IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 8, 2020, accepted January 10, 2020, date of publication January 15, 2020, date of current version January 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2966726

Variable Convolution and Pooling Convolutional Neural Network for Text Sentiment Classification

MIN DONG[1], YONGFA LI[1], XUE TANG[1], JINGYUN XU[2], SHENG BI[1], AND YI CAI[2], (Member, IEEE)
[1]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
[2]School of Software Engineering, South China University of Technology, Guangzhou 510006, China

Corresponding authors: Sheng Bi (picy@scut.edu.cn) and Yi Cai (ycai@scut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61703168 and Grant 61703284, in part by the Guangdong Province Science and Technology Plan Projects under Grant 2017A030310163 and Grant 2017A010101031, in part by the Shenzhen Basic Research Project under Grant JCYJ20160429161539298, and in part by the Shenzhen Peacock Project under Grant KQTD20140630154026047.

ABSTRACT With the popularity of the internet, the expression of emotions and methods of communication are becoming increasingly abundant, and most of these emotions are transmitted in text form. Text sentiment classification research mainly includes three methods based on sentiment dictionaries, machine learning and deep learning. In recent years, many deep learning-based works have used TextCNN (text convolution neural network) to extract text semantic information for text sentiment analysis. However, TextCNN only considers the length of the sentence when extracting semantic information. It ignores the semantic features between word vectors and only considers the maximum feature value of the feature image in the pooling layer without considering other information. Therefore, in this paper, we propose a convolutional neural network based on multiple convolutions and pooling for text sentiment classification (variable convolution and pooling convolution neural network, VCPCNN). There are three contributions in this paper. First, a multiconvolution and pooling neural network is proposed for the TextCNN network structure. Second, four convolution operations are introduced in the word embedding dimension or direction, which are helpful for mining the local features on the semantic dimensions of word vectors. Finally, average pooling is introduced in the pooling layer, which is beneficial for saving the important feature information of the extracted features. The verification test was carried out on four emotional datasets, including English emotional polarity, Chinese emotional polarity, Chinese subjective and objective emotion and Chinese multicategory. Our apporach is effective in that its result was up to **1.97%** higher than that of the TextCNN network.

INDEX TERMS Text sentiment classification, deep learning, CNN.

I. INTRODUCTION

Sentiment analysis, also called opinion mining, is a popular issue in the field of text analysis. It takes the subjective information in text as its research object, and attempts to recognize the categories of sentiment orientation of those opinions and attitudes that the subjective information reflects. The attitude may be self-judgment or self-assessment, or it may be an emotion or an emotional state. Therefore, the study of text sentiment classification needs to divide the text into subjective and objective parts first and then classify its emotional categories.

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

Text sentiment classification research is currently a challenging task that faces many challenges in emotional opacity, text formatting, text language, etc. Emotional opacity mainly refers to the ambiguous meaning of the text and the reliance on the context. For example,“this playground is very big” and “this is a very big noise”. Both sentences have the same phrase “very big”, but they express two opposite sentiment polarities. In terms of text formatting, the massive quantity of texts we analyze is mainly from the internet with variable lengths and formats. As to the text language, texts in Chinese greatly differ from texts in English in their own attributes: Chinese texts need word segmentations, whereas English texts do not need to be segmented becasue of the spaces between each word.

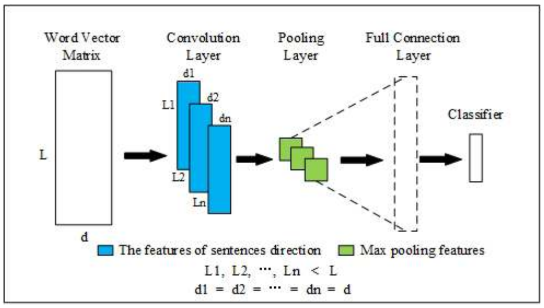


FIGURE 1. The network structure of TextCNN, L is the length of sentences, d is the dimension of the word vector.

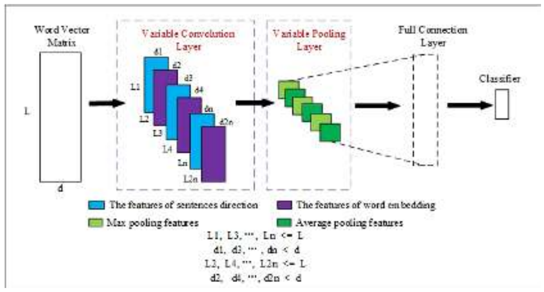


FIGURE 2. The overall structure of VCPCNN, L is the length of sentences, d is the dimension of the word vector.

The current ordinary method of text sentiment classification research is based on the sentiment dictionary, but this method needs to establish a high-quality sentiment dictionary and can be classified by hand-crafting. This kind of dictionary consumes a large number of resources and entails some disadvantages, such as it dose not contain sufficient words to cover a large scale. Another method is based on machine learning, which requires setting a high-efficiency machine learning model that has a higher classification accuracy than the method based on the sentiment dictionary. However, it requires people who have a relative knowledge background to design the features. It has a very limited resolution, and it is difficult to obtain comprehensive and deep text features. In recent years, in the field of natural language processing, the deep learning model has advanced. It has achieved hierarchical autogeneration features and end-to-end classification without artificial design.

TextCNN [1] (structure is shown in figure 1) has achieved good results in this respect. But, as to text sentiment classification, TextCNN only performs one-dimensional convolutions on sentence length. The width of convolution kernel is the dimension of the word vector. Thus, convolution layer can only extract semantic features on sentence direction and lose semantic information on the dimension of word vector. As is shown in figure 1, among all features and dimensions, the only variable is the sentence length L ($L1, L2, \ldots, Ln < L$), the dimension and direction of word embedding are equal to the width d($d1 = d2 = \ldots = dn$) of original word vector matrix. Meanwhile, pooling layer of TextCNN adopted a single max-pooling operation.

For the shortcomings of losing features on text word embedding dimension and inadequacy of single pooling in TextCNN, the paper propose VCPCNN to improve its convolution layer and pooling layer. The overall structure of VCPCNN is shown in figure 2, including input layer of word vector matrix, variable convolution layer, variable pooling layer, fully connected layer and categorization. Variable convolution layer refers to extracting features from sentence length L and word embedding dimension d respectively, the size of feature extracted from sentence length is L_i x d_i (i belongs to (1,3,5,…,n) and $Li <= L$, $di < d$), the size of feature extracted from word embedding dimension is L_j x d_j(j belongs to (2,4,6,…,2n) and $Lj <= L$, $dj < d$). Variable pooling layer refers to adopting max-pooling and average pooling to pooling features extracted from variable convolution layer respectively, and then combined these results as input of fully connected layer. Specific parameter configuration and convolution method of convolution kernel on convolution layer were introduced in detail in Sect III.

The rest of the paper is organized as follows. Section II reviews the developmental history of the text sentiment analysis from the traditional sentiment dictionary and machine learning, to the contribution of deep learning in text sentiment classification. Section III introduces the basic structure of TextCNN and then describes the specific structural design of VCPCNN. Section IV analyzes the dataset, model parameters and the results of the experiment. Section V compares VCPCNN with the existing methods, summarizes the enhancing effect that deep learning achieved in the four datasets, especially in the Chinese dataset, and describes the contribution of this method in Chinese sentiment classification.

II. RELATED WORK

With the development of sentiment classification research, its requirements are becoming more detailed. Thus, the particle size of sentiment categories is also divided into more detailed categories from traditional binary sentiment (positive and negative) to ternary sentiment (positive, neutral and negative). For example, the sentiment categories of UCI's [2] emotion identification sentence dataset are simply divided into positive sentiment and negative sentiment. The Stanford emotion dataset [3] divided emotion into five categories: very positive, positive, neutral, negative and very negative. The Kaggle [4] competition announced a film comment dataset that divided emotion into negative, slightly negative, neutral, slightly positive, and positive. The dataset published by Yelp [5] includes over five million user comments on a scale of 1-5. However, human emotions are complicated. Simply dividing emotions cannot realize the actual tasks well. Therefore, Liew *et al.* [6] introduced and used the EmoTweet-28 dataset, which contains 28 emotional categories. Wu *et al.* [7] subdivided emotions into 32 categories, making the expression of emotions in the same direction more refined.

In the past, sentiment classification research mainly relied on high-quality emotional dictionaries, such as the English dictionary "WordNet [8], [9]", HowNet in the Chinese mainland and NTUSD [10] published by Taiwan University. However, for some specific tasks of sentiment classification, we tend to merge, expand and rebuild the emotional dictionary to improve the classification effect. Keshavarz and Abadeh [11] combined corpora-based and lexicon-based methods to build adaptive sentiment lexicons to improve the polarity classification of sentiments in microblogs. Chekima and Alfred [12] utilized existing sentiment analysis resources and tools from English along with the automated machine translation capability to automatically build a Malay sentiment dictionary. Hao *et al.* [13] proposed an approach for automatically constructing a sentiment dictionary based on microblog emoticons by collecting microblog texts that are annotated by emoticons and forming a sentiment text corpus.

The development of machine learning has led to its gradual application to the analysis of textual sentiment [14]–[16]. Arunachalam *et al.* [17] and Verma and Thakur [18] et al. discussed common machine learning algorithms (such as Bayesian, LDA, and dynamic ontology classification) in text classification and emotional polarity information mining. Among them, the application of common machine learning algorithms in text sentiment analysis includes Bayesian [19], [20], LDA [21], [22], NB [23], SVM [24], [25], the maximum entropy method [26], and KNN [27]. Liu *et al.* [28] provided a method for multiclass sentiment classification based on an improved one-vs-one (OVO) strategy, and the support vector machine (SVM) algorithm was proposed. However, all of them have a common problem because they need to manually construct feature engineering.

In recent years, deep learning has made great progress in text sentiment analysis [29], [30] but still faces many challenges, as described in reference [31], [32]. **At the sentence-level**, Vieira and Moura [33] proposed a convolutional neural network for sentence-level classification tasks with different hyperparameter settings. Fu *et al.* [34] proposed a semi-supervised method (called CHL-PRAE) that combines the HowNet lexicon to train phrase recursive autoencoders. Xu and Cai [35] designed a neural network to incorporate context-relevant knowledge into a convolutional neural network for short text classification. All of them provide the best performance for sentence-level sentiment analysis. However, they also consider the emotional characteristics in the direction of the sentences. **At the multichannel-level**, Yoon and Kim [36] improved the sentiment classifier for predicting document-level sentiments from Twitter by using multichannel lexicon embeddings and applying the multichannel method on lexicon to improve lexicon features. Chen *et al.* [37] exposed the rich structures of sentences to deep neural networks, using a multichannel information crossing(MIX) model. In reference [38], multichannel was the use of CNN and LSTM to generate information channels for Vietnamese sentiment analysis. **In multiview**, Cai and Hao [39] proposed a multiview and attention-based BI-LSTM method for Weibo emotion recognition. Huang *et al.* [40]. implemented multiview learning based on heterogeneous deep neural networks for document-level opinion mining. **In addition**, Aziz Sharfuddin *et al.* [41] used a deep recurrent neural network with Bi-LSTM to classify Bengali text emotionally and achieved remarkable results. Hassan and Mahmood [42] also attempted to combine CNN and RNN and proposed a convolutional recurrent deep learning model for extracting local features of text and solving the problems of long-term dependencies. Zhang *et al.* [43] inspired by the methodology of three-way decisions, proposed 3W-CNN by combining traditional methods. In order to increase the performance of text classification, Lei *et al.* [44] propose a task-oriented representation that captures word-class relevance as task-relevant information. Whether in the direction of sentence length, multichannel features, or multiperspective features, the above methods are used to extract various forms of features at the level of sentence features to improve the accuracy of classification. However, all of them ignore the feature information on the dimension. The VCPCNN proposed in this paper makes up for this deficiency. Therefore, we propose the VCPCNN model to compensate for that shortcoming.

Yin and Schütze [45] also proposed a convolutional neural network structure (MVCNN) for sentence classification using the multichannel input method and a variable-size convolution operation. The multichannel input means that a small batch of sentences is integrated into sentences with the same length, and the unknown words of the corresponding channel are randomly initialized or obtain good initialization from the mutual learning stage described in the next section. Variable-size convolution refers to extracting features in the same layer by using convolution kernels of different sizes on the basis of the preservation of traditional multicore attributes. However, whether it is multichannel or variable convolution, MVCNN extracts features only in the sentence direction dimension while ignoring the features in the word embedding dimension. The main contribution of VCPCNN proposed in this paper is that the factor of the word embedding dimension is considered, which is the essential difference from MVCNN. The multiconvolution and multipooling operations performed in this paper are not only in the sentence length, but also applied more in the word embedding dimension to extract richer semantic feature information to compensate for the lack of semantic information in the word embedding dimension. In addition, the average pooling operation proposed in the pooling layer is beneficial for preserving the important feature information of the extracted features.

Based on the research of the above-related works, we know that the study of sentiment analysis uses a variety of methods. However, the methods mentioned above are mainly used to extract the semantic feature information from the sentence dimension and ignore the information feature of the word vector dimension. The VCPCNN is also an improved

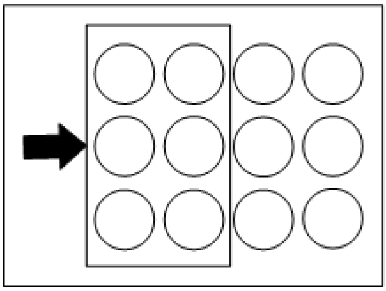


FIGURE 3. TextCNN convolution in the direction of sentence length.

multiconvolution and pooling structure proposed on the basis of TextCNN. This paper introduces four convolutions to mine the word vector's local features of the semantic dimension in the word insertion direction or dimension. Additionally, on the basis of max-pooling, this paper adds average pooling to fetch more detailed local feature information. Section III introduces the specific structure.

III. IMPLEMENTATION DETAILS

We first introduce the basis of VCPCNN improvement (TextCNN model) and then focus on the design and specific application of the four network structures corresponding to VCPCNN.

A. TextCNN MODEL

TextCNN proposed by Kim [1] is a three-layer convolution neural network based on multiple convolution kernels. Figure 1 shows the network structure of TextCNN.

TextCNN has only one convolution layer and only one-dimensional convolution in the direction of the sentence length. The width of the convolution kernel is the dimension of the word vector. As shown in Figure 3, assuming that the length of an input sentence is 4, the word embedding dimension is 3, and the convolution kernel is 2. We use different convolution kernel widths to extract richer features.

However, the word vector is a semantic vector that has different semantic information in all dimensions. Therefore, a convolving operation can be introduced in a single-word embedding dimension or an entire-word embedding direction and can introduce a single semantic dimension of a word vector or a local feature of an adjacent semantic dimension. In this way, the model can extract not only the local features in the direction of the sentence length but also the local structural features of the semantic dimension of the word vector.

In addition, TextCNN only uses max-pooling in the pooling layer, considering only the largest eigenvalues of each feature map without regarding other factors. Therefore, some important information may be lost.

B. VCPCNN MODEL

Considering the above two prominent problems of TextCNN, this paper improves the network structure of TextCNN and proposes VCPCNN to make up for these shortcomings.

TABLE 1. The amount of parameters corresponding to the four improved structures.

	VCPCNN-1D	VCPCNN-2D
SAME	$n \times d \times 1$	$d \times 1 \times 1$
DIFF	$n \times 1 \times k$	$d \times 1 \times k$

k represents the amount of convolution kernels.

The experimental results in Section IV also show that VCPCNN achieves good performance in this respect.

In addition to one-dimensional convolutions in the direction of the sentence length, VCPCNN also adopts convolution kernels of two sizes for convolution in the word embedding dimension. Assuming that the input sentence's length is n, and the dimension of word embedding is k, then the first convolution kernel size is $n \times d$, and d is the width of the convolution kernel in the word embedding. This structure is defined as VCPCNN-1D. The second convolution kernel size is $d \times 1$, which is defined as VCPCNN-2D, and d is the width of the convolution kernel in the length of the sentences.

There are two convolution cases for the word embedding dimensions. In the first case, assuming there is no relationship between each word embedding dimension, the convolution operation is performed in each word embedding dimension separately. Thus, convolution kernels are different in different word embedding dimensions. The second case assumes that there is a relationship between adjacent word embedding dimensions; that is, ordinary convolutions are performed in the direction of word embedding, and the convolution kernels are the same in all positions in the direction of word embedding. This paper defines the first convolution to "DIFF convolution" and the second is "SAME convolution".

Therefore, the structures of VCPCNN-1D and VCPCNN-2D also correspond to the convolution kernel DIFF and SAME, as shown in Figure 4. However, the width of the convolution kernel in the word embedding dimension direction can be 1 in the DIFF type; thus, the convolution kernel size of the VCPCNN-1D_DIFF network structure must be $n \times 1$. Additionally, VCPCNN-1D_SAME's convolution kernel size is $n \times d$. The convolution kernel sizes of VCPCNN-2D_SAME and VCPCNN-2D_DIFF are both $d \times 1$. Table 1 shows the parameters of the four improvement network structures in the word embedding dimension.

It can be seen from Table 1 that the parameters of the DIFF structure greatly increase relative to the SAME, and the multiple is the word vector dimension. Moreover, the number of parameters of the VCPCNN-2D_SAME Model is much lower than of the other three models.

Since the structure of VCPCNN is similar to TextCNN, both of them are only different in the convolution layer and the pooling layer. Therefore, this paper only introduces the convolutional and pooling layers of the four improved network structures.

Assume that the input matrix of the convolution layer is $x \in R^{n \times k}$, where n represents the length of the input sentences, and k represents the dimensions of the word vector.

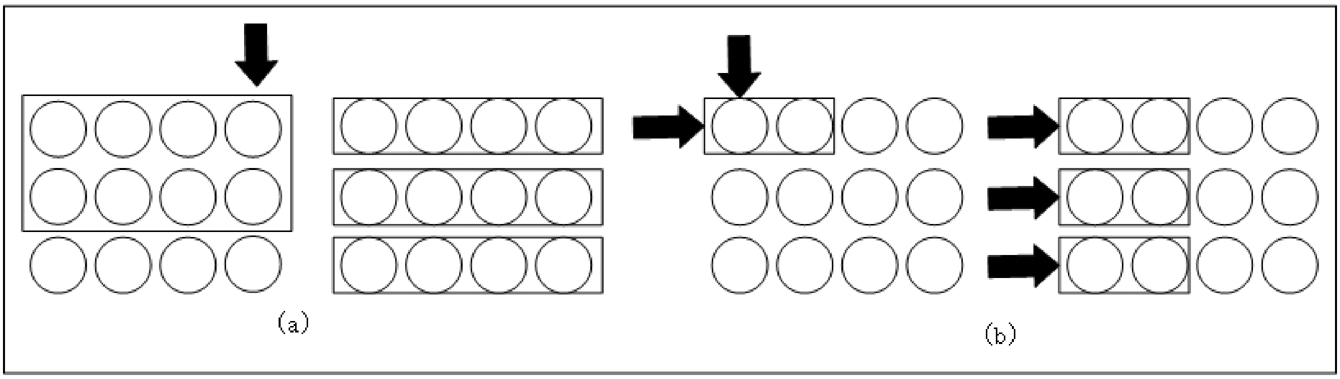


FIGURE 4. The horizontal direction represents the length of sentence assumed to be 4, the vertical direction is the word embedding dimension assumed to be 3, and the convolution kernel width is unified to 2. (a) the convolution of VCPCNN-1D_SAME and VCPCNN-1D_DIFF in word embedding dimension; (b) the convolution of VCPCNN-2D_SAME and VCPCNN-2D_DIFF in word embedding dimension.

In the direction of sentence length, x_i represents the word vector of the i-th word, $x_{i:j}$ represents the link of the word vector from the i-th word (involved) to the j-th word. The input matrix x can be expressed as the link of n word vectors in the k dimension. Refer to (1).

$$x_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n \tag{1}$$

In the direction of the word embedding dimension, x^i represents the i-th word's sentence vector in the word vector dimension, $x^{i:j}$ represents the sentence vector link in the word dimension from the i-th to the j-th word. The input matrix x can be expressed as the link of k sentence vectors in the n-th dimension. Refer to (2).

$$x^{1:k} = x^1 \oplus x^2 \oplus \ldots \oplus x^k \tag{2}$$

In formulas (1) and (2), $\oplus$ represents the vector link.

The convolution layer of TextCNN produces a feature c^i_{seq} to the i-th word sequence $x_{i:i+d-1}$, convolution kernel $w_{seq} \in R^{d\times k}$ with d width. Refer to (3).

$$c^i_{seq} = f(w_{seq} \cdot x_{i:i+d-1} + b_{seq}) \tag{3}$$

Among them, $b_{seq} \in R$ is a bias, and f is a nonlinear function. We apply the convolution kernel w_{seq} to all possible windows $\{x_{1:d}, x_{2:d+1}, \ldots, x_{n-d+1:n}\}$ of the sentence to generate the corresponding feature map. Refer to (4).

$$c_{seq} = [c^1_{seq}, c^2_{seq}, \ldots, c^{n-d+1}_{seq}] \tag{4}$$

In $c_{seq} \in R^{n-d+1}$, c_{seq} represents the convolution results in the direction of the sentence length.

C. VCPCNN-1D STRUCTURE

In addition to the one-dimensional convolution in the direction of the sentence length (refer to (3)), VCPCNN-1D also adds the one-dimensional convolution in the direction of the word embedding dimension. The main difference between the SAME and DIFF structures is the use of different convolution methods and different numbers of convolution kernels in the length of the sentence, which results in different feature combinations in the direction of the word vector dimension. The convolution configurations of the two structures of SAME and DIFF are shown in Table 1. The number of convolution kernels of the former remains unchanged at 1, and the size of the convolution is n x d. The convolution of the latter remains at the word embedding dimension. The width of 1 is convolved, and K features of the same size are used for feature extraction. The specific prominent differences are shown in the positions of the red marks in Figure 5 and Figure 6.

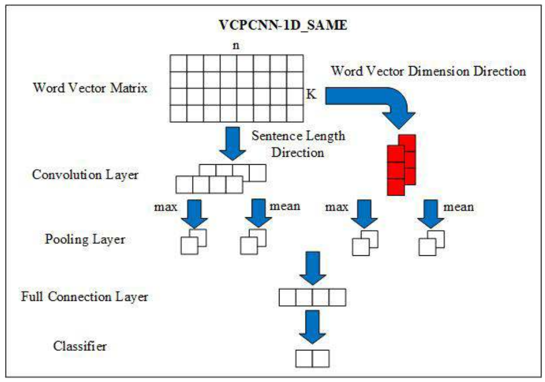


FIGURE 5. Shows the specific structure of VCPCNN-1D_SAME.

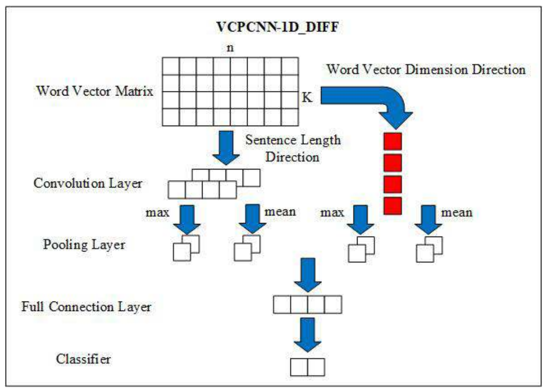


FIGURE 6. Shows the specific structure of VCPCNN-1D_DIFF.

1) VCPCNN-1D_SAME

Assuming a convolution kernel $w_{emd} \in R^{n\times d}$, d represents the width of the convolution kernel in the direction of the word embedding, and n represents the length of input sentences. All the positions of the convolution kernel w_{emd} in the direction of the word embedding are the same. We convolved the word embedding sequence $x^{i:\,i+d-1}$ of the input matrix by using the convolution kernel w_{emd} and obtained a feature map of c^i_{emd}. Refer to (5).

$$c^i_{emd} = f(w_{emd} \cdot x^{i:\,i+d-1} + b_{emd}) \tag{5}$$

Among them, $b_{emd} \in R$ is a bias, and f is a nonlinear function. We apply the convolution kernel w_{emd} to all possible windows $\{x^{1:\,d}, x^{2:\,d+1}, \ldots, x^{k-d+1:\,k}\}$ of the sentence to generate corresponding feature map. Refer to (6). $c_{emd} \in R^{k-d+1}$

$$c_{emd} = [c^1_{emd}, c^2_{emd}, \ldots, c^{k-d+1}_{emd}] \tag{6}$$

Figure 5 shows the specific structure of VCPCNN-1D_SAME.

2) VCPCNN-1D_DIFF

Because it is a DIFF type, every word embedding dimension corresponds to its own convolution kernel. There are k convolution kernels, and the size of each kernel is $n \times 1$. Assuming the convolution kernel in the i-th word embedding dimension is $w^i_{emd} \in R^{n\times 1}$, n represents the length of the input sentence. Having convolved the sentence vector $x^i \in R^{n\times 1}$ in the i-th word embedding dimension of the input matrix by using convolution kernel w^i_{emd}, we obtain a feature c^i_{emd}. Refer to (7).

$$c^i_{emd} = f(w^i_{emd} \cdot x^i + b^i_{emd}) \tag{7}$$

Among them, $b^i_{emd} \in R$ is a bias, and f is a nonlinear function. We apply the convolution kernel w^i_{emd} to the corresponding i-th word embedding dimension and generate the final feature map. Refer to (8). $c_{emd} \in R^k$.

$$c_{emd} = [c^1_{emd}, c^2_{emd}, \ldots, c^k_{emd}] \tag{8}$$

Figure 6 shows the specific structure of VCPCNN-1D_DIFF.

For VCPCNN-1D, the convolution in the word embedding direction is either a SAME type or a DIFF type, and the output feature maps are all one-dimensional vectors with different lengths.

In the pooling layer, two kinds of pooling (max-pooling and average pooling) are performed for the convolution output vector $c_{seq} \in R^{n-d+1}$ in the length direction of the sentence and the convolution output vector c_{emd} in the direction of the word embedding dimension, and then we obtain the following four eigenvalues.

$$\hat{c_{seq}} = max(c_{seq}) \tag{9}$$
$$\bar{c_{seq}} = mean(c_{seq}) \tag{10}$$
$$\hat{c_{emd}} = max(c_{emd}) \tag{11}$$
$$\bar{c_{emd}} = mean(c_{emd}) \tag{12}$$

Regarding the four output features above, this paper combined the results of max-pooling and average pooling of the two convolutions and obtained two combinations: $(\hat{c_{seq}}, \bar{c_{seq}})$ and $(\hat{c_{emd}}, \bar{c_{emd}})$. One combination (c_1, c_2) has three different combinations.

1) Add by bit, $+$ represents add by bit.

$$c_{add} = c_1 + c_2 \tag{13}$$

2) Add by bit, $-$ represents add by bit.

$$c_{sub} = c_1 - c_2 \tag{14}$$

3) Take absolute value after minus by bit.

$$c_{Abs\_Sub} = |c_1 - c_2| \tag{15}$$

After combining the two combinations, we obtained two eigenvalues and linked them. Refer to (16). $\oplus$ represents the link. v is the final output vector of the pooling layer.

$$v = v_1 \oplus v_2 \tag{16}$$

D. VCPCNN-2D STRUCTURE

In addition to the one-dimensional convolution of formula (3), VCPCNN-2D adds a convolution operation with a convolution kernel size of $d \times 1$ in the convolutional layer.

1) VCPCNN-2D_SAME

Assume that a convolution kernel with width d is $w \in R^{d\times 1}$. Since the type is SAME, the convolution kernel is the same in all positions of the input matrix, which is equivalent to a normal two-dimensional convolution. The convolution kernel w is convolved for the j-th sequence $x^i_{j:\,j+d-1}$ of the i-th word embedding dimension of the input matrix to yield a feature c^i_j, referring to (17).

$$c^i_j = f(w \cdot x^i_{j:\,j+d-1} + b) \tag{17}$$

Among them, $b \in R$ is the bias, and the convolution kernel w is applied to all possible windows $c^{i,d} = \{x^i_{1:\,d}, x^i_{2:\,d+1}, \ldots, x^i_{n-d+1:\,n}\}$ in the i-th word vector dimension of the input matrix to produce the corresponding feature map. See formula (18).

$$c^i = [c^i_1, c^i_2, \ldots, c^i_{n-d+1}] \tag{18}$$

Among them, x^j represents the j-th word sentence vector in the word vector dimension, $x^j_{p:\,q}$ represents the vector of the p-th to q-th words in the j-th word vector dimension of the sentence, and $c^j \in R^{n-d+1}$ is the convolution output vector in the j-th word vector dimension. The output vector on each word vector dimension is connected to obtain the final output feature map $c_{emd} \in R^{(n-d+1)\times k}$; see formula (19).

$$c_{emd} = [c^1, c^2, \ldots, c^k] \tag{19}$$

Figure 8 shows the specific structure of VCPCNN-2D_SAME.

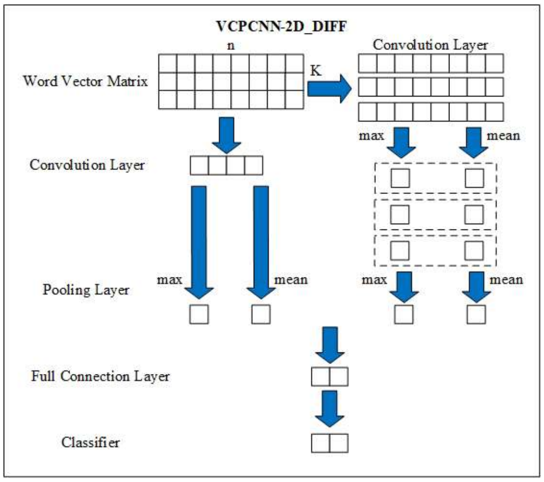


FIGURE 7. Shows the specific structure of VCPCNN-2D_DIFF.

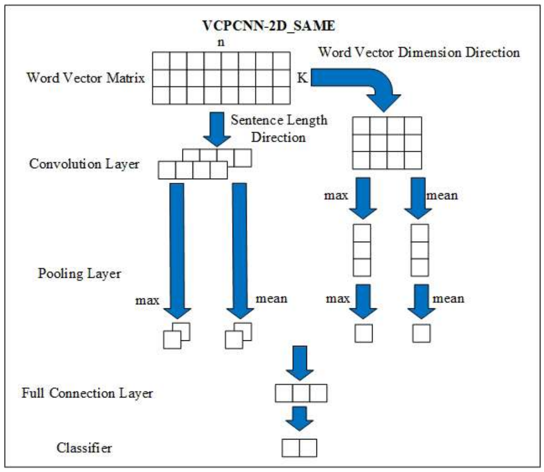


FIGURE 8. Shows the specific structure of VCPCNN-2D_SAME.

2) VCPCNN-2D_DIFF

Assume that a convolution width is $w_j \in R^{d\times 1}$ and represents a one-dimensional convolution kernel applied to the j-th dimension of the input matrix. The word sequence $x^j_{i:\,i+d-1}$ of the input matrix is convolved using the convolution kernel w_j to yield a feature c^j_i. See formula (20).

$$c^j_i = f(w_j \cdot x^j_{i:\,i+d-1} + b_j) \tag{20}$$

Among them, $b_j \in R$ is the bias. The convolution kernel w_j on the j-th word vector dimension is applied to all possible windows $\{x^j_{1:\,d}, x^j_{2:\,d+1}, \ldots, x^j_{n:\,d+n-1}\}$ in the j-th word vector dimension of the sentence to produce the corresponding feature map. See formula (21).

$$c^j = [c^j_1, c^j_2, \ldots, c^j_n] \tag{21}$$

Among them, x^j represents the j-th word sentence vector in the word vector dimension, $x^j_{p:\,q}$ represents the vector of the p-th to q-th words in the j-th word vector dimension of the sentence, and $c^j \in R^n$ is the convolution output vector in the j-th word vector dimension. The output vector on each word vector dimension is connected to obtain the final output feature map $c_{emd} \in R^{n\times k}$; see formula (22).

$$c_{emd} = [c^1, c^2, \ldots, c^k] \tag{22}$$

Figure 7 shows the specific structure of VCPCNN-2D_DIFF.

In particular, VCPCNN-2D_DIFF uses a wide convolution, and the other three network structures take a narrow convolution.

Whether the VCPCNN-2D's type is SAME or DIFF, its convolution result is a two-dimensional feature map c_{emd}.

First, pool each word embedding dimension. It is assumed that the convolution output vector of the i-th word embedding dimension is c^i_{emd}, and then maximum and average pooling are performed separately, and two pooling results of $\hat{p}^i_{emd} = max(c^i_{emd})$ and $\bar{p}^i_{emd} = mean(c^i_{emd})$ are obtained ($\hat{p}^i_{emd}, \bar{p}^i_{emd} \in R$). Finally, using $\widetilde{p}^i_{emd} = \hat{p}^i_{emd} \circ \bar{p}^i_{emd}$ to combine (the symbol "$\circ$" represents any combination, such as bitwise addition, bitwise subtraction, and absolute value of bitwise subtraction).

After connecting the two pooled results of each word embedding dimension, the output feature map is $\widetilde{p}_{emd} \in R^k$, and then it is finally pooled to obtain $\hat{\widetilde{p}}_{emd} = max(\widetilde{p}_{emd})$ and $\bar{\widetilde{p}}_{emd} = mean(\widetilde{p}_{emd})$. The max-pooling and average pooling are performed only once for $c_{seq} \in R^{n-h+1}$, and then we obtain $\hat{c}_{seq}$ and $\bar{c}_{seq}$. The operations of formulas (13) to (16) are also performed for these four output features, and finally, the output vector is obtained.

The above describes the four structural analyses corresponding to VCPCNN. We apply them to the four emotional databases (English emotional polarity, Chinese emotional polarity, Chinese subjective and objective emotions, and Chinese multicategory emotions) corresponding to this paper for experimentation and compare them with existing advanced text sentiment classification models to select the best performance on the corresponding datasets. The specific experimental analysis and experimental conclusions are shown in Section IV.

IV. EXPERIMENTAL ANALYSIS

A. DATASETS

All of the sentiment datasets that we used are based on sentences. The English dataset used is the emotional identification sentence dataset of UCI [2], and the Chinese datasets belong to two evaluation task datasets from the 2014 Natural Language Processing and Chinese Computing Conference: Chinese microblog text sentiment analysis and emotion classification based on deep learning. The specific information of each dataset is shown in Table 2.

UCI's Sentiment Labeled Sentence dataset includes comments from three sites, imdb.com, amazon.com, and yelp.com, each with 1,000 sentences. These comments involve two kinds of sentiment polarities: positive and negative. The number of positive and negative samples is 1,500 for each dataset.

TABLE 2. The specific information of four sentiment datasets.

Datasets	English Polarity Emotion	Chinese Polarity Emotion	Chinese Subjective Objective	Chinese Multi-Category Emotional
Number of Categories	2	2	2	7
Name of Categories	Positive, Negative	Positive, Negative	Subjective, Objective	Like, Happy, Sad, Disgusted, Angry, Fear, Surprised
Number of Samples	1500:1500	5000:5000	29731: 15690	1899 : 3130 : 299 : 2805 : 4259 : 2478 : 820
Total Data Volume	3000	10000	45421	15690
Data Sources	Website	Microblog	Microblog	Microblog

The first dataset in the Chinese dataset is the training set of the NLPCC2014 evaluation task one, which contains a total of 14,000 microblogs. Its text format is XML. Each microblog has several sentences, for a total of 45,421 sentences. Each sentence has its corresponding subjective and objective category. The objective sentences have no corresponding emotional category. The subjective sentences have two emotional category identifiers: the main emotional category of the sentence, and the secondary emotional category. The emotional categories of the dataset are divided into seven categories: happiness, sadness, disgust, anger, fear, and surprise. The main sentiment category is used in this paper, and the other is discarded. According to whether the sentence is a subjective sentence, all sentences can be compiled into a subjective sentiment dataset and an objective sentiment dataset. Then, all subjective sentences are extracted and classified into a multicategory emotional dataset based on the main emotional dataset. The second dataset is a dataset of the NLPCC2014 evaluation task 2, in the format of XML, including positive and negative sentiment polarities. There are 10,000 comment sentences, 5,000 for each sentiment polarity. The preprocessing process for the datasets is as follows:

1) Remove noise: There are some noise in the text that are not related to the sentiment classification, such as ''@ Mr. Wang'', website links, etc.
2) Participle: This is a unique process for Chinese text. The word segmentation tool is jieba participle.

In addition, each input sentence needs to be aligned with the same length. If the length of the sentence is short, the missing part of the sentence is filled with <PAD>. If the length of the sentence is too long, the sentence is truncated.

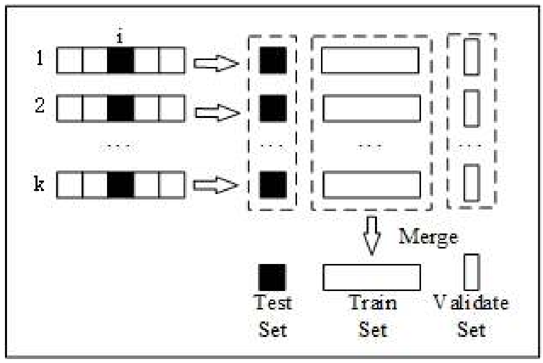


FIGURE 9. The specific process of class *i*.

We use a five-fold cross-validation method to reduce the impact of randomness on the experimental results and divided the text into five equal parts. One of them is taken as a test set, and the remaining parts were divided into two sets: a training set and a verification set. We adopt a ratio of 7: 1: 2 to divide the training set, the verification set and the test set. For example, the specific process of the class is shown in Figure 9.

B. PARAMETER CONFIGURATION

We adopt four sentiment datasets, shown in Table 2, as the test data, using MVCNN [45], RCNN [46], DCNN [47], bidirectional LSTM (BI-LSTM) and TextCNN as comparison models. All of these models are implemented by TensorFlow. However, the DCNN implemented in our paper is different from the original paper [47] in that the input sentence length needs to be fixed. The MVCNN uses single-channel input, but its structure is the same as that of the original paper. Table 3 shows the specific parameters of the two models. The four structures of VCPCNN proposed in this paper are improved on the basis of TextCNN, with only one convolution layer; the specific configuration of their model parameters is shown in Table 4.

The word vector dimensions of this experiment are all fixed at 50; the English dataset uses the 50-dimensional Senna word vector, and the Chinese dataset uses the 50-dimensional word vector trained by the word2vec tool for the Chinese Wikipedia corpus. We use cross entropy as the loss function, Adam as the optimizer, 0.001 as the learning rate, and a small batch gradient descent method with a sample size of 100 per batch. The input sentence lengths are fixed at 200 for all datasets and models.

TABLE 3. The parameter configuration of DCNN and MVCNN.

Model	Conv1			k-max1	Conv2			k-max2
	Type	Size	Number	Parameter k	Type	Size	Number	Parameter k
DCNN	1D Wide Convolution	5×1	10	100	1D Wide Concolution	3×1	30	3
MVCNN								

Note: "Conv1" and "Conv2" represent the first and second convolutional layers of the model; "k-max1" and "k-max2" represent the first k-max pooling layer and the second k -max pooling layer respectively; "1D" means one-dimensional convolution and "2D" means two-dimensional convolution.

TABLE 4. The parameter configuration of TextCNN and VCPCNN.

Model	Conv1		
	Type	Size	Number
TextCNN	1D Narrrow Convolution	$5 \times d$	30
		$3 \times d$	30
VCPCNN-1D_SAME	1D Narrrow Convolution	$n \times 5$	30
		$n \times 3$	30
VCPCNN-1D_DIFF	1D Narrrow Convolution	$n \times 1$	$30 \times d$
VCPCNN-2D_SAME	2D Narrrow Convolution	3×1	30
		5×1	30
VCPCNN-2D_DIFF	1D Wide Convolution	3×1	$30 \times d$
		5×1	$30 \times d$

n represents the length of the input sentence, d represents the dimensions of word vector.

C. EXPERIMENTAL TRAINING PROCESS

The accuracy rate is considered as the evaluation index of this experiment. Assuming that the sample size is m, the i-th sample $x_i$true category is y_i, and the category of the model prediction is $pred_i$. $I(x)$ is expressed as an indicator function, and is 1 if the condition is satisfied; otherwise, 0. The accuracy is calculated as equation (23).

$$precision = \frac{1}{m}\sum_{i=1}^{m} I(y_i == pred_i) \tag{23}$$

The minibatch gradient descent method is adopted for the training process in this paper. First, the iteration number n, and the batch size m are determined. In each iteration, a batch of training data is randomly obtained, and the loss is calculated for the forward direction. Assuming that the loss of the sample x_i is $loss_i$, then the loss of the entire batch is $loss = \sum_{i=1}^{m} loss_i$. The gradient of all parameters is then inversely calculated and updated until the training set is traversed. Additionally, the verification iteration interval EVAL_NUM used in this experiment is 1; that is, in each iteration, the verification is performed once the training is completed. The number of iterations WAIT_EPOCH for the early stop is set to 6; that is, if the minimum verification loss has not been updated for 6 iterations, the training is stopped in advance.

D. RESULT ANALYSIS

Our experiments are carried out on four different datasets to verify the four structures corresponding to the VCPCNN proposed in this paper. They are compared with other existing advanced technologies to select the optimal network structure on different datasets. Therefore, it better serves the needs of different types of sentiment classification.

1) ACCURACY ANALYSIS

Each network structure of VCPCNN has three pooling methods from formula (13) to (16) for the two pooling results of the same convolution. Therefore, the results of VCPCNN in Table 5, Table 6, Table 7, and Table 8 are the most accurate values of the three combinations. All the testing results are the average value of the five testing sets' precision rate in five-fold cross validation. Note that the columns with "+" or "−" in the following tables indicate the percentage changes relative to the TextCNN model in the precision rate of the four structures corresponding to VCPCNN.

It can be seen from Table 5 and Table 8 that the classification effect of TextCNN on the English polarity emotion dataset and Chinese multicategory dataset is better than that of other existing models. We also found that the worst

TABLE 5. The classification precision of all models in english polarity emotion dataset.

Model	English Polarity Emotion	
TextCNN	0.7690	
DCNN	0.6770	
MVCNN	0.7317	
RCNN	0.7623	
Bi-LSTM	0.7283	
VCPCNN-1D_SAME	0.7796	+1.06%
VCPCNN-1D_DIFF	0.7637	−0.53%
VCPCNN-2D_SAME	0.7813	+1.23%
VCPCNN-2D_DIFF	**0.7887**	**+1.97%**

TABLE 6. The classification precision of all models in chinese polarity emotion dataset.

Model	Chinese Polarity Emotion	
TextCNN	0.7354	
DCNN	0.7076	
MVCNN	0.7066	
RCNN	0.7351	
Bi-LSTM	0.7317	
VCPCNN-1D_SAME	0.7479	+1.25%
VCPCNN-1D_DIFF	**0.7483**	**+1.29%**
VCPCNN-2D_SAME	0.7395	+0.41%
VCPCNN-2D_DIFF	0.7408	+0.54%

TABLE 7. The classification precision of all models in chinese subjective objective dataset.

Model	Chinese Subjective Objective	
TextCNN	0.7518	
DCNN	0.7325	
MVCNN	0.7173	
RCNN	0.7507	
Bi-LSTM	0.7563	
VCPCNN-1D_SAME	**0.7621**	**+1.03%**
VCPCNN-1D_DIFF	0.7586	+0.68%
VCPCNN-2D_SAME	0.7609	+0.91%
VCPCNN-2D_DIFF	0.7610	+0.92%

TABLE 8. The classification precision of all models in chinese multi-category dataset.

Model	Chinese Multi-Category	
TextCNN	0.4790	
DCNN	0.4005	
MVCNN	0.3499	
RCNN	0.4588	
Bi-LSTM	0.4679	
VCPCNN-1D_SAME	0.4868	+0.78%
VCPCNN-1D_DIFF	0.4785	-0.05%
VCPCNN-2D_SAME	0.4904	+1.14%
VCPCNN-2D_DIFF	**0.4959**	**+1.69%**

performances of VCPCNN-1D_DIFF in the four network structures we proposed are -0.53% and -0.05%, respectively, and the other three structures are improved. Among them, VCPCNN-2D_DIFF has the largest improvement, which is over 1.9%. Therefore, we choose VCPCNN-2D_DIFF as the optimal model for text sentiment analysis on the English polarity emotion dataset and Chinese multicategory dataset.

Table 6 shows the precision rate of classification of various models on the Chinese polarity emotion dataset. It is found that the four improved structures we proposed are better than the classification results of TextCNN, and VCPCNN-1D_DIFF, which increased by 1.29%, performs best. Therefore, we have reason to choose this model as the best model for sentiment analysis on the Chinese polarity emotion dataset.

In Table 7, we find that the four structures corresponding to VCPCNN are better than that of the TextCNN model in the Chinese subjective objective dataset. The performance of VCPCNN_1D_SAME improved 1.03%, so the model structure is taken as its optimal structure on the dataset.

In conclusion, VCPCNN is basically better than TextCNN, except for the structure of VCPCNN-1D_DIFF, which has a worse effect than TextCNN on the English emotional polarity and Chinese multicategory datasets. Among the four network structures proposed in this paper, VCPCNN-1D_DIFF has the worst classification effect. However, VCPCNN-2D_DIFF has the best classification effect in the English emotional polarity and Chinese multicategory emotional datasets. Its accuracy is also only second to the highest VCPCNN-1D_SAME in the Chinese subjective and objective emotional dataset. VCPCNN-1D_SAME and VCPCNN-2D_SAME perform second, and both perform equally well. The accuracy of VCPCNN-1D_SAME is higher than that of VCPCNN-2D_SAME in Chinese sentiment polarity and Chinese subjectivity and objectivity. However, the accuracy of VCPCNN-2D_SAME is higher on the other two datasets.

In our analysis, perhaps because each dimension in the vector space of the word vector is independent, one-dimensional convolution is performed on each word embedding dimension. It is the essence of the semantic vector space that fits the word vector. Although VCPCNN-1D_DIFF is the same as VCPCNN-2D_DIFF, the convolution kernel is different in each word embedding dimension. The convolution kernel width of VCPCNN-1D_DIFF in each word embedding dimension is the entire sentence length so that local features in the direction of the sentence length cannot be extracted for each word embedding dimension. While VCPCNN-2D_SAME can extract these features, its convolution kernels are shared in each word embedding dimension, which cannot meet the requirements of the spatial dimension of the word vector.

The experimental results above show that the VCPCNN proposed in this paper is better than the multichannel input and variable convolution model (MVCNN) for the semantic information of the word embedding dimension considered to extract more features. Additionally, the overall accuracy of the two types of DIFF in the four structures proposed in this paper is better than the two types of SAME for the most part. This is mainly because the former convolution kernels are more fine-grained (that is, smaller convolution kernels are adopted), which is convenient for extracting more subtle local features, and the number of convolution kernels is slightly more than the latter. Table 4 for the parameter configuration is shown.

TABLE 9. Classification accuracy of initialization models with different word vector dimensions.

Word Vectors	Dimensions	Accuracy
Yelp_All	50	**0.8360**
Yelp_Rest	50	0.8267
Yelp_ALl	100	**0.8360**
Yelp_Rest	100	0.8293

Note: Yelp_All means all comments from Yelp, and Yelp_Rest refers to Yelp's comments except the comments with a score of 3 (score 3 for neutral scores)

2) TIME PERFORMANCE ANALYSIS

Furthermore, the time performance is tested on the four model structures and the TextCNN model proposed in this paper with the same four datasets (only counting the time of training 25 epoch+verifying 1 epoch in five-fold cross validation). The statistical results are shown in Table 9. The time unit in the table is minutes (min). Figure 10 shows that the four structures of the VCPCNN model are better than our improved object, the TextCNN model, in terms of time performance. Table 4 shows the reason that the model has only one convolution layer, so the size of the model parameters is mainly reflected in the size and number of convolution kernels. However, VCPCNN uses multiple and small convolution kernels, so to a certain extent, the model parameters are lower. The average pooling operation is also introduced in the pooling layer. The pooling by "VALID" further reduces the parameters of the full connection.

3) OTHERS

In addition to some factors of the model structure, there are some other external factors that influenced the experimental results; for example, the length of the data model sentence and the dimension size of the pretrained words embedded. Therefore, the next experiment will be carried out from these aspects, and the corresponding results will be analyzed.

The default length of the sentence in this paper is 200. To verify the effect of different lengths on the performance of the model, the four structural models proposed in this paper are tested on the four datasets in the range of [50,300] with a step size of 50. The results of the experiment are shown in Figures 11, 12, 13, and 14.

The experimental data of the four graphs above reflect that the accuracy of the model is generally a trend of rising first and then decreasing, and the best accuracy of each model is basically stabilized at the interval [150,200]. It can be concluded that when the length of the sentence is short, the model

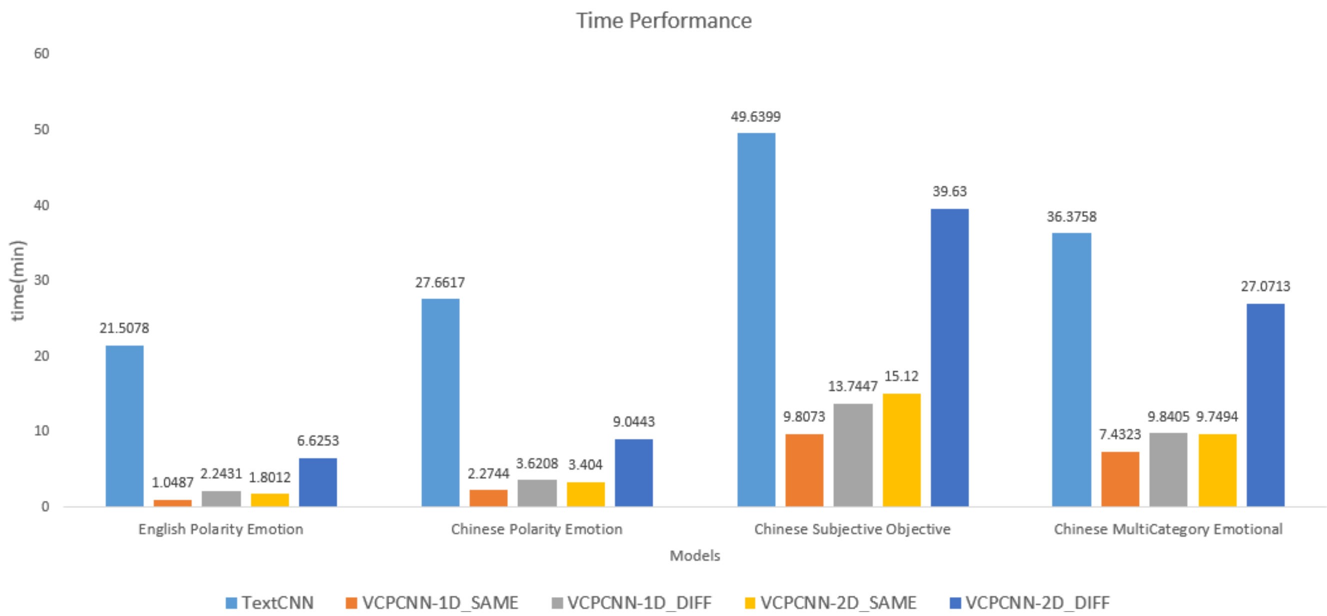


FIGURE 10. Time performance comparison.

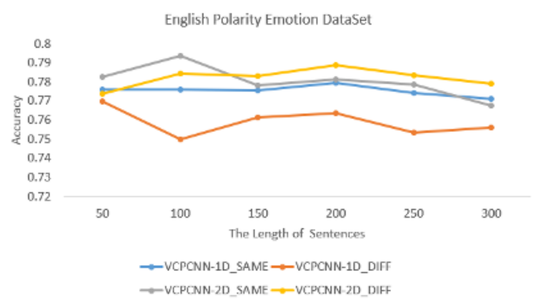


FIGURE 11. The effects of sentence lengths on english sentiment polarity dataset.

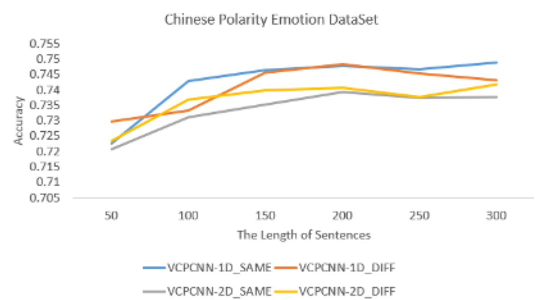


FIGURE 12. The effects of sentence lengths on chinese sentiment polarity dataset.

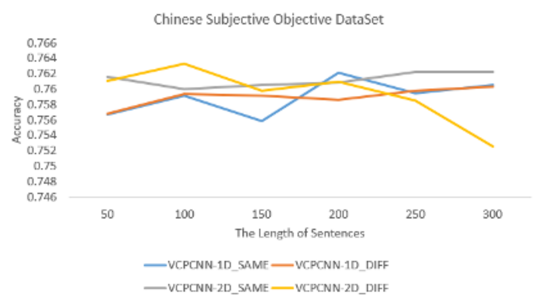


FIGURE 13. The effects of sentence lengths on subjective and objective sentiment dataset.

can extract relatively little feature information, which cannot reflect the sentiment information well. When the sentence length is long, it should theoretically extract more abundant semantic information. However, only a very small number of models in the experiment showed a certain upward trend. The possible reason for the analysis is that when the sentence length is long, it does not truly represent the actual situation in actual language expression, because people tend to express themselves through medium and short sentences, so when the sentence length is too long, the information that should be expressed may be blurred instead. Thus, the accuracy rate declines as the length of the sentence increases. Additionally, as the length of the sentence increases, the time cost increases accordingly. Therefore, it is reasonable to set the length of the sentence to 200.

The experiments in this paper are carried out to analyze the possible effects of pretrained word vector dimensions on sentiment classification. The pretrained word vectors mainly use the large-scale users' comments dataset published by Yelp (mainly for the word vector of emotional information), and the word vector dimension is set to 50 and 100 to initialize the model. The experimental results are shown in Table 9. The results in the table are represented in average accuracy.

In theory, the higher the dimension of the word vector, the richer the semantic information included, and the better the effect should be. However, the results from Table 9 show

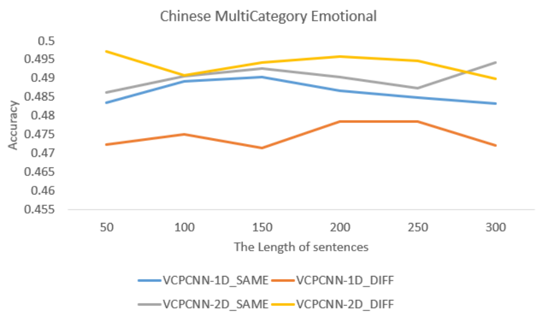


FIGURE 14. The effects of sentence lengths on multi-categories sentiment dataset.

that the size of the word vector dimension seems to have a small effect on the classification accuracy. The accuracy of the Yelp_All word vector remains the same regardless of whether the word vector dimension is 50 or 100. Yelp_Rest is approximately 0.3% higher in the 100 dimension than in the 50 dimension. The influence of the dataset and the actual context may take responsibility for it. The increase in dimension cannot improve the accuracy at the cost of pretraining time. Therefore, 50 is chosen as the default dimension size in this paper.

V. CONCLUSION

A multiconvolution and pooling method for text sentiment classification is proposed in this paper, which is based on the TextCNN network structure. For better text feature extraction, the model introduces four different convolution operations in the word embedding dimension and adds average pooling in the pooling layer to extract more detailed local features. The experimental results show that compared with methods such as DCNN, MVCNN, RCNN and Bi-LSTM, our method is greatly improved, especially in the Chinese multicategory dataset. The VCPCNN-2D_DIFF structure is **14.60%** higher than the MVCNN method, although the VCPCNN-2D_SAME structure decreases less than 1% in the English emotional polarity dataset compared with RCNN. The effects of our four structures proposed in this paper on Chinese sentiment datasets are generally better than English emotion. Therefore, compared with TextCNN, VCPCNN is more suitable for Chinese sentiment classification.

REFERENCES

[1] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*. [Online]. Available: https://arxiv.org/abs/1408.5882

[2] (2015). *The UCI Sentiment Labelled Sentences Dataset*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences

[3] (2013). *Stanford Sentiment Treebank Dataset*. [Online]. Available: http://nlp.stanford.edu/~socherr/stanfordSentimentTreebank.zip

[4] (2015). *Kaggle Sentiment Analysis on Movies Reviews*. [Online]. Available: https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews/data

[5] (2018). *Yelp Dataset*. [Online]. Available: https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews/data.

[6] J. S. Y. Liew, H. R. Turtle, and E. D. Liddy, "EmoTweet-28: A fine-grained emotion corpus for sentiment analysis," in *Proc. 10th Int. Conf. Lang. Resour. Eval. (LREC)*, 2016, pp. 1149–1156.

[7] X. Wu, H. Tong, and M. Klyen, "Fine-grained sentiment analysis with 32 dimensions," in *Proc. Int. Conf. Asian Lang. Process. (IALP)*, Dec. 2017, pp. 387–390.

[8] C. Fellbaum, "Wordnet: An electronic lexical resource," in *Oxford Handbook of Cognitive Science*, 2017, pp. 301–314. [Online]. Available: https://books.google.com/books?hl=zh-CN&lr=&id=NTsoDwAAQBAJ&oi=fnd&pg=PA301&dq=Wordnet:+An+electronic+lexical+resource&ots=AyhpNo6g3c&sig=lRNhesPQk4pxe1AhGoeJZPdGlMg#v=onepage&q&f=false

[9] U. R. Dhungana and S. Shakya, "PolyWordNet: A lexical database," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, vol. 2, Aug. 2016, pp. 1–7.

[10] S.-M. Wang and L.-W. Ku, "Antusd: A large chinese sentiment dictionary," in *Proc. 10th Int. Conf. Lang. Resour. Eval. (LREC)*, 2016, pp. 2697–2702.

[11] H. Keshavarz and M. S. Abadeh, "ALGA: Adaptive lexicon learning using genetic algorithm for sentiment analysis of microblogs," *Knowl.-Based Syst.*, vol. 122, pp. 1–16, Apr. 2017.

[12] K. Chekima and R. Alfred, "Non-english sentiment dictionary construction," *Adv. Sci. Lett.*, vol. 24, no. 2, pp. 1416–1420, Feb. 2018.

[13] X. Hao, Y. Jia, and Q. Gu, "An automatic construction approach for sentiment dictionary based on weibo emoticons," in *Proc. Int. Conf. Netw., Commun., Comput. Eng. (NCCE)*, May 2018.

[14] T. Chen, R. Xu, Y. He, Y. Xia, and X. Wang, "Learning user and product distributed representations using a sequence model for sentiment analysis," *IEEE Comput. Intell. Mag.*, vol. 11, no. 3, pp. 34–44, Aug. 2016.

[15] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, "Machine learning-based sentiment analysis for Twitter accounts," *Math. Comput. Appl.*, vol. 23, no. 1, p. 11, Feb. 2018.

[16] B. Saha and S. K. Ghosh, "Product based sentiment classification using machine learning on e-commerce," *Int. J. Innov. Knowl. Concepts*, vol. 7, no. 1, pp. 130–138, 2019.

[17] N. Arunachalam, S. J. Sneka, and G. Madhumathi, "A survey on text classification techniques for sentiment polarity detection," in *Proc. Innov. Power Adv. Comput. Technol. (i-PACT)*, Apr. 2017, pp. 1–5.

[18] B. Verma and R. S. Thakur, "Sentiment analysis using lexicon and machine learning-based approaches: A survey," in *Proc. Int. Conf. Recent Adv. Comput. Commun.* Springer, 2018, pp. 441–447. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-10-8198-9_46

[19] J. Song, K. T. Kim, B. J. Lee, S.-Y. Kim, and H. Y. Youn, "A novel classification approach based on Naïve Bayes for Twitter sentiment analysis," *Trans. Internet Inf. Syst.*, vol. 11, no. 6, pp. 2996–3011, 2017.

[20] F. Wang, X. Deng, and L. Hou, "Chinese news text multi classification based on naive bayes algorithm," in *Proc. 2nd Int. Symp. Comput. Sci. Intell. Control*, 2018, p. 31.

[21] Z. Chen, S. Teng, W. Zhang, H. Tang, Z. Zhang, J. He, X. Fang, and L. Fei, "LSTM sentiment polarity analysis based on LDA clustering," in *Proc. CCF Conf. Comput. Supported Cooperat. Work Social Comput.* Springer, 2018, pp. 342–355. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-13-3044-5_25

[22] N. Burns, Y. Bi, H. Wang, and T. Anderson, "Enhanced twofold-LDA model for aspect discovery and sentiment classification," *Int. J. Knowl.-Based Organizations*, vol. 9, no. 4, pp. 1–20, 2019.

[23] J. K. Rout, K.-K.-R. Choo, A. K. Dash, S. Bakshi, S. K. Jena, and K. L. Williams, "A model for sentiment and emotion analysis of unstructured social media text," *Electron. Commerce Res.*, vol. 18, no. 1, pp. 181–199, Mar. 2018.

[24] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul./Aug. 2008.

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

[26] X. Xie, S. Ge, F. Hu, M. Xie, and N. Jiang, "An improved algorithm for sentiment analysis based on maximum entropy," *Soft Comput.*, vol. 23, no. 2, pp. 599–611, Jan. 2019.

[27] S. S. Istia and H. D. Purnomo, "Sentiment analysis of law enforcement performance using support vector machine and K-nearest neighbor," in *Proc. 3rd Int. Conf. Inf. Technol., Inf. Syst. Electr. Eng. (ICITISEE)*, Nov. 2018, pp. 84–89.

[28] Y. Liu, J.-W. Bi, and Z.-P. Fan, "A method for multi-class sentiment classification based on an improved one-vs-one (OVO) strategy and the support vector machine (SVM) algorithm," *Inf. Sci.*, vols. 394–395, pp. 38–52, Jul. 2017.

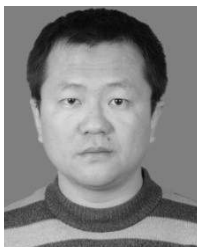
[29] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdiscipl. Rev. Data Mining Knowl. Discovery*, vol. 8, no. 4, p. e1253, 2018.
[30] J. Zhou, J. X. Huang, Q. Chen, Q. V. Hu, T. Wang, and L. He, "Deep learning for aspect-level sentiment classification: Survey, vision and challenges," *IEEE Access*, vol. 7, pp. 78454–78483, 2019.
[31] D. M. El-Din Mohamed Hussein, "A survey on sentiment analysis challenges," *J. King Saud Univ.-Eng. Sci.*, vol. 30, no. 4, pp. 330–338, Oct. 2018.
[32] S. Shayaa, N. I. Jaafar, S. Bahri, A. Sulaiman, P. Seuk Wai, Y. Wai Chung, A. Z. Piprani, and M. A. Al-Garadi, "Sentiment analysis of big data: Methods, applications, and open challenges," *IEEE Access*, vol. 6, pp. 37807–37827, 2018.
[33] J. P. A. Vieira and R. S. Moura, "An analysis of convolutional neural networks for sentence classification," in *Proc. XLIII Latin Amer. Comput. Conf. (CLEI)*, Sep. 2017, pp. 1–5.
[34] X. Fu, W. Liu, Y. Xu, and L. Cui, "Combine HowNet lexicon to train phrase recursive autoencoder for sentence-level sentiment analysis," *Neurocomputing*, vol. 241, pp. 18–27, Jun. 2017.
[35] J. Xu and Y. Cai, "Incorporating context-relevant knowledge into convolutional neural networks for short text classification," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 10067–10068.
[36] J. Yoon and H. Kim, "Multi-channel lexicon integrated CNN-BiLSTM models for sentiment analysis," in *Proc. 29th Conf. Comput. Linguistics Speech Process. (ROCLING)*, 2017, pp. 244–253.
[37] H. Chen, F. X. Han, D. Niu, D. Liu, K. Lai, C. Wu, and Y. Xu, "Mix: Multi-channel information crossing for text matching," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 110–119.
[38] Q.-H. Vo, H.-T. Nguyen, B. Le, and M.-L. Nguyen, "Multi-channel LSTM-CNN model for Vietnamese sentiment analysis," in *Proc. 9th Int. Conf. Knowl. Syst. Eng. (KSE)*, Oct. 2017, pp. 24–29.
[39] X. Cai and Z. Hao, "Multi-view and attention-based BI-LSTM for weibo emotion recognition," in *Proc. Int. Conf. Netw., Commun., Comput. Eng. (NCCE)*, May 2018.
[40] P. Huang, X. Xie, and S. Sun, "Multi-view opinion mining with deep learning," *Neural Process. Lett.*, vol. 50, no. 2, pp. 1451–1463, Oct. 2019.
[41] A. Aziz Sharfuddin, M. Nafis Tihami, and M. Saiful Islam, "A deep recurrent neural network with BiLSTM model for sentiment classification," in *Proc. Int. Conf. Bangla Speech Lang. Process. (ICBSLP)*, Sep. 2018, pp. 1–4.
[42] A. Hassan and A. Mahmood, "Convolutional recurrent deep learning model for sentence classification," *IEEE Access*, vol. 6, pp. 13949–13957, 2018.
[43] Y. Zhang, Z. Zhang, D. Miao, and J. Wang, "Three-way enhanced convolutional neural networks for sentence-level sentiment classification," *Inf. Sci.*, vol. 477, pp. 55–64, Mar. 2019.
[44] X. Lei, Y. Cai, J. Xu, D. Ren, Q. Li, and H.-F. Leung, "Incorporating task-oriented representation in text classification," in *Proc. Int. Conf. Database Syst. Adv. Appl.* Springer, 2019, pp. 401–415. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-18579-4_24
[45] W. Yin and H. Schütze, "Multichannel variable-size convolution for sentence classification," 2016, *arXiv:1603.04513*. [Online]. Available: https://arxiv.org/abs/1603.04513
[46] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015.
[47] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," 2014, *arXiv:1404.2188*. [Online]. Available: https://arxiv.org/abs/1404.2188

MIN DONG received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2005. She has been a Fellow with the School of Computer Science and Engineering, South China University of Technology, since July 2003. She was appointed as an Associate Professor in September 2009. Her research interests encompass the Internet of Things (IoT), software architecture, and data mining and analysis.

YONGFA LI received the B.S. degree in computer science and technology from the Shenyang University of Technology, China, in 2013. He is currently pursuing the M.S. degree with the Computer Science and Engineering, South China University of Technology, China. His current research interests include natural language processing, deep learning, machine learning, and human action recognition.

XUE TANG received the bachelor's degree in software engineering from the Software College, South China University of Technology, in 2015, and the master's degree in computer science and technology from the School of Computer Science, South China University of Technology, in 2018.

JINGYUN XU is currently pursuing the Ph.D. degree in software engineering with the South China University of Technology, Guangzhou, China. Her research interests include natural language processing, word representation, text classification, and sentiment analysis.



SHENG BI received the M.Sc. and Ph.D. degrees from the South China University of Technology, Guangzhou, China, in 2003 and 2010, respectively. He has been a Fellow with the School of Computer Science and Engineering, South China University of Technology, since July 2003. He was appointed as an Associate Professor in September 2014. He is currently a Research Fellow of the Shenzhen Academy of Robotics. His research interests are intelligent robots, FPGA fast processing algorithms, embedded intelligent terminal, and smartphone development.

YI CAI (Member, IEEE) was born in Beihai, Guangxi, China, in 1980. He received the B.S. degree in polymer materials from Sichuan University, Chengdu, China, in 2003, and the M.S. and Ph.D. degrees in computer science from The Chinese University of Hong Kong, Hong Kong, in 2005 and 2009, respectively.

He was a Visiting Scholar with the KEG of Tsinghua University, Nanyang Technological University. From 2009 to 2011, he held a postdoctoral position at the City University of Hong Kong, Hong Kong. He is currently a Professor with the School of Software Engineering, South China University of Technology, Guangzhou, China. He published 90 articles, including CIKM, ER, WI, WISE, ICTAI, and JCST, and RW books. He reviewed articles from conferences and journals related to information retrieval, semantic web, recommender systems, data mining and database, including TKDE, TOIT, WWWJ, JCST, CIKM, and ER. His research interests are recommendation systems, personalized search, semantic web, and data mining. He is a Program Committee Member of conferences, including ICWE, WAIM, ICEBE, and NDBC.