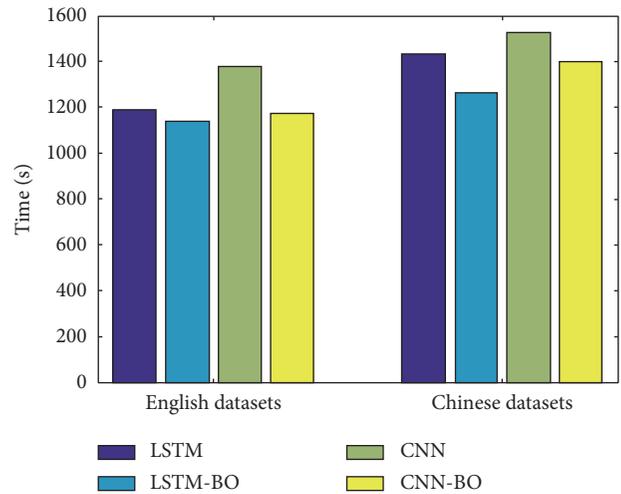


FIGURE 15: Time performance graph of Chinese and English datasets.

iterations in the Chinese dataset is 0.4616, which is 0.2332 lower than the RNN model. The analysis in this paper can prevent the overfitting of the model to a certain extent, and it can effectively extract the text features and reduce the loss.

(iii) Time performance analysis

Figure 19 shows the time consumption of the two models in the Chinese and English datasets. It can be seen from Figure 19 that the running time of W-RNN model is over 159 seconds than the RNN model in the English dataset and over 184 seconds

than the RNN model in the Chinese dataset. The reason why the W-RNN is slower than RNN is that the W-RNN model has time consumption on calculating word weights and sorting according to the weight of words.

4.2.3. Discussion on W-RNN Model for Optimizing Loss Function. Based on the improved scheme of Section 2.1, loss function, the W-RNN model loss function is optimized, and the weight-recurrent neural network-binary-optimize (W-RNN-BO) model is constructed, and experiments are carried out on the Chinese and English datasets, respectively. The

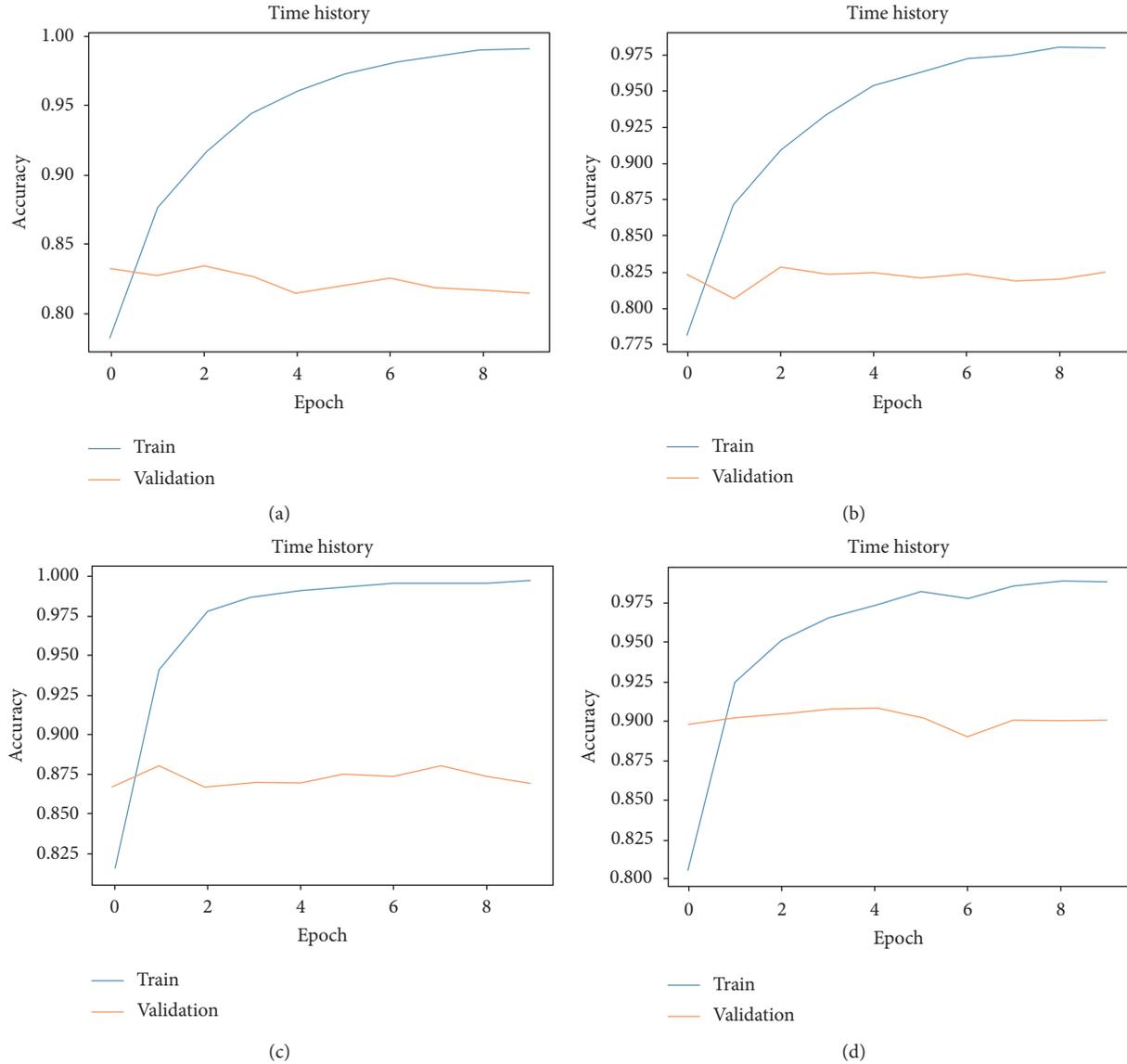


FIGURE 16: Accuracy of each model in Chinese and English datasets. (a) Changes in the accuracy of RNN in the English dataset. (b) Changes in the accuracy of W-RNN in the English dataset. (c) Changes in the accuracy of RNN in the Chinese dataset. (d) Changes in the accuracy of W-RNN in the Chinese dataset.

number of iterations is 10 times. The experimental results are shown in Table 13.

Table 13 shows the accuracy, loss rate, and time performance of the W-RNN-BO and W-RNN models in the Chinese and English datasets; Figure 20 shows a line graph showing the change in the accuracy of each model as the number of iterations increases in the Chinese and English datasets. The following results can be obtained from the above experimental results:

- (i) In terms of accuracy, it can be seen from Figure 20 that the accuracy of the W-RNN-BO model is higher than that of the W-RNN model after the second iteration in the Chinese and English datasets. And after the 10th iteration, the W-RNN-BO model is 0.59% and 0.78% higher than the W-RNN

model in the Chinese and English datasets, respectively.

- (ii) In terms of loss rate, it can be seen from Table 13 that the loss convergence effect of the W-RNN-BO model is better. After the 10th iteration in the Chinese and English datasets, the loss rate is reduced to 0.1405 and 0.1430, respectively, which is lower than W-RNN of 0.3211 and 0.3070.
- (iii) In terms of time performance, the W-RNN-BO model was reduced by 79 seconds in the English dataset compared to the W-RNN model and 105 seconds in the Chinese dataset compared to the W-RNN model.

This experiment also fully proves the effectiveness of the improved loss function described in Section 2.1, so the model possesses better generalization ability and improves

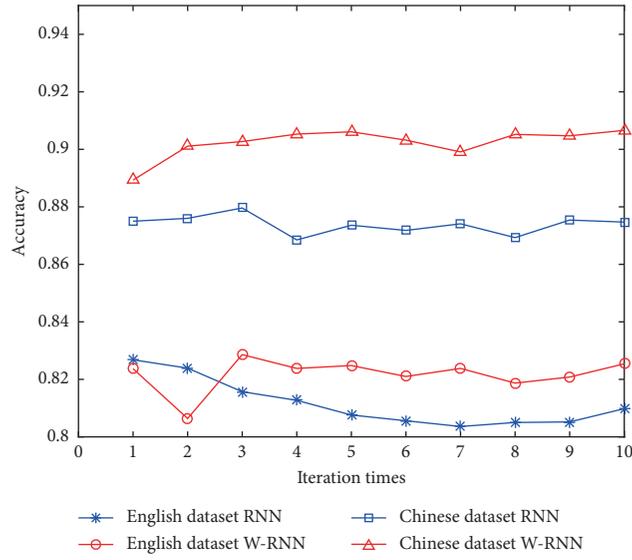


FIGURE 17: Accuracy of each model in Chinese and English datasets.

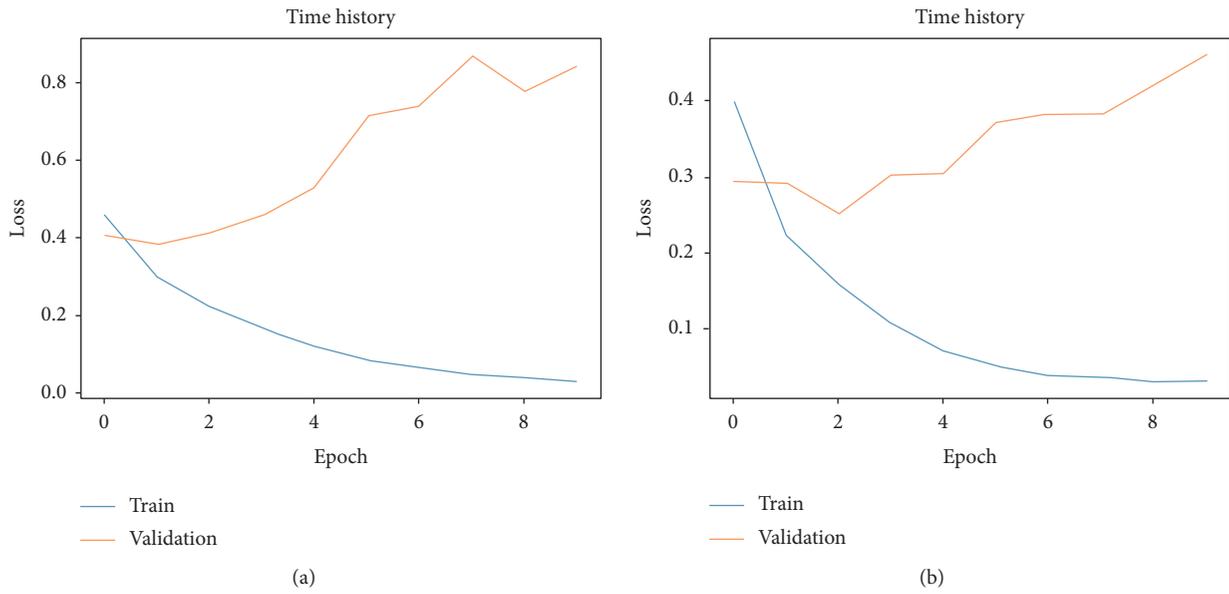


FIGURE 18: Continued.

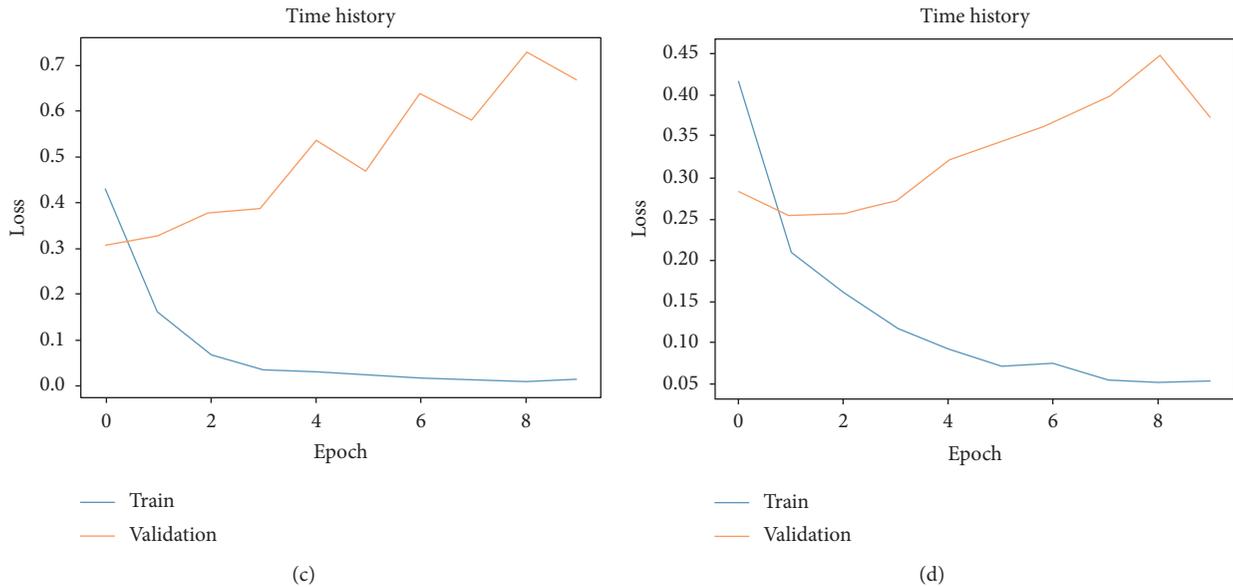


FIGURE 18: Loss of each model in Chinese and English datasets. (a) Change in loss rate of RNN in English dataset. (b) Change in loss rate of W-RNN in English dataset. (c) Change in loss rate of RNN in Chinese dataset. (d) Change in loss rate of W-RNN in Chinese dataset.

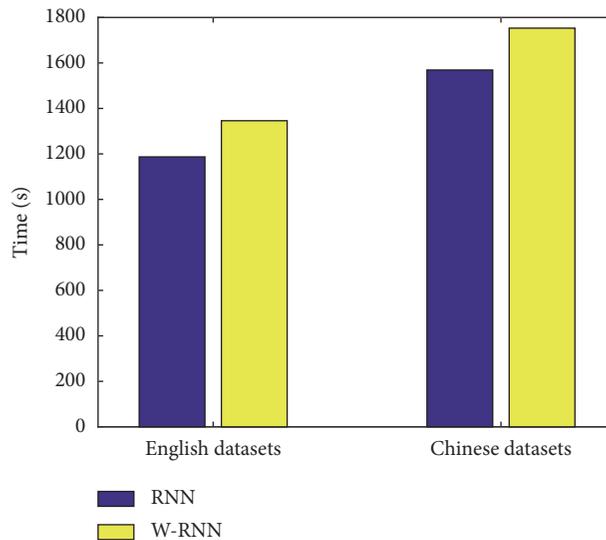


FIGURE 19: Time performance of each model in Chinese and English datasets.

TABLE 11: Analysis of output results of W-RNN model.

(1) The layout of this book is very poor, and four or five pages in front are blank.
 [(7, "poor", 0.36663848), (17, "blank", 0.16964875), (13, "pages", 0.16570723), (6, "very", 0.11767213), (4, "book", 0.07908833), (5, "is", 0.0750145), (4, "this", 0.06969571), (0, "the", 0.04458785), (16, "front", 0.023623824), (16, "are", 0.006633401), (10, "four", -0.0), (12, "five", -0.015168488), (2, "of", -0.022220552), (1, "layout", -0.08092123)]

(2) This picture book is very good, my son likes it and has been reluctant to return it.
 [(1, "picture", 0.30500817), (5, "good", 0.2046071), (4, "very", 0.17955697), (2, "book", 0.17825627), (4, '9', 0.16365236), (14, "reluctant", 0.15772212), (7, "my", 0.11790274), (16, "return", 0.10876195), (6, '.', 0.0819352), (3, "is", 0.064508274), (0, "this", 0.013084531), (12, "has", 0.0077801645), (8, "son", 0.0015705228), (15, "to", -0.023336783), (11, "and", -0.16156882), (13, "been", -0.39944094)]

TABLE 12: Comparison of classification results on different network models.

Dataset	Model	Accuracy (%)	Loss (%)	Time (s)
English dataset	RNN	80.98	0.8170	1187
	W-RNN	82.54	0.4500	1346
Chinese dataset	RNN	87.47	0.6948	1569
	W-RNN	90.66	0.4616	1753

TABLE 13: Comparison of classification results on different network models.

Dataset	Model	Accuracy (%)	Loss (%)	Time (s)
English dataset	W-RNN	82.54	0.4500	1346
	W-RNN-BO	83.32	0.1430	1267
Chinese dataset	W-RNN	90.66	0.4616	1753
	W-RNN-BO	91.25	0.1405	1648

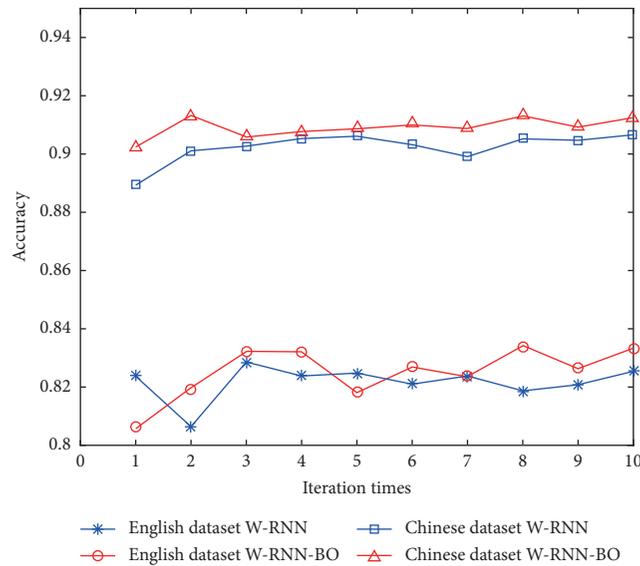


FIGURE 20: Variation rate of each model under different iterations.

the accuracy, loss rate, time, and other performances in the emotional classification task.

5. Conclusion

In order to solve the shortcomings of traditional deep neural networks in sentiment analysis tasks, three emotion classification models are proposed in this paper, based on deep neural networks. Firstly, based on the LSTM and CNN models, the traditional cross entropy loss function is improved. The LSTM-BO and CNN-BO models are designed so that the improved model can more effectively fit the prediction error samples and prevent the overfitting phenomenon. In addition, combined with the characteristics of the circulating neural network, by analysing the influence of the circulating neural network, by analysing the influence of the input words on the final classification, the importance of each

word to the classification results is obtained, and the W-RNN model is constructed. The model gives higher weight to words with stronger emotional tendency and reduces the loss of emotional information. In order to verify the effectiveness of the three sentiment classification models, qualitative and quantitative sentiment analysis experiments of two kinds of datasets in Chinese and English were designed. The experimental results show that the three models proposed in this paper improve the accuracy of text sentiment classification to a certain extent and also perform better in loss rate and time performance.

In the next work, we will consider the characteristics of CNN with its extraction of text features and RNN with its capacity on series tasks, which can be combined with self-attention to construct a better model for text feature extraction and classification.

Data Availability

The data used to support the findings of this study not available in [28] can be obtained from the corresponding author upon reasonable request.

Conflicts of Interest

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

Acknowledgments

The authors are grateful to Chi-Hua Chen and Fangying Song for providing necessary advice. This research was partially supported by the National Key Research and Development Program of China (2018YFB1201500), National Natural Science Foundation of China (grant no. 61773313), Natural Science Basic Research Program of Shaanxi (program no. 2020JM-709), and Scientific Research Foundation of National University of Defense Technology (grant no. ZK18-03-43).

References

- [1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10*, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 79–86, July 2002.
- [2] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 417–424, July 2002.
- [3] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 142–150, July 2011.
- [4] C. Du and L. Huang, "Sentiment analysis method based on piecewise convolutional neural network and generative adversarial network," *International Journal of Computers, Communications & Control*, vol. 14, 2019.
- [5] Y. Zhang, Y. Tong, and Y. Jiang, "Study of sentiment classification for Chinese microblog based on recurrent neural network," *Chinese Journal of Electronics*, vol. 25, no. 4, pp. 601–607, 2016.
- [6] A. McCallum, D. Freitag, and F. C. Pereira, "Maximum entropy markov models for information extraction and segmentation," *ICML*, vol. 17, pp. 591–598, 2000.
- [7] Y. F. Pan, X. Hou, and C. L. Liu, "Text localization in natural scene images based on conditional random field," in *Proceedings of the 10th International Conference on Document Analysis and Recognition*, IEEE, Barcelona, Spain, pp. 6–10, July 2009.
- [8] T. Joachims, *Learning to Classify Text Using Support Vector Machines*, Springer, Berlin, Germany, 2002.
- [9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [10] C.-H. Chen, F. Song, F.-J. Hwang, and L. Wu, "A probability density function generator based on neural networks," *Physica A: Statistical Mechanics and its Applications*, vol. 541, 2019.
- [11] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [12] Y. Zhao, B. Qin, T. Liu, and D. Tang, "Social sentiment sensor: a visualization system for topic detection and topic sentiment analysis on microblog," *Multimedia Tools and Applications*, vol. 75, no. 15, pp. 8843–8860, 2016.
- [13] Y. Kim, "Convolutional neural networks for sentence classification," 2014, <http://arxiv.org/abs/1408.5882>, 2014.
- [14] A. Kennedy and D. Inkpen, "Sentiment classification of movie reviews using contextual valence shifters," *Computational Intelligence*, vol. 22, pp. 110–112, 2011.
- [15] Y. Kim, "Convolutional neural networks for sentence classification," 2014, <http://arxiv.org/abs/1408.5882>.
- [16] T. Zagibalov and J. Carroll, "Automatic seed word selection for unsupervised sentiment classification of Chinese text," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1073–1080, July 2008.
- [17] D. Trichtler, *Loss Function*, *Encyclopedia of Biostatistics*, John Wiley & Sons, 2005.
- [18] K. P. Kording and D. M. Wolpert, "The loss function of sensorimotor learning," *Proceedings of the National Academy of Sciences*, vol. 101, no. 26, pp. 9839–9842, 2004.
- [19] X. Li, Y. Q. Lu, and D. Tao, "Robust subspace clustering by Cauchy loss function," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 2067–2078, 2019.
- [20] A. Jati, N. Kumar, R. Chen et al., "Hierarchy-aware loss function on a tree structured label space for audio event detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, May 2019.
- [21] S. Cui, M. Chen, and C. Liu, "DsUnet: a new network structure for detection and segmentation of ultrasound breast lesions," *Journal of Medical Imaging and Health Informatics*, vol. 10, no. 3, pp. 661–666, 2020.
- [22] C. Zhao, S. Wang, and D. Li, *Deep Transfer Learning for Social Media Cross-Domain Sentiment Classification*, *Chinese National Conference on Social Media Processing*, pp. 232–243, Springer, Berlin, Germany, 2017.
- [23] A. Dey, M. Jenamani, and J. J. Thakkar, "Cross-D-vectorizers: a set of feature-spaces for cross-domain sentiment analysis from consumer review," *Multimedia Tools and Applications*, vol. 78, no. 16, pp. 23141–23159, 2019.
- [24] P. Zola, P. Cortez, C. Ragno, and E. Brentari, "Social media cross-source and cross-domain sentiment classification," *International Journal of Information Technology and Decision Making*, vol. 18, no. 1, pp. 1469–1499, 2019.
- [25] A. Dey, M. Jenamani, and J. J. Thakkar, "Sentiment weight of N-grams in dataset (SEND): a feature-set for cross-domain sentiment classification, 2017 ninth international conference on advances in pattern recognition (ICAPR)," pp. 1–6, IEEE, 2017.
- [26] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: a unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, San Juan, PR, USA, June 2015.
- [27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [28] C. Potts, "On the negativity of negation," *Semantics and Linguistic Theory*, vol. 20, pp. 636–659, 2010.