

Research Article

A Text Emotion Analysis Method Using the Dual-Channel Convolution Neural Network in Social Networks

Di Wu,^{1,2} Jianpei Zhang ,¹ and Qingchao Zhao ¹

¹College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, Heilongjiang, China

²College of Computer and Control Engineering, Qiqihar University, Qiqihar 161006, Heilongjiang, China

Correspondence should be addressed to Jianpei Zhang; zhangjianpei@hrbeu.edu.cn

Received 22 July 2020; Revised 5 August 2020; Accepted 10 August 2020; Published 5 October 2020

Guest Editor: Yi-Zhang Jiang

Copyright © 2020 Di Wu 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.

In order to solve the problem that the existing deep learning method has insufficient ability in feature extraction in the text emotion classification task, this paper proposes a text emotion analysis using the dual-channel convolution neural network in the social network. First, a double-channel convolutional neural network is constructed. Combined with emotion words, parts of speech, degree adverbs, negative words, punctuation, and other word features that affect the text's emotional tendency, an extended text feature is formed. Then, using the CNN's multichannel mechanism, the extended text features based on the word vector features and the semantic features based on the word vectors are, respectively, input into the CNN model. After each convolution operation of the convolution channel, the BN technology is used to normalize the internal data of the network and the padding technology is used to improve the ability of the model to extract edge features of the data and the speed of the model. Finally, a dynamic k -max continuous pooling strategy is adopted to realize the dimensionality reduction of features and enhance the model's ability to extract features. The experimental results show that the accuracy and $F1$ values obtained by the proposed method can be as high as 94.16% and 92.61%, respectively, which are better than several comparison algorithms.

1. Introduction

Text emotion classification in social networks is an important subtask of emotion analysis, and its goal is to judge the emotional orientation of texts [1–3]. It belongs to the category of opinion mining and is used to analyze users' views, emotions, and attitudes towards entities (such as products, services, individuals, and events) based on the user's behavior in social networks (such as comments and word-of-mouth) and on the basis of logic, linguistics, and psychology. The most common methods are feature-based classification methods and emotion knowledge-based classification methods. The feature-based classification method makes good use of the effective features of the text and has a good classification effect in datasets with small data volume and high annotation quality [4, 5]. However, the model generalization ability of this kind of method is weak [6], which requires a lot of annotation data and artificially designed features. The classification method based on emotion knowledge classifies text by using the emotion

dictionary [7], the domain dictionary [8], and a series of judgment rules and determines the emotional polarity of all words by calculating the point mutual information of each word and the reference word [9, 10], so as to obtain a better classification effect. However, this kind of method is not good for text classification in a large number of emerging languages, which requires a lot of work to summarize and mine rules from data.

Aiming at the problem of text emotion analysis, a text emotion analysis method for e-commerce that reviews text using the depth double-channel convolutional neural network (DDCCNN) is proposed and the sparse text features are improved. The main innovations of the proposed method are summarized as follows:

- (1) In view of the sparse features in the text dataset, the existing deep learning methods have insufficient feature extraction capabilities in text emotion classification tasks. A double-channel convolutional neural network is constructed, and the multichannel

mechanism of the CNN is used to represent the text based on the word vector features. Input the CNN model separately from the text representation based on extended features, fully considering the extraction of the edge features of the dataset.

- (2) In this model, the BN layer is added after each convolution operation of the convolution channel. The BN technology is used to normalize the internal data of the network, which improves the accuracy of model training.

2. Related Work

Convolutional neural network is a multilayer artificial neural network that was originally used to solve image recognition problems. With the continuous improvement of the model structure, the CNN has now become one of the most effective tools for solving text classification problems. For the problem of text emotion analysis, scholars have proposed many methods. For example, Feuerriegel et al. [11] proposed an emotion analysis method based on transfer learning. By pretraining for different tasks and then adjusting the output layer to the emotion recognition task, the accuracy of the model is improved. Li and Wu [12] proposed an online prediction method for text mining and emotion analysis, which automatically improves the accuracy of the model by automatically analyzing the emotion polarity of the text and obtaining the value of each text. Massie et al. [13] proposed a vocabulary-based feature extraction method for text emotion classification. By expanding domain-specific emotion dictionaries, the vocabulary not only adapts to vocabulary changes in a certain field but also provides accurate quantitative evaluation, improving the accuracy of the model rate. The document model based on deep learning proposed in [14] is used for text emotion recognition. By combining features related to emotion recognition and the CNN, a feature extractor is constructed, and the personality tendency of the article author is judged by extracting features. He and Xia [15] proposed a joint binary neural network (JBNN) multilabel text emotion analysis method by feeding the representation of the text to a set of logic functions and performing multiple binary classifications in the neural network while simultaneously improving the accuracy of the model. However, the generalization ability of these methods is weak, which requires a lot of annotation data and artificially designed features. Rahmani et al. [16] proposed an emotion analysis method based on improved pretraining word embedding, which improves the accuracy of the model by improving the accuracy of pretraining word embedding in emotion analysis. Philipe et al. [17] proposed a stable emotion analysis method based on off-the-shelf methods, which improves the robustness of the model by reducing the large variability of cross-domain unsupervised methods. Wang et al. [18] proposed a text emotion analysis method of the joint factor graph model. By considering the relationship between different emotions and using confidence propagation algorithm for emotion recognition, the accuracy of the model is improved. Neviarouskaya et al. [19] proposed a rule-based emotion recognition method using the natural language processing technology, through symbol

prompt processing, abbreviation detection conversion, and sentence analysis to perceive the information expressed in the written language emotion and improve the accuracy of the model. Parinaz et al. [20] proposed an attention-based encoder and decoder, which improves the accuracy of the model by extracting potential dependencies between targets. However, these methods ignore the effective use of existing emotional resources and features. Song et al. [21] proposed an attention-based long short-term memory network text emotion recognition method, which trains important word vectors in the embedded space through joint coding to improve the accuracy of the model. Li et al. [22] proposed a semisupervised machine learning algorithm, which dynamically generated new corpora in different random subspaces to solve the problem of data imbalance and improve the accuracy of the model. Srinivasan and Anita [23] proposed a method of hierarchical training to improve the emotion recognition of text. By extracting an emotional dictionary from an annotated emotional corpus and then hierarchically training the model, the accuracy of the model is improved. However, these methods are prone to overfitting when the data volume is large.

Li et al. [24] proposed a text emotion classification method (emotion-feature-enhanced deep neural network, SDNN) based on the emotion attention mechanism of the deep neural network. The attention mechanism is used to select keyword phrases related to emotion words to improve the efficiency of the model. However, when the amount of data is small, the model does not work well. Liu et al. [25] proposed a method of surviving a large amount of supervised data for short-text emotion analysis (large-scale convolutional neural network, LSCNN). By generating large-scale artificial training data, the model's Pan-Chinese ability is improved. The running speed is slow.

Based on the above analysis, it can be seen that although deep learning-based methods have strong feature learning capabilities and also demonstrate certain advantages in performance, the text representation of such methods is single and ignores the effectiveness of existing emotional resources and features use.

3. Proposed Text Emotion Classification Model

3.1. Text Emotion Classification Model Based on the Convolutional Neural Network. Because the original short text has irregular problems, it leads to inaccurate part-of-speech tagging. Therefore, the trained word vectors may also be inaccurately expressed. If directly input to the convolutional neural network, it will cause the model to overfit. This paper uses a dual-mode channel input strategy. The proposed DDCCNN consists of an input layer, a convolutional layer, a pooling layer, and a classifier layer. It introduces extended features for text representation in a weighted form and compares it with text based on word vector features. The representation is integrated into the training process of the model, and at the same time, the dimensionality reduction expression of the features is realized through continuous pooling. The overall framework is shown in Figure 1.

In Figure 1, the DDCCNN structure has four layers, from top to bottom, in order.

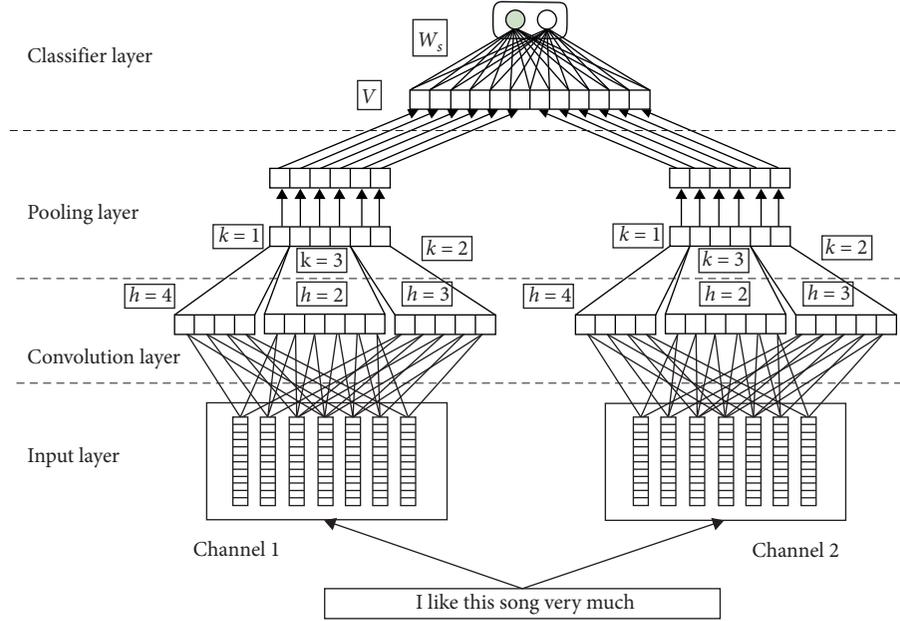


FIGURE 1: Proposed text emotion classification model.

3.1.1. Classifier Layer. In order to get the classified category estimate, we need to use the classification function softmax \hat{c} to perform normalization operations. The specific process is as follows:

$$p(y_k) = \frac{\exp(\omega_k \cdot \hat{c} + b_k)}{\sum_{i=1}^n \exp(\omega_i \cdot \hat{c} + b_i)}, \quad (1)$$

where ω_i and b_i are the parameters and offsets of the corresponding output category i of the fully connected layer and n is the number of categories. Assuming that the input text is X_k , its category is y_k , and θ is the model parameter, there is $p(y_k) = P(y_k | X_k, \theta)$. A stochastic gradient descent algorithm is used to minimize the negative log-likelihood function, and each training update each parameter in the network through back propagation until the model reaches the fit. Then the objective function of network training is

$$\theta := - \sum_{k=1}^N \log P(y_k | X_k, \theta), \quad (2)$$

where N is the training sample size.

Assume that the length of the input sentence D in the network is 7, and the dimension of the word vector and the extended feature vector are both 12. As can be seen from Figure 1, each layer of the network from the bottom up includes trainable parameters (weights). The 2 channels in the convolutional layer contain 3 convolution kernels ω_c of size 2×12 , 3×12 , and 4×12 , respectively. These 3 convolution kernels and the input matrix are, respectively, convolved to obtain 3 feature maps c , the sizes are 6×1 , 5×1 , and 4×1 . The pooling layer is composed of two feature maps of size 6×1 , which are obtained by the convolutional layer through dynamic k -max pooling operation, and the k values

are 3, 2, and 1, respectively. Finally, the output feature maps of the two channels in the pooling layer are spliced to form the final feature vector \mathbf{v} , which is then connected to a softmax classifier to output the final category label.

3.1.2. Pooling Layer. The pooling layer samples by setting a fixed step in the pooling area. Assuming that the height of the pooling area is h and the width is m , the entire feature map is divided into several $m \times h$ subareas, and then, the maximum pooling operation is performed on each subarea, and the corresponding feature value after pooling is the output:

$$c_p = \max(c_i). \quad (3)$$

Finally, the extracted text features are stitched together to form abstract text features, which can be expressed as

$$c' = [c_{p1,1}, \dots, c_{p1,m}, c_{p2,1}, \dots, c_{p2,m}, c_{p3,1}, \dots, c_{p3,m}], \quad (4)$$

where suppose the number of convolution kernels in each group is m ; $c_{pi,j}$ represents the j^{th} feature obtained by convolution of the convolution kernel size when the convolution kernel height is i , in order to further abstract the features while mining deeper semantics.

3.1.3. Convolutional Layer. The convolution layer performs a convolution operation on the word vector of the input layer through a convolution kernel and operates on each fixed-size window to generate an abstract feature output. When performing convolution operations on text, a convolution kernel of size $h \times n$ is generally selected, where h is the set word vector dimension, and n represents the number of words selected for each convolution operation. In this paper, three sets of convolution kernels with convolution

kernel sizes of 3, 4, and 5 are selected. The convolution operation is expressed as

$$c_i = f(W_1 \cdot L_{1,i:i+h} + W_2 \cdot L_{2,i:i+h} + b), \quad (5)$$

where c_i represents the i^{th} eigenvalue obtained after the convolution operation; $f(\cdot)$ represents the activation function ReLU; W_1 and W_2 , respectively, represent the weight parameter matrix of the convolution kernel on the two mode channels, $W_1, W_2 \in R^{h \times k}$; $h \times k$ represents the size of the convolution kernel window, each scan of a word sequence area with a height of h and a dimension of k will produce a feature value; $L_{1,i:i+h}$ represents the word vector matrix from the i^{th} word to the $(i+h)^{\text{th}}$ word in the text; $L_{2,i:i+h}$ represents the word vector matrix from the i^{th} word to the $(i+h)^{\text{th}}$ word in the text; and b represents the offset term of the convolutional layer. A convolution operation is performed on the word vector matrix in each window in the short text to obtain a feature map, expressed as

$$c = (c_1, c_2, \dots, c_{N-h+1}), \quad (6)$$

where N represents the number of words in short text; h is the length of the convolution window; and c represents the feature map formed by the convolution operation of the word vector matrix represented by the short text.

3.1.4. Input Layer. The output of the pooling layer will be used as the input of the last fully connected layer, expressed as

$$c'' = f'(W' \cdot c' + b'), \quad (7)$$

where c' is the vector obtained by pooling; W' is the weight parameter matrix of the fully connected layer; b' represents the offset term of the fully connected layer; $f'(\cdot)$ represents the activation function ReLU; and finally, c'' converts the straightening to a long vector \hat{c} .

3.2. Model Optimization. Batch normalization (BN) can improve the training speed of the network to a certain extent [26]. The basic idea is to use each batch of samples to train the network and then normalize the output data of each layer of the network.

Suppose the set of input data of a layer of network is $\{x_1, x_2, \dots, x_k\}$. The number of samples in the batch is k , and the BN technology first normalizes the batch of data:

$$\hat{x}_i = \frac{(x_i - E[x])}{(\sqrt{\text{Var}[x] + \varepsilon})}, \quad (8)$$

where $E[x]$ represents the sample mean of the batch input; $\text{Var}[x]$ represents the sample variance of the batch input; and ε is a numerical constant added to the variance, generally a positive number close to zero is used to ensure the reconstruction transformation response numerical stability.

The normalized data conforms to the distribution with a mean of 0 and a variance of 1. Most of them fall in the unsaturated region of the activation function, which prevents the gradient from disappearing and accelerates the

model training speed; However, after normalization, the original expression of the upper layer network is inevitably destroyed, so the reconstruction parameters γ and β are introduced to transform the batch of normalized data according to the reconstruction function to obtain the final output data $\{y_1, y_2, \dots, y_k\}$ of the layer:

$$y_i = \text{BN}_{\gamma, \beta}(\hat{x}_i) = \gamma \hat{x}_i + \beta. \quad (9)$$

Using BN technology to normalize the internal data of the network is equivalent to introducing a BN network layer into the network. In this model, the BN layer is added directly after each convolution operation of the convolution channel. At the same time, in order to ensure the smoothness of the through channel, this operation is not added to the through channel.

Padding controls the dimension of the input and output feature matrix of the convolution layer by padding zeros around the feature matrix. Enhance the feature extraction of edge data, and at the same time, perform dimension control on text data, so that the deep network of the feature extraction part can be trained. Let the input feature matrix, convolution kernel, padding, and step size output feature matrix have the same length on different axes i, c, p, s , and out, respectively. When padding is used, the length relationship is as shown in the following equation (10):

$$\text{out} = \begin{cases} \left\lceil \frac{(i-c)}{s} \right\rceil + 1, & i \geq c + s, \text{ no,} \\ \left\lceil \frac{(i+2p-c)}{s} \right\rceil + 1, & i \geq c + s - 2p, \text{ yes.} \end{cases} \quad (10)$$

Because some data have the characteristics of less a priori information, the dimension is smaller after data pre-processing, when padding is not used, the dimension of the feature extraction part will quickly decline, and the deep network cannot be trained. Therefore, the padding technique of $p = 1$ is used in this model, and the combination of $c = 3$ and $s = 1$ convolution kernel can ensure that the dimension of the feature matrix remains unchanged.

4. Improve Word Vector Training

This section will introduce in detail how to use the part-of-speech probability factor λ to improve word vector training and show the improved text matrix expression of different models based on the scene weight factor η [27].

4.1. Part-Of-Speech Probability Factor. Due to the sparseness of the semantic features of short text, it is insensitive to the part-of-speech factors of the text, which affects the subsequent classification effect.

Splice part-of-speech probabilities of words with words to form Word-coefficient part of speech (Word-CPOS). For example, (China, 0.21) and (Ethnic, 0.44), and then convert the original text into a Word-CPOS sequence, which is used as the input of the word vector model.

This paper proposes a part-of-speech probability factor to quantify the contribution of words with different parts of speech to the text. In fact, the word parts of speech are divided into nouns, verbs, adjectives, adverbs, and other parts of speech. The assumed value of λ is $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$, and the calculation method is

$$\left. \begin{aligned} \lambda_1 &= \frac{\text{num}(n)}{W} \\ \lambda_2 &= \frac{\text{num}(v)}{W} \\ \lambda_3 &= \frac{\text{num}(\text{adj})}{W} \\ \lambda_4 &= \frac{\text{num}(\text{adv})}{W} \\ \lambda_5 &= \frac{\text{num}(\text{other})}{W} \end{aligned} \right\}, \quad (11)$$

in the formula,

$$\begin{aligned} W &= \text{num}(n) + \text{num}(v) \\ &\quad + \text{num}(\text{adj}) + \text{num}(\text{adv}) \\ &\quad + \text{num}(\text{other}), \end{aligned} \quad (12)$$

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 = 1,$$

with $\text{num}(n)$, $\text{num}(v)$, $\text{num}(\text{adj})$, $\text{num}(\text{adv})$, and $\text{num}(\text{other})$, respectively, to represent the number of nouns, verbs, adjectives, adverbs, and other part-of-speech words; and W represents the number of words in the text after the word segmentation operation.

4.2. Word Vector Scene Factor η . The word vector tool word2vec includes the CBOW model and SkipGram model. Where CBOW is more suitable for small data, and SkipGram performs better for large corpus. The two models have different word-cutting effects, and the income also changes according to different usage scenarios.

Therefore, this paper constructs and introduces a scene weighting factor η , which is related to the size of the corpus. Its construction method is shown in formula (13):

$$\eta(\eta_1, \eta_2) = \begin{cases} \eta_1 = 1.5, \eta_2 = 0.5, Z \leq 1 \text{ GB}, \\ \eta_1 = 0.5, \eta_2 = 1.5, Z \geq 1 \text{ GB}. \end{cases} \quad (13)$$

In the formula, η_1 represents the weight of the word vector based on the CBOW training model; η_2 represents the weight of the word vector based on the SkipGram training model; and Z represents the memory size of the corpus size in GB.

4.3. Text Matrix A after Introducing λ and η . Suppose a sentence in the text is $S = (x_1, x_2, \dots, x_N)$. In the formula, x_i represents a word in the text, and the part-of-speech probability is used to obtain the part-of-speech probability

$Q = (w_1, w_2, \dots, w_N)$ of each word. Then stitch it into a Word-CPOS sequence:

$$\text{SQ} = ((x_1, w_1)(x_2, w_2), \dots, (x_n, w_N)). \quad (14)$$

Input SQ into the pretrained Word-CPOS model. Each Word-CPOS item will be converted into n -dimensional Word-CPOS vector sq_n , and then, this sentence will be converted into the $W \times n$ -dimensional matrix. Since there is more than one sentence in the short text, the sentence in the short text needs to be cascaded to obtain the text matrix A . Finally, according to the size of the corpus, choose the scene weighting factor η to get the Word-CPOS input vector.

4.4. Semantic Features Based on Word Vectors. Existing research shows that in the case of supervised learning of large-scale labeled data, using unsupervised learning to initialize the input value of the neural network is an effective way to improve training results and accelerate training convergence. Therefore, this article uses the word2vec tool to calculate the word vector of the Weibo text as the initial value of the input of the convolutional neural network.

The word2vec is an open source and efficient tool for computing distributed word vectors on Google. This tool takes a corpus as input and calculates a word vector representation. It uses a vocabulary constructed from text data as training data and then learns the high-dimensional vector representation of the phrase, that is, mapping the phrase into a finite-dimensional high-dimensional space. Compared with the traditional one hot encoding representation model, the word vector is a dense vector representation, and it is easier to calculate the distance metric between the vectors, so it is more suitable for natural language processing and machine learning and at the same time for the shortness and arbitrariness of Weibo text. It has better robustness.

The word2vec tool contains two models for computing word vectors, namely, the continuous word bag model (CBOW) and the skip word model (SkipGram). The tool is based on the neural network language model, by removing hidden layers and using hierarchical softmax and negative sampling techniques to reduce computational complexity and optimize training results. The word2vec tool is used to train the Chinese corpus after word segmentation, and the vector representation of the specified dimension of each phrase can be calculated. Using the word vector of the phrase can easily calculate the cosine distance of the word vector as a measure of similarity between the phrases, so the word vector represents the deep semantic connection between the phrases in the corpus.

4.5. Dynamic k -max Continuous Pooling Strategy. In order to make up for the lack of information loss caused by the maximum pooling [29] strategy, this paper proposes a dynamic k -max continuous pooling strategy. The basic idea is that during the downsampling of the feature map in the pooling layer, the effect of the height of the sliding window of

the convolution kernel on the size of the generated feature map is appropriately considered. That is, the height of the convolution kernel is used as an important basis for the value of the downsampling number k of the feature map. The higher the height of the convolution kernel, the less the number of downsampling. The lower the height of the convolution kernel, the more downsampling. The calculation formula of k value is as follows:

$$k = \lceil \frac{s}{h} \rceil, \quad (15)$$

where s represents the sentence length, and h represents the height of the convolution kernel. Intuitively, the value of k is proportional to the length of the sentence and inversely proportional to the height of the convolution kernel. Compared with the maximum pooling strategy, the algorithm in this paper uses dynamic k -max continuous pooling strategy that has obvious advantages. The comparison of the two pooling strategies is shown in Figures 2 and 3.

In Figures 2 and 3, the length of the input sentence is 6, and the height h of the convolution kernel is 2 and 3, respectively. The part with the heavier background color in the feature map represents the most important feature. As can be seen from Figures 2 and 3, when the height of the convolution kernel is 2 and 3, the maximum pooling operation only retains an important feature in each feature map and continuous pooling according to dynamic k -max. The operation retains top 3 and top 2 important features with the largest values, respectively. In summary, the advantages of the dynamic k -max continuous pooling strategy are as follows: extract multiple important semantic combination features and retain the relative order information between the features.

4.6. Model Training. The DDCCNN proposed in this paper can handle texts of different lengths. The matrix of the convolution kernel adopts the feature vectors obtained by two-channel convolution of three convolution kernels of sizes 2 and 3 as the input of the softmax classifier. The final output of the predicted value of the emotion category is defined as follows:

$$\hat{y}^{(i)} = \frac{1}{\left(1 + e^{-(w_i TC^{(4)} + b_i)}\right)}. \quad (16)$$

In the formula, $\hat{y}^{(i)}$ represents the predicted value of the i^{th} -sample; and w_i and b_i represent the weight and offset to be trained, respectively. Then the objective function of the model is

$$\text{Loss} = - \sum_x \left[y^{(i)} \ln(\hat{y}^{(i)}) + (1 - y^{(i)}) \ln(1 - \hat{y}^{(i)}) \right]. \quad (17)$$

From equation (17), it can be known that $\text{Loss} \geq 0$. Suppose that the label of training sample is $y = 1$, then $\text{Loss} = -\ln(\hat{y})$, then $\hat{y} \approx 1$, and then $\text{loss} = 0$; when the label of the training sample is $\text{loss} = -\ln(1 - \hat{y})$, we can know loss , then $\hat{y} \approx 0$, then $\text{loss} = 0$.

5. Experimental Results and Analysis

In order to verify the effectiveness of the proposed DDCCNN text emotion analysis method, a full experimental evaluation was carried out on the Weibo text data collection collected by ourselves. This paper compares the experiments based on JBNN proposed in [15], SDNN proposed in [24], and LSCNN and DDCCNN proposed in [25]. The experiment was conducted on the Win10 operating system with 8G memory, Intel Core i5-3220 3.3 GHz, programming language python 2.7, and the deep learning system TensorFlow.

5.1. Experimental Dataset. The dataset used in the experiment includes word2vec training data and the Microblog dataset collected by myself. In this paper, the training of word vectors uses an unspliced corpus, and the training of Word-CPOS vectors uses a corpus that has been stitched with part-of-speech probabilities. This article uses Google's open source tool word2vec to obtain a text word vector representation, which includes SkipGram and CBOW methods. At the same time, two methods are used for word vector training for subsequent comparison experiments. And uniformly use the SkipGram method to train the part-of-speech mosaic matrix, and set the weight of the input text representation of the convolutional neural network according to the scene weight factor. Word vector and Word-CPOS vector training parameters are as follows: the text represents the dimension of the vector 100 dimensions; the size of the context window is 10; and the number of vector training iterations is 10.

For the Microblog dataset, no suitable public dataset was found, so about 10000 microblog texts were obtained by self-collection. Because the data on microblog platform is based on Internet information, ordinary users cannot access the Microblog database. But they provide a common API interface for ordinary users to mine microblog information. In this paper, the web data crawler collection software based on Python language is divided into two parts: crawling microblog module and microblog information storage module. Crawler crawling microblog module through the combination of web technology and regular matching technology to complete the analysis and acquisition of network pages, and microblog information storage module through the crawling data into the required data structure stored in the database. In the experiment, 80% of the text is extracted as the training set, and 20% of the text is used as the test set.

First of all, in the experiment, the original corpus was cleaned, and punctuation marks, English characters, and other special characters were removed to ensure the standardization of the corpus. Then, perform word segmentation on the corpus. Finally, the part-of-speech probabilistic annotation is performed on the corpus to construct a Word-CPOS vector. In the experiment, ReLU was used as the activation function, the Adam optimizer method was used as the gradient update rule, the number of filters used the single-factor variable method, the experiment established

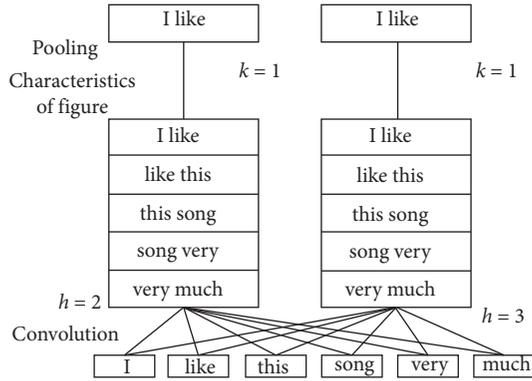


FIGURE 2: Maximum pooling strategy diagram.

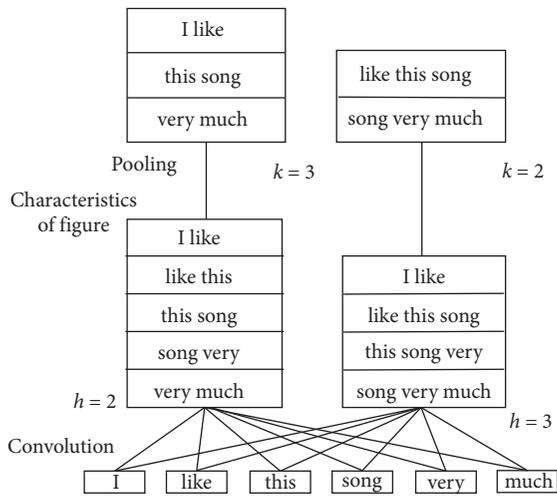


FIGURE 3: Dynamic k -max continuous pooling diagram.

the optimal parameter as 100, and the convolution window size as 3, 4, and 5. At the same time, in order to avoid the phenomenon of overfitting and improve the generalization ability of the model, the performance of the neural network structure is improved through dropout. By randomly ignoring the neurons in the convolutional layer to average the prediction probability, it can also reduce the interaction between the neurons in the hidden layer and optimize the structure of the model.

5.2. Evaluation Index. In this paper, the evaluation criteria for text emotion classification are mainly measured from the classification accuracy and $F1$. For a given sample size of N , the actual label of sample x_i is y_i , the classification label is \hat{y} , and the formula for calculating accuracy is

$$\text{Accuracy} = \frac{\sum_{i=1}^N |y_i = \hat{y}|}{N}. \quad (18)$$

$F1$ is the reconciled average value of the precision rate and recall rate. Assuming the precision rate and recall rate, the specific calculation formula is as follows:

$$P = \frac{TP}{(TP + FP)},$$

$$R = \frac{TP}{(TP + FN)}, \quad (19)$$

$$F1 = \frac{2 * P * R}{(P + R)},$$

where TP means the positive example data correctly classified by the classifier; TN means the negative example data correctly classified by the classifier; FP means the negative example data incorrectly marked as positive example data; and FN indicates positive example data that was incorrectly marked as negative example data.

5.3. The Effect of the Number of Convolution Layers on the Results. When using traditional convolutional neural networks, in order to ensure comparability, in addition to the last layer of convolutional layers, other layers also correspondingly add ReLU activation function to perform nonlinear transformation and add the BN layer and padding technology for optimization. The correct rate of different network structures is shown in Table 1:

From Table 1, we can see that the existing several convolutional neural networks achieve the best results at 32 layers, with correct rates of 90.75%, 91.32%, and 92.15%, but starting from 40 layers, the existing several convolutional neural networks. The accuracy rate of the network has been declining. This is because as the depth of the convolutional neural network increases, the extracted feature dimensions are large, and the model cannot be learned well. The prediction accuracy of the double-channel convolutional neural network gradually increases with the increase of depth. When the network reaches 56 layers, the prediction accuracy rate has reached 94.16%, and due to batch normalization, the performance of the double-channel convolutional neural network has remained stable.

5.4. Comparison with Current Advanced Algorithms. In order to verify the effectiveness and superiority of the algorithm in this paper and the effect of emotion classification under different models, two sets of comparative experiments were conducted with JBNN, SDNN, and LSCNN. Experiment 1 uses a callback function to select the model with the highest correct rate as the model's correct rate and $F1$ value. Figure 4 shows the classification results of the experimental dataset under different neural networks. Experiment 2 compared DDCCNN with (Bag of words, BOW) + (Support vector machine, SVM) algorithm and word2vec + SVM algorithm. BOW + SVM means that BOW is used to represent text features, and SVM is used as a classifier. The accuracy rate and $F1$ value results of different algorithms are shown in Figure 4.

It can be seen from Figure 4 that the correct rate, $F1$ value, and recall rate of DDCCNN have achieved 94.16%, 92.61%, and 92.71%, respectively. The other three methods have achieved relatively low classification results. The other

TABLE 1: Comparison of correct rates of different network structures (%).

Number of layers	JBNN	SDNN	LSCNN	DDCCNN
16	90.13	90.47	92.01	93.98
24	90.34	91.28	92.08	94.07
32	90.75	91.32	92.15	94.26
40	89.24	90.06	90.27	94.18
48	88.73	89.57	90.15	94.22
56	88.24	89.14	89.33	94.26

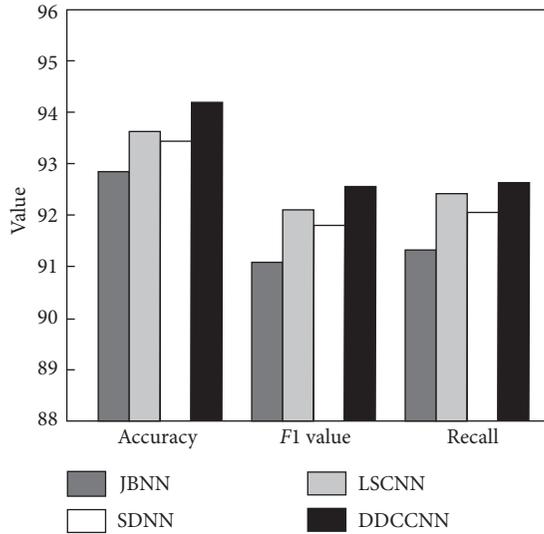


FIGURE 4: Comparison of classification results of different models.

three methods have achieved relatively low classification results. From the analysis reason, it can be known that the existing JBNN, SDNN, and LSCNN ignore the semantic information of the sentence, and for the large number of features extracted after convolution, the learning ability is poor, resulting in the model's accuracy and $F1$ value slightly lower. DDCCNN extracts the edge features of the data, expands the coverage of the features by weighting, and merges the word vector features, thereby extracting richer semantic information and making the model more accurate.

It can be seen from Figure 5 that the performance of the word2vec+SVM algorithm is higher than BOW+SVM. This is because the traditional BOW ignores the semantic information of the sentence, so the SVM classification effect is relatively poor. After word2vec training the word vector, the semantic information makes SVM classification performance significantly improved. It can be seen from the experiment that the correct rate and $F1$ value of the DDCCNN model are significantly better than those using the SVM classifier, which shows that the shallow machine learning algorithm has limited fitting ability in the case of too much data, and the DDCCNN model is effective in emotion analysis. The use of k -max pooling strategy effectively reduces the loss of text information and further improves the text classification performance of the model.

In order to show the model effect of text emotion classification under different iteration times, Figures 6–8,

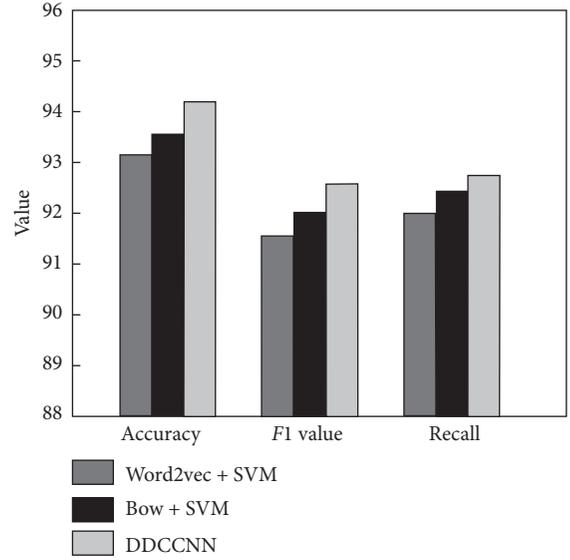


FIGURE 5: Comparison of classification results of different algorithms.

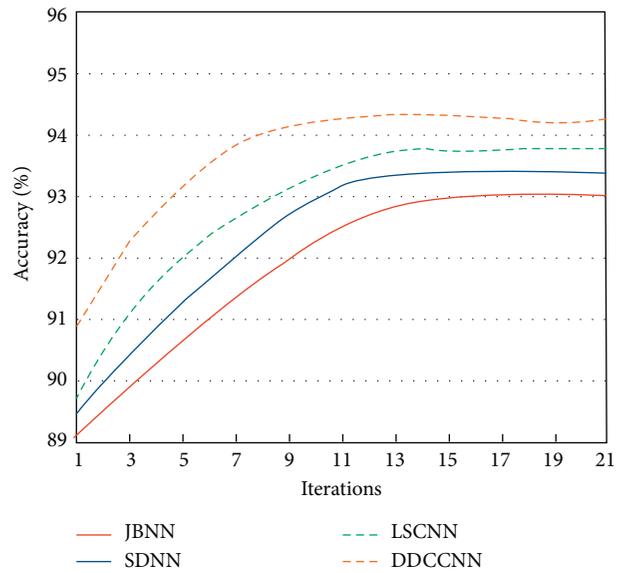


FIGURE 6: Comparison of model correctness for different iterations.

respectively, show the correct rate, $F1$ value, and recall rate of each model under different iteration times.

It can be seen from Figures 6–8 that the accuracy rate of the JBNN model, the $F1$ value, and the recall rate are the lowest. The fluctuation rate of the accuracy rate of the LSCNN model is larger than that of other models. The quality is poor, and the overall semantic information cannot be expressed well. The emotion classification accuracy rate and $F1$ value of the DDCCNN model have always been higher than other algorithms because the simultaneous use of extended features and dynamic k -max continuous pooling strategy can enhance the ability of the model to obtain high-level text features while enhancing the emotional meaning of text. In addition, the DDCCNN has a small fluctuation

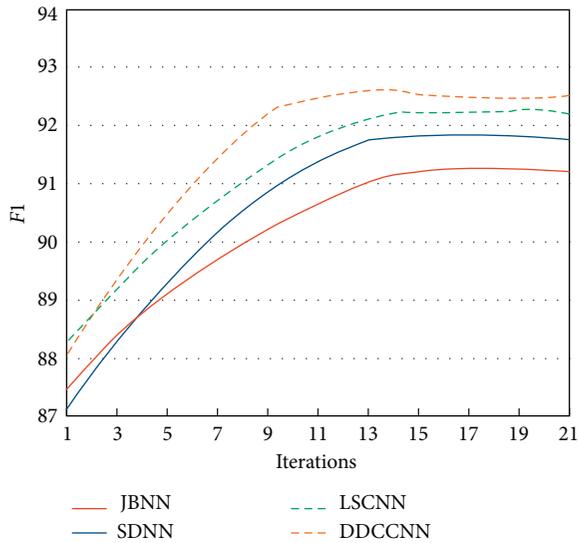


FIGURE 7: Comparison of $F1$ values of models with different iteration times.

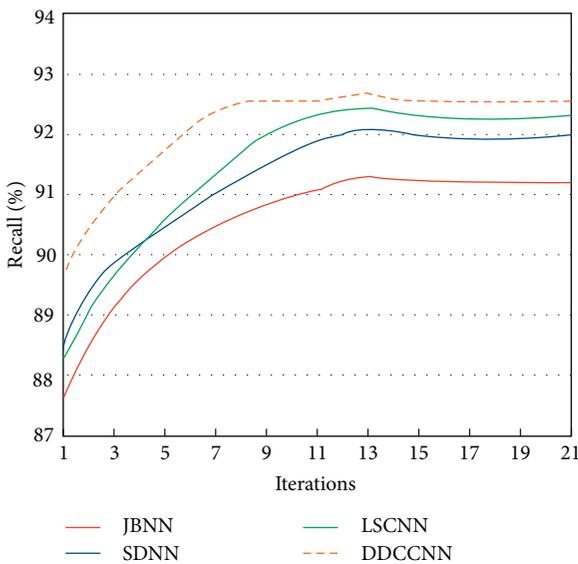


FIGURE 8: Comparison of recall rates of models with different iteration times.

range, and when the number of iterations is 13, the DDCCNN model achieves the highest accuracy and $F1$ value. At this time, in order to avoid overfitting the model, the iteration can be stopped.

6. Conclusions

In view of the sparse features in the text dataset, the existing deep learning methods have insufficient feature extraction capabilities in text emotion classification tasks. A new text emotion analysis method for e-commerce review using deep learning double-channel convolutional neural networks is proposed. In the proposed method, by adding zeros around the feature matrix, controlling the input and output feature matrix dimensions of the convolution layer, and enhancing

the feature extraction of edge data, an improved word vector training method is introduced, which effectively improves the model's ability to extract features, and expand the coverage of features by weighting. Experimental results show that the accuracy of the proposed DDCCNN is better than the comparison algorithm.

Text emotion recognition has broad application prospects in the era of artificial intelligence. Although the technology based on a deep learning double-channel convolutional neural network has proposed a new method that is different from traditional text emotion recognition to solve the problem of sparse features in text datasets and has achieved good results. However, the study found that the number of positive texts containing negative words was mistakenly classified as negative texts in the total number of misclassified collections is higher. Therefore, in the future text emotion analysis task, we will try to solve the problem of the impact of negative words on forward classified text. Introduce multiple features, such as word order and different word vectors, to further improve the model classification performance.

Data Availability

The data used to support this study are made available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the study.

Acknowledgments

This paper was supported by the National Natural Science Foundation of China (61672179), Research Fund for the Doctoral Program of Higher Education of China (20122304110012), and Basic Business Special Project of Heilongjiang Education Department (135109313).

References

- [1] M. Al-Ayyoub, H. Al-Sarhan, M. Al-So'Ud et al., "Framework for affective news analysis of Arabic news: 2014 Gaza attacks case study," *Journal of Universal Computer Science*, vol. 23, no. 3, pp. 327–352, 2017.
- [2] T. Allee, M. Elsig, and A. Lugg, "The ties between the world trade organization and preferential trade agreements: a textual analysis," *Journal of International Economic Law*, vol. 20, no. 2, pp. 333–363, 2017.
- [3] H. H. Gao, W. Q. Huang, and X. X. Yang, "Applying probabilistic model checking to path planning in an intelligent transportation system using mobility trajectories and their statistical data," *Intelligent Automation And Soft Computing (Autosoft)*, vol. 25, no. 3, pp. 547–559, 2019.
- [4] H. H. Gao, W. Q. Huang, Y. C. Duan et al., "Research on cost-driven services composition in an uncertain environment," *Journal of Internet Technology (JIT)*, vol. 20, no. 3, pp. 755–769, 2019.
- [5] C. Zhang and P. Yue, "Diffused feature study of Weibo hot topic," *Journal of Geomatics*, vol. 43, no. 2, pp. 115–118, 2018.

- [6] H. H. Gao, Y. C. Duan, L. X. Shao et al., "Transformation-based processing of typed resources for multimedia sources in the IoT environment," *Wireless Networks*, vol. 25, no. 3, 2019.
- [7] H. H. Gao, Y. S. Xu, Y. Y. Yin et al., "Context-aware QoS prediction with neural collaborative filtering for internet-of-things services," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4532–4542, 2019.
- [8] S. Li, M. Shao, and Y. Fu, "Person re-identification by cross-view multi-level dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2963–2977, 2018.
- [9] S. Zhao, Z. Su, and G. Miao, "Application of English education information management system based on convolution neural network classification algorithm," *The International Journal of Electrical Engineering & Education*, 2020.
- [10] X. J. Ma, H. H. Gao, H. H. Xu, and M. J. Bian, "An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 249, 2019.
- [11] S. Feuerriegel, B. Kratzwald, M. Kraus et al., "Deep learning for affective computing: text-based emotion recognition in decision support," *Decision Support Systems*, vol. 115, pp. 24–35, 2018.
- [12] N. Li and D. D. Wu, "Using text mining and sentiment analysis for online forums hotspot detection and forecast," *Decision Support Systems*, vol. 48, no. 2, pp. 354–368, 2010.
- [13] S. Massie, N. Wiratunga, A. Bandhakavi et al., "Lexicon based feature extraction for emotion text classification," *Pattern Recognition Letters*, vol. 93, pp. 133–142, 2017.
- [14] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria, "Deep learning-based document modeling for personality detection from text," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 74–79, 2017.
- [15] H. H. He and R. Xia, "Joint binary neural network for multi-label learning with applications to emotion classification," *Springer Cham*, vol. 8, no. 14, 2018.
- [16] R. Rahmani, A. Ghodsi, H. Veisi et al., "Emotion analysis based on improved pre-trained word embeddings," *Expert Systems with Application*, vol. 117, pp. 139–147, 2019.
- [17] F. Philipe, D. H. Melo, M. Dalip, and M. Junior, "A stable emotion analysis method based on the combination of off the shelf approaches," *Journal of the Association for Information Science and Technology*, vol. 70, no. 3, pp. 242–255, 2019.
- [18] Z. Wang, S. Y. M. Lee, S. Li, and G. Zhou, "Emotion analysis in code-switching text with joint factor graph model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 3, pp. 469–480, 2017.
- [19] A. Neviarouskaya, H. Prendinger, and M. Ishizuka, "Recognition of affect conveyed by text messaging in online communication," *Online Communities and Social Computing*, pp. 141–150, 2007.
- [20] S. Parinaz, I. Diana, and Z. Xiaodan, "Exploring deep neural networks for multitarget stance detection," *Computational Intelligence*, vol. 35, no. 1, pp. 82–97, 2018.
- [21] M. Song, H. Park, and K.-s. Shin, "Attention-based long short-term memory network using sentiment lexicon embedding for aspect-level sentiment analysis in Korean," *Information Processing & Management*, vol. 56, no. 3, pp. 637–653, 2019.
- [22] S. Li, Z. Wang, G. Zhou, and S. Y. M. Lee, "Semi-supervised learning for imbalanced emotion classification," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1826–1831, Barcelona, Spain, 2011.
- [23] R. Srinivasan and R. Anita, "Text to speech conversion with emotion detection," *International Journal of Applied Engineering Research*, vol. 13, no. 14, pp. 11512–11517, 2018.
- [24] W. K. Li, P. Y. Liu, Q. Y. Zhang, and Liu, "An improved approach for text sentiment classification based on a deep neural network via a sentiment attention mechanism," *Future Internet*, vol. 11, no. 4, p. 96, 2019.
- [25] M. Liu, W. Wu, Z. Gu, Z. Yu, F. Qi, and Y. Li, "Deep learning based on batch normalization for P300 signal detection," *Neurocomputing*, vol. 275, pp. 288–297, 2018.
- [26] Y. F. Zhou and N. Chen, "The LAP under facility disruptions during early post-earthquake rescue using PSO-GA hybrid algorithm," *Fresenius Environmental Bulletin*, vol. 28, no. 12, pp. 9906–9914, 2019.
- [27] J. M. Ceballos and Á. F. Núñez, "Algorithm to compute minimal matrix representation of nilpotent lie algebras," *International Journal of Computer Mathematics*, vol. 97, no. 1-2, pp. 275–293, 2020.
- [28] Y. Zhou, H. Yu, Z. Li, J. Su, and C. Liu, "Robust optimization of a distribution network location-routing problem under carbon trading policies," *IEEE Access*, vol. 8, no. 1, pp. 46288–46306, 2020.
- [29] S. H. Wang, Y. D. Lv, Y. Sui et al., "Alcoholism detection by data augmentation and convolutional neural network with stochastic pooling," *Journal of Medical Systems*, vol. 42, no. 1, pp. 1–11, 2018.
- [30] M. S. A. Duais and F. S. Mohamad, "Paper dynamically-adaptive weight in batch back propagation algorithm via dynamic training rate for speedup and accuracy training," *Journal of Telecommunications & Information Technology*, vol. 4, no. 4, pp. 82–89, 2018.