Journal of Organizational and End User Computing
Volume 33 • Issue 6

A BERT-Based Hybrid Short Text Classification Model Incorporating CNN and Attention-Based BiGRU

Tong Bao, Information Center, Jiangsu Academy of Agricultural Sciences & Institute of Science and Technology Information, Jiangsu University, China

Ni Ren, Information Center, Jiangsu Academy of Agricultural Sciences & Institute of Science and Technology Information, Jiangsu University, China*

Rui Luo, Information Center, Jiangsu Academy of Agricultural Sciences, China

Baojia Wang, Information Center, Jiangsu Academy of Agricultural Sciences, China

Gengyu Shen, Information Center, Jiangsu Academy of Agricultural Sciences, China

Ting Guo, Information Center, Jiangsu Academy of Agricultural Sciences, China

ABSTRACT

Short text classification is a research focus for natural language processing (NLP), which is widely used in news classification, sentiment analysis, mail filtering, and other fields. In recent years, deep learning techniques are applied to text classification and have made some progress. Different from ordinary text classification, short text has the problem of less vocabulary and feature sparsity, which raise higher request for text semantic feature representation. To address this issue, this paper proposes a feature fusion framework based on the bidirectional encoder representations from transformers (BERT). In this hybrid method, BERT is used to train word vector representation. Convolutional neural network (CNN) captures static features. As a supplement, a bi-gated recurrent neural network (BiGRU) is adopted to capture contextual features. Furthermore, an attention mechanism is introduced to assign the weight of salient words. The experimental results confirmed that the proposed model significantly outperforms the other state-of-the-art baseline methods.

KEYWORDS

Deep Learning, Fusion Framework, Natural Language Processing, Short Text Classification

1. INTRODUCTION

Due to the development and widespread use of the internet and mobile devices, users are always encountering and processing massive amounts of text data, such as news insights, product reviews, and messages. These large amounts of text data contain information on human social attributes, content preferences, and psychology. The careful mining and scientific analysis of these text data can generate extremely high social value. As the most basic task in the process of text data mining and analysis, text classification has been widely used in various industry fields, such as topic tagging, public opinion analysis, mail filtering, and recommendation systems (Lin, Z., et al., 2016; Ren, Y.

DOI: 10.4018/JOEUC.294580

*Corresponding Author

F., et al., 2016; Kiliroor, C. C., & Valliyammai, C. 2019; Sulthana, A. R., & Ramasamy, S. 2019). Generally, short text mainly includes news headlines, social issues, product reviews, etc. Most of these texts are unstructured with the characteristics of large size, sparseness, and irregularity. Therefore, extracting the features of short texts and correctly classifying them has become one of the current challenges in the field of natural language processing (NLP).

Deep learning is a branch of machine learning. Deep learning simulates the mechanism of the human brain by establishing a deep neural network and interprets and analyzes data, such as images, voices, and texts. In text classification, the most basic but critical part is to convert text into digital vectors that computers can understand, this process is called, "The Representation of Text." The earliest technology of text representation was one-hot encoding where the dimension of the word index is set to 1 and all of the others are set to 0. However, this representation suffers from the problem of high sparsity and dimensional explosion; More importantly, it does not consider the weight of words to text. TF-IDF (Yu, C. T., & Salton, G. 1976) is an optimized one-hot model that evaluates the importance of a word in a document or corpus, but there are still problems of dimensionality, and the model cannot reflect the sequence of information. Therefore, follow-up work has focused on constructing distributed dense word vectors with low dimensions. Word2Vec (Mikolov, T., et al., 2013) is a kind of neural network language model that considers contextual semantic information while avoiding the problem of dimensionality, whics has significantly better effects than previous models. In addition, FastText (Joulin, A., et al., 2016) is a word vector calculation and text classification tool open-sourced by Facebook in 2016. while working on classification tasks, FastText can often achieve accuracy comparable to deep networks, but it is faster than deep neural networks in training time. However, both Word2vec and FastText are static models and cannot solve the problem of polysemous words. To address this issue, Pre-trained language models, such as Embedding from Language Models(ELMo) (Peters, M. E., et al., 2018), Generate Pretraining Model(GPT) (Radford, A., et al., 2018) and the Bidirectional Encoder Representations from Transformers model (BERT) (Devlin, J., et al., 2018), have replaced Word2Vec as the current trend of word representation. ELMo uses the bidirectional long short-term memory(BiLSTM) (Hochreiter, S., & Schmidhuber, J. 1997) structure to obtain a general semantic representation through pretraining, and migrates the representation as a feature to the specific task. In addition, BERT and GPT use the transformer structure for pretraining. The fine-tuning method can be applied to training downstream special tasks by reducing the pretrained parameters, which not only saves time and computing resources but also quickly achieves better results.

Based on the above factors and the features of short text, this paper proposes a hybrid short text classification model-based BERT, namely BCBGA. First, the word vector of the short text is encoded through the pretraining model BERT, and then convolutional neural networks (CNNs)(Kim, Y. 2014) are adopted to extract local static features from different locations. To further improve the accuracy, a bi-gated recurrent neural network (BiGRU)(Cho, K., et al., 2014) is applied to obtain the contextual semantics information and then combines it with the attention mechanism to enhance the weight distribution of different words. Finally, the two parts of the features are input into the SoftMax function through the fully connected layer to obtain the classification result. The experimental results show that the model proposed in this paper has better performance than other baseline models. The contributions of this research are as follows:

1. To further improve short text classification accuracy, a new hybrid architecture named BCBGA is proposed to address the problem of sparse word features in short texts, which can effectively improve classification accuracy.
2. To enhance the ability of word vector representation, we use the BERT model to train word vectors during the text vectorization process.
3. The model also leverages the distinct advantages of CNN and BiGRU. The CNN model extracts local features of the text from the spatial perspective. A BiGRU obtains the sequence features of

the sentence. Furthermore, an attention mechanism is introduced to highlight the contribution of keywords in sentences.

4. We conduct experiments on large-scale public datasets and further analyze the influence of different parameters of the BCBGA on the classification result.

The rest of the paper is structured as follows: In Section 2, we briefly summarize the related work to this study. Section 3 presents the model frameworks of this paper. In Section 4, we describe the experiments and their results. Finally, we conclude the paper and introduce directions for future research in Section 5.

2. RELATED WORK

As one of the classic tasks of natural language processing, text classification has been extensively studied by many researchers. Currently, text classification methods are mainly divided into traditional machine learning methods and deep learning methods. Traditional machine learning methods, such as TF-IDF (Yu, C. T., & Salton, G. 1976), K-nearest neighbour (KNN) (Bijalwan, V., et al., 2014), and the naive Bayesian model (NBM) (Goudjil, M., et al., 2018) contain methods that have achieved reliable classification results. However, due to the excessive reliance on manually defined features, the above methods all have a common defect with poor transferability and time consumption problems.

In recent years, many researchers have conducted in-depth research on deep neural networks and applied them to text classification. Compared with traditional machine learning, deep learning can usually achieve great performance by simply passing the data directly to the network activated by the nonlinear function, which eliminates the tedious and challenging feature engineering stage. Among these approaches, CNNs and Recurrent Neural Networks (RNNs) have been widely used in various text classification tasks. (Kim, Y. 2014) proposed a method that uses different size convolution kernels to extract local static features in the text. After the convolution operation, the pooling layer was used to extract the most obvious features, and finally, the features were input to the fully connected layer for classification. Subsequently, (Zeng, S., et al., 2020) used multiple convolution filters to combine and pool together and then analyzed the influence of convolution kernels, combined kernels, and word embedding on the classification results. For instance, there are many differences between the word structures of Chinese and English. Thus, the model needs to be improved to adapt to the characteristics of Chinese words, which makes the classification of Chinese more challenging. (Guo, B., et al., 2019) proposed a novel term weighting scheme, in which multiple weights are assigned to each term and applied to word embeddings to enhance the classification performance of CNNs. (Xu, W. H., et al., 2019) recommend a method that uses the CNN-based skip-gram method for Chinese text classification and accesses Sogou news corpus. The experimental results indicate that the CNN with the skip-gram model performs more efficiently than the CNN-based one-hot method. The CNN only uses convolution and pooling to provide advantageous feature extraction capabilities, but the convolution operation cannot consider the positioned information of the text sequence, while the sequence structure of RNN can extract the context features of the text. However, RNN has the problem of gradient disappearance during the training process, which affects the results of the experiment. Therefore, to improve their respective weaknesses and take advantage of the individual strengths, the hybrid network that combines CNN and RNN has been mentioned by an increasing number of researchers. (Liu, B., et al., 2020) studied a hybrid network combining a CNN and BiGRU with fully connected layers, which can capture both global and local textual semantics at a fast convergence speed. (Luo, L. X. 2019) used the LDA model to train the topic distribution of short text and adopted GRU-CNN to strengthen the relationship between words and text to achieve highly accurate text classification. (Jin, N., et al., 2020) proposed a sentiment classification model named MTL-MSCNN-LSTM, which applied LSTM and multiscale CNN to jointly execute an encoding sentence that takes into account global and local features of the text. Experiments show that this method has better classification performance than the baseline model.

The attention mechanism has been introduced to assign the weights according to the contribution of different words to the article (Mnih, V., et al., 2014), which has a positive effect on text classification. (Qiao, X., et al., 2019) proposed the word-character attention model (WCAM). This method uses word-level attention to capture words that have a closer semantic relationship to the text and character-level attention extract word that have obvious judging properties in the text. (Jang, B., et al., 2020) proposed a hybrid model combining LSTM and CNN model that comprehensively considers the strengths of LSTM and CNN with an additional attention mechanism. (Zhang, Y. S., et al., 2019) proposed a coordinated CNN-LSTM-attention(CCLA) model, used CCLA units to learn the vector representations of sentences and transfer them to a SoftMax regression classifier to identify the sentiment tendencies in the text. (Zhang, D. J., et al., 2019) used bidirectional gated recurrent units (BiGRUs) and integrated a novel attention pooling mechanism with the max-pooling operation, so that the model could focus on the critical words and retain the most meaningful features of the text.

BERT is a pretrained language model trained on a massive corpus, and it was open sourced by Google in 2018. Compared with the static word vector generated by Word2Vec, BERT is based on the structure of a multilayer transformer encoder, which uses the attention mechanism to make words directly encode each other, regardless of the direction and distance. The pretraining part of BERT includes two unsupervised tasks: the masked language model (MLM) and next sentence prediction (NSP). The MLM will first randomly mask out 15% of the words of the input text, and then the transformer will predict the words of these masks and adjust the parameters of the model to make the prediction accuracy as high as possible. Therefore, BERT will rely more on contextual information to predict the masked word, which promotes the model to better understand the relevance of the different sentences. The NSP task is used to identify whether two sentences are continuous in the article. The purpose of adding this task is that many current NLP tasks, such as Question Answering (Q&A) or Natural Language Inference (NLI), need to understand the relationship between two sentences. The combination of these two tasks enables the model to more accurately describe semantic information at the sentence or even text level. Fine-tuning is the process of modifying a few initialized parameters that have been pretrained on massive corpora so that the parameters can be adapted to different datasets. In a sense, fine-tuning truly realize transfer learning in the field of NLP and provides great convenience for further study by researchers. This paper uses the BERT-BASE, Chinese, which is a Chinese version trained on the Wikipedia corpus for fine-tuning the downstream tasks.

3. THE PROPOSED MODEL BCBGA

This section describes the specific model architecture(BCBGA) proposed in this paper. The model can be divided into three layers, including the word embedding layer, the hybrid neural network layer, and the full connection layer. Firstly, in the word embedding layer, to enhance the ability of word vector representation, BERT is applied to replace Word2Vec to train the word vector representation. Second, CNN is applied to extract static features and then obtain the highest contribution feature through max pooling. At the same time, BiGRU-attention is adopted to obtain contextual semantic information and highlight the weights of the keywords. Finally, the two parts of features are spliced through the fully connected layer and then input into SoftMax to obtain the classification result. The overall structure of the proposed model is shown in Figure 1.

3.1. BERT Model

BERT uses a bidirectional transformer structure based on the multi-head-attention for feature extraction. (Vaswani, A., et al., 2017). Usually, the core of Encoder-Decoder structure to solve this kind of sequence problem is realized based on the sequence structure in the RNN, but the structure of RNN has the shortcomings that it cannot be parallelized and run slowly. To address this issue,

Figure 1. Overall structure of BCBGA

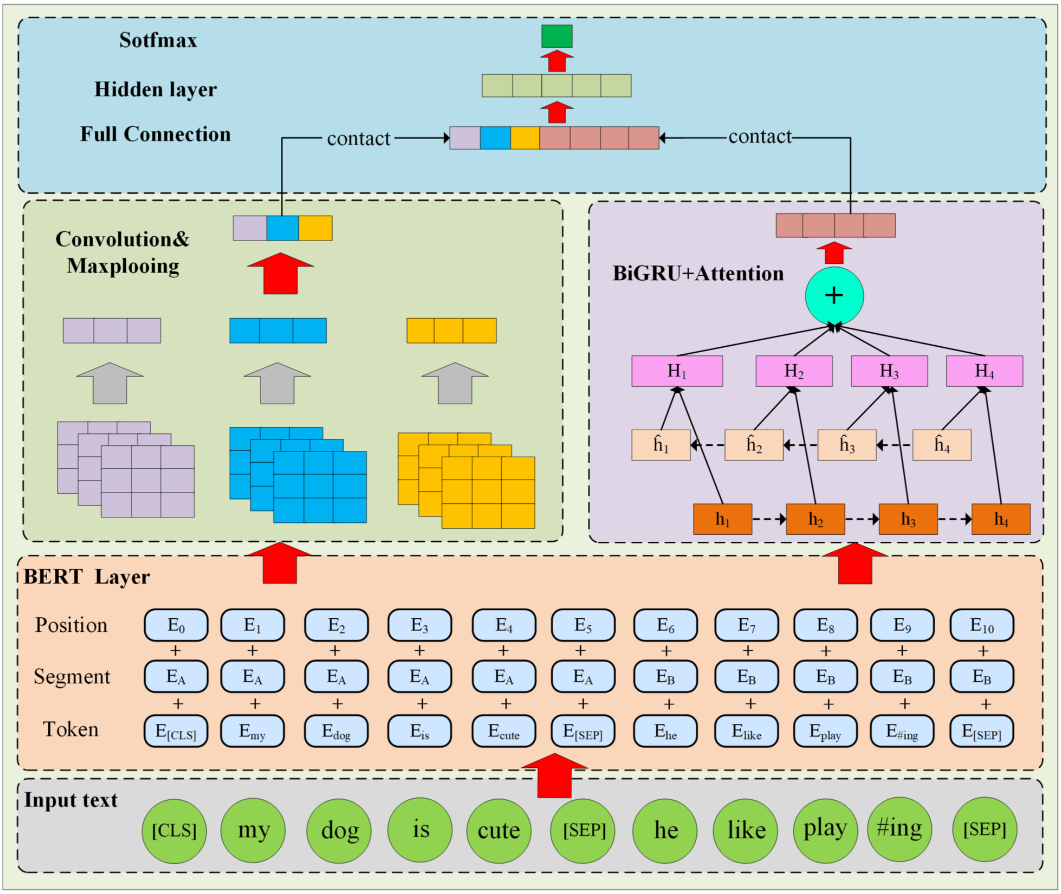
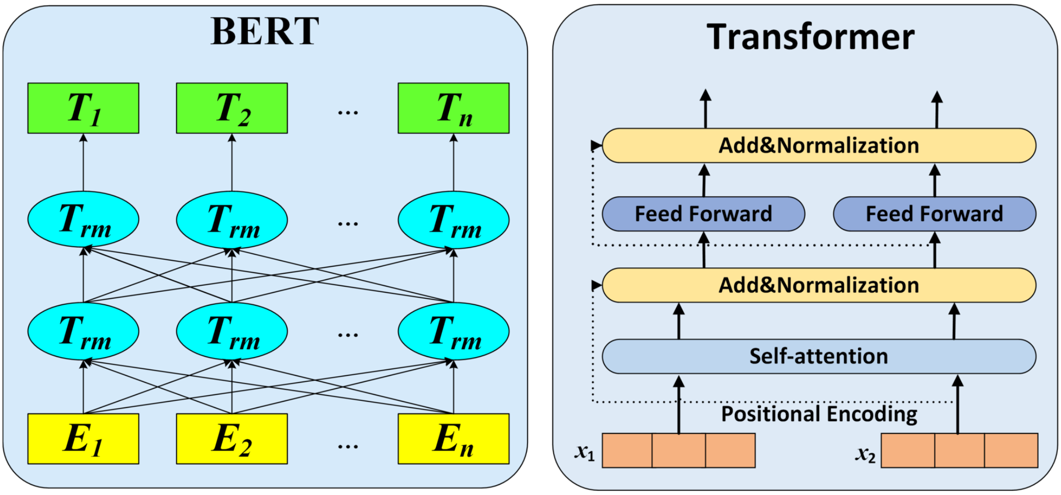

Sotfmax
Hidden layer
Full Connection
contact
contact
Convolution&
Maxplooing
BiGRU+Attention
+
H1
H2
H3
H4
ĥ1
ĥ2
ĥ3
ĥ4
h1
h2
h3
h4
BERT Layer
Position
Segment
Token
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 E10
EA EA EA EA EA EA EB EB EB EB EB
E[CLS] Emy Edog Eis Ecute E[SEP] Ehe Elike Eplay E#ing E[SEP]
Input text
[CLS] my dog is cute [SEP] he like play #ing [SEP]

Figure 2. The overall framework of BERT and the Transformer



transformer uses self-attention mechanism to replace the sequence structure of RNN, which effectively solving the long-term dependence problem in natural language processing. The structure of the BERT and Transformer is shown in Figure 2.

In the task of text classification, BERT will insert [CLS] as the starting symbol at the head of the input sentence, and insert the [SEP] symbol as the separator for multiple sentences. In addition, BERT uses position encoding to represent the position information of words as follows:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{1}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{2}$$

where pos is the position of the word in the sentence, i is a certain dimension of the word vector, *2i* is an even dimension, *2i*+1 is an odd dimension, and model is the dimension of the word vector.

As mentioned above, the word encoding of BERT can be divided into three parts, namely, word vector (token embedding), sentence vector (segment embedding), and position vector (position embedding). Then, each word will be transformed into matrix vectors *Q, K,* and *V* through different linear transformations, and calculated independently of each other to obtain the association relationship between words as follows:

$$\text{Attention}\left(Q,K,V\right) = \text{sof}tmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

where *Q*, *K*, and *V* are the query matrix, keyword matrix, and value matrix, respectively. The dimensions of the *Q* matrix and *K* matrix are d_k, and *T* is the transposed matrix.

3.2. CNN Model

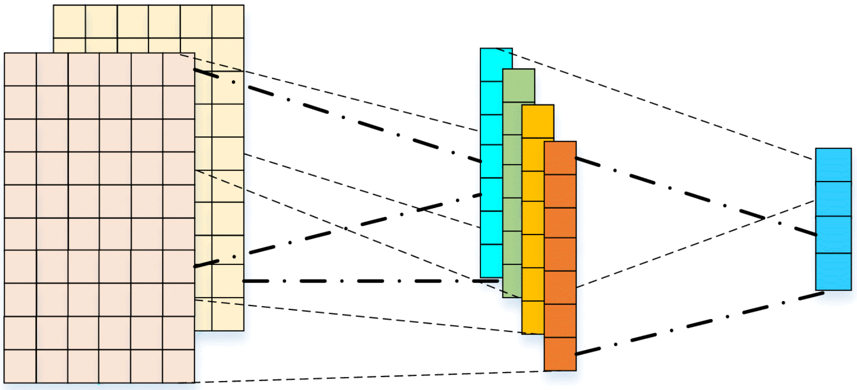
CNNs have made great progress in the field of machine vision. Moreover, they have also gradually gained ground in the field of NLP. The structure of a CNN mainly includes an input layer, convolutional layer, and pooling layer. In the task of text classification, the input layer is a word vector matrix, and the convolutional layer uses convolution kernels of different sizes to perform convolution operations on the word vector matrix to extract corresponding local features. The main function of the pooling layer is to reduce data dimensionality and prevent overfitting. The overall framework of a CNN is shown in Figure 3.

The specific calculation process of a CNN is as follows:

For a given input sentence word vector matrix $M \in R^{L\times d}$, *L* is the sentence length, *d* is the word vector dimension, and a convolution kernel w_i of different sizes is used to perform matrix M. The convolution operation is shown in Equation 4.

$$c = f\left(w_i \cdot M + b\right) \tag{4}$$

Figure 3. The overall framework of CNN



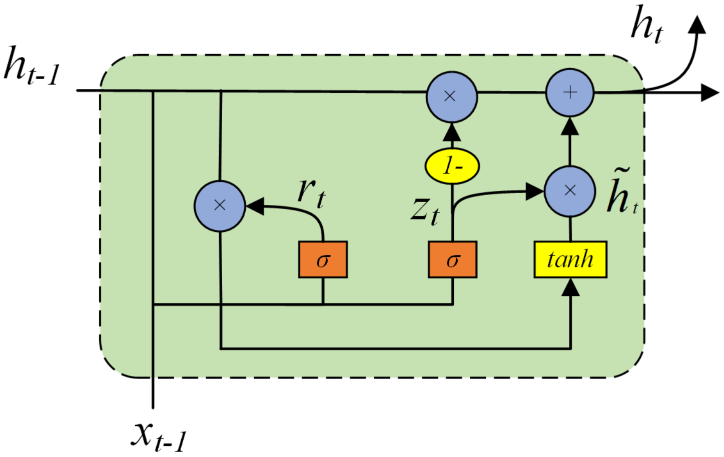
where f is the activation function of the convolutional layer and b is the bias term. In this article, the vector, after convolution, undergoes max-pooling to retain the maximum feature k as the input of the fully connected layer. The max-pooling layer calculation is shown in Equation 5.

$$\boldsymbol{k} = \boldsymbol{max}\left(\boldsymbol{c}_1, \boldsymbol{c}_2 \cdots, \boldsymbol{c}_n\right) \tag{5}$$

3.3. BiGRU Model

LSTM and GRUs, as variants of recurrent neural networks (RNNs), both introduce a special gated structure to avoid the problems of gradient disappearance and gradient explosion that exist in traditional RNNs. GRU optimizes the gate structure based on LSTM and simplifies the input gate, forget gate, and output in LSTM into update gates and reset gates, which reduces the number of model parameters and increases computing efficiency. The structure of GRU is shown in Figure 4.

Figure 4. The structure of GRU



where r_t represents the reset gate, which determines the degree to which the previous neuron information is forgotten. Z_t represents the update gate, which determines how much past information should be passed to the future, the model is updated as follows:

$$r_t = \sigma\left(W_r \cdot \left[h_{t-1}, x_t\right]\right) \tag{6}$$

$$z_t = \sigma\left(W_z \cdot \left[h_{t-1}, x_t\right]\right) \tag{7}$$

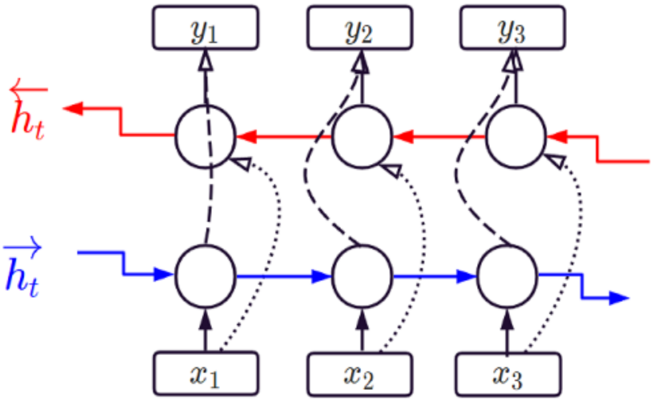
$$\tilde{h}_t = \tanh\left(W \cdot \left[r_t * h_{t-1}, x_t\right]\right) \tag{8}$$

$$h_t = \left(1 - z_t\right) * h_{t-1} + z_t * \tilde{h}_t \tag{9}$$

where σ is the sigmoid activation function; W_r, W_z ,and W are the weight matrices of the reset gate, update gate and hidden layer, respectively; x_t is the input of the model at time t; h_{t-1} is the state of the hidden layer at time t-1; $\tilde{h}_t$ is the sum of the past and currently hidden layer state at time t; and h_t is the hidden layer output.

The one-way GRU model is always output from the front to the back, and cannot capture the contribution of the following information to the semantics. In this paper, we use BiGRU to obtain text information from two different directions and jointly use them as the final output. The structure of the BiGRU is shown in Figure 5.

Figure 5. The framework of the BiGRU



As seen in Figure 5, the current hidden layer state of the BiGRU is determined by the current input x_t, the forward hidden layer state output $\underset{h_{t-1}}{\rightarrow}$ and the reverse hidden layer state output $\underset{h_{t-1}}{\leftarrow}$ at time t-1. The BiGRU is spliced by two unidirectional GRUs, and the hidden layer state of the BiGRU at time t can be obtained by the weighted summation of $\underset{h_{t-1}}{\rightarrow}$ and $\underset{h_{t-1}}{\leftarrow}$:

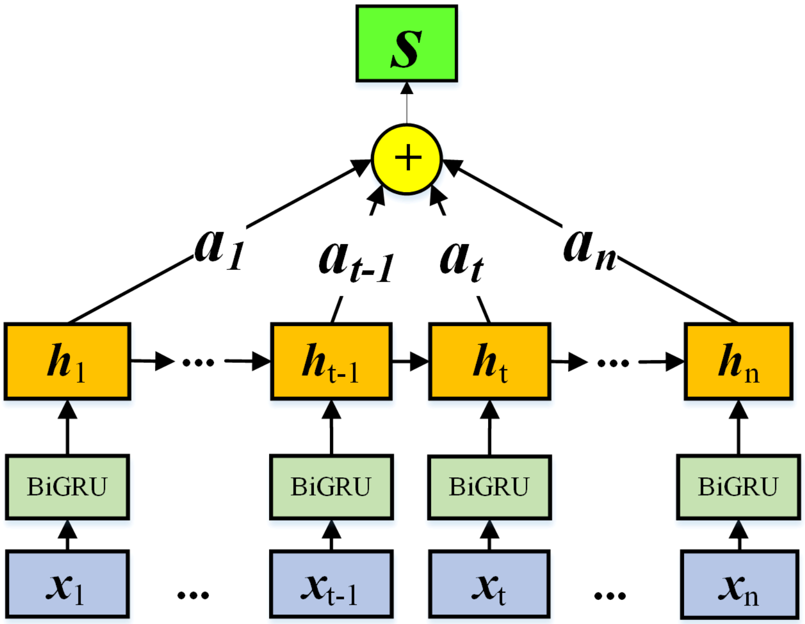
$$h_t = w_t \cdot GRU\left(x_t, \vec{h}_{t-1}\right) + v_t \cdot GRU\left(x_t, \overleftarrow{h}_{t-1}\right) + b_t \tag{10}$$

where the GRU() function represents the nonlinear transformation of the input word vector and encodes the word vector into the corresponding GRU hidden layer state. w_t and v_t represent the weights corresponding to the hidden layers $\underset{h_t}{\rightarrow}$ and $\underset{h_t}{\leftarrow}$, respectively, and b_t represents the bias corresponding to the hidden layer state at time *t*.

3.4. Attention Mechanism

In short text classification, each word has a different contribution to text features, especially verbs and nouns usually occupy a large proportion of text meaning. The attention mechanism can increase the weight coefficient of such keywords in the classification process and obtain better results. The structure of the attention mechanism is shown in Figure 6.

Figure 6. The overall framework of the attention mechanism



where the word vectors x_1, x_{t-1}, x_t, and x_n output the corresponding hidden layer feature vectors h_1, h_{t-1}, h_t, and h_n through the BiGRU and then multiply them with the weight coefficients a_1, a_{t-1}, a_t, and a_n and accumulate them as the output *V* of the attention layer. The calculation process is shown in Equations 11–13.

$$\boldsymbol{e}_i = \boldsymbol{v}_i \tanh\left(\boldsymbol{w}_i \boldsymbol{h}_i + \boldsymbol{b}_i\right) \tag{11}$$

$$\boldsymbol{a}_i = \frac{\exp\left(\boldsymbol{e}_i\right)}{\sum_{j=1}^{n}\left(\boldsymbol{e}_j\right)} \tag{12}$$

$$V = \sum_{i=1}^{t} \boldsymbol{a}_i \boldsymbol{h}_i \tag{13}$$

where $\boldsymbol{e}_i$ is the hidden layer state vector. $\boldsymbol{w}_i$ and $\boldsymbol{v}_i$ are the weight coefficient matrices at the time i. $\boldsymbol{b}_i$ is the bias. $\boldsymbol{a}_i$ is the attention score of the word, which determines the importance of the word to the sentence. $\boldsymbol{V}$ is the output vector of the cumulative summation of the weights of each word.

4. EXPERIMENTS

In this section, experiments are conducted on the Chinese News short text dataset to evaluate the performance of the proposed model. The dataset, parameter settings, evaluation metrics and baseline methods used in the experiment are introduced, and then the experimental results are discussed.

4.1. Dataset

The experiment uses the THUCNews (M. Sum, J., et al., 2016) short text dataset to train the classification model. The THUCNews is generated by filtering historical data of Sina News RSS subscription channels from 2005 to 2011. It contains 740,000 news documents (2.19 GB), all in UTF-8 plain text format. This paper uses 10 categories, finance, realty, stocks, education, science, society, politics, sports, games, and entertainment. Each category contains 20000 samples. After word segmentation, the length of the text sequence ranges between 20–30, and the details of the dataset are shown in Table 1.

Table 1. Dataset

Total	Training	Validation	Test	Category	Length(max)	Vocabulary_Size
200000	180000	10000	10000	10	30	4762

4.2. Experimental setup

The different word vectors directly affect the classification results. In this paper, all baseline models are compared using Word2Vec and BERT to train word vectors. The word vector dimension of Word2Vec is set to 300 and set to 768 of BERT. The batch size is set to 128, and the dropout is 0.5. Other parameters in the hybrid neural network layer are given in Table 2.

Table 2. Parameter's settings

Hyperparameter	Value
Number of convolution kernels	256
Filter size	(2,3,4)
Learning rate	3e-5
Optimizer	Adam
Epoch	10
Batch size	128
BiGRU hidden size	128
BiGRU hidden layer	2

4.3. Evaluation Metrics

The classification results in the experiment are evaluated using P(precision), R(recall), and F1(micro average). The equations of these parameter are defined as:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \ \mathrm{FP}} \tag{14}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \ \mathrm{FN}} \tag{15}$$

$$\mathrm{F1} = \frac{2 \times P \times R}{P + R} \tag{16}$$

where *TP*(true positives) represents the number of samples that are positive and judged as positive by the classifier. *FP*(false positives) represents the number of samples that are negative but judged as positive by the classifier. *FN*(false negative) represents the number of samples that are positive but judged as negative by the classifier. F1 is used to comprehensively average P and R.

4.4. Model Comparison

To verify the effectiveness of the model proposed in this paper, we compared it to the following current state-of-the-art text classification baselines:

(1) FastText: The word vector and text classification tool proposed by (Joulin, A., et al., 2016),which uses character-level n-grams to represent a word, and introduces a hierarchical SoftMax level to speed up the calculation process.
(2) CNN: The most traditional CNN model in text classification proposed by (Kim, Y. 2014), and many researchers have revamped it to further improve their accuracy.
(3) BiGRU: An improved model based on RNN (Liu, P., et al., 2016), which combines forward GRU and backward GRU, and finally merges bidirectional sequence features as output.

(4) DPCNN: The word-level deep pyramid CNN model for text classification proposed by (Johnson, R., & Zhang, T. 2017). This method can effectively model the long-term dependence in the text.
(5) RCNN: (Lai, S., et al., 2015) proposed a fusion model that uses LSTM or GRU to replace the convolutional layer, and then combined it with a pooling layer for classification.
(6) Transformer (Vaswani, A., et al., 2017): This method uses self-attention mechanism to calculate the score between each word in the sentence and adds position coding to retain sequence information.

We also horizontally compared with other models experimented on the data set of this paper. Including the LibSVM method used in THUCTC (M. Sum, J., et al., 2016); The Word2Vec model with the improved TFIDF algorithm for combining weights proposed by Chen, Z. (2019); The model based on bidirectional temporal convolutional network and attention mechanism (Bi-TCA) proposed by (Zuo, Y., et al., 2020). The NIN+dropout model proposed by (Fu, Y.-P., et al., 2018); A hybrid model based on Ernie CNN and bilstm attention (MEBCA) proposed by (Zhaoye, X., et al., 2021).

4.5. Results and Analysis

4.5.1 Overall Comparison

This section compared the classification results of baseline models and the methods proposed by other researchers. In particular, for baseline methods, we compared the effects of Word2Vec and BERT word vector models on classification performance. After conducting multiple rounds of experiments, The comparison results of baseline methods that use Word2Vec training word vector are shown in Table 3. The horizontal comparison results based on the BERT model are shown in Table 4. The F1 value of each model on the dataset is shown in Figure 7. Finally, the specific performance of the model in each category is shown in Table 5 and Figure 8.

Table 3. Model comparison results based on Word2Vec

Type	Model	P	R	F1
	FastText	**91.48%**	**91.43%**	**91.45%**
Word2Vec+baseline methods	Transformer	89.31%	89.13%	89.22%
	CNN	91.06%	91.02%	91.04%
	DPCNN	91.05%	90.97%	91.23%
	RCNN	91.32%	91.30%	91.28%
	BiGRU	90.80%	90.73%	90.76%
	BiGRU +Attention	90.65%	90.48%	90.56%
	CNN+BiGRU+Attention	91.79%	91.61%	91.70%

As in Table 3, when using Word2Vec to train word vectors, CNN+BiGRU+Attention achieved better results than other baseline models. Compared to the variants of the single GRU and CNN models, CNN+BiGRU+Attention performs better. This is because CNN+BiGRU+Attention takes the respective advantages of CNN and BiGRU networks to combine the local and contextual information features of the question sentence. In addition, the introduction of the attention mechanism highlights the important words in the sentence, which has a significant effect on the improvement of the classification results. Furthermore, the two models based on CNN are even better than the BiGRU. This proves that when Word2Vec is used as word vector, the BiGRU cannot utilise its advantages in long-distance feature extraction when the text sequence is short, but CNN can quickly converge the

Table 4. Overall model comparison results based on BERT

Type	Model	P	R	F1/Accuracy
pure	**BERT**	**93.68%**	**93.65%**	**93.66%**
BERT+baseline methods	CNN	94.06%	93.98%	94.02%
	DPCNN	94.04%	93.99%	94.01%
	RCNN	94.51%	94.48%	94.49%
	BiGRU	94.22%	94.20%	94.21%
	BiGRU+Attention	94.37%	94.31%	94.34%
Other models	LibSVM(M. Sum, J., et al., 2016)	88.60%	82.90%	85.60%
	TFIDF+word2vec Chen, Z. (2019)	94.26%	94.57%	94.41%
	Bi-TCA(Zuo, Y., et al., 2020)	×	×	91.42%
	NIN+dropout(Fu, Y.-P., et al., 2018)	×	×	94.87%
	MECBA(Zhaoye, X., et al., 2021)	94.71%	94.69%	94.61%
Ours	BCBGA	**95.06%**	**95.02%**	**95.04%**

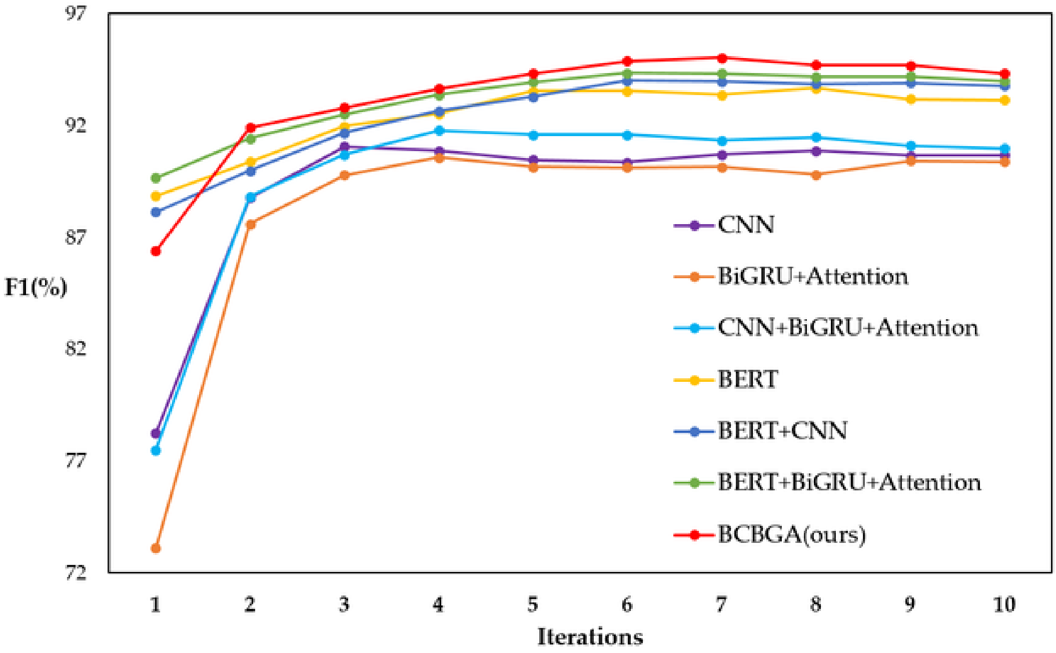
model through a convolution operation and extract local key features, which makes the CNN more suitable for short text classification than sequence networks.

Table 4 shows the classification results of the model based on BERT. The results indicate that the method using BERT as the word vector has a dominant performance in short text classification. The precision and F1 value of the pure BERT baseline model have reached 93.68% and 93.66% respectively. Compared to the best model, the CNN+BiGRU+Attention discussed in the previous discussion, it improved by 1.89% and 1.96% respectively. At the same time, in comparison to with BERT+CNN, the precision and F1 value of BERT+ BiGRU increased by 0.16% and 0.19%, respectively. This proves that when using BERT as the word vector, GRUs showed better adaptability than CNNs. From another perspective, this also indicates that sequence models such as GRUs are more sensitive to different word vector representations. In comparison with the models proposed by other researchers, our model relative improved the F1 value and accuracy of LibSVM and Bi-TCA models by 9.44% and 3.62%, respectively. NIN+dropout and MECBA also have very competitive classification performance with the accuracy and F1 value reaching 94.87% and 94.61% respectively. But in the end, the BCBGA proposed in this paper obtain the best performance with the Precision and F1 value reaching 95.06% and 95.04%, respectively.

Combining Table 3 and 4. In comparison with CNN+BiGRU+Attention, the BCBGA model uses BERT as the word vector significantly improves the classification performance. This proves that the word vector generated by BERT has better characterization capabilities than Word2Vec. Compared with BERT+CNN and BERT+BiGRU+Attention, BCBGA increased the F1 value by 1.02% and 0.7%, respectively. This is because in short text classification, although CNN cannot capture the position information of text sequences, it can extract important local features in sentences through convolution and pooling, which is helpful for improving classification accuracy. Moreover, BiGRU+Attention can supplement the sequence information of the sentence and highlight the vocabulary that has a crucial influence on semantics, which is another reason to improve the classification performance of BCBGA. In summary, the above experiments prove that all of the components of BCBGA can have a positive influence on the classification results.

Figure 7 shows that selecting BERT as the word vector representation significantly improved the effect at the beginning of training. Among them, The CNN has the fastest convergence speed and converges in epoch 3. However, the BCBGA model proposed in this paper started to take the

Figure 7. The F1 value of each model on the THUCNews dataset



lead in epoch 2 and continued to lead in the subsequent training process. Finally, the BCBGA model reached the highest F1 value in epoch 7, and then as the training time increased, the performance no longer improved. Overall, the F-value curve of BCBGA is relatively smooth, the training process is stable and accompanied by a high accuracy rate.

Table 5. F1 value of the six models in 10 categories

Categories	Word2Vec			BERT		
	CNN	BiGRU +Attention	Hybrid*	CNN	BiGRU +Attention	Hybrid* (ours)
finance	90.23%	90.23%	91.09%	93.33%	93.72%	**94.43%**
realty	92.82%	91.80%	92.12%	93.94%	95.39%	**95.91%**
stocks	86.47%	84.12%	84.98%	89.47%	90.56%	**90.93%**
education	95.28%	94.28%	95.90%	96.75%	**97.02%**	96.69%
science	86.13%	85.19%	86.68%	90.42%	90.85%	**91.47%**
society	90.57%	89.82%	91.10%	93.46%	94.45%	**94.96%**
politics	89.70%	87.91%	89.95%	92.90%	92.42%	**93.73%**
sports	95.52%	97.36%	97.84%	98.25%	98.65%	**98.95%**
games	91.01%	92.97%	93.43%	95.59%	96.04%	**96.64%**
entertainment	92.35%	93.27%	93.83%	95.64%	95.70%	**96.61%**

*Hybrid: CNN+BiGRU+Attention model

Figure 8. Confusion matrices for different models

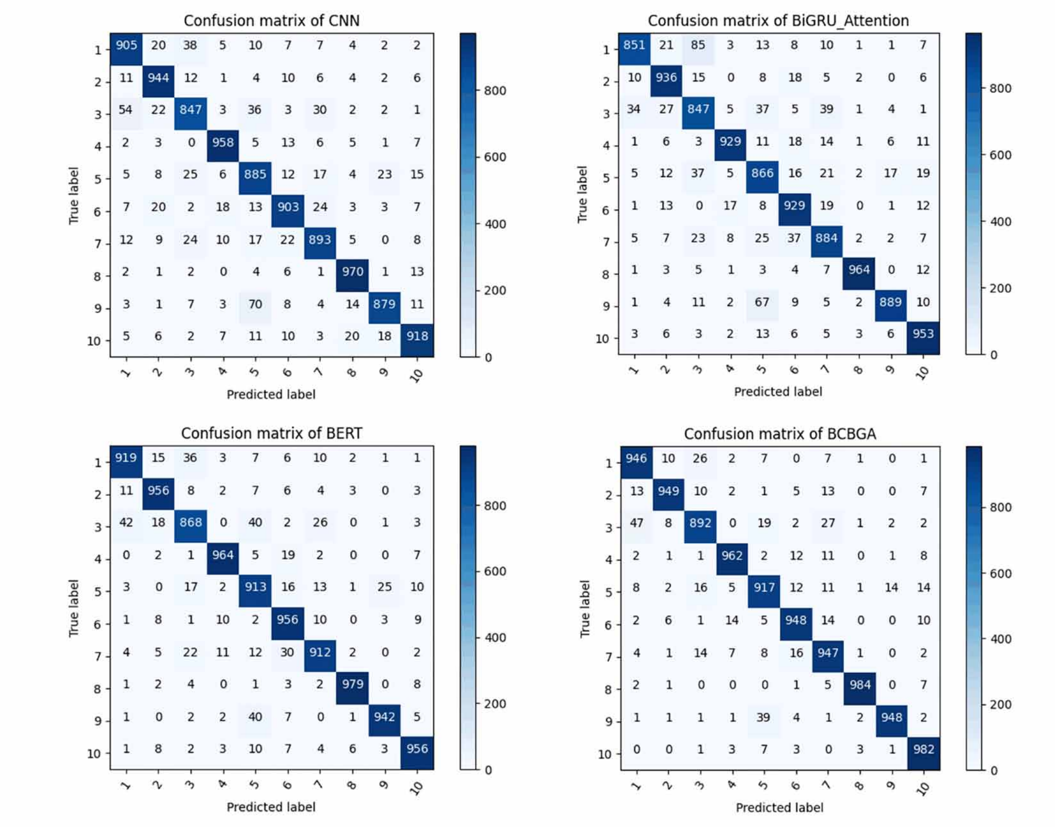


Table 5 shows the detailed performance of the different models in 10 domain categories. it is visible that BCBGA still achieved superior performance in all categories except education. In the experiment with Word2Vec as the word vector, the performances of CNN and BiGRU+Attention are very close in each category. The hybrid model, CNN+BiGRU+Attention, further improves the classification accuracy, but the improvement is limited. After using BERT as the word vector, the performance of either a single network or a hybrid model in each category has been greatly improved.

The confusion matrix in Figure 8 shows the classification of 1000 samples in the test set. The classification results of CNN and BiGRU+Attention are relatively scattered, and the identified labels are not balanced, especially in categories 3 (stocks), 5 (science), and 7 (politics). However, this situation has been changed in the model based on BERT. This is because the BERT model's powerful representation ability can distinguish the meaning of words in different sentences. In short texts, the meaning of the words has a major effect on the classification results. Hence, the confusion matrix shows that while BERT improves the classification accuracy, it also increases the concentration of classification, especially for categories 8 (sports), 9 (games), and 10 (entertainment).

4.5.2 Classification Effect of Different BCBGA Parameters

The setting of hyperparameters in the neural network has an important influence on the final experimental results. To further improve the performance of the BCBGA model, the convolution kernel size, filter width, number of GRU hidden layers, learning rate, and training set size are further explored. The hidden size of BiGRU is set to 128, the number of convolution kernels is set to 256, and other parameters remain unchanged. All parameters are evaluated for multiple rounds of experiments, The average result is as follows:

As shown in Table 6, when using a single convolution kernel, the classification performance of the model worsens. As the size of the convolution kernel increases, the accuracy of the model increases. When the size of the convolution kernel reaches 3, the accuracy of the model reaches its highest. After that, increasing the size of the convolution kernel no longer improves the performance of the model. Through further analysis, it is known that the size of the convolution kernel used in the model is directly proportional to the number of captured features, which means the larger the convolution kernel used in the model, the more comprehensive the information obtained, and the classification performance is improved accordingly. However, as the convolution kernel increases, the number of parameters also increases, which causes the model to use more memory and training time, and the accuracy of the model does not see much improvement. In view of this case, 3 kernel was selected for training in the BCBGA algorithm model.

Table 6. Classification performance of different convolution kernel sizes

kernel size	P	R	F1
1	90.70%	90.16%	90.43%
2	94.22%	94.19%	94.20%
3	95.06%	95.02%	95.04%
5	94.61%	94.59%	94.37%
7	94.59%	94.58%	93.88%

Table 7. Classification performance of different convolution filter widths

filter width	P	R	F1
[2,3,4]	94.78%	94.75%	94.76%
[3,4,5]	95.06%	95.02%	95.04%
[4,5,6]	94.13%	94.06%	94.89%
[5,6,7]	94.44%	94.43%	94.43%

The size of the convolution filter determines how many spatial features can be obtained in one convolution, and thus, choosing a suitable filter size has an important influence on the effect of convolution. Generally, combining convolution kernel sizes with similar results produces a better performance(Li, X., et al., 2021). Therefore, the combination of convolution filters in this paper includes 4 groups, the widths are [2,3,4], [3,4,5], [4,5,6], and [5,6,7]. The results of different combinations are shown in Table 7.

As can be seen in Table 7 that the convolution filter has the best performance when the width is set to [3,4,5]. Although the performance of other widths has slightly decreased, it still maintains an F1 value of over 94.43%. This also proves that when the convolution kernel size is set to 3, the model has excellent overall performance.

Table 8. Classification performance of different GRU layers

BiGRU hidden layer	P	R	F1
1	94.39%	94.37%	94.38%
2	95.06%	95.02%	95.04%
3	93.95%	93.93%	93.94%

Table 9. Classification performance of different learning rates

learning rate	P	R	F1
0.0001	0.00%	0.00%	0.00%
0.00001	95.03%	95.01%	95.02%
0.00003	95.06%	95.02%	95.04%
0.00005	95.02%	94.99%	95.00%
0.000001	94.76%	94.70%	94.73%

Table 10. Classification performance of different training data size

datasize	P	R	F1
10k	90.27%	90.22%	90.24%
50k	92.17%	92.15%	92.16%
100k	94.73%	94.70%	94.71%
150k	94.98%	94.96%	94.97%
180k	95.06%	95.02%	95.04%

Table 8 shows the influence of different GRU layers on the classification results. The classification performance is best when the number of hidden layers is set to 2, followed by 1. As the hidden layer is increased to 3, the classification performance is significantly decreased. This is because the increase in the hidden layer introduces more parameters and calculations, which hurts the fitting of the model.

The learning rate is an important hyperparameter in supervised learning and deep learning, which determines whether the objective function can converge to a local minimum and when to converge to the minimum. If the learning rate is too large, the loss function may directly exceed the global optimum. If the learning rate is too small, the change speed of the loss function is very slow and it is easy to be trapped in the local minimum. Table 9 shows the contrast results of various learning rates. It can be seen that as the learning rate decreased, the F1 value of the model first increased and then decreased. When the learning rate is set to 0.00003, the F1 value of the model was the highest, which was selected in the BCBGA model for experimentation.

The size of the training dataset is another factor that determines the effect of model fitting. Table 10 shows the effect of the training data size on the classification performance in the THUCNews datasets. It can be seen that when the training data set size is 10k, the F1 value reaches 90.24%. The model is still fitting and there is still room for improvement. As the number of data increases to 50k and 100k, the F1 value increases by 1.92% and 4.47% respectively. When the dataset increases from 100k to 180k, the performance of the model is still improving but the rate of improvement has become slower. This indicates that the model has reached almost the best fit on this dataset, and increasing the size of the dataset can no longer effectively improve the classification performance.

5. CONCLUSIONS AND FUTURE SCOPE

Short texts usually contain only a few words with practical meaning, and these words play a crucial role in the classification process. In this paper, a multifeature fusion short text classification model based on BERT, BCBGA, is proposed. The method includes four components: BERT, CNN, BiGRU, and Attention mechanism. BERT is used to train dynamic word vectors to enhance the word representation ability of short texts. CNN captures the static information of the text from a spatial perspective and retains the most contributing features through max pooling. The BiGRU aids in learning the sequence features of sentences. An attention mechanism is introduced to highlight words that provide crucial cues for classification. The fully connected classification layer fuses multiple features to obtain the final classification result. Experiments of the proposed model are conducted on THUCNews short text datasets and compared with mainstream state-of-the-art text classification models. The experiments show that the proposed model achieves better classification performance compared to the baseline methods.

Future research will focus on the following aspects: (1) verifying the effectiveness of the method on multiple datasets; (2) adding an attention mechanism to the CNN and further optimizing the feature extraction layer; and (3) design models to improve parallelism and reduce training time.

Funding

This research was funded by National Social Science Foundation of China [grant numbers: 19BTQ032].

Declaration of Conflicting Interests

The authors declare no conflict of interest with respect to the research, authorship and/or publication of this article.

REFERENCES

Bijalwan, V., Kumar, V., Kumari, P., & Pascual, J. (2014). KNN based machine learning approach for text and document mining. *International Journal of Database Theory and Application*, *7*(1), 61–70.

Chen, Z. (2019). *Short text classification based on word2vec and improved TDFIDF merge weighting*. Paper presented at the 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE).

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). *Learning phrase representations using RNN encoder-decoder for statistical machine translation*. arXiv preprint arXiv:1406.1078.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *Bert: Pre-training of deep bidirectional transformers for language understanding*. arXiv preprint arXiv:1810.04805.

Fu, Y.-P., Liu, Y., & Zhang, Z.-J. (2018). Sentence Classification Using Novel NIN. *Journal of Computers*, *29*(5), 250–259.

Goudjil, M., Koudil, M., Bedda, M., & Ghoggali, N. (2018). A novel active learning method using SVM for text classification. *International Journal of Automation and Computing*, *15*(3), 290–298.

Guo, B., Zhang, C. X., Liu, J. M., & Ma, X. Y. (2019). Improving text classification with weighted word embeddings via a multi-channel TextCNN model. *Neurocomputing*, *363*, 366–374. doi:10.1016/j.neucom.2019.07.052

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Jang, B., Kim, M., Harerimana, G., Kang, S. U., & Kim, J. W. (2020). Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism. Applied Sciences-Basel, 10(17). doi. *Artn*, *5841*. Advance online publication. doi:10.3390/App10175841

Jiang, L., Li, C., Wang, S., & Zhang, L. (2016). Deep feature weighting for naive Bayes and its application to text classification. *Engineering Applications of Artificial Intelligence*, *52*, 26–39.

Jin, N., Wu, J. X., Ma, X., Yan, K., & Mo, Y. C. (2020). Multi-Task Learning Model Based on Multi-Scale CNN and LSTM for Sentiment Classification. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 77060–77072. doi:10.1109/Access.2020.2989428

Johnson, R., & Zhang, T. (2017). Deep pyramid convolutional neural networks for text categorization. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). *Bag of tricks for efficient text classification*. arXiv preprint arXiv:1607.01759.

Kiliroor, C. C., & Valliyammai, C. (2019). Social network based filtering of unsolicited messages from e-mails. *Journal of Intelligent & Fuzzy Systems*, *36*(5), 4037–4048. doi:10.3233/JIFS-169964

Kim, Y. (2014). *Convolutional Neural Networks for Sentence Classification*. arXiv:1408.5882. https://ui.adsabs.harvard.edu/abs/2014arXiv1408.5882K

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). *Recurrent convolutional neural networks for text classification*. Paper presented at the Twenty-ninth AAAI conference on artificial intelligence.

Li, X., Cui, M., Li, J., Bai, R., Lu, Z., & Aickelin, U. (2021). A hybrid medical text classification framework: Integrating attentive rule construction and neural network. *Neurocomputing*, *443*, 345–355.

Lin, Z., Jin, X. L., Xu, X. K., Wang, Y. Z., Cheng, X. Q., Wang, W. P., & Meng, D. (2016). An Unsupervised Cross-Lingual Topic Model Framework for Sentiment Classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*(3), 432–444. Advance online publication. doi:10.1109/TASLP.2015.2512041

Liu, B., Zhou, Y., & Sun, W. (2020). Character-level text classification via convolutional neural network and gated recurrent unit. *International Journal of Machine Learning and Cybernetics*, *11*(8), 1939–1949. doi:10.1007/s13042-020-01084-9

Liu, P., Qiu, X., & Huang, X. (2016). *Recurrent neural network for text classification with multi-task learning*. arXiv preprint arXiv:1605.05101.

Luo, L. X. (2019). Network text sentiment analysis method combining LDA text representation and GRU-CNN. *Personal and Ubiquitous Computing*, *23*(3-4), 405–412. doi:10.1007/s00779-018-1183-9

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). *Distributed representations of words and phrases and their compositionality*. Paper presented at the Advances in neural information processing systems.

Mnih, V., Heess, N., & Graves, A. (2014). *Recurrent models of visual attention*. Paper presented at the Advances in neural information processing systems.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). *Deep contextualized word representations*. arXiv preprint arXiv:1802.05365.

Qiao, X., Peng, C., Liu, Z., & Hu, Y. F. (2019). Word-character attention model for Chinese text classification. *International Journal of Machine Learning and Cybernetics*, *10*(12), 3521–3537. doi:10.1007/s13042-019-00942-5

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving language understanding by generative pre-training*. Academic Press.

Ren, Y. F., Wang, R. M., & Ji, D. H. (2016). A topic-enhanced word embedding for Twitter sentiment classification. *Information Sciences*, *369*, 188–198. doi:10.1016/j.ins.2016.06.040

Sulthana, A. R., & Ramasamy, S. (2019). Ontology and context based recommendation system using Neuro-Fuzzy Classification. *Computers & Electrical Engineering*, *74*, 498–510. doi:10.1016/j.compeleceng.2018.01.034

Sum, M., Li, J., Guo, Z., Zhao, Y., Zheng, Y., Si, X., & Liu, Z. (2016). *Thuctc: An efficient Chinese text classifier*. GitHub Repository. Available: https://github.com/thunlp/THUCTC

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). *Attention is all you need*. Paper presented at the Advances in neural information processing systems.

Xu, W. H., Huang, H., Zhang, J., Gu, H., Yang, J., & Gui, G. (2019). CNN-based Skip-Gram Method for Improving Classification Accuracy of Chinese Text. *Transactions on Internet and Information Systems (Seoul)*, *13*(12), 6080–6096. doi:10.3837/tiis.2019.12.016

Yu, C. T., & Salton, G. (1976). Precision weighting—An effective automatic indexing method. *Journal of the Association for Computing Machinery*, *23*(1), 76–88. doi:10.1145/321921.321930

Zeng, S., Ma, Y., Zhang, X., & Du, X. (2020). Term-based pooling in convolutional neural networks for text classification. *China Communications*, *17*(4), 109–124.

Zhang, D. J., Hong, M. B., Zou, L., Han, F., He, F. Z., Tu, Z. G., & Ren, Y. F. (2019). Attention Pooling-Based Bidirectional Gated Recurrent Units Model for Sentimental Classification. *International Journal Of Computational Intelligence Systems*, *12*(2), 723–732. doi:10.2991/ijcis.d.190710.001

Zhang, Y. S., Zheng, J., Jiang, Y. R., Huang, G. J., & Chen, R. Y. (2019). A Text Sentiment Classification Modeling Method Based on Coordinated CNN-LSTM-Attention Model. *Chinese Journal of Electronics*, *28*(1), 120–126. doi:10.1049/cje.2018.11.004

Zhaoye, X., Xiaoqun, L., & Peijie, S. (2021). *Hybrid Chinese text classification model based on pretraining model.* Paper presented at the Journal of Physics: Conference Series.

Zuo, Y., Jiang, L., Sun, H., Ma, C., Liang, Y., Nie, S., & Zhou, Y. (2020). *Short text classification based on bidirectional TCN and attention mechanism.* Paper presented at the Journal of Physics: Conference Series.