

Received 7 November 2022, accepted 22 November 2022, date of publication 30 November 2022,  
date of current version 6 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3225659

## RESEARCH ARTICLE

# Word-Level and Pinyin-Level Based Chinese Short Text Classification

XINJIE SUN<sup>1,2</sup> AND XINGYING HUO<sup>1</sup>

<sup>1</sup>Institute of Computer Science, Liupanshui Normal University, Liupanshui 553004, China

<sup>2</sup>Guizhou Xinjie Qianxun Software Service Company Ltd., Liupanshui 553004, China

Corresponding author: Xinjie Sun (sxj123lps@163.com)

This work was supported in part by the Guizhou Province under Grant KY[2020]112, in part by the Liupanshui Normal University High level Talent Research Launch Fund under Grant LPSSYKYJJ201708, in part by the Liupanshui Science and Technology Bureau Fund Project under Grant 52020-2018-04-15, in part by the Liupanshui Normal University Major Comprehensive Reform Pilot Project under Grant LPSSYzyzhggsd202003, in part by the Science and Technology Foundation of Guizhou Province under Grant ZK[2022]528, and in part by the Youth Science and Technology Talent Growth Project of Department of Education in Guizhou Province under Grant KY[2022]054.

**ABSTRACT** Short text classification is an important branch of Natural Language Processing. Although CNN and RNN have achieved satisfactory results in the text classification tasks, they are difficult to apply to the Chinese short text classification because of the data sparsity and the homophonic typos problems of them. To solve the above problems, word-level and Pinyin-level based Chinese short text classification model is constructed. Since homophones have the same Pinyin, the addition of Pinyin-level features can solve the homophonic typos problem. In addition, due to the introduction of more features, the data sparsity problem of short text can be solved. In order to fully extract the deep hidden features of the short text, a deep learning model based on BiLSTM, Attention and CNN is constructed, and the residual network is used to solve the gradient disappearance problem with the increase of network layers. Additionally, considering that the complex deep learning network structure will increase the text classification time, the Text Center is constructed. When there is a new text input, the text classification task can be quickly realized by calculating the Manhattan distance between the embedding vector of it and the vectors stored in the Text Center. The Accuracy, Precision, Recall and F1 of the proposed model on the simplifyweibo\_4\_moods dataset are 0.9713, 0.9627, 0.9765 and 0.9696 respectively, and those on the online\_shopping\_10\_cats dataset are 0.9533, 0.9416, 0.9608 and 0.9511 respectively, which are better than that of the baseline method. In addition, the classification time of the proposed model on simplifyweibo\_4\_moods and online\_shopping\_10\_cats is 0.0042 and 0.0033 respectively, which is far lower than that of the baseline method.

**INDEX TERMS** Short text classification, data sparsity, homophonic typos problem, word-level, Pinyin-level, text center.

## I. INTRODUCTION

The continuous development of social media has gradually made it the main platform for netizens<sup>1</sup> to express their views and opinions. A large number of active netizens publish a large number of micro-blogs, tweets and other short texts bearing user information everyday, which contains abun-

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan<sup>id</sup>.

<sup>1</sup>It refers to all people who conduct network activities through computers and the Internet.

dant valuable information reflecting public opinion, social hot spots and user interests. However, how to quickly and accurately mine important information from massive texts according to the personalized needs of society and users is still facing huge challenges. The research of short text classification technology can help the system more efficiently “understand” and “manage” all kinds of short texts, which has an important role in promoting the development of social intelligence.

As an important research content of Natural Language Processing (NLP) [1], text classification has been widely

used in search engines [2], [3], information filtering [4], [5], subject tracking [6], [7], mail classification [8], [9] and other fields. The traditional machine learning classification method divides the whole text classification task into feature engineering and classifier. Feature engineering is divided into three parts: text preprocessing, feature extraction and text representation. The ultimate goal is to convert text into a format that can be understood by the computer and encapsulate enough information for classification, that is, strong feature expression ability. Commonly used machine learning classifiers include Naive Bayes (NB) [10], K-Nearest Neighbor (KNN) [11], Decision Tree (DT) [12], Support Vector Machine (SVM) [13], etc. Feature engineering is a very complicated process, and the quality of feature selection directly determines the classification results of machine learning classifiers. With the continuous development of deep learning models, more and more researchers use deep learning models for text classification.

Deep learning method does not require complex feature engineering, but relies on its powerful ability to fit data distribution to automatically learn rules by training massive data. As Convolutional Neural Network (CNN) [14] can extract text features through one-dimensional convolution operation and retain the most important features through pooling operation, many researchers use CNN for text classification [15]. Text data can be regarded as words that have sequence relationships, which are difficult for CNN to capture [16]. Therefore, Recurrent Neural Network (RNN) [17] is used in text classification due to its powerful ability in processing variable-length sequence input. However, RNN has the problem of gradient disappearance when the text length is too long, and it is difficult to capture long-distance global dependencies [18], which is solved by Long Short-Term memory (LSTM) [19]. Considering that LSTM only propagate forward and can easily ignore important content after the current time node, Bidirectional Long Short-Term Memory (BiLSTM) [20] is proposed to use both forward and backward temporal features for text classification.

Although CNN and RNN have achieved satisfactory results in the text classification tasks, most of them are for English and there are only a few studies on Chinese text classification [21], [22], [23]. With the rapid development of Chinese social network, the classification of Chinese short text becomes more and more important. Since Chinese is based on pictograms and there are many homonyms in Chinese, homophonic typos problem<sup>2</sup> often occurs in Chinese short texts, which is difficult to be handled by traditional text classification models. Considering that Pinyin in Chinese is a system that takes Latin letters as the modern standard Chinese phonetics, which can be used to solve the homophonic typos problem of Chinese short texts. Therefore, word-level and Pinyin-level based short text classification model, WP-STC, is constructed in this paper. Since WP-STC contains both

<sup>2</sup>It means that Chinese words are wrongly written into other words with the same Pinyin but other meanings.

word-level and Pinyin-level features, it can also effectively solve the data sparsity problem<sup>3</sup> of short texts.

Considering that RNNs assign the same attention to all contexts and the text classification results of deep learning model alone are always not satisfactory [15], a deep learning model based on BiLSTM, Attention mechanism and CNN is proposed in this paper. BiLSTM is used to obtain bidirectional temporal features, Attention mechanism is used to assign different weights to the context, and CNN is used to extract local features and reduce dimension. In addition, the residual network is used to solve the gradient disappearance problem with the increase of network layers [25].

Additionally, considering that the complex deep learning network structure will increase the text classification time, the concept of Text Center is innovatively proposed. When there is a new text input, the text classification task can be quickly realized by calculating the Manhattan distance between the embedding vector of it and the vectors stored in the Text Center.

The main contributions of this paper are as follows:

(1) Word-level and Pinyin-level based Chinese short text classification model is proposed, which can well solve the data sparsity and the frequent occurrence of homophonic typos problems of Chinese short texts.

(2) A deep learning model based on BiLSTM, Attention mechanism, CNN and residual network is proposed. BiLSTM is used to obtain bidirectional temporal features, Attention is used to assign different weights to the context, and CNN is used to extract local features and reduce dimension. In addition, the residual network is used to solve the gradient disappearance problem with the increase of network layers.

(3) The concept of Text Center is innovatively proposed, which can greatly reduce the classification time by just comparing the embedding vector of the input text with the vectors stored in the Text Center.

(4) Multi-group comparison experiments on two public baseline datasets prove that the proposed model not only get the best classification Accuracy, Precision, Recall and F1, but also greatly reduces the classification time.

The remainder of this work is organized as follows. Section II introduces the latest research results of text classification. Section III introduces the background of LSTM, Attention mechanism and CNN. Section IV introduces the construction of the proposed model. Section V introduces the experimental results of the proposed model and the comparison models. Section VI provides summary and future work.

## II. RELATED WORK

### A. TEXT CLASSIFICATION

Commonly used text classification methods include machine learning and deep learning.

<sup>3</sup>It refers to the situation that a large number of data are missing or zero in the data matrix.

## 1) MACHINE LEARNING

Luo et al. [13] implemented the SVM in classifying English text and documents. Experimental results performed on a set of 1033 text document demonstrated that the classifier provided the best results when the feature set size is small. Richard et al. [26] reviewed how the NLP techniques of TF-IDF combined with the supervised machine learning model of SVM and word embedding approaches such as Word2vec could be used to categorize/label protocol deviations across multiple therapeutic areas. Annalisa et al. [27] presented an updated survey of 12 machine learning text classifiers applied to a public spam corpus. They proposed a new pipeline to optimise hyperparameter selection and improve the models' performance by applying specific methods (based on NLP) in the preprocessing stage. Luo [13] implemented the SVM model in classifying English text and documents. Abdalla et al. [28] proposed similarity measures with machine learning models and presented benchmarking studies for integration methodology over balanced/imbalanced datasets.

## 2) DEEP LEARNING

Machine learning methods highly rely on feature engineering, which is very complex and time-consuming, therefore. Since deep learning methods do not need complex feature engineering, more and more researchers use them for text classification.

Cheng et al. [29] proposed a novel text classification model based on hierarchical self-attention mechanism capsule network, which was composed of the capsule network and hierarchical self-attention network. The experimental results on 5 text classification datasets showed that the proposed model achieved the best classification results compared with other baseline models. Kong et al. [30] proposed a hierarchical BERT with an adaptive fine-tuning strategy (HAdaBERT). HAdaBERT consisted of an attention-based gated memory network as the global encoder and a BERT-based model as the local encoder. Experimental results on different corpora indicated that HAdaBERT outperformed the state-of-the-art pretrained language models. Sergio et al. [31] proposed Stacked DeBERT, which improved the robustness of incomplete data by designing a novel encoding scheme in BERT. Stacked DeBERT took advantage of stacks of multilayer perceptrons for the reconstruction of missing words' embeddings by extracting more abstract and meaningful hidden feature vectors, and bidirectional transformers for improved embedding representation. Liu et al. [32] proposed a Co-attention Network with Label Embedding (CNLE) that jointly encoded the labels and text into their mutually attended representations, which was able to attend to the relevant parts of both. Experiments showed that CNLE achieved competitive results on 2 multi-label and 7 multi-class classification benchmarks. Zhang [33] proposed a news text classification method based on the combination of deep learning (DL) algorithms. Gao et al. [34] introduced four methods to scale

BERT to perform document classification on clinical texts several thousand words long.

## B. SHORT TEXT CLASSIFICATION

Although machine learning and deep learning models are widely used text classification models, considering the data sparsity of short texts, simple machine learning or deep learning network cannot classify them well. Therefore, researchers use more complex models for short text classification.

Yang et al. [35] proposed a novel heterogeneous Graph Neural Network (GNN)-based method for semi-supervised short text classification, leveraging full advantage of limited labeled data and large unlabeled data through information propagation along the graph. Wang et al. [36] proposed SHINE, which used Graph Neural Network (GNN) for short text classification. Wang et al. [37] proposed a short text classification method based on semantic extension and CNN. Škrlj et al. [39] proposed tax2vec, a parallel algorithm for constructing taxonomy-based features, which could well solve the data sparsity problem of short texts. Zhou et al. [40] proposed a semantic extension-based classification algorithm for short texts and both ordinary 1D convolution and atrous convolution are performed. The model achieved the best classification results and had lower computational complexity than BERTbase. Yu et al. [41] proposed Deep Pyramid Temporal Convolutional Network for short text classification, which was mainly consisting of concatenated embedding layer, causal convolution, 1/2 max pooling down-sampling and residual blocks.

## C. CHINESE SHORT TEXT CLASSIFICATION

Although deep learning models have achieved satisfactory results in short text classification tasks, most of them are for English and there are only a few studies on Chinese short text classification.

Hao et al. [21] proposed a Mutual-Attention CNN framework, which integrated features at the word and character levels for Chinese short text classification. Lyu et al. [22] introduced HowNet<sup>4</sup> as an external knowledge base, and proposed a language knowledge enhancement graph converter to deal with Chinese word ambiguity problem. Feng et al. [23] proposed a sentiment classification model for short Chinese texts. The model combined word features with part-of-speech features, position features and dependent syntactic features to form three new combined features, which were input into the multi-channel CNN, and combined with the multi-head attention mechanism to more fully integrate the multi-head attention mechanism. Yang et al. [42] proposed a character-word graph attention network to explore the interactive information between characters and words for Chinese text classification.

The existing Chinese short text classification models are basically based on the characteristics of words or characters, and it is difficult to deal with the homophonic typos problem that often occurs in Chinese short texts. In order to overcome

<sup>4</sup><https://www.cnki.net/>

the above problem, word-level and Pinyin-level based Chinese short text classification model, WP-STC, is constructed in this paper.

### III. BACKGROUND

#### A. LSTM

LSTM [19] is a special type of Recurrent Neural Network (RNN) [43] designed to deal with the gradient disappearance problem faced by RNNs. Like other types of RNNs, LSTM generates its outputs based on the input of the current time step and the output of the previous time step, and sends the current output to the next time step. Each LSTM cell consists of a memory cell  $c_t$  that maintains its state at arbitrary time intervals and three nonlinear gates, including an input gate  $i_t$ , a forget gate  $f_t$ , and an output gate  $o_t$ , which are designed to regulate the flow of information into and out of the memory cell. An LSTM containing a hidden layer is defined as follows [19].

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (2)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \quad (5)$$

$$h_t = o_t \otimes \tanh(c_t) \quad (6)$$

where  $i_t$ ,  $f_t$ ,  $o_t$  and  $c_t$  denote the input gate, forget gate, output gate, and memory cell activation vector at moment  $t$ , respectively.  $\sigma(\cdot)$  denotes the logistic sigmoid function,  $\otimes$  denotes the element-level multiplication operation. The trainable parameters  $W_i, W_f, W_o, W_c \in \mathbb{R}^{H \times d}$ ,  $U_i, U_f, U_o, U_c \in \mathbb{R}^{H \times H}$ ,  $b_i, b_f, b_o$  and  $b_c$  denote the bias terms.  $d$  and  $H$  denote the dimension and the hidden layer of the input.

#### B. ATTENTION MECHANISM

The Attention mechanism [38] can highlight important information from contextual information by setting different weights, thus paying more attention to the parts that are similar to the elements of the input and suppressing other useless information.

Let  $h = (h_1, h_2, \dots, h_T)$  denote the input of the Attention network, then the correlation  $e_{ij}$  between the  $j$ th input  $h_j$  and the current hidden state  $s_{t-1}$  is calculated as follows [38]:

$$e_{ij} = \text{score}(s_{t-1}, h_j) \quad (7)$$

where  $\text{score}()$  denotes the weight-focused multiplication. Assuming that  $v$  is the trainable parameter, the associated likelihood  $a_{ij}$  and the content vector  $c_t$  for time step  $t$  are calculated as follows [38]:

$$a_{ij} = \frac{\exp(v^T e_{ij})}{\sum \exp(v^T e_{ij})} \quad (8)$$

$$c_t = \sum a_{ij} h_t \quad (9)$$

#### C. CNN

The CNN [14] for text classification includes multiple one-dimensional (1D) convolution layers and pooling layers. 1D convolution is used to extract deep features, and pooling is used to further obtain important features and reduce dimensions. For the Chinese text  $S$  containing  $s$  embedding vectors, assuming that the length of each embedding vector is  $e$ , then  $S$  is a matrix with the size of  $s \times e$ . Multiple linear filters with the size of  $h \times e$  are used to perform the convolution operation to generate the feature map  $M = [m_0, m_1, \dots, m_{s-h}]$ .  $h$  is the length of the linear filter and  $1 \leq h \leq s$ . Let  $S_{i:j}$  represents the matrix from the  $i$ th word to the  $j$ th word in  $S$ , then the  $i$ th feature sequence of  $M$  is generated by the following formula [14]:

$$m_i = f(WS_{i:i+h-1} + b) \quad (i = 0, 1, \dots, s-h) \quad (10)$$

where  $f()$  denotes the activation function Relu,  $W$  denotes the weight and  $b$  denotes the bias term. After the convolution operation, the max pooling is used to reduce the dimension and extract the key features  $b$  [50]:

$$b = \max \{m_i\} \quad (11)$$

### IV. MODEL CONSTRUCTION

WP-STC is constructed in this paper for Chinese short text classification, and the structure is shown in Figure 1. In the first step, the Chinese text of input layer is transformed into word-level and Pinyin-level features. In the second step, the word-level and Pinyin-level features are transformed into embedding vectors by pre-trained Word2vec [44]. In the third step, a deep learning model, Bi-Att-CNN, which is based on BiLSTM, Attention and CNN is constructed, and the word-level and Pinyin-level embedding vectors are both fed into Bi-Att-CNN to obtain the hidden vectors. In the fourth step, the word-level and Pinyin-level hidden vectors are added together to fed into the fully connected network and the softmax function is used to get the classification results. At the same time, the word-level and Pinyin-level hidden vectors are used to get the Text Center of different text categories. When there is a new text input, the text classification task can be quickly realized by calculating the Manhattan distance between the embedding vector of it and the vectors stored in the Text Center. The implementation process of the model is described in detail below.

#### A. ACQUISITION OF WORD-LEVEL AND PINYIN-LEVEL FEATURES

The processes of converting Chinese text in input layer into word-level and Pinyin-level features are shown in Figure 2. Given a Chinese short text  $T$  of length  $z$ , firstly,  $T$  is transformed into word-level feature  $T_{word} = \{w_1, w_2, \dots, w_b\}$  containing  $b$  words by jieba,<sup>5</sup> and then each Pinyin of  $T_{word}$  is obtained by pyPinyin<sup>6</sup> to get the Pinyin-level feature

<sup>5</sup>The most commonly used Chinese word segmentation tool.

<sup>6</sup>Python package used to get Pinyin of Chinese word.

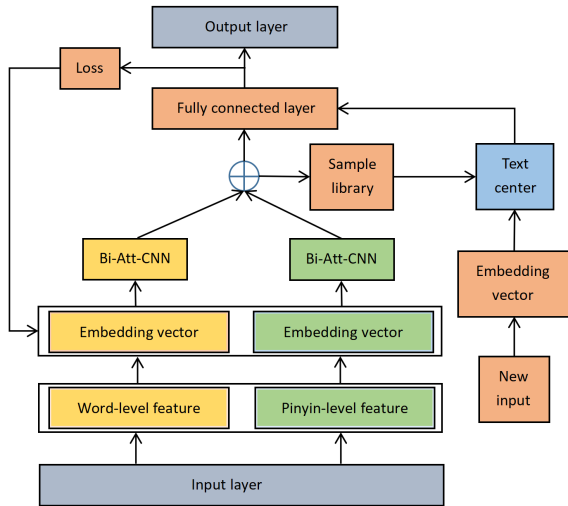


FIGURE 1. The structure of WP-STC.

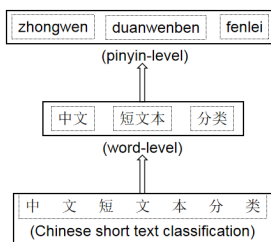


FIGURE 2. The processes of converting Chinese text into word-level and Pinyin-level features.

$T_{Pinyin} = \{p_1, p_2, \dots, p_z\}$ . For every word has a Pinyin, then  $b = z$ .

**B. ACQUISITION OF EMBEDDING VECTOR**

Word2vec [45] is the most commonly used static word vector model, which can convert word and Pinyin into fixed length word vectors and a pre-trained Word2vec [44] is used to transform both  $T_{word}$  and  $T_{Pinyin}$  into embedding vectors  $V_{word} = \{v^{w_1}, v^{w_2}, \dots, v^{w_b}\}$  and  $V_{Pinyin} = \{v^{p_1}, v^{p_2}, \dots, v^{p_z}\}$ . For computational convenience, assume that the length of each word and Pinyin embedding vector is  $p$ , then  $V_{word} \in \mathbb{R}^{b \times p}$ ,  $V_{Pinyin} \in \mathbb{R}^{z \times p}$ .

**C. CONSTRUCTION OF BI-ATT-CNN**

The deep learning model based on BiLSTM, Attention and CNN, Bi-Att-CNN, is constructed to get the hidden vectors of  $V_{word}$  and  $V_{Pinyin}$ , and the structure of Bi-Att-CNN is shown in Figure 3. Bi-Att-CNN can well extract the deep hidden features of the embedding vectors of the embedding layer. The specific processes are as follows. Firstly,  $V_{word}$  and  $V_{Pinyin}$  from the embedding layer are fed into BiLSTM to obtain the forward and backward hidden vectors  $\vec{h}_{word}$ ,  $\overleftarrow{h}_{word}$ ,  $\vec{h}_{Pinyin}$  and  $\overleftarrow{h}_{Pinyin}$ :

$$\vec{h}_{word} = \overrightarrow{LSTM}(V_{word}) \tag{12}$$

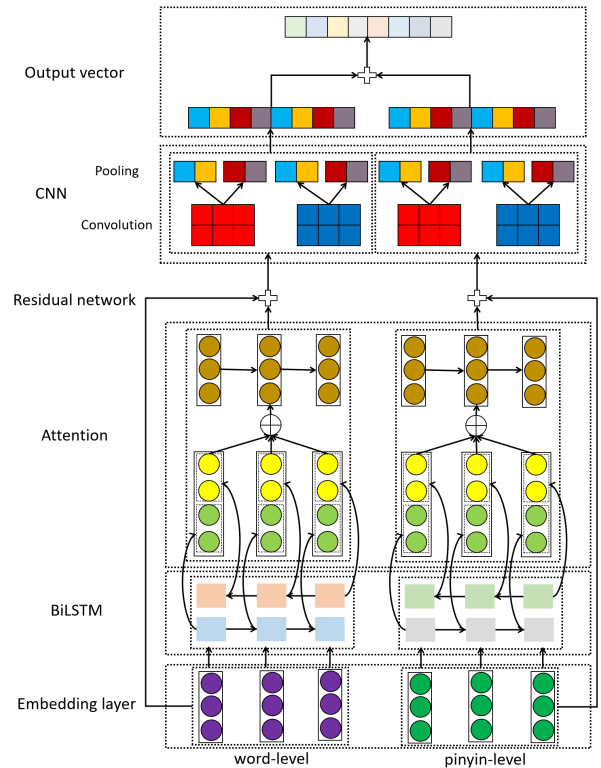


FIGURE 3. The structure of Bi-Att-CNN.

$$\overleftarrow{h}_{word} = \overleftarrow{LSTM}(V_{word}) \tag{13}$$

$$\vec{h}_{Pinyin} = \overrightarrow{LSTM}(V_{Pinyin}) \tag{14}$$

$$\overleftarrow{h}_{Pinyin} = \overleftarrow{LSTM}(V_{Pinyin}) \tag{15}$$

Secondly, the forward and backward hidden vectors are concatenated to obtain the bi-directional timing vectors  $h_{word}$  and  $h_{Pinyin}$ :

$$h_{word} = \begin{bmatrix} \vec{h}_{word} & \overleftarrow{h}_{word} \end{bmatrix} \tag{16}$$

$$h_{Pinyin} = \begin{bmatrix} \vec{h}_{Pinyin} & \overleftarrow{h}_{Pinyin} \end{bmatrix} \tag{17}$$

Thirdly,  $h_{word}$  and  $h_{Pinyin}$  are fed into the Attention network separately, so as to assign different attentions to the context. Specifically, at first, the hidden representations  $x_{word}$  and  $x_{Pinyin}$  of  $h_{word}$  and  $h_{Pinyin}$  are obtained respectively:

$$x_{word} = \tanh(W_{word}h_{word} + b_{word}) \tag{18}$$

$$x_{Pinyin} = \tanh(W_{Pinyin}h_{Pinyin} + b_{Pinyin}) \tag{19}$$

where  $W_{word}$  and  $W_{Pinyin}$  denote weights and  $b_{word}$  and  $b_{Pinyin}$  denote bias terms. Next, the importances of the context are calculated based on the similarities between  $y_{word}$ ,  $y_{Pinyin}$  and  $x_{word}$ ,  $x_{Pinyin}$ , where  $y_{word}$  and  $y_{Pinyin}$  are the randomly initialized context vectors. After obtaining the weights, the softmax function is used to normalize them to obtain the weight vectors  $r_{word}$  and  $r_{Pinyin}$ :

$$r_{word} = \frac{\exp(x_{word}y_{word})}{\sum \exp(x_{word}y_{word})} \tag{20}$$

$$r_{Pinyin} = \frac{\exp(x_{Pinyin}y_{Pinyin})}{\sum \exp(x_{Pinyin}y_{Pinyin})} \quad (21)$$

Finally, the comment vectors  $f_{word}$  and  $f_{Pinyin}$  containing all contextual attention are obtained by weighted summation of  $r_{word}$  and  $r_{Pinyin}$ :

$$f_{word} = \sum r_{word}h_{word} \quad (22)$$

$$f_{Pinyin} = \sum r_{Pinyin}h_{Pinyin} \quad (23)$$

Fourthly, the residual network is used to solve the gradient disappearance problem with the increase of network layers by adding the output vectors of the Attention network and the initial embedding vectors:

$$fr_{word} = r_{word} + V_{word} \quad (24)$$

$$fr_{Pinyin} = r_{Pinyin} + V_{Pinyin} \quad (25)$$

Fifthly,  $fr_{word}$  and  $fr_{Pinyin}$  are fed into the convolutional networks of CNN to obtain the results of convolutional operation  $t_{word}$  and  $t_{Pinyin}$ :

$$t_{word} = \text{relu}(W_{word}^{cnn}fr_{word} + b_{word}^{cnn}) \quad (26)$$

$$t_{Pinyin} = \text{relu}(W_{Pinyin}^{cnn}fr_{Pinyin} + b_{Pinyin}^{cnn}) \quad (27)$$

where  $W_{word}^{cnn}$  and  $W_{Pinyin}^{cnn}$  denote weights and  $b_{word}^{cnn}$  and  $b_{Pinyin}^{cnn}$  denote bias terms.

Sixthly,  $t_{word}$  and  $t_{Pinyin}$  are fed into the max pooling networks of CNN to get the result of pooling operation  $g_{word}$  and  $g_{Pinyin}$ :

$$g_{word} = \max(t_{word}) \quad (28)$$

$$g_{Pinyin} = \max(t_{Pinyin}) \quad (29)$$

Seventhly,  $g_{word}$  and  $g_{Pinyin}$  are added to get the output vector  $g = g_{word} + g_{Pinyin}$  of Bi-Att-CNN network.

#### D. CLASSIFICATION RESULTS

The output vector  $g$  of the Bi-Att-CNN network is fed to the fully connected network, and then the softmax function is used to get the final classification result  $label$ .

$$label = \text{argmax}(\text{softmax}(\tanh(gW^g))) \quad (30)$$

where  $W^g$  denotes the weight.

#### E. TEXT CENTER

Since the number of samples used for training is always large, it is very time-consuming to traverse all training samples and compute their distances from the new input text when a new text input is available. In order to reduce the computational complexity, the concept of Text Center is proposed. When there is a new text input, the text classification task can be quickly realized by calculating the Manhattan distance [46] between the embedding vector of it and the vectors stored in the Text Center. The original and improved calculation methods are shown in Figure 4 and Figure 5, respectively.

We analyze the calculation process of the original method and Text Center separately. Suppose the example of the sample library is denoted as  $X = \{X_1, X_2, \dots, X_k\}$ ,  $k$  is the length

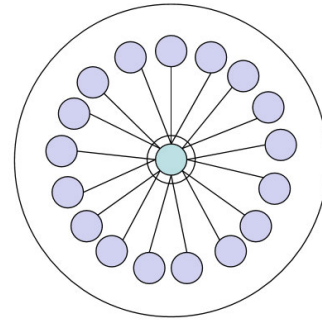


FIGURE 4. Calculation process of original text classification method.

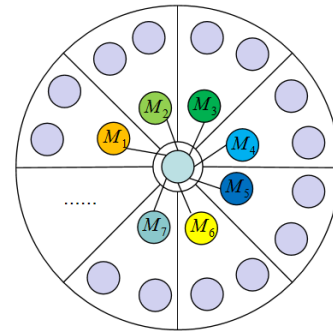


FIGURE 5. Calculation process of Text Center.

of the sample library,  $X_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$ ,  $n$  is the length of the feature dimension,  $x_{label} = \{1, 2, \dots, a\}$ ,  $a$  denotes the total class of the text. The word vector corresponding to the new short text to be classified is assumed to be  $Y = \{y_1, y_2, \dots, y_n\}$ . The process of text classification by the original method is as follows.

$$\begin{aligned} Test_i &= |X_i - Y| \\ &= |x_{i1} - y_1| + |x_{i2} - y_2| + \dots + |x_{in} - y_n| \\ y_{label} &= x_{label} \min(Test_i) \end{aligned} \quad (31)$$

It is not difficult to obtain the time complexity to be  $o(k \times n)$ . In order to reduce the amount of computation and speed up the classification, the time complexity can be reduced by establishing the Text Center of the sample library. The specific process is as follows.

First, the Text Center  $T_1-T_a$  are established for each of the  $a$  categories of text:

$$\begin{aligned} T_1 &= \{t_{11}, t_{12}, \dots, t_{1n}\} \\ t_{1i} &= \frac{1}{k_1} \sum_{i=1}^{k_1} x_{1i}, X_{label} = 1 \\ T_2 &= \{t_{21}, t_{22}, \dots, t_{2n}\} \\ t_{2i} &= \frac{1}{k_2} \sum_{i=1}^{k_2} x_{2i}, X_{label} = 2 \\ &\dots\dots\dots \\ T_a &= \{t_{a1}, t_{a2}, \dots, t_{an}\} \end{aligned}$$

$$t_{ai} = \frac{1}{k_a} \sum_{i=1}^{k_a} x_{ai} \cdot X_{label} = a \quad (32)$$

where  $k_i$  denotes the number of samples labeled  $i$ ,  $i = 1, 2, \dots, a$ . The improved similarity is calculated as follows:

$$\begin{aligned} TestT_1 &= |T_1 - Y| \\ &= |t_{11} - y_1| + |t_{12} - y_2| + \dots + |t_{1n} - y_n| \\ TestT_2 &= |T_2 - Y| \\ &= |t_{21} - y_1| + |t_{22} - y_2| + \dots + |t_{2n} - y_n| \\ &\dots\dots \\ TestT_a &= |T_a - Y| \\ &= |t_{a1} - y_1| + |t_{a2} - y_2| + \dots + |t_{an} - y_n| \\ y_{label} &= x_{label} \min(TestT_1, TestT_2, \dots, TestT_a) \end{aligned} \quad (33)$$

It is easy to see that only  $a$  calculations need to be performed to derive the classification results for the new input text.

## V. EXPERIMENT

### A. DATASET

Two Chinese baseline datasets are selected, and the details are as follows.

simplifyweibo\_4\_moods<sup>7</sup>: The dataset contains more than 200,000 Weibo<sup>8</sup> records labeled with four types of emotions, including about 50,000 joyful, angry, disgusted and depressed Weibo records each.

online\_shopping\_10\_cats<sup>9</sup>: The dataset contains more than 60,000 reviews for products in 10 categories, with about 30,000 positive and 30,000 negative reviews each.

### B. BASELINE METHODS

KNN [47]: A commonly used machine learning algorithm for text classification.

Decision Tree (DT) [48]: A commonly used machine learning algorithm for sentiment classification.

SVM [49]: A commonly used machine learning algorithm for text classification.

TextCNN [50]: This model extracts local features by convolution operation and dimensionality reduction by pooling operation, and is a commonly used deep learning algorithm for text classification.

LSTM [19]: This model effectively solves the gradient disappearance problem of RNN by setting multiple gates, and is a commonly used deep learning algorithm for text classification.

BiLSTM [20]: This model takes into account the forward and backward temporal content, and is a commonly used deep learning algorithm for text classification.

HGAT [35]: This model is a heterogeneous GNN-based method for semi-supervised short text classification, leverag-

ing full advantage of limited labeled data and large unlabeled data through information propagation along the graph.

SHINE [36]: This model first models the short text as a hierarchical heterogeneous graph consisting of word-level component graphs which introduce more semantic and syntactic information. Then, the model dynamically learns a short document graph to facilitate effective label propagation among similar short texts.

T-SNE [51]: This model proposes a method to generate a word-level emotion distribution vector for short text classification.

BERT-TER [52]: This model proposes a dual-channel system for multi-class short text emotion recognition, and develops a technique to explain its training and predictions.

The baseline methods are reproduced according to the references, and the optimal values of each baseline method after repeated parameter adjustment are obtained as the experimental results.

### C. EVALUATION CRITERIA

Accuracy, Precision, Recall and F1 are used as evaluation criteria, which are calculated as follows:

$$\begin{aligned} Accuracy &= \frac{TP + TN}{TP + FP + TN + FN} \\ Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \\ F1 &= \frac{2 * Precision * Recall}{Precision + Recall} \end{aligned} \quad (34)$$

where  $P$  and  $N$  denote the number of forward and reverse samples, respectively,  $TP$  and  $TN$  denote the number of correctly predicted forward and reverse samples, respectively, and  $FP$  and  $FN$  denote the number of incorrectly predicted forward and reverse samples, respectively [53].

### D. EXPERIMENTAL ENVIRONMENT

Deep learning model based on BiLSTM, Attention and CNN, Bi-Att-CNN, is constructed in this paper. There are three layers of BiLSTM networks, and the number of neurons is 256, 256, 128, respectively. Relu is used as activation function, and 0.5 dropout is used at the end of each layer. There are two layers of CNN networks. The first layer contains 128 convolutional kernels of size 3, and the size of pooling template is 3. The second layer contains 64 convolution kernels of size 3, and the size of pooling template is 3. Two layers of fully connected networks are contained. The number of neurons in the first layer is 256, and that in the second layer is equal to the number of text categories in the dataset. The length of the embedding vector obtained through Word2vec is 100. The learning rate is set to 0.0001, the batch size is set to 64, and the epoch is set to 100.

<sup>7</sup><https://zhuanlan.zhihu.com/p/80029681>

<sup>8</sup>One of the largest social platforms in China.

<sup>9</sup><https://github.com/SophonPlus/ChineseNlpCorpus/raw/master/datasets/>

**TABLE 1. Classification results of the comparison models on the simplifyweibo\_4\_moods dataset (The best results are in bold).**

Method	Accuracy	Precision	Recall	F1	Time (s)
KNN	0.7981	0.7851	0.8003	0.7926	89.5
DT	0.8110	0.8121	0.8069	0.8095	13.6
SVM	0.8319	0.8259	0.8372	0.8315	179.6
TextCNN	0.8399	0.8407	0.8330	0.8369	403.5
LSTM	0.8167	0.8183	0.8124	0.8153	372.6
BiLSTM	0.8487	0.8426	0.8541	0.8483	411.2
HGAT	0.8569	0.8577	0.8498	0.8538	493.3
SHINE	0.8497	0.8514	0.8452	0.8483	384.1
T-SNE	0.8569	0.8577	0.8541	0.8559	418.0
BERT-TER	0.9008	0.9032	0.8988	0.9010	1462.7
WP-STC	0.9545	0.9565	0.9614	0.9589	522.4
WP-STC-Center	<b>0.9713</b>	<b>0.9627</b>	<b>0.9765</b>	<b>0.9696</b>	<b>0.0042</b>

**TABLE 2. Classification results of the comparison models on the online\_shopping\_10\_cats dataset (The best results are in bold).**

Method	Accuracy	Precision	Recall	F1	Time (s)
KNN	0.7643	0.7681	0.7662	0.7671	48.3
DT	0.7571	0.7607	0.7594	0.7600	7.3
SVM	0.7769	0.7669	0.7732	0.7700	97.0
TextCNN	0.7933	0.7905	0.7942	0.7923	217.9
LSTM	0.7870	0.7893	0.7876	0.7885	201.2
BiLSTM	0.8609	0.8652	0.8630	0.8641	222.0
HGAT	0.8528	0.8569	0.8554	0.8561	266.4
SHINE	0.8569	0.8458	0.8528	0.8493	207.4
T-SNE	0.8948	0.8917	0.8958	0.8937	225.7
BERT-TER	0.9162	0.9189	0.9169	0.9179	649.9
WP-STC	0.9317	0.9341	0.9295	0.9318	282.1
WP-STC-Center	<b>0.9533</b>	<b>0.9416</b>	<b>0.9608</b>	<b>0.9511</b>	<b>0.0033</b>

**E. EXPERIMENTAL RESULTS**

In this section, multiple sets of experiments are conducted to verify the classification effects of WP-STC, WP-STC with Text Center and baseline methods. To facilitate understanding, WP-STC-Center is used to represent WP-STC with Text Center. For all experiments, the ratios of train set, test set and validation set on the two baseline datasets are set to 7:2:1. In addition, in order to make the classification results more accurate, 5-fold cross-validation is adopted in all experiments.

WP-STC, WP-STC-Center and baseline methods are applied to the two baseline datasets, respectively, and the classification results are shown in Tables 1 and 2, respectively. It can be seen that the classification results of WP-STC on the two baseline datasets are better than all baseline methods. The Accuracy, Precision, Recall and F1 are improved by at least 0.0538, 0.0533, 0.0625 and 0.0579, respectively. However, although WP-STC outperforms the state-of-the-art baseline method, it takes the second longest time, while WP-STC-Center takes the shortest time and has the best classification Accuracy, Precision, Recall and F1, which indicates that the text-centered classification model proposed in this paper can not only improve the classification Accuracy, Precision, Recall and F1, but also reduce the classification time.

**F. MODEL ANALYSIS**

**1) IMPORTANCE ANALYSIS OF EACH COMPONENT**

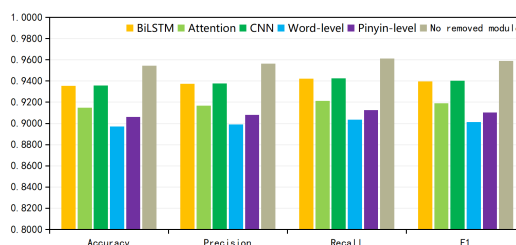
The following experiments are conducted to verify the importance of each module in WP-STC and WP-STC-Center. WP-STC and WP-STC-Center are constructed to

**TABLE 3. Classification results of different distance methods in simplifyweibo\_4\_moods dataset (The best results are in bold).**

Distance	Accuracy	Precision	Recall	F1	Time (s)
Euler	0.9606	0.9599	0.9722	0.9660	0.0158
Minkowski	0.9655	0.9623	0.9753	0.9688	0.0571
Chebyshev	0.9638	0.9517	0.9694	0.9605	0.3293
Cosine similarity	0.9587	0.9506	0.9677	0.9591	0.0635
Manhattan	<b>0.9713</b>	<b>0.9627</b>	<b>0.9765</b>	<b>0.9696</b>	<b>0.0042</b>

**TABLE 4. Classification results of different distance methods in online\_shopping\_10\_cats dataset (The best results are in bold).**

Distance	Accuracy	Precision	Recall	F1	Time (s)
Euler	0.9488	0.9305	0.9479	0.9391	0.0136
Minkowski	0.9526	0.9411	0.9582	0.9496	0.0511
Chebyshev	0.9479	0.9396	0.9503	0.9449	0.2979
Cosine similarity	0.9371	0.9317	0.9429	0.9373	0.0583
Manhattan	<b>0.9533</b>	<b>0.9416</b>	<b>0.9608</b>	<b>0.9511</b>	<b>0.0033</b>



**FIGURE 6. Classification results of removing modules from WP-STC on the simplifyweibo\_4\_moods dataset (legend indicates the removed modules).**

implement Chinese short text classification by word-level and Pinyin-level feature extraction, and a deep learning model based on BiLSTM, Attention and CNN is built to improve the classification results. Figures 6-9 show the classification results after removing one module of the model. It can be seen that when one module of WP-STC is removed, the Accuracy, Precision, Recall and F1 on the simplifyweibo\_4\_moods dataset decrease by 0.0186-0.0573, 0.0187-0.0574, 0.0188-0.0577, 0.0188-0.0575, respectively, and that on the online\_shopping\_10\_cats dataset decrease by 0.0093-0.0375, 0.0094-0.038, 0.0093-0.0377, 0.0091-0.0378, respectively. When one module of WP-STC-Center is removed, the Accuracy, Precision, Recall and F1 on the simplifyweibo\_4\_moods dataset decrease by 0.0141-0.0486, 0.014-0.0481, 0.0142-0.0488, 0.0136-0.0484, respectively, and that on the online\_shopping\_10\_cats dataset decrease by 0.013-0.0334, 0.0132-0.033, 0.0135-0.0336, 0.0133-0.0333, respectively. Specifically, the classification results of both WP-STC and WP-STC-Center decrease significantly when the word-level or Pinyin-level feature is removed, and the classification results decrease less when BiLSTM, Attention or CNN is removed. Although removing one module in WP-STC and WP-STC-Center leads to a decrease in the classification results, they still outperform most of the baseline methods compared with Tables 1 and 2. This indicates that the word-level and Pinyin-level based Chinese short text classification model and each layer of the constructed deep learning model can improve the classification results.



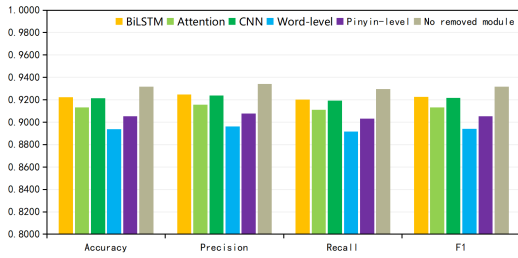


FIGURE 7. Classification results of removing modules from WP-STC on the online\_shopping\_10\_cats dataset (legend indicates the removed modules).

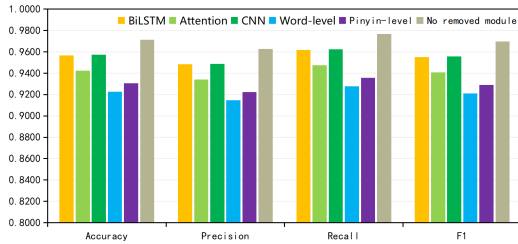


FIGURE 8. Classification results of removing modules from WP-STC-Center on the simplifyweibo\_4\_moods dataset (legend indicates the removed modules).

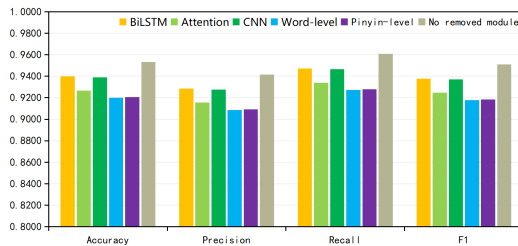


FIGURE 9. Classification results of removing modules from WP-STC-Center on the online\_shopping\_10\_cats dataset (legend indicates the removed modules).

2) COMPARISON OF DIFFERENT DISTANCE METHODS

Text Center is constructed in this paper for fast implementation of classification of new input text. Considering the simplicity and accuracy of Manhattan distance [46], the category of new input text is obtained by calculating Manhattan distance between the embedding vector of the new input text and the vectors stored in the Text Center. Since there are various methods to calculate the distance between vectors, the following experiments are conducted to demonstrate the superiority of Manhattan distance. Besides Manhattan distance, Euler distance [54], Minkowski distance [55], Chebyshev distance [56], and cosine similarity [57] are also selected, and the experimental results are shown in Tables 3 and 4, respectively. From Tables 3 and 4, it can be seen that Manhattan distance achieves the optimal Accuracy, Precision, Recall, F1 and classification time on both baseline datasets.

3) IMPORTANCE ANALYSIS OF BI-ATT-CNN

To further demonstrate the contribution of Bi-Att-CNN to text classification results, a group of comparative experiments are

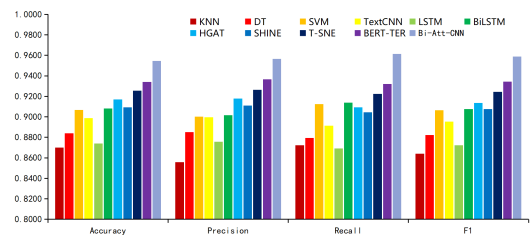


FIGURE 10. Classification results on simplifyweibo\_4\_moods dataset after replacing Bi-Att-CNN with baseline methods in WP-STC (legend indicates the methods of replacing Bi-Att-CNN).

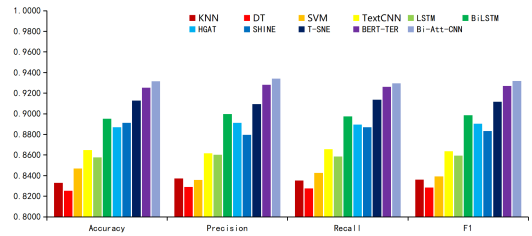


FIGURE 11. Classification results on online\_shopping\_10\_cats dataset after replacing Bi-Att-CNN with baseline methods in WP-STC (legend indicates the methods of replacing Bi-Att-CNN).

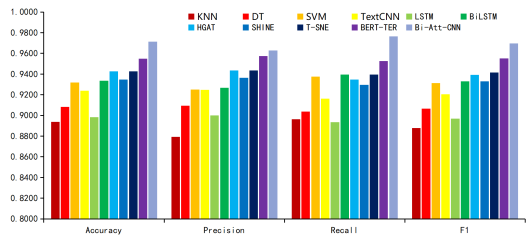


FIGURE 12. Classification results on simplifyweibo\_4\_moods dataset after replacing Bi-Att-CNN with baseline methods in WP-STC-Center (legend indicates the methods of replacing Bi-Att-CNN).

conducted. Keep other structures in WP-STC and WP-STC-Center unchanged, and replace Bi-Att-CNN with baseline methods respectively. The classification results are shown in Figures 10-13 respectively. It can be seen that when Bi-Att-CNN is replaced by baseline methods, the classification results are greatly improved compared with baseline methods, but still lower than WP-STC and WP-STC-Center using Bi-Att-CNN, which not only shows that the word-level and Pinyin-level classification models constructed in this paper greatly improve the classification results of Chinese short texts, but also shows that the deep learning model based on BiLSTM, Attention and CNN constructed in this paper outperforms all the baseline methods.

4) CLASSIFICATION RESULTS FOR IMBALANCED DATASETS

The baseline datasets selected in this paper are all balanced datasets, but the actual situation is often that the forward short texts are much larger than the reverse short texts. Therefore, the classification results on the unbalanced datasets are

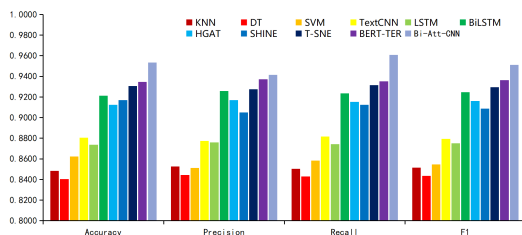


FIGURE 13. Classification results on online\_shopping\_10\_cats dataset after replacing Bi-Att-CNN with baseline methods in WP-STC-Center (legend indicates the methods of replacing Bi-Att-CNN).

TABLE 5. Classification results of WP-STC in simplifyweibo\_4\_moods dataset with different proportions of forward and reverse short texts (The best results are in bold).

Proportion	Accuracy	Precision	Recall	F1
1:1	<b>0.9545</b>	<b>0.9565</b>	<b>0.9614</b>	<b>0.9589</b>
2:1	0.9450	0.9469	0.9518	0.9493
3:1	0.9402	0.9422	0.9470	0.9446
4:1	0.9384	0.9403	0.9451	0.9427
5:1	0.9365	0.9384	0.9432	0.9408
6:1	0.9346	0.9365	0.9413	0.9389
7:1	0.9328	0.9347	0.9395	0.9371
8:1	0.9234	0.9253	0.9301	0.9277
9:1	0.9142	0.9161	0.9208	0.9184
10:1	0.9050	0.9069	0.9116	0.9092

TABLE 6. Classification results of WP-STC in online\_shopping\_10\_cats dataset with different proportions of forward and reverse short texts (The best results are in bold).

Proportion	Accuracy	Precision	Recall	F1
1:1	<b>0.9317</b>	<b>0.9341</b>	<b>0.9295</b>	<b>0.9318</b>
2:1	0.9224	0.9248	0.9202	0.9225
3:1	0.9178	0.9201	0.9156	0.9179
4:1	0.9159	0.9183	0.9138	0.9160
5:1	0.9141	0.9165	0.9119	0.9142
6:1	0.9123	0.9146	0.9101	0.9124
7:1	0.9105	0.9128	0.9083	0.9105
8:1	0.9086	0.9110	0.9065	0.9087
9:1	0.9068	0.9091	0.9047	0.9069
10:1	0.9050	0.9073	0.9029	0.9051

TABLE 7. Classification results of WP-STC-Center in simplifyweibo\_4\_moods dataset with different proportions of forward and reverse short texts (The best results are in bold).

Proportion	Accuracy	Precision	Recall	F1
1:1	<b>0.9713</b>	<b>0.9627</b>	<b>0.9765</b>	<b>0.9696</b>
2:1	0.9645	0.9560	0.9697	0.9628
3:1	0.9577	0.9493	0.9629	0.9560
4:1	0.9510	0.9426	0.9561	0.9493
5:1	0.9463	0.9379	0.9514	0.9446
6:1	0.9416	0.9332	0.9466	0.9399
7:1	0.9369	0.9286	0.9419	0.9352
8:1	0.9322	0.9239	0.9372	0.9305
9:1	0.9275	0.9193	0.9325	0.9258
10:1	0.9229	0.9147	0.9278	0.9212

tested by the following experiments. The ratios of forward and reverse short texts in both dataset are set to 1:1-10:1, respectively, and WP-STC and WP-STC-Center are used for text classification. The classification results are shown in Tables 5-8, respectively. From Tables 5-8, it can be seen that the classification results of both WP-STC and WP-STC-Center gradually decrease as the ratio of forward and reverse

TABLE 8. Classification results of WP-STC-Center in online\_shopping\_10\_cats dataset with different proportions of forward and reverse short texts (The best results are in bold).

Proportion	Accuracy	Precision	Recall	F1
1:1	<b>0.9533</b>	<b>0.9416</b>	<b>0.9608</b>	<b>0.9511</b>
2:1	0.9485	0.9397	0.9560	0.9478
3:1	0.9438	0.9378	0.9512	0.9445
4:1	0.9391	0.9360	0.9465	0.9412
5:1	0.9344	0.9341	0.9417	0.9379
6:1	0.9297	0.9322	0.9370	0.9346
7:1	0.9251	0.9304	0.9323	0.9313
8:1	0.9204	0.9285	0.9277	0.9281
9:1	0.9158	0.9266	0.9230	0.9248
10:1	0.9112	0.9248	0.9184	0.9216

short texts gradually increases. Although the classification results has decreased, they still remain at high values. Even when the ratio of forward and reverse short texts is 10:1, the classification Accuracy, Precision, Recall and F1 of WP-STC on both datasets are all greater than 0.9, and the classification Accuracy, Precision, Recall and F1 of WP-STC-Center on both datasets are all greater than 0.91. Therefore, WP-STC and WP-STC-Center are applicable to both balanced and unbalanced datasets.

### 5) ANALYSIS OF THE EFFECT OF SOLVING HOMOPHONIC TYPOS PROBLEM

WP-STC can solve the homophonic typos problem that often occurs in Chinese short texts by adding the Pinyin-level feature. The following experiments are carried out to verify the above conclusion. 1000 short text comments from Weibo<sup>10</sup> are crawled as a test set, and a word in each short text is manually and randomly selected to change into a homonym. The trained WP-STC in simplifyweibo\_4\_moods dataset with and without Pinyin-level features are used to test the classification effect. It can be seen from the experimental results that the classification Accuracy of WP-STC with Pinyin-level feature is 0.9227, and that of WP-STC without Pinyin-level feature is 0.7866. Therefore, it can be concluded that WP-STC can well solve the homophonic typos problem of Chinese short texts.

## VI. CONCLUSION

Word-level and Pinyin-level based Chinese short text classification model is constructed in this paper, and a deep learning model based on BiLSTM, Attention and CNN is proposed. In order to reduce the classification time, the concept of Text Center is innovatively proposed. Multiple experiments on two baseline datasets demonstrate that WP-STC with Text Center not only outperforms the state-of-the-art text classification model in terms of classification Accuracy, Precision, Recall and F1, but also greatly reduces the classification time.

However, since different Text Centers are constructed for different datasets, the classification results of data outside this dataset is not ideal after being input into the Text Center. Therefore, the future work of us will focus on improving the

<sup>10</sup>One of the largest social platforms in China.

robustness of the model, so that the Text Center does not rely on the construction of a single dataset.

## REFERENCES

- [1] K. R. Chowdhary, "Natural language processing," in *Fundamentals of Artificial Intelligence*. 2020, pp. 603–649.
- [2] M. Khadhraoui, H. Bellaaj, M. B. Ammar, H. Hamam, and M. Jmaiel, "Survey of BERT-base models for scientific text classification: COVID-19 case study," *Appl. Sci.*, vol. 12, no. 6, p. 2891, Mar. 2022.
- [3] A. Gupta and R. Bhatia, "Ensemble approach for web page classification," *Multimedia Tools Appl.*, pp. 25219–25240, Apr. 2021.
- [4] Y. Choi, J. Lee, and Y. Choe, "Online travel information filtering: Role of commercial cues in trust and distrust mechanisms," *J. Travel Tourism Marketing*, vol. 38, no. 7, pp. 710–724, Sep. 2021.
- [5] J. Zhang, S. Gao, J. Xia, G. Li, X. Qi, and B. Gao, "Distributed adaptive cubature information filtering for bounded noise system in wireless sensor networks," *Int. J. Robust Nonlinear Control*, vol. 31, no. 10, pp. 4869–4896, Jul. 2021.
- [6] M. Frueh, A. Schilling, S. Gatidis, and T. Kuestner, "Real time landmark detection for within- and cross subject tracking with minimal human supervision," *IEEE Access*, vol. 10, pp. 81192–81202, 2022.
- [7] Z. Gu, T. Yin, and Z. Ding, "Path tracking control of autonomous vehicles subject to deception attacks via a learning-based event-triggered mechanism," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5644–5653, Dec. 2021.
- [8] M. Hina, M. Ali, A. R. Javed, F. Ghabban, L. A. Khan, and Z. Jalil, "SeFACED: Semantic-based forensic analysis and classification of E-mail data using deep learning," *IEEE Access*, vol. 9, pp. 98398–98411, 2021.
- [9] D. E. Salhi, A. Tari, and M. T. Kechadi, "Email classification for forensic analysis by information gain technique," *Int. J. Softw. Sci. Comput. Intell.*, vol. 13, no. 4, pp. 40–53, Oct. 2021.
- [10] S. Gan, S. Shao, L. Chen, L. Yu, and L. Jiang, "Adapting hidden naive Bayes for text classification," *Mathematics*, vol. 9, no. 19, p. 2378, Sep. 2021.
- [11] N. G. Ramadhan, "Indonesian online news topics classification using Word2Vec and K-nearest neighbor," *RESTI J. Syst. Eng. Inf. Technol.*, vol. 5, pp. 1083–1089, Dec. 2021.
- [12] N. Yuvaraj, V. Chang, B. Gobinathan, A. Pinagapani, S. Kannan, G. Dhiman, and A. R. Rajan, "Automatic detection of cyberbullying using multi-feature based artificial intelligence with deep decision tree classification," *Comput. Elect. Eng.*, vol. 92, Jun. 2021, Art. no. 107186.
- [13] X. Luo, "Efficient English text classification using selected machine learning techniques," *Alexandria Eng. J.*, vol. 60, no. 3, pp. 3401–3409, Jun. 2021.
- [14] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, *arXiv:1511.08458*.
- [15] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, and L. He, "A survey on text classification: From shallow to deep learning," 2020, *arXiv:2008.00364*.
- [16] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Aug. 2017, pp. 1–6.
- [17] M. Fan, Z. Si, X. Xie, Y. Liu, and T. Liu, "Text backdoor detection using an interpretable RNN abstract model," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4117–4132, 2021.
- [18] J. R. Brennan, C. Dyer, A. Kuncoro, and J. T. Hale, "Localizing syntactic predictions using recurrent neural network grammars," *Neuropsychologia*, vol. 146, Sep. 2020, Art. no. 107479.
- [19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [20] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Bidirectional long short-term memory networks for relation classification," in *Proc. 29th Pacific Asia Conf. Lang., Inf. Comput.*, 2016, pp. 73–78.
- [21] M. Hao, B. Xu, J. Y. Liang, B. W. Zhang, and X. C. Yin, "Chinese short text classification with mutual-attention convolutional neural networks," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 19, no. 5, pp. 1–13, 2020.
- [22] B. Lyu, L. Chen, S. Zhu, and Y. Kai, "LET: Linguistic knowledge enhanced graph transformer for Chinese short text matching," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 13498–13506.
- [23] Y. Feng and Y. Cheng, "Short text sentiment analysis based on multi-channel CNN with multi-head attention mechanism," *IEEE Access*, vol. 9, pp. 19854–19863, 2021.
- [24] M. Alessandro and R. Basili, "Complex linguistic features for text classification: A comprehensive study," in *Proc. Eur. Conf. Inf. Retr.*, 2004, pp. 181–196.
- [25] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1181–1193, May 2019.
- [26] E. Richard and B. Reddy, "Text classification for clinical trial operations: Evaluation and comparison of natural language processing techniques," *Therapeutic Innov. Regulatory Sci.*, vol. 55, no. 2, pp. 447–453, Mar. 2021.
- [27] A. Occhipinti, L. Rogers, and C. Angione, "A pipeline and comparative study of 12 machine learning models for text classification," *Exp. Syst. Appl.*, vol. 201, Sep. 2022, Art. no. 117193.
- [28] H. I. Abdalla and A. A. Amer, "On the integration of similarity measures with machine learning models to enhance text classification performance," *Inf. Sci.*, vol. 614, pp. 263–288, Oct. 2022.
- [29] Y. Cheng, H. Zou, H. Sun, H. Chen, Y. Cai, M. Li, and Q. Du, "HSAN-capsule: A novel text classification model," *Neurocomputing*, vol. 489, pp. 521–533, Jun. 2022.
- [30] J. Kong, J. Wang, and X. Zhang, "Hierarchical BERT with an adaptive fine-tuning strategy for document classification," *Knowl.-Based Syst.*, vol. 238, Feb. 2022, Art. no. 107872.
- [31] G. C. Sergio and M. Lee, "Stacked DeBERT: All attention in incomplete data for text classification," *Neural Netw.*, vol. 136, pp. 87–96, Apr. 2021.
- [32] M. Liu, L. Liu, J. Cao, and Q. Du, "Co-attention network with label embedding for text classification," *Neurocomputing*, vol. 471, pp. 61–69, Jan. 2022.
- [33] M. Zhang, "Applications of deep learning in news text classification," *Sci. Program.*, vol. 2021, pp. 1–9, Aug. 2021.
- [34] S. Gao, M. Alawad, M. T. Young, J. Gounley, N. Schaefferkoetter, H. J. Yoon, X.-C. Wu, E. B. Durbin, J. Doherty, A. Stroup, L. Coyle, and G. Tourassi, "Limitations of transformers on clinical text classification," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 9, pp. 3596–3607, Sep. 2021.
- [35] T. Yang, L. Hu, C. Shi, H. Ji, X. Li, and L. Nie, "HGAT: Heterogeneous graph attention networks for semi-supervised short text classification," *ACM Trans. Inf. Syst.*, vol. 39, no. 3, pp. 1–29, Jul. 2021.
- [36] Y. Wang, S. Wang, Q. Yao, and D. Dou, "Hierarchical heterogeneous graph representation learning for short text classification," 2021, *arXiv:2111.00180*.
- [37] H. Wang, K. Tian, Z. Wu, and L. Wang, "A short text classification method based on convolutional neural network and semantic extension," *Int. J. Comput. Intell. Syst.*, vol. 14, no. 1, pp. 367–375, 2021.
- [38] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, Mar. 2021.
- [39] B. Škrlić, M. Martinc, J. Kralj, N. Lavrač, and S. Pollak, "Tax2vec: Constructing interpretable features from taxonomies for short text classification," *Comput. Speech Lang.*, vol. 65, Jan. 2021, Art. no. 101104.
- [40] Y. Zhou, J. Li, J. Chi, W. Tang, and Y. Zheng, "Set-CNN: A text convolutional neural network based on semantic extension for short text classification," *Knowl.-Based Syst.*, vol. 257, Dec. 2022, Art. no. 109948.
- [41] S. Yu, D. Liu, Y. Zhang, S. Zhao, and W. Wang, "DPTCN: A novel deep CNN model for short text classification," *J. Intell. Fuzzy Syst.*, vol. 41, no. 6, pp. 7093–7100, 2021.
- [42] S. Yang and Y. Liu, "A character-word graph attention networks for Chinese text classification," in *Proc. IEEE Int. Conf. Big Knowl. (ICBK)*, Dec. 2021, pp. 462–469.
- [43] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014, *arXiv:1409.2329*.
- [44] S. Li, Z. Zhao, R. Hu, W. Li, T. Liu, and X. Du, "Analogical reasoning on Chinese morphological and semantic relations," 2018, *arXiv:1805.06504*.
- [45] T. Mikolov, W.-T. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proc. NAACL-HLT*, 2013, pp. 746–751.
- [46] Y. Sun, S. Li, Y. Wang, and X. Wang, "Fault diagnosis of rolling bearing based on empirical mode decomposition and improved Manhattan distance in symmetrized dot pattern image," *Mech. Syst. Signal Process.*, vol. 159, Oct. 2021, Art. no. 107817.
- [47] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.
- [48] D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man Cybern.*, vol. 21, no. 3, pp. 660–674, May 1991.

- [49] W. S. Noble, "What is a support vector machine?" *Nature Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, 2006.
- [50] D. Alsaleh and S. Larabi-Marie-Sainte, "Arabic text classification using convolutional neural network and genetic algorithms," *IEEE Access*, vol. 9, pp. 91670–91685, 2021.
- [51] Z. Li, H. Xie, G. Cheng, and Q. Li, "Word-level emotion distribution with two schemas for short text emotion classification," *Knowl.-Based Syst.*, vol. 227, Sep. 2021, Art. no. 107163.
- [52] P. Kumar and B. Raman, "A BERT based dual-channel explainable text emotion recognition system," *Neural Netw.*, vol. 150, pp. 392–407, Jun. 2022.
- [53] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation," in *Proc. Australas. Joint Conf. Artif. Intell.*, 2006, pp. 1015–1021.
- [54] C. Remlinger, J. Mikael, and R. Elie, "Conditional loss and deep Euler scheme for time series generation," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 8098–8105.
- [55] A. Sundararajan and B. Ollis, "Regression and generalized additive model to enhance the performance of photovoltaic power ensemble predictors," *IEEE Access*, vol. 9, pp. 111899–111914, 2021.
- [56] Y. Sun, S. Li, and X. Wang, "Bearing fault diagnosis based on EMD and improved Chebyshev distance in SDP image," *Measurement*, vol. 176, May 2021, Art. no. 109100.
- [57] J. Chen, Z. Guo, and J. Hu, "Ring-regularized cosine similarity learning for fine-grained face verification," *Pattern Recognit. Lett.*, vol. 148, pp. 68–74, Aug. 2021.



**XINJIE SUN** received the M.E. degree from the School of Computer Science and Engineering, Northeastern University, Shenyang, China, in 2013. He is currently a Teacher with Liupanshui Normal University. His research interests include next generation internet and artificial intelligence.



**XINGYING HUO** received the Ph.D. degree in electrical science and technology from Beijing Jiaotong University, Beijing, China, in 2018. From 2016 to 2017, she was a Visiting Student with Houston University, TX, USA. In 2018, she joined as the Faculty Member of the Department of Computer Science, Liupanshui Normal University, where she became an Associate Professor, in 2019. Her current research interests include leaky-wave structures and radio-wave propagation.

...