IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 7, 2022, accepted March 22, 2022, date of publication March 28, 2022, date of current version April 1, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3162614

A Long-Text Classification Method of Chinese News Based on BERT and CNN

XINYING CHEN, PEIMIN CONG, AND SHUO LV
School of Computer and Communication Engineering, Dalian Jiaotong University, Dalian 116028, China
Corresponding author: Xinying Chen (chenxy1979@163.com)

This work was supported by the Science and Technology Innovation Fund Program of Dalian under Grant 2021JJ13SN81.

ABSTRACT Text Classification is an important research area in natural language processing (NLP) that has received a considerable amount of scholarly attention in recent years. However, real Chinese online news is characterized by long text, a large amount of information and complex structure, which also reduces the accuracy of Chinese long text classification as a result. To improve the accuracy of long text classification of Chinese news, we propose a BERT-based local feature convolutional network (LFCN) model including four novel modules. First, to address the limitation of Bidirectional Encoder Representations from Transformers (BERT) on the length of the max input sequence, we propose a named Dynamic LEAD-n (DLn) method to extract short texts within the long text based on the traditional LEAD digest algorithm. In Text-Text Encoder (TTE) module, we use BERT pretrained language model to complete the sentence-level feature vector representation of a news text and to capture global features by using the attention mechanism to identify correlated words in text. After that, we propose a CNN-based local feature convolution (LFC) module to capture local features in text, such as key phrases. Finally, the feature vectors generated by the different operations over several different periods are fused and used to predict the category of a news text. Experimental results show that the new method further improves the accuracy of long text classification of Chinese news.

INDEX TERMS Bidirectional encoder representations from transformers, Chinese long text classification, convolutional neural network, natural language processing, representation learning.

I. INTRODUCTION

Text Classification (TC) is a key task in Natural Language Processing (NLP), which aims to assign predefined labels or classes to texts. And news categorization plays a key role in the TC. It refers to building a model that obtains the topic of a text. So, it is an indispensable part in application fields such as information retrieval and recommendation systems. Specially, this paper resolves the problem of news categorization for Chinese long texts.

In the past few years, deep learning methods have made great progress in NLP areas such as sentiment analysis [1], news classification [2], and machine translation [3]. On one hand, Convolutional Neural Networks (CNNs) work well in the case of detecting local and position invariant patterns (e.g., VGG [4], ResNets [5]). Kalchbrenner *et al.* [6] and Kim [7] applied CNNs to the TC task. It is used to capture local salient features such as keywords in text and to complete the TC task. However, CNNs may fail to capture the global features. For example, the word ''apple'' sometimes denotes a fruit, but sometimes a brand of product which needs the correlated words (i.e., contextual information) to make an accurate judgment. Therefore, Hughes *et al.* [8], Liu *et al.* [9], and Zhang *et al.* [10] proposed some novel neural network models based on CNN and associated with other techniques. On the other hand, Pre-trained Language Models (PLMs), represented by Bidirectional Encoder Representations from Transformers (BERT) [11], have achieved good results for short TC tasks. It benefits from the attention mechanism, which is effective to capture global features of sentences or documents by identifying correlated words in text. Soon after, a few variant models of BERT were generated, such as RoBERTa [12], ALBERT [13], DistillBERT [14], and SpanBERT [15]. Especially, Cui *et al.* [16] introduced several PLMs for Chinese in conjunction with the Whole Word Masking (WWM) mechanism. However, the BERT model only allows input of sequences with less than 512 tokens.

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

Many efforts have been made to address the limitation of BERT on the length of the max input sequence (i.e., 512). Sun *et al.* [17] used a truncation method that only retained the first few tokens in a long text. Pappagari *et al.* [18] and Jin *et al.* [19] split long texts into multiple segments, which are fed into a model. However, all the above approaches have a potential to undermine the semantic integrity of a sentence. Akhter *et al.* [20] used a single-layer multisize filters convolutional neural network (SMFCNN) to study document classification. However, it is difficult for a single model to learn both local and global text features. Jin *et al.* [19] combined CNN and LSTM for Chinese long text classification based on BERT. However, they stacked the models simply. As a text feature vector output by BERT is not able to express the most original text features. Therefore, the original text representation should be fed to a model such as CNN to extract features when combining other models with the BERT models.

In summary, there are two main challenges in the task of categorizing long text of Chinese news. Generally, it is relatively long that the length of texts to be classified. It is not less than 1000 tokens and can even be up to tens of thousands of tokens. Consequently, it also directly affects the accuracy of long TC tasks. The another challenge is that the local and global features are often confused in long text of Chinese news. That is, the models are expected to have the ability to recognize local and global features. In summary, the challenges still make Chinese long text classification problem difficult. At the same time, in this paper, we carefully analyze the real Chinese news text data. We find that the meanings expressed by sentences at the beginning and end of Chinese news texts are closer to the topic. The traditional LEAD [21] digest algorithm is a good choice, which is a simple extractive summarization. However, the LEAD requires manually setting the number of extracted sentences, and the length of each sentence is variable. Therefore, we improve the LEAD summarization algorithm to extract more relatively important sentences at the beginning and end of the document.

Based on the above, we propose a novel model called Local Feature Convolutional Network (LFCN) model to improve the performance of Chinese long classification to address the two challenges. First, in this paper, we improve the LEAD to extract as many sentences as possible on a maximum length of 256, which is called Dynamic LEAD-n (DLn). In this way, it overcomes the limitation of BERT on the length of max input sequence. Likewise, it reduces the time and resources of training by setting the max input sequence length to 256. On the basis, this paper utilizes BERT to complete the sentence-level feature vector representation of news texts, and learn their contextual representations (i.e., global features of sentences and documents). Then, we use convolution operation to capture local salient features such as key phrases in text. Finally, the text feature vectors generated by the different operations for several different periods are selected in LCFN. They are fused as the final representation of the news texts, and used for the TC task.

The main contributions of this paper are as follows:

1) This paper proposes a LEAD-based extractive summarization algorithm called DLn. It is a simple and efficient way to extract long Chinese texts.

2) This paper proposes a LFCN architecture based on BERT and CNN aiming at the long text classification problem of Chinese news. It can capture local and global text features.

3) In this paper, we build a more time-sensitive Chinese news TC dataset, called MCNews. The MCNews dataset was obtained from major internet news and official media via the web in May 2021.

4) Extensive experiments have been conducted on the THUCNews and MCNews datasets. The experimental results show that the new method is reasonable and effective.

II. RELATED WORK

Professor Hans first introduced the concept of TC in 1958 and applied the ideas of probability statistics to the TC tasks [22]. As technology continues to evolve, deep learning methods are becoming dominant in a variety of TC tasks. As the best embodiment of representation learning (e.g., [23], [24]), deep learning can automatically construct features and solve the problem of artificial feature construction. Representation learning is the process of learning a parametric mapping from the original input data field to a feature vector or tensor. It is hoped that more abstract and useful concepts will be captured and extracted to improve performance on a range of downstream tasks [25]. In NLP, the applications of representation learning mainly include unsupervised/supervised pretraining of text, distributed representation, and transfer learning. For example, representatives of unsupervised learning and distributed representations are greedy layer-wise unsupervised pretraining [26] and word embeddings [27], respectively. Besides, there are some other representation learning methods, such as joint versus independent multiview hashing for cross-view retrieval, which is used for data with an increasing number or a large number of views. In this paper, most of the study is based on the BERT PLMs and CNNs. Therefore, this paper briefly reviews the two related approaches.

A. CNN-BASED MODELS

CNNs are good at learning local and location-invariant features [28], such as "Taylor Swift", "Metaverse" and other topic-related keywords. CNN-based TC model was earlier proposed by Kim [7]. They built a simple CNN model based on word2vec [29]. And CNNs with only one layer of convolution performed well like this.

In recent years, researchers have proposed numerous novel CNN-based models. Wang *et al.* [30] proposed a CNN-based framework that combines explicit and implicit representations of short texts for classification. Trusca and Spanakis [31] proposed a Hybrid Tiled Convolutional Neural Network (HTCNN) model that applies filters only to words that appear in similar contexts and their adjacent words. The study by Alam *et al.* [32] discussed the CNN approach to short TC by given a short text. The model generated a textual

representation by using words and entities. Yan *et al.* [33] proposed a CNN-based model to improve the accuracy of TC tasks. Zhao *et al.* [34] developed a method for interpreting CNNs to solve TC problems. As a result, CNN has become one of the favoured architectures for TC tasks.

B. PRE-TRAINED LANGUAGE MODELS

Recently, a large body of work has shown that PLMs on large corpora can learn generic language representations, which facilitates NLP downstream tasks and avoids the need to train new models from scratch [35]. For specific NLP tasks, only an additional output layer is added on top of the PLMs and then simple fine-tuning is performed.

PLMs can be divided into two categories, autoregressive and autoencoding PLMs [28]. One of the earliest autoregressive PLMs was GPT, released by the OpenAI team [36]. It is a unidirectional model that predicts a text sequence word by word from left to right (or right to left), meaning that the prediction of each word depends on the previous prediction. The OpenAI team has released several new versions of GPT (e.g., GPT-2 [37], GPT-3 [38]). There are also a few other autoregressive PLMs, such as ELMO [39] and XLNet [40].

One of the most widely used autoencoding PLMs is Google's BERT [11] based on bi-directional Transformer [41]. Unlike GPT, BERT uses an unsupervised Masked Language Model (MLM) task for training. This unsupervised task masks several tokens at random and then predicts those tokens that are masked. In the same year, Facebook changed BERT's design selection and training strategy and released an improved version of BERT, RoBERTa [12]. Subsequently, Cui *et al.* [16] released a series of Chinese BERT PLMs, such as BERT-wwm, BERT-wwm-ext, and RoBERTa-wwm-ext, in conjunction with Google's subsequent WWM mechanism. Sun *et al.* [42] proposed a Chinese PLM incorporating glyph and pinyin information, namely ChineseBERT. ChineseBERT achieves state-of-the-art performance on several Chinese NLP tasks.

III. METHOD

In order to solve the problem of news classification for Chinese long text, this paper proposes a LFCN model, whose network architecture is shown in Figure 1. First, in this paper, we use the DLn digest algorithm to extract short text pairs in long texts. Next, in the Text-Text Encoder (TTE) layer, we use the BERT embedding method to convert the texts into input vectors. And we use the multilayer bidirectional Transformer feature extractor of BERT to capture global expressions of feature relationships between words. In the Local Feature Convolutional (LFC) layer, we use convolution operation to extract local salient features such as key phrases contained in texts. Then, the multiple feature vectors of text output from the above two layers are fused. And the result is used as the final feature vector representation of a text. Finally, there is a classifier layer, where we use a single layer feedforward network to predict the category of a news text from the feature vector. The details of the implementation of LFCN and its rationale are described in the subsequent subsections of this article.

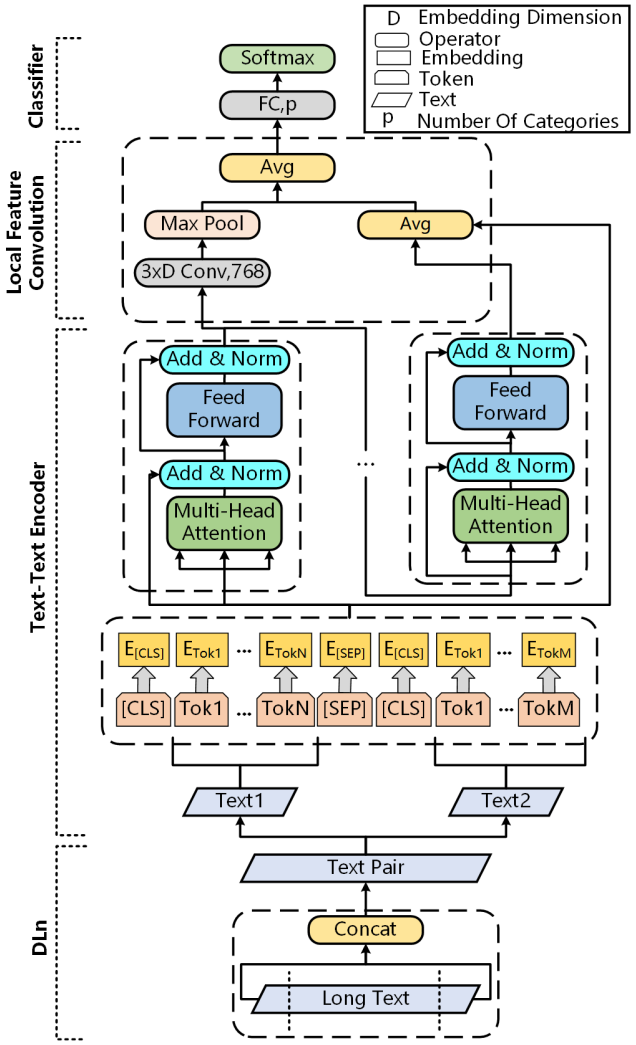

FIGURE 1. Architecture of local feature convolution network (LFCN).

A. PROBLEM DEFINITION

In the problem of Chinese news long text classification, the dataset is denoted as a set $\boldsymbol{D} = \{(t_1, c_1), (t_2, c_2), \cdots, (t_i, c_i), \cdots, (t_n, c_n)\}$, where n is the size of the dataset, t_i represents a news text, and c_i represents a category of the news text. Each piece of data (t_i, c_i) is a text and its label. A text t_i is denoted as a sequence of token (i.e., $t_i = \{x_1, x_2, \cdots, x_i, \cdots, x_{l_i}\}$), where x_i is the ith token in the text, l_i is the length of the ith text, and its size varies with the length of text. The categories of news texts include "Sports", "Business", "House", "Health", etc. The goal of news text classification is to assign preset category labels to each text.

B. PROBLEM ANALYSIS

Online news text data has the following characteristics: long text length, diversity of categories, text features between different categories are easily confused [43]. Text features can

be grouped into two categories, local and global features. One of the most representative local features is key phrases. They could be phrases that express specific topics like "NBA". In Chinese texts, there are often some ambiguities that require more correlated words to be recognized, which are called global features. It shows that the major difference between local and global features is that the model requires a different number of relevant words (i.e., different receptive field sizes) in order to capture them. Texts often contain many local and global features, which brings a new challenge to recognize them.

C. DATA PREPROCESSING

There are few existing long text datasets of Chinese news with low timeliness. To address this, a Chinese news long text classification dataset, called MCNews, is constructed in this paper. Due to the large number of useless characters in the obtained online news text data, we perform a series of preprocessing operations on the native text data. Data preprocessing includes the collection, cleaning, and segmentation of news text data. In addition, we deal with the long news texts.

1) COLLECTION OF NEWS TEXT DATA

The MCNews dataset was obtained for this paper in May 2021 from major Internet news (Sina, Netease, Thepaper.cn, etc.) and official media (Xinhua, People's Daily, China News, etc.) via the Internet. The dataset includes about 15 200 news text data from nine categories, including "Business", "House", "Education", "Technology", "Sports", "Game", "Entertainment", "Military", and "Auto". The data distribution for each category and data sources in the MCNews dataset are shown in Figure 2 and Figure 3, respectively.

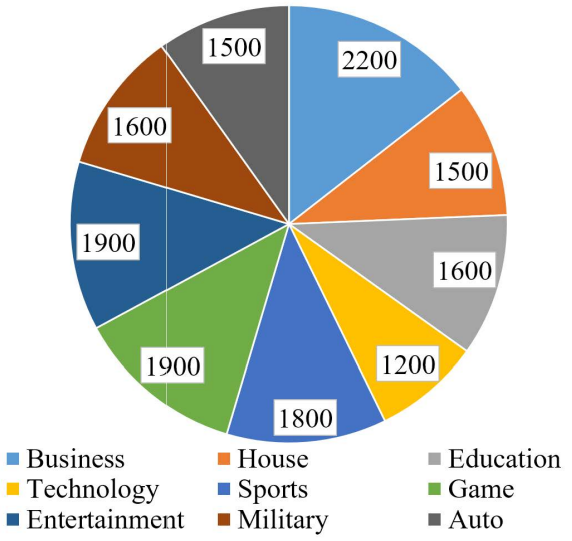

FIGURE 2. Data distribution of each category in the MCNews dataset.

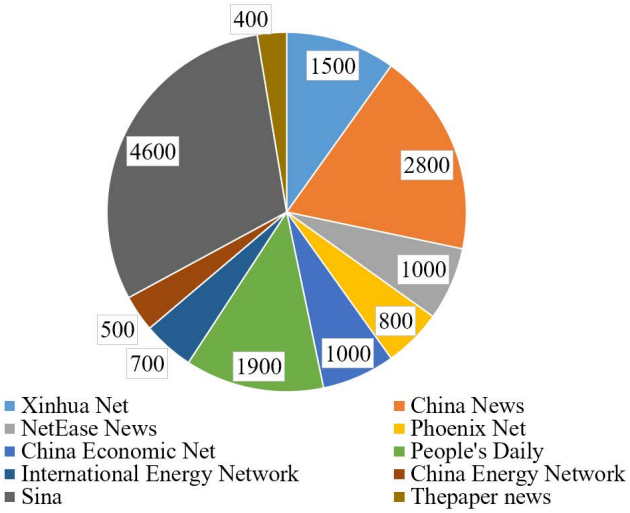

FIGURE 3. Distribution of data sources in the MCNews dataset.

2) CLEANING AND SEGMANTATION OF MCNEWS DATASET

Data cleaning mainly includes removing useless characters and incomplete text data. Specifically, we remove blank characters, links, and other useless characters. Finally, the data from each category is divided into training and validation sets in a 7:3 ratio.

3) DEALING WITH THE LONG TEXTS

To address the limitation of BERT on the length of the max input sequence and to reduce computational cost, this paper proposes a simple and efficient digest algorithm based on LEAD, called DLn. For example, LEAD-3 means to distill the first three sentences [21]. As shown in the "DLn" module of Figure 1. The DLn digest algorithm is processed in detail as follows.

Step 1: Determine the sentence separators. The presence of a separator means the end of a complete sentence. And the sentence separators used in this paper include: ':', ';', '!', '?', '。', '，' ':', ';', '，' '?', '，', '!', and '，' '.'

Step 2: Extract the short texts. In this paper, we take short texts of equal length from the beginning and end of a long text, respectively. In particular, we set the max input sequence length of the model to 256. Thus, after removing the special classification symbol [CLS] and the separator [SEP], the short text in the above contains 127 words.

Step 3: Delete the incomplete sentences in the short texts. As the length of each sentence is not fixed, the last sentence or the first sentence in an extracted short text may be incomplete. In order to ensure the semantic integrity of each sentence, we have removed incomplete sentences. The DLn digest algorithm pseudo-code is shown in Algorithm 1. A visualization of the DLn digest algorithm is shown in Figure 4.

D. TEXT-TEXT ENCODER LAYER

BERT [11] is a popular pretrained language model. It consists of two unsupervised tasks in the pretraining phase, namely

Algorithm 1 DLn

Input: Original long text: original_text
Output: Short text: text_pair

```
Define an array of sentence separator marks = [: ; ! ? ∘ : ; ? ! , ,]
Calculate the length of the original long text: len
if len > 254 then
    Extract the first 127 tokens text1 and the last 127 tokens text2 in long text
    Get the last token t1 of text1 and the first token t2 of text2
    if t1 not in marks then
        Delete the incomplete sentences at the end of text1 to get a new text1
    end if
    if t2 not in marks then
        Delete the incomplete sentences at the top of text2 to get a new text2
    end if
end if
Encapsulate return values text_pair = text1, text2
return text_pair
```

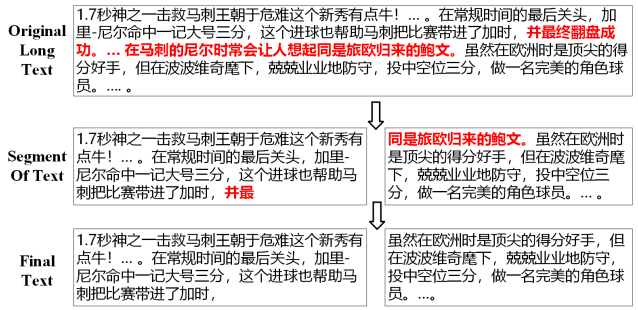

FIGURE 4. An example of the DLn summary algorithm.

MLM and Next Sentence Prediction (NSP). In this way, it can learn a more comprehensive contextual representation by identifying correlated words in sentences or documents. The input embeddings for BERT are the sum of the token, segment, and position embeddings. It uses position embedding to encode the token sequences without a need for recurrent architecture, and it uses segment embedding to differentiate the input sentences [44].

In this paper, the TTE module uses BERT to encode both Text T1 and Text T2. As shown in the "Text-Text Encoder" module in Figure 1. Some scholars (e.g., Sun *et al.* [17]) do not distinguish between short text pairs taken at the beginning and end of long texts, treating them as a "single sentence" classification task when using the BERT embedding method. A simple visual example is shown in Figure 5.

In this paper, we differentiate the generated short text pairs and treat them as a "sentence pair" classification task. Specifically, we propose to encapsulate short text pairs together into a single token sequence

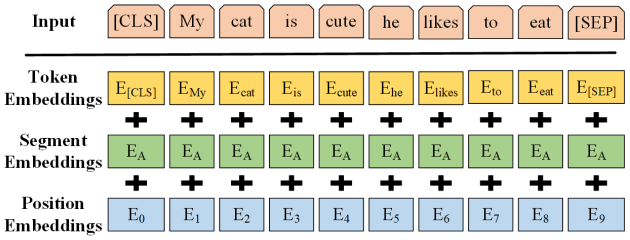

FIGURE 5. BERT input representation for the "single sentence" classification task.

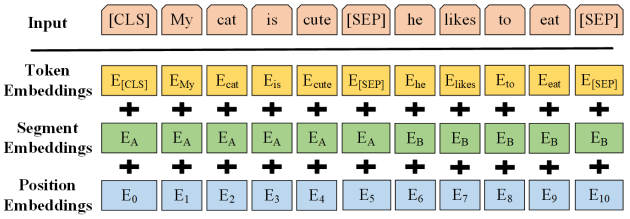

FIGURE 6. BERT input representation for the "sentence pair" classification task.

$([CLS], t1_1, t1_2, \cdots, t1_i, \cdots, t1_m, [SEP], t2_1, t2_2, \cdots, t2_i, \cdots, t2_n, [SEP])$, where $[CLS]$ represents the special classification marker, $[SEP]$ represents the special segmented paragraph marker, $t1_i$ and $t2_i$ represent the ith token of the corresponding text. A visual example is shown in Figure 6. The input embeddings are then constructed as in BERT [11].

Formally, the input is defined as $\boldsymbol{X} = (x_1, x_2, \cdots, x_i, \cdots, x_l)$, where $x_i \in \mathbb{R}^{1\times d}$ is the embedding constructed by summing the ith token through the corresponding token, segment, and position embeddings, d is the maximum embedding dimension of the hidden layer, and l is the length of max input sequence. In layer j, a short text pair embedding is denoted as $\boldsymbol{E}^{(j)} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_i, \cdots, \boldsymbol{e}_l)$, where $\boldsymbol{e}_i \in \mathbb{R}^{1\times d}$ coincides with the corresponding x_i dimension. $\boldsymbol{E}^{(j)}$ is obtained by (1)-(8).

$$\boldsymbol{Z} = \boldsymbol{E}^{j-1} \tag{1}$$

$$\boldsymbol{H}_i(\boldsymbol{Z}) = \text{softmax}\left(\frac{\left(\boldsymbol{Z}\boldsymbol{W}_i^Q\right)\left(\boldsymbol{Z}\boldsymbol{W}_i^K\right)^T}{\sqrt{\left(\frac{d}{h}\right)}}\right)\left(\boldsymbol{Z}\boldsymbol{W}_i^V\right) \tag{2}$$

$$\text{MultiHead}(\boldsymbol{Z}) = \text{Concat}\left(\boldsymbol{H}_1, \ldots, \boldsymbol{H}_i, \ldots, \boldsymbol{H}_h\right)\boldsymbol{W}^o \tag{3}$$

$$LN\left(x_i\right) = \alpha * \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \tag{4}$$

$$\boldsymbol{S} = LN(\boldsymbol{Z} + MultiHead(\boldsymbol{Z})) \tag{5}$$

$$\text{ReLU}(x) = \begin{cases} x, & x > 0 \\ 0, & x \le 0 \end{cases} \tag{6}$$

$$FCFN(\boldsymbol{S}) = \text{ReLU}\left(\boldsymbol{S}\boldsymbol{W}_1 + \boldsymbol{b}_1\right)\boldsymbol{W}_2 + \boldsymbol{b}_2 \tag{7}$$

$$\boldsymbol{E}^{(j)} = LN(\boldsymbol{S} + FCFN(\boldsymbol{S})) \tag{8}$$

where $\left\{\boldsymbol{W}_i^Q, \boldsymbol{W}_i^K, \boldsymbol{W}_i^V\right\} \in \mathbb{R}^{\mathrm{d}\times\frac{\mathrm{d}}{\mathrm{h}}}$, $\boldsymbol{W}^O \in \mathbb{R}^{\mathrm{d}\times\mathrm{d}}$, $\boldsymbol{W}_1 \in \mathbb{R}^{\mathrm{d}\times\mathrm{h\_s}}$, $\boldsymbol{W}_2 \in \mathbb{R}^{\mathrm{h\_s}\times\mathrm{d}}$, $\boldsymbol{b}_1 \in \mathbb{R}^{1\times\mathrm{h\_s}}$ and $\boldsymbol{b}_2 \in \mathbb{R}^{1\times d}$ are

learnable parameter matrices. h is the number of attention heads, h_s is the hidden layer size in a fully connected feed-forward network (FCFN), LN represents layer normalization, x_i denotes the ith sample data of the input, μ and σ^2 denote the mean and variance of the feature vectors of each sample data, respectively, ε denotes a decimal (added to the variance to avoid dividing by zero), α and β denote the scaling and translation parameters, respectively. In particular, $\boldsymbol{E}^{(0)} = \boldsymbol{X}$. Throughout the model, 12 layers like this are stacked.

In the TTE layer, the attention mechanism is effective to identify the connections between words in the text. In a nutshell, attention in language models can be interpreted as a vector of importance weights [28]. In this way, the model can better capture the global text features.

E. LOCAL FEATURE CONVOLUTION LAYER

After the text vector has passed through the TTE layer, the model learns a large number of global features by identifying the correlation between words or tokens in text. For the model to capture more local and global features in text, this paper adopts the idea proposed by Kim [7] to capture local salient features such as keywords in the sequence, i.e., using multiple filters for one-dimensional convolutional operation. For example, when the convolutional kernel size (local receptive field) is 3, it may be interpreted as follows: the model learns the degree of association of a word with the preceding and following words through convolutional operations. When fusing BERT with CNN models, some scholars [19], [44] directly or hardly directly input feature vectors from the BERT output to the CNNs, which can destroy some original features in text. Therefore, in the TTE module, the output feature vector of the first layer is fed into the LFC layer and used to capture local salient features such as key phrases in text, while preserving the original features of the text as far as possible. As shown in the "Local Feature Convolution" module in Figure 1.

Formally, assume that $\boldsymbol{e}_i \in \mathbb{R}^{1\times d}$ is the d-dimensional input embedding of the ith token in the corresponding TTE layer input sequence, and that a sequence of length l is represented as a sequence vector $\boldsymbol{E}_{1:l} \in \mathbb{R}^{l\times d}$, as shown in (9).

$$\boldsymbol{E}_{1:l} = \boldsymbol{e}_1 \oplus \boldsymbol{e}_2 \oplus \cdots \oplus \boldsymbol{e}_i \oplus \cdots \oplus \boldsymbol{e}_l \tag{9}$$

where $\oplus$ is the concatenation operator. In general, $\boldsymbol{E}_{i:i+j}$ denotes the consecutive tokens $e_i, e_{i+1}, \cdots, e_{i+j}$. A one-dimensional convolution operation involves a filter $\boldsymbol{W} \in \mathbb{R}^{m\times d}$, which is applied to a window of m tokens to produce a new feature. For example, the feature $f_i \in \mathbb{R}$ is generated from the ith to the $(i+m-1)$th token, as shown in (10).

$$f_i = f\left(\boldsymbol{W} \cdot \boldsymbol{E}_{i:i+m-1} + b\right) \tag{10}$$

where f is a non-linear function (e.g., ReLU, Tanh, etc.), and $b \in \mathbb{R}$ is a bias term. This filter is applied to each possible token window in the sequence $\{\boldsymbol{E}_{1:m}, \boldsymbol{E}_{2:m+1}, \cdots, \boldsymbol{E}_{l-m+1:l}\}$ to generate a feature map $\boldsymbol{F} \in \mathbb{R}^{l-m+1}$, as shown in (11).

$$\boldsymbol{F} = [f_1, f_2, \cdots, f_{l-m+1}] \tag{11}$$

We then apply max-over-time pooling to the feature map $\boldsymbol{F}$ and take the maximum value $\hat{f} = \max\{\boldsymbol{F}\}$ as the corresponding feature of a particular filter. Finally, each filter produces one salient feature, which is concatenated to generate a high-level feature vector $\hat{\boldsymbol{F}} \in \mathbb{R}^y$, as shown in (12).

$$\hat{\boldsymbol{F}} = \left[\hat{f}_1, \hat{f}_2, \cdots, \hat{f}_y\right] \tag{12}$$

where y is the number of filters. At the same time, we use the feature vector $\hat{\boldsymbol{F}}$ as the output feature vector of the LFC module and use it for the final TC task.

Next, in order to fuse feature vectors that express local and global text features, this paper proposes a way of fusing textual feature vectors (shown as "Avg" in Figure 1). Li *et al.* [45] found that using the output feature vectors from the first and twelfth layers in the TTE (i.e., $\boldsymbol{E}^{(1)}$ and $\boldsymbol{E}^{(12)}$) for the classification task helped to improve the TC performance. Considering the specificity of the LFCN model in this paper, based on it, we then use the feature vector $\hat{\boldsymbol{F}}$ output from the LFC module for the classification task. In other words, after the $\boldsymbol{E}^{(1)}$ and $\boldsymbol{E}^{(12)}$ feature vectors are dimensioned down by downsampling, we merge the feature vectors $\boldsymbol{E}^{(1)}$, $\boldsymbol{E}^{(12)}$, and $\hat{\boldsymbol{F}}$, and use them for TC task.

Formally, in this paper, the feature vectors $\boldsymbol{E}^{(1)}$ and $\boldsymbol{E}^{(12)}$ are summed and averaged to obtain the fused feature vector $\tilde{\boldsymbol{F}} \in \mathbb{R}^{1\times d}$, as shown in (13).

$$\tilde{\boldsymbol{F}} = \left(\boldsymbol{E}^{(1)} + \boldsymbol{E}^{(12)}\right)/2 \tag{13}$$

$\tilde{\boldsymbol{F}}$ is the final feature vector output by the TTE module. It is consistent with the dimensionality of the feature vector $\boldsymbol{E}^{(1)}$ and $\boldsymbol{E}^{(12)}$. And the feature vector $\tilde{\boldsymbol{F}} \in \mathbb{R}^{1\times d}$ is summed and averaged again with the feature vector $\hat{\boldsymbol{F}} \in \mathbb{R}^{1\times y}$ output from the LFC module to obtain a new fused feature vector $\bar{\boldsymbol{F}} \in \mathbb{R}^{1\times y}$, as shown in (14).

$$\bar{\boldsymbol{F}} = (\tilde{\boldsymbol{F}} + \hat{\boldsymbol{F}})/2 \tag{14}$$

The number of filters y is equal to the embedding dimension d of the hidden layer, which is consistent with the dimension of the feature vector $\tilde{\boldsymbol{F}}$ and $\hat{\boldsymbol{F}}$. Finally, we feed the feature vector $\bar{\boldsymbol{F}}$ to the classifier layer to complete the final TC task.

F. CLASSIFIER LAYER

We input the feature vector $\bar{\boldsymbol{F}}$ into a fully connected layer and map the feature vector to the sample category space via a feature space transformation [46]. This results in a higher level of abstraction of the feature vector, which is used for the classification task. As shown in the "Classifier" module in Figure 1.

Formally, $\bar{\boldsymbol{F}} \in \mathbb{R}^{1\times d}$ belongs to a little higher-level feature vector. And a higher-level feature vector is represented by $\boldsymbol{R} \in \mathbb{R}^{1\times k}$, as shown in (15).

$$\boldsymbol{R} = \bar{\boldsymbol{F}}\boldsymbol{W}_3 + \boldsymbol{b}_3 \tag{15}$$

where $\boldsymbol{W}_3 \in \mathbb{R}^{d\times k}$ is the matrix parameter, $\boldsymbol{b}_3 \in \mathbb{R}^{1\times k}$ is the bias term, k is the total number of news text categories, d is the maximum embedding dimension of the hidden layer, and $\boldsymbol{W}_3$ and $\boldsymbol{b}_3$ are the parameters that need to be learned and updated. Finally, we use the softmax function to predict the category of news texts. Formally, assume $r_1, r_2, \cdots, r_i, \cdots, r_k$ represent the probability of each category of each text (i.e., $\boldsymbol{R} = [r_1, r_2, \cdots, r_i, \cdots, r_k]$), the feature vector $\boldsymbol{R} \in \boldsymbol{R}^{1\times k}$ is converted into a probability vector $\boldsymbol{P} \in \boldsymbol{R}^{1\times k}$ using the softmax function, as shown in (16) and (17).

$$\boldsymbol{P} = \text{softmax}(\boldsymbol{R}) = [p_1, p_2, \cdots, p_i, \cdots, p_k] \tag{16}$$

$$p_i = \frac{\exp(r_i)}{\sum_{i=1}^{k} \exp(r_i)} \tag{17}$$

where p_i represents the probability that a news text belongs to class i and e is an exponential function with base e. It makes the larger values in the vector more distinctive.

In summary, the LFCN architecture based on BERT and CNN proposed in this paper has three novelties as following:

1) Deal with Chinese long texts. In order to solve the limitation of BERT PLMs on the length of input sequence and to improve the accuracy of news long TC task, this paper proposes a simple and efficient DLn digest algorithm for long text of online news. The core idea of the algorithm is to obtain a number of complete sentences at the beginning and end of a long text.

2) Selection of deep learning models. PLMs, represented by BERT, can learn generic language representations. And they have achieved good results on short TC tasks. CNNs are good at capturing local features such as key phrases. Therefore, in order to equip the model with the ability to capture both local and global features, the BERT and CNN models are fused in this paper. Specially, to preserve the original features of a text as possible, we input the output feature vector of the first layer of the TTE to the LFC module.

3) Selection and processing of feature vectors. In order to enable the model to learn more textual features in news text, in this paper, we have selected and fused two feature vectors output by the first and twelfth layers of TTE module and a feature vector output by LFC module respectively. In this way, the fused feature vector expresses the local and global features in text.

IV. EXPERIMENTS

To improve the accuracy of the Chinese news long text classification task, a BERT-based LFCN model is proposed in this paper. In this section, to verify the validity of the new method, we show the results of our experiments on the THUCNews and MCNews datasets as well as the experimental procedure.

A. EVALUATION INDICATORS

In order to verify the performance of the different classification methods, this paper evaluates all classification methods based on a confusion matrix. These include: True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) [47]. Specially, this paper uses four multicategory task evaluation metrics, Accuracy, Weighted-Precision (WP), Weighted-Recall (WR), WeightedF1 (WF1) [48]. As shown in (18)-(21).

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{m} TP_i \tag{18}$$

$$WP = \frac{1}{N}\sum_{i=1}^{m} n_i \frac{TP_i}{TP_i + FP_i} \tag{19}$$

$$WR = \frac{1}{N}\sum_{i=1}^{m} n_i \frac{TP_i}{TP_i + FN_i} \tag{20}$$

$$WF1 = \frac{1}{N}\sum_{i=1}^{m} n_i \frac{2P_iR_i}{P_i + R_i} \tag{21}$$

where N represents the total number of samples, m represents the total number of categories, n_i represents the number of samples in the ith category, TP_i, FP_i and FN_i denote the number of positively classified, positively misclassified, and negatively misclassified samples of the model in the ith category, respectively. P_i and R_i are the precision P and recall R of the ith category, respectively.

B. DATASETS

All experiments in this paper are conducted on the THUCNews and MCNews datasets. Next, we briefly describe the datasets. The overall picture of the dataset is shown in Table 1.

1) THUCNews datasets. This dataset is part of THUCTC (http://thuctc.thunlp.org/). In this paper, we use a version of 65 000 news items evenly distributed across 10 areas: ''Sports'', ''Business'', ''House'', ''Home'', ''Education'', ''Technology'', ''Fashion'', ''Politics'', ''Games'', and ''Entertainment'' [16]. It contains 50 000/5 000/10 000 training/validation/test set data with an average text length of 913, 882, and 969, respectively, with the longest text containing 27 467 tokens.

2) MCNews datasets. The distribution of data features is consistent with that provided by the A7 question in the 10th China Software Cup (http://www.cnsoftbei.com/). It includes 15 200 non-uniformly distributed online news items in nine fields, including ''Business'', ''House'', ''Education'', ''Technology'', ''Sports'', ''Games'', ''Entertainment'', ''Military'', and ''Auto''. It was also divided into training and validation sets in the ratio of 7:3. The average length of the text was 1 197 and 1 238, respectively. And the longest text contained 30 495 tokens.

C. EXPERIMENTAL SETTINGS

In this paper, we use the default AdamWeightDecayOptimizer optimizer in the BERT model. We set the max input sequence length to 256, the batch size to 32, the number of training epochs to 4 and the initial learning rate to 2×10^{-5}. We use an early stop mechanism that will terminate training

TABLE 1. Dataset statistics.

Dataset	AW	L	S	N	C
THUCNews(Train)	913	27 467	8	50 000	10
THUCNews(Dev)	882	10 919	15	5 000	10
THUCNews(Test)	969	14 720	13	10 000	10
MCNews(Train)	1 197	30 495	2	11 000	9
MCNews(Dev)	1 238	21 814	11	4 200	9

C is the count of news categories. AW, L, and S denote the average, longest, and shortest number of tokens of the text respectively. N denotes the total number of news text data.

if the accuracy on the validation set has not improved after two epochs of training. To accommodate the uncertainty of MLM, five sets of identical experiments are conducted for each BERT model, with each set of experiments saving the best performing model on the validation set.

D. BASELINE MODELS

In this paper, eight state-of-the-art deep learning models are selected as baseline models, including CNN, ERNIE, BERT, BERT-wwm, BERT-wwm-ext RoBERTa-wwm-ext, MacBERT, and ChineseBERT.

CNN: The method was proposed by Kim [7] and it uses two-channel word embeddings. During training phase, all embeddings are randomly initialized using word2vec [49].

MacBERT: MacBERT (MLM as correction BERT) [50] improves upon RoBERTa by using the MLM as correlation (Mac) masking strategy and the sentence-order prediction (SOP) task.

BERT: This approach, proposed by Devlin *et al.* [11], is a novel language model using a bidirectional encoder representation in Transformer.

BERT-WWM: Cui *et al.* [16] used the WWM mechanism, and they introduced several Chinese BERT PLMs. The BERT-wwm, BERT-wwm-ext, and RoBERTa-wwm-ext PLMs have been chosen for this paper.

ERNIE: ERNIE (Enhanced Representation through kNowledge IntEgration) [51] applied different masking strategies to optimize the masking process of BERT, which includes char-level masking, phrase-level masking, and entity-level masking.

ChineseBERT: ChineseBERT [42] incorporates both the glyph and pinyin information of Chinese characters into language model pretraining, and achieves new SOTA performances on multiple Chinese NLP tasks.

E. MAIN EXPERIMENTAL RESULTS

The experimental results of the proposed model and the base models on the THUCNews and MCNews datasets are shown in Table 2.

The experimental results show that the LFCN model proposed in this paper outperforms the baseline models on both the THUCNews and MCNews datasets. It is worth noting that the LFCN model has the highest accuracy of 99.0% and 96.2% on the THUCNews and MCNews datasets, 1.1% and 2.2% higher than the baseline models, respectively.

TABLE 2. Main experimental results (Accuracy/%).

	THUNews		MCNews
Model	Dev	Test	Dev
CNN	94.9	96.5	91.0
ERNIE	98.2	97.7	-
BERT	97.7	97.8	93.4
BERT-wwm	98.0	97.8	94.4
BERT-wwm-ext	97.7	97.7	93.4
RoBERTa-wwm-ext	98.3	97.8	94.0
MacBERT	98.2	97.7	-
ChineseBERT	98.1	97.9	-
LFCN (ours)	**98.8**	**99.0**	**96.2**

We also find that a single CNN model does not perform very well because CNNs can only capture local features. Overall, the BERT model and its variants outperform the CNN model because the BERT model is based on a multihead self-attention, which can capture global features in text easily and understand the overall meaning of the text better. Further more, the WWM mechanism improves the performance of the BERT model to a certain extent. The reason is that in Chinese, the same word in different contexts conveys different meanings.

F. EXPERIMENTAL PROCEDURE AND ANALYSIS

1) DEALING WITH THE LONG TEXTS

In order to address the limitation of the BERT model on the max input sequence. In this paper, we try three different methods to extract short texts. These include: a) Top: only keep the complete sentence of the first 254 tokens, b) Tail: only keep the complete sentence of the last 254 tokens, c) Both: keep the complete sentences of the first and last 127 tokens in both ''Top'' and ''Tail'' modes. We conduct experiments on the THUCNews and MCNews datasets, and use the BERT-wwm, BERT-wwm-ext, and RoBERTa-wwm-ext PLMs as initial checkpoints, respectively. The experimental results are shown in Table 3.

For both the BERT-wwm and BERT-wwm-ext models, the ''Both'' approach is preferred over the ''Tail'' and ''Top'' approaches in many cases. There is no great distinction between the ''Top'' and the ''Tail'' approaches, but on the whole the ''Tail'' approach outperforms the ''Top'' approach. For the RoBERTa-wwm-ext model, the ''Both'' approach always outperforms the ''Tail'' and ''Top'' approaches. In the field of Chinese news, editors may prefer to write the core content at the beginning and end of the articles. Therefore, in the follow-up experiments, this paper adopts the ''Both'' method to extract short texts, which is called DLn digest algorithm.

TABLE 3. Accuracy of the three different extraction methods (%).

Baseline model	Method	THUCNews Dev	THUCNews Test	MCNews Dev
BERT-wwm	Top	$97.74_{(97.59)}$	$97.22_{(96.88)}$	$93.98_{(93.81)}$
	Tail	$96.48_{(96.09)}$	$97.42_{(96.92)}$	$\mathbf{95.08}_{(94.54)}$
	Both	$\mathbf{98.50}_{\mathbf{(98.45)}}$	$\mathbf{98.87}_{\mathbf{(98.79)}}$	$94.94_{\mathbf{(94.82)}}$
BERT-wwm-ext	Top	$98.16_{(97.99)}$	$97.63_{(97.26)}$	$93.96_{(93.87)}$
	Tail	$96.16_{(95.85)}$	$97.78_{(97.21)}$	$94.64_{\mathbf{(94.47)}}$
	Both	$\mathbf{98.70}_{\mathbf{(98.51)}}$	$\mathbf{98.87}_{\mathbf{(98.76)}}$	$\mathbf{94.76}_{(94.29)}$
RoBERTa-wwm-ext	Top	$97.88_{(97.83)}$	$97.31_{(97.24)}$	$94.38_{(94.21)}$
	Tail	$96.42_{(96.24)}$	$97.63_{(97.57)}$	$94.90_{(94.76)}$
	Both	$\mathbf{98.82}_{\mathbf{(98.65)}}$	$\mathbf{98.93}_{\mathbf{(98.78)}}$	$\mathbf{95.33}_{\mathbf{(94.90)}}$

Form of experimental results: Max value (mean value), same below.

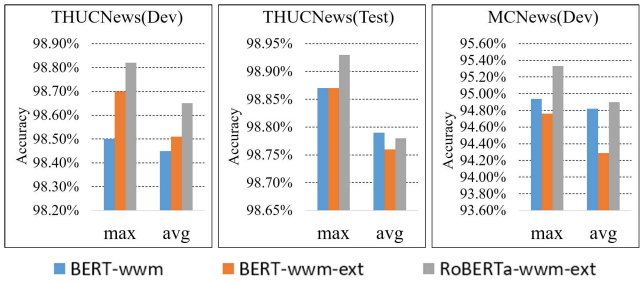

FIGURE 7. Experimental results of DLn combining different PLMs.

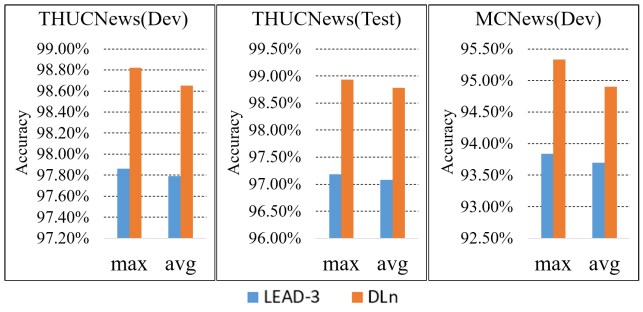

FIGURE 8. Experimental results of LEAD-3 and DLn extraction methods.

2) SELECTING A PLM AS INITIAL CHECKPOINT

The experimental results of the BERT-wwm, BERT-wwm-ext, and RoBERTa-wwm-ext models after using the DLn digest algorithm are shown in Figure 7.

It is clear that the RoBERTa-wwm-ext model outperforms the BERT-wwm and BERT-wwm-ext models in the vast majority of cases. Therefore, in the subsequent experiments, this paper uses the RoBERTa-wwm-ext PLM as the starting checkpoint for the TTE layer.

3) COMPARING WITH LEAD-3

In this paper, we have compared DLn with the traditional LEAD-3 digest algorithm, and the experimental results are shown in Table 4 and Figure 8.

Overall, the accuracy of the DLn method is higher than the LEAD-3 method. In particular, the DLn method improves the accuracy by a maximum of 1.75% on the THUCNews dataset and 1.49% on the MCNews dataset compared to the LEAD-3 method.

4) EFFECT OF EACH MODULE OF LFCN

Given the limited ability of a single model to capture text features, in this paper, we combine the BERT (i.e., RoBERTa-wwm-ext) PLM with a CNN (i.e., LFC module), called LFCN. Ablation experiments are also conducted on the

TABLE 4. Experimental results of two different methods (Accuracy/%).

Extraction method	THUNews Dev	THUNews Test	MCNews Dev
LEAD-3	$97.86_{(97.79)}$	$97.18_{(97.08)}$	$93.84_{(93.70)}$
DLn	$\mathbf{98.82}_{\mathbf{(98.65)}}$	$\mathbf{98.93}_{\mathbf{(98.78)}}$	$\mathbf{95.33}_{\mathbf{(94.90)}}$

THUCNews and MCNews datasets. The results of the experiment are shown in Table 5 and Figure 9.

As shown in Figure 9, the model with "DLn+TTE+LFC" tends to perform best in most cases. The accuracy and WF1 on the THUCNews dataset are as high as 98.98%. Overall, the use of the LFCN model (i.e., the combination of "DLn+TTE+LFC" method) proposed in this paper is able to deliver varying degrees of performance improvement on the THUCNews and MCNews datasets.

5) EFFECT OF THE NUMBER OF BERT HIDDEN LAYERS

The number of hidden layers in the BERT model (i.e., TTE module) affects its ability to capture text features. Therefore, we investigate the effectiveness of different numbers of hidden layers. Then, we train the LFCN model and record the performance in terms of accuracy.

TABLE 5. Experimental results of the ablation experiments (%).

	THUCNews(Dev)		THUCNews(Test)		MCNews(Dev)	
Model	Accuracy	WF1	Accuracy	WF1	Accuracy	WF1
DLn+TTE	**98.82**$_{(98.65)}$	**98.82**$_{(98.65)}$	98.93$_{(98.78)}$	98.93$_{(98.78)}$	95.46$_{(94.96)}$	95.33$_{(94.90)}$
TTE+LFC	98.24$_{(98.07)}$	98.23$_{(98.07)}$	97.20$_{(97.02)}$	97.18$_{(97.00)}$	93.44$_{(93.42)}$	93.35$_{(93.33)}$
DLn+TTE+LFC	98.80$_{\mathbf{(98.69)}}$	98.80$_{\mathbf{(98.69)}}$	**98.98**$_{\mathbf{(98.83)}}$	**98.98**$_{\mathbf{(98.83)}}$	**95.56**$_{\mathbf{(95.13)}}$	**95.43**$_{\mathbf{(95.02)}}$

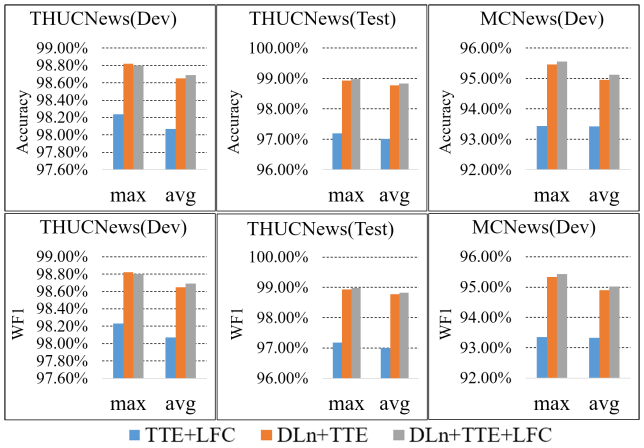

FIGURE 9. Experimental results of the ablation experiment.

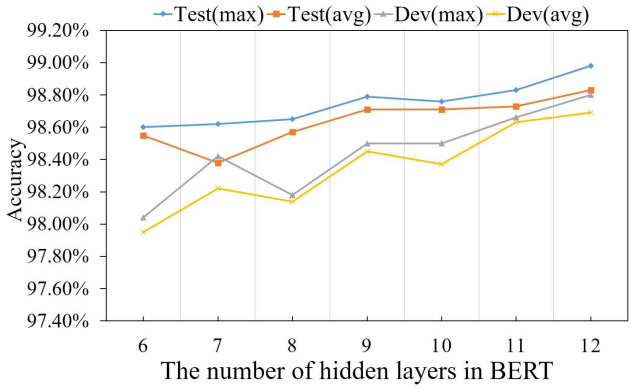

FIGURE 10. Effect of the number of hidden layers in BERT.

TABLE 6. Effect of the number of hidden layers in BERT (Accuracy/%).

	THUNews	
Number of hidden layers	Dev	Test
6	98.04$_{(97.95)}$	98.60$_{(98.55)}$
7	98.42$_{(98.22)}$	98.62$_{(98.38)}$
8	98.18$_{(98.14)}$	98.65$_{(98.57)}$
9	98.50$_{(98.45)}$	98.79$_{(98.71)}$
10	98.50$_{(98.37)}$	98.76$_{(98.71)}$
11	98.66$_{(98.63)}$	98.83$_{(98.73)}$
12	**98.80**$_{\mathbf{(98.69)}}$	**98.98**$_{\mathbf{(98.83)}}$

Figure 10 and Table 6 show the performance of the LFCN model with different numbers of hidden layers. It is not difficult to find that the LFCN achieves the best performance when the number of hidden layers is 12 (the maximum number of hidden layers for $BERT_{BASE}$ is 12).

6) EFFECT OF THE BERT INPUT REPRESENTATION

Finally, experiments are conducted in this paper with two different input embedding methods (shown in Figure 5 and Figure 6), noted as "signal" and "dual", respectively. The experimental results are shown in Table 7 and Figure 11.

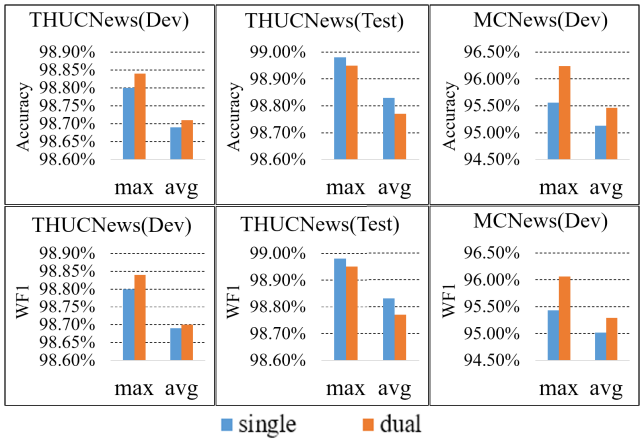

FIGURE 11. Experimental results for different input methods.

It is easy to find from Figure 11 that there is no significant distinction between the two types of "signal" and "dual" input representation. However, the "dual" input method outperforms the "single" input method in most cases.

TABLE 7. Experimental results for different input methods (%).

	THUCNews(Dev)		THUCNews(Test)		MCNews(Dev)	
Model	Accuracy	WF1	Accuracy	WF1	Accuracy	WF1
single	98.80$_{(98.69)}$	98.80$_{(98.69)}$	**98.98**$_{\mathbf{(98.83)}}$	**98.98**$_{\mathbf{(98.83)}}$	95.56$_{(95.13)}$	95.43$_{(95.02)}$
dual	**98.84**$_{\mathbf{(98.71)}}$	**98.84**$_{\mathbf{(98.70)}}$	98.95$_{(98.77)}$	98.95$_{(98.77)}$	**96.24**$_{\mathbf{(95.46)}}$	**96.06**$_{\mathbf{(95.29)}}$

In summary, the long TC method proposed in this paper not only solves the limitation of the BERT PLMs on the length of the max input sequence, but also improves the performance of long TC. It is noteworthy that the LFCN model proposed in this paper has the highest accuracy of 98.98% and 96.24% on the THUCNews and MCNews datasets, respectively.

V. CONCLUSION

To solve the problem of classifying long text in Chinese news, we propose a LFCN architecture based on BERT and CNN. In the data preprocessing phase, we propose a DLn long text digest algorithm to extract short text pairs from long texts and represent them using the BERT input of the ''sentence pair'' classification task. In particular, we fuse three feature vectors generated by TTE and LFC module into a textual feature vector. This feature vector is used as the final representation of the news text. And it is fed into the classification layer to predict the corresponding category of the news text. Supported by abundant comparative experiments, the accuracy of our proposed method is higher than that of the base models in most cases. Notably, the highest accuracy of 99.0% and 96.2% are achieved on the THUCNews and MCNews datasets, an improvement of 1.1% and 2.2% relative to the base model, respectively.

The Chinese news text classification problem is just one branch of extracting the valuable information contained in the massive amount of news text data. In future research, tasks such as sentiment classification can also be carried out in conjunction with the methods proposed in this paper. Nowadays, information is abundant. And the valuable information contained in data is well worth exploring.

REFERENCES

[1] Y. Cheng, H. Sun, H. Chen, M. Li, Y. Cai, Z. Cai, and J. Huang, ''Sentiment analysis using multi-head attention capsules with multi-channel CNN and bidirectional GRU,'' *IEEE Access*, vol. 9, pp. 60383–60395, 2021, doi: 10.1109/ACCESS.2021.3073988.

[2] S. Sanagavarapu, S. Sridhar, and S. Chitrakala, ''News categorization using hybrid BiLSTM-ANN model with feature engineering,'' in *Proc. IEEE 11th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2021, pp. 134–140.

[3] T. Luong, H. Pham, and C. D. Manning, ''Effective approaches to attention-based neural machine translation,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.

[4] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' in *Proc. ICLR*, San Diego, CA, USA, 2015, pp. 1–14.

[5] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[6] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, ''A convolutional neural network for modelling sentences,'' in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics (Long Papers)*, vol. 1, 2014, pp. 655–665.

[7] Y. Kim, ''Convolutional neural networks for sentence classification,'' in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1746–1751.

[8] M. Hughes, I. Li, S. Kotoulas, and T. Suzumura, ''Medical text classification using convolutional neural networks,'' 2017, *arXiv:1704.06841*.

[9] Y. Liu, J. Zhang, C. Gao, J. Qu, and L. Ji, ''A sensitivity analysis of attention-gated convolutional neural networks for sentence classification,'' 2019, *arXiv:1908.06263*.

[10] 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, doi: 10.1016/j.ins.2018.10.030.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ''BERT: Pre-training of deep bidirectional transformers for language understanding,'' 2018, *arXiv:1810.04805*.

[12] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, ''RoBERTa: A robustly optimized BERT pretraining approach,'' 2019, *arXiv:1907.11692*.

[13] P.-H. Chi, P.-H. Chung, T.-H. Wu, C.-C. Hsieh, Y.-H. Chen, S.-W. Li, and H.-Y. Lee, ''Audio albert: A lite bert for self-supervised learning of audio representation,'' in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Jan. 2021, pp. 344–350.

[14] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, ''DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter,'' 2019, *arXiv:1910.01108*.

[15] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, ''SpanBERT: Improving pre-training by representing and predicting spans,'' *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 64–77, Dec. 2020.

[16] Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang, ''Pre-training with whole word masking for Chinese BERT,'' 2019, *arXiv:1906.08101*.

[17] C. Sun, X. Qiu, Y. Xu, and X. Huang, ''How to fine-tune BERT for text classification?'' in *Proc. CCL*, vol. 11856, Kunming, China, 2019, pp. 194–206.

[18] R. Pappagari, P. Zelasko, J. Villalba, Y. Carmiel, and N. Dehak, ''Hierarchical transformers for long document classification,'' in *Proc. ASRU*, Dec. 2019, pp. 838–844.

[19] Y. Jin, Q. Zhu, X. Deng, and L. Hu, ''Weighted hierarchy mechanism over BERT for long text classification,'' in *Proc. ICAIS*, vol. 12736, Dublin, Ireland, 2021, pp. 566–574.

[20] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, A. Mehmood, and M. T. Sadiq, ''Document-level text classification using single-layer multisize filters convolutional neural network,'' *IEEE Access*, vol. 8, pp. 42689–42707, 2020, doi: 10.1109/ACCESS.2020.2976744.

[21] Y. Liu, ''Fine-tune BERT for extractive summarization,'' 2019, *arXiv:1903.10318*.

[22] H. P. Luhn, ''The automatic creation of literature abstracts,'' *IBM J. Res. Develop.*, vol. 2, no. 2, pp. 159–165, Apr. 1958, doi: 10.1147/rd.22.0159.

[23] X. Zhang, F. Yin, G. Ma, B. Ge, and W. Xiao, ''M-SQL: Multi-task representation learning for single-table Text2sql generation,'' *IEEE Access*, vol. 8, pp. 43156–43167, 2020, doi: 10.1109/ACCESS.2020.2977613.

[24] P. Li, J. Gao, B. Zhai, J. Zhang, and Z. Chen, ''Multi-view representation learning via dual optimal transportation,'' *IEEE Access*, vol. 9, pp. 144976–144984, 2021, doi: 10.1109/ACCESS.2021.3123078.

[25] P. H. Le-Khac, G. Healy, and A. F. Smeaton, ''Contrastive representation learning: A framework and review,'' *IEEE Access*, vol. 8, pp. 193907–193934, 2020, doi: 10.1109/ACCESS.2020.3031549.

[26] Y. Bengio, P. Lamblin, and D. Popovici, ''Greedy layer-wise training of deep networks,'' in *Proc. NIPS*, Vancouver, BC, Canada, 2006, pp. 153–160.

[27] L. Cagliero and M. L. Quatra, ''Inferring multilingual domain-specific word embeddings from large document corpora,'' *IEEE Access*, vol. 9, pp. 137309–137321, 2021, doi: 10.1109/ACCESS.2021.3118093.

[28] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, ''Deep learning based text classification: A comprehensive review,'' *ACM Comput. Surv.*, vol. 54, no. 3, pp. 62:1–62:40, 2021, doi: 10.1145/3439726.

[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean, ''Efficient estimation of word representations in vector space,'' in *Proc. ICLR*, Scottsdale, AZ, USA, 2013, pp. 1–12.

[30] J. Wang, Z. Wang, D. Zhang, and J. Yan, ''Combining knowledge with deep convolutional neural networks for short text classification,'' in *Proc. IJCAI*, Melbourne, VIC, Australia, 2017, pp. 2915–2921.

[31] M. M. Trusca and G. Spanakis, ''Hybrid tiled convolutional neural networks for text sentiment classification,'' 2020, *arXiv:2001.11857*.

[32] M. Alam, Q. Bie, R. Türker, and H. Sack, ''Entity-based short text classification using convolutional neural networks,'' in *Proc. EKAW*, vol. 12387, Bolzano, Italy, 2020, pp. 136–146.

[33] Y. Yan, W. Li, G. Chen, and W. Liu, ''An improved text classification method based on convolutional neural networks,'' in *Proc. Int. Conf. Control, Robot. Intell. Syst.*, Oct. 2020, pp. 185–190.

[34] W. Zhao, R. Singh, T. Joshi, A. Sudjianto, and V. N. Nair, ''Self-interpretable convolutional neural networks for text classification,'' 2021, *arXiv:2105.08589*.

[35] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," 2020, *arXiv:2003.08271*.
[36] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. (2018). *Improving Language Understanding by Generative Pre-Training*. [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf
[37] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. (2019). *Language Models are Unsupervised Multitask Learners*. [Online]. Available: https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf
[38] T. B. Brown *et al.*, "Language models are few-shot learners," 2020, *arXiv:2005.14165*.
[39] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., (Long Papers)*, vol. 1, 2018, pp. 2227–2237.
[40] Z.-L. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. NeurIPS*, Vancouver, BC, Canada, 2019, pp. 5754–5764.
[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.
[42] Z. Sun, X. Li, X. Sun, Y. Meng, X. Ao, Q. He, F. Wu, and J. Li, "ChineseBERT: Chinese pretraining enhanced by glyph and pinyin information," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process. (Long Papers)*, vol. 1, 2021, pp. 2065–2075.
[43] N.-F. Sun and C.-Y. Du, "News text classification method and simulation based on the hybrid deep learning model," *Complex*, vol. 2021, 2021, Art. no. 8064579, doi: 10.1155/2021/8064579.
[44] D.-C. Yang and C. Du, "Stance detection with stance-wise convolution network," in *Proc. NLPCC*, vol. 12430. Zhengzhou, China, 2020, pp. 555–567.
[45] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, and L. Li, "On the sentence embeddings from pre-trained language models," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 9119–9130.
[46] J. Deng, L. Cheng, and Z. Wang, "Attention-based BiLSTM fused CNN with gating mechanism model for Chinese long text classification," *Comput. Speech Lang.*, vol. 68, Jul. 2021, Art. no. 101182, doi: 10.1016/j.csl.2020.101182.
[47] W.-P. Jing, X. Song, D. Di, and H. Song, "GeoGAT: Graph model based on attention mechanism for geographic text classification," *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, vol. 20, no. 5, pp. 76:1–76:18, 2021, doi: 10.1145/3434239.
[48] Z. Wang, L. Wang, C. Huang, and X. Luo, "BERT-based Chinese text classification for emergency domain with a novel loss function," 2021, *arXiv:2104.04197*.
[49] Y. Liu, P. Li, and X. Hu, "Combining context-relevant features with multi-stage attention network for short text classification," *Comput. Speech Lang.*, vol. 71, Jan. 2022, Art. no. 101268, doi: 10.1016/j.csl.2021.101268.
[50] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu, "Revisiting pre-trained models for Chinese natural language processing," 2020, *arXiv:2004.13922*.
[51] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, and H. Wang, "ERNIE 2.0: A continual pre-training framework for language understanding," 2019, *arXiv:1907.12412*.

XINYING CHEN received the B.E. degree in computer science and technology and the M.E. degree in computer software and theory from Jilin University, China, in 2002 and 2005, respectively, and the Ph.D. degree in computer application technology from Dalian Maritime University, China. She is currently an Associate Professor with the School of Computer and Communication Engineering, Dalian Jiaotong University. Her research interests include data mining, intelligent information processing, the Internet of Things, and the Semantic Web of Things.

PEIMIN CONG received the B.E. degree from City Institute, Dalian University of Technology, China, in 2020. He is currently pursuing the M.E. degree with the School of Computer and Communication Engineering, Dalian Jiaotong University, China. His research interests include data mining, deep learning, and natural language processing.

SHUO LV received the B.E. degree from the Chengdu University of Information Technology, China, in 2021. She is currently pursuing the M.E. degree with the School of Computer and Communication Engineering, Dalian Jiaotong University, China. Her research interests include natural language processing and deep learning.