Improved Chinese Short Text Classification Method Based on ERNIE_BiGRU Model

Xue Cheng, Chen Zhang and Qingxu Li
School of Cyber Security, Gansu University of Political Science and Law, Lanzhou 730070 China
1726899676@qq.com

Abstract. This paper is aimed at traditional word embedding models and Bidirectional Encoder Representations from Transformers (BERT) that cannot learn text semantic knowledge, as well as convolutional neural network (CNN) and Bidirectional long short-term memory (BiLSTM) unable to distinguish the importance of words, proposing an improved Chinese short text classification method based on ERNIE_BiGRU model. Firstly, learning text knowledge and information through the Enhanced Representation through Knowledge Integration (ERNIE) enhances the model's semantic representation capabilities. Secondly, considering that CNN can only extract local features of the text while ignoring the semantic relevance between contextual information, and the Bidirectional Gating Recurrent Unit (BiGRU) is simpler, has fewer network parameters and faster calculation speed than the BiLSTM, the combination of CNN and BiGRU enables the model to capture both local phrase-level features and contextual structure information. Finally, according to the importance of features, the attention mechanism is used to assign different weights to improve the classification effect of the model. The experimental results show that the ERNIE_CNN_BiGRU_Attention (ECBA) model used in this paper has achieved good results in the task of Chinese short text classification.

1. Introduction
Text classification is the most common and important type of task in Natural Language Processing (NLP). In today’s era of advanced Internet technology, a large amount of data is generated at all times, many of which include a large amount of valuable short text data. For example, user preferences, needs and other related information can be extracted from common short text data such as Weibo and Twitter, and then complete the sentiment analysis, recommendation system, question answering system and other tasks. Compared with long texts, short text features are sparser and more context-dependent, so the processing of short text data is a very meaningful task. The traditional method for text classification tasks is to use machine learning (ML) algorithms. In 2002, Pang et al[1] firstly proposed to use ML to solve sentiment classification tasks and improved the classification results. The words in the text can be converted into multi-dimensional word vectors by ML, these word vectors contain n-grams, part of speech, sentiment and other features, and then these multi-dimensional word vectors are introduced into the classification model to learn classification features and perform classification. This method is not only simple but also effective. However, there are still shortcomings that the semantic relationship between the text and the dependence relationship between words can’t be reflected[2]. In response to the above problems, Bengio et al[3] in 2003 proposed a method to represent text words with real number vectors, which contain the positional relationship and semantic information between words in this
method. In 2013, Mikolov et al[4] proposed a shallow and two-layer neural network to train and generate word vectors, including continuous bag-of-words (CBOW) and Skip-gram models. Because part of the features will be lost after word segmentation in the text pre-processing stage, this static word embedding method can’t fully express the original semantic information[5]. Therefore, the semantic representation of text information greatly affects the effect of subsequent tasks. At present, pre-training models based on large corpora have gradually been widely used in NLP tasks. Considering that static word vectors can’t be changed according to different language environments, leading to the emergence of polysemy, PETERS M et al[6] proposed Embedding from Language Model (ELMo), which captures semantic features at the high level and grammatical features at the bottom level. On this basis, Google proposed the BERT model. In 2018, J Devlin[7] and others used a large-scale database to train, so that BERT achieves the optimal result. This paper uses ERNIE to complete the semantic representation of the text. This model improves the pre-training logic of the BERT model and uses a large amount of Chinese corpus for training, making the model more suitable for various Chinese text tasks.

The main contributions of this paper are as follows:

- Considering that traditional word embedding models have lost the original semantic information and the BERT model can only learn contextual semantic information, this paper uses ERNIE, which can learn semantic knowledge, to enhance the capabilities of semantic representation of the model.
- Considering that the BiGRU is simpler, has fewer network parameters and faster calculation speed than the BiLSTM model, this paper uses GRU instead of LSTM, and in order to obtain context structure information, BiGRU is used instead of the GRU.
- Considering that the CNN and BiGRU will extract different features, this paper combines the two models and uses the attention mechanism to assign weights to different features.

2. Literature review

With the continuous deepening of research, many methods in the image field are used in NLP. In 2014, Kim Y et al[8] used pre-trained Word2Vec word vectors combined with multiple convolution kernels of different sizes to extract features and verified its good performance in multiple NLP tasks. Although the CNN used by Kim Y et al. can extract more comprehensive text features, it still fails to focus on the dependencies between contexts. In 2015, Zhang et al[9] considered that using words as the smallest unit for semantic representation has certain drawbacks, because different word segmentation rules can get different classification results. For example, the sentence "南京市长江大桥" can be divided into "南京市长/江/大桥" or "南京市/长江大桥", thus proposed a CNN model with characters as input for text classification, making the model more versatile.

In 2016, Liu et al[10] proposed different sharing mechanisms for different text classification tasks based on the RNN model, and achieved good classification results. Compared with CNN, RNN can learn contextual semantic information, but if the input sequence is too long, RNN will have the disadvantages of vanishing gradient or exploding gradient. In response to this problem, Hochreiter and Schmidhuber[11] improved the structure of RNN to produce LSTM and GRU. In 2018, Li Yang and others[12] concatenate the features extracted by CNN and BiLSTM, combined the advantages of the features extracted by the two models, and greatly improved the classification performance of the model. Considering that the importance of information in the text is different to the classification results, important words should be focus on more attention. Therefore, Bahdanau et al. were the first to use attention in the field of NLP. In 2017, the Google team used self-attention to learn text representation[13]. As the name suggests, self-Attention is to calculate the weight of oneself to oneself, in order to more fully consider the semantic and grammatical relationship between contexts.

3. Model construction

The ERNIE_CNN_BiGRU_Attention (ECBA) model used in this article mainly includes the embedding layer, CNN layer, BiGRU layer, and Attention layer. The overall framework of the model is shown in Figure 1.
ERNIE uses a large corpus for training to learn the relationship between words, entities and entities in the corpus, thereby learning the semantic knowledge information in the text. This model is mainly to improve the mask strategy in BERT. Figure 2 and Figure 3 show the Mask strategy of the BERT and ERNIE models respectively.

Figure 1. ECBA model framework

3.1. embedding layer
ERNIE uses a large corpus for training to learn the relationship between words, entities and entities in the corpus, thereby learning the semantic knowledge information in the text. This model is mainly to improve the mask strategy in BERT. Figure 2 and Figure 3 show the Mask strategy of the BERT and ERNIE models respectively.

Figure 2. BERT's Mask strategy

Figure 3. ERNIE's Mask strategy
The BERT model hardly pays attention to semantic knowledge, especially in Chinese texts. The BERT model can determine that the masked word is [肃] through the co-occurrence of [甘] and [省], but the model has not learned the knowledge about Gansu Province. By learning the expression of words and entities, the ERNIE model can learn the relationship between [敦煌] and [甘肃省], such as [敦煌] is...
located in [甘肃省] and [敦煌] is a national historical and cultural city. Comparing with the BERT model that can only learn text semantic information, the ERNIE model introduces knowledge information, which enhances the semantic expression ability of the model.

ERNIE1.0 made the following improvements to the Mask strategy of the BERT model:
- Mask based on phrase level.
- Mask based on entity level.

The entities here include nouns such as person's name, location, organization, product, etc.

3.2. CNN layer

CNN is mainly composed of convolutional layer, pooling layer, and fully connected layer. The convolution layer can use multiple convolution kernels of different sizes to complete feature extraction. The pooling layer retains only the main features and reduces network parameters and calculations. The fully connected layer is used as a classifier, as shown in Figure 4.

![Figure 4. TextCNN model](image)

This paper firstly uses ERNIE to represent the text into matrix form. Considering that the pooling operation may lose part of the feature information, this paper only uses the main structure convolution layer of CNN to extract local word features, the size of the convolution kernel is \([2, 3, 4, 5]\). After many experiments, it is concluded that the model classification effect is best when the convolution kernel size is 5.

3.3. BiGRU layer

Aiming at the problem of vanishing gradient of the long sequence in RNN, LSTM has been proposed, which feedback connections between hidden units that are associated with discrete time steps, which allow long-term sequence dependencies to be learned and a transaction label to be predicted given the sequence of past transactions. LSTM unit consists of a memory cell that stores information which is updated by three special gates: the input gate, the forget gate and the output gate[14]. Chao et al[15] proposed GRU that also uses a gating mechanism to memorize the previous node information. It combines the forget gate and input gate of LSTM into an update gate, which reduces network parameters and makes training faster. The principle of GRU is shown in Figure 5.

Among them, \(x_t\) is the input data, \(h_t\) is the output of the GRU unit, \(r_t\) is the reset gate, and \(z_t\) is the update gate[16]. The calculation from \(h_{t-1}\) to \(h_t\) is jointly determined by \(r_t\) and \(z_t\)[17]. The specific unit calculation can be expressed by formulas (1) ~ (4).

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

\[
r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{2}
\]

\[
\hat{h}_t = \tanh(W \cdot [r_t \times h_{t-1}, x_t]) \tag{3}
\]

\[
h_t = (1 - z_t) \times h_{t-1} + z_t \times \hat{h}_t \tag{4}
\]
Among them, $W_z$ and $W_r$ represent the update gate and reset gate respectively, and $W$ are the weight matrix.

![GRU schematic](image)

**Figure 5.** GRU schematic

Considering that the core of GRU is to use the hidden state to save the input information that has passed it, and a single GRU can only learn the information before the current word, so this paper uses BiGRU in order to the influence of the information before and after the current word can be considered at any point in time\[17\]. The global structure information $I$ of the phrase can be expressed by formula (5).

$$I = \begin{bmatrix} \text{GRU} & : & \text{GRU} \end{bmatrix}$$  \hfill (5)

### 3.4. Attention layer

In this paper, the global structure information of BiGRU’s output input into the Attention model, and weights can be assigned to each word through calculation, and then the global semantic information of the sentence is multiplied by the respective weights to obtain the weighted text vector, the calculation method of the weight is shown in formula (6). The BiGRU_Attention structure is shown in Figure 6.

![BiGRU_Attention model structure](image)

**Figure 6.** BiGRU_Attention model structure

$$\alpha_i = \frac{\exp(Score_i)}{\sum_i \exp(Score(k))}$$  \hfill (6)

### 4. Experiments and result analysis

#### 4.1. Experimental data and design
The THUCNews data set is composed of news documents in UTF-8 text format. This paper extracts 200,000 news headlines from the original data set for short text classification experiments. There are 10 categories. In this paper, the data set is divided into training set, validation set and test set. In order to avoid data imbalance, the amount of data in each category in the data set is equal.

Training BERT and ERNIE requires a lot of time and computational cost. Therefore, BERT and ERNIE used in this paper are the published pre-training models "BERT-Base, Chinese" and "ERNIE 1.0". The ERNIE_BiGRU_Attention (EBGA) and ECBA used in this paper are all based on the pretrained network ERNIE, adding some layers to get own network. Generally, there are two ways to use the pre-training model:

- Constantly update the parameters of the entire network.
- Fix the parameters of the pre-training model, and only update the parameters that you have added.

4.1.1. Hyperparameter settings. This experiment is completed using the Pytorch deep learning framework. Table 1 lists the training parameter settings of EBGA.

| Hyperparameter | values | Hyperparameter | values |
|----------------|--------|----------------|--------|
| optimizer      | BertAdam | hidden_size    | 768    |
| lr             | 5e-5    | pad_size       | 32     |
| weight_decay   | 0.01    | BiGRU_layers   | 2      |
| batch_size     | 128     | require_improvement | 1000 |
| epoch          | 10      | loss_fun       | CrossEntropyLoss |

4.1.2. Evaluation metrics. For multi-classification tasks in NLP, \( P \), \( R \), and \( F_1 \) score are commonly used to evaluate the model. Among them, \( F_1 \) score firstly calculates the \( F_1 \) value of each category, and then calculates the average value of each category. Table 2 shows the general form of confusion matrix.

| Reality | forecast | Positive | negative |
|---------|----------|----------|----------|
| True    | TP       | FN       |
| False   | FP       | TN       |

\( P \) and \( R \) can be expressed by formula (7) and formula (8).

\[
P = \frac{TP}{TP + FP} \tag{7}
\]

\[
R = \frac{TP}{TP + FN} \tag{8}
\]

The maximum value of \( F_1 \) score is 1, and the minimum value is 0, which is defined as formula (9).

\[
F_1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{9}
\]

4.1.3. Results analysis. During the experiment, every 100 batches output the accuracy and loss values on the training set and the validation set and record and save them. Finally, the trend of change is presented in the form of more intuitive pictures. Stop training after the model reaches the optimal level.
In this paper, the training process of EBGA used two modes, such as pre-training model parameter update and fixed. The values of evaluation indicators of the model are shown in Table 3, and the accuracy change of the verification set is shown in Figure 7, in the figure, every 0.5 epoch output an accuracy as the node describing the performance trend of the model.

| Pre-training model parameter | $P$    | $R$    | $F_{1\text{-score}}$ | $\text{acc}$ |
|------------------------------|--------|--------|----------------------|-------------|
| Fine-tuned                   | 0.9478 | 0.9476 | 0.9476               | 0.9476      |
| fixed                        | 0.8851 | 0.8853 | 0.8848               | 0.8853      |

Figure 7. The results of parameters of fine-tuned and fixed

It can be seen from Figure 7 that the performance of the model with updating the entire model parameters is better, and the performance of the model reaches the best after 3 rounds. At the same time, it can be seen from Table 3 that performance of the model was improved by about 4% through fine-tuning parameters of model.

4.2. Comparative experiment design

In order to verify the performance of the model used in this paper, compare the model used in this paper with the CNN, BiGRU, BiGRU_Attention, BERT, BERT_BiGRU, ERNIE, and ERNIE_BiGRU models. These models are represented as models 1-7. Models 1-3 divide the text by characters.

4.2.1. Hyperparameter settings. The experimental parameter settings of models 1-3 are shown in Table 4. The parameter settings of CNN and LSTM are shown in Table 5 and Table 6.

| Hyperparameter | values | Hyperparameter | values |
|----------------|--------|----------------|--------|
| optimizer      | Adam   | loss_fun       | CrossEntropyLoss |
| lr             | 1e-3   | batch_size     | 128    |
| epoch          | 10     | pad_size       | 32     |
Table 5. Network parameters of CNN model

| Hyperparameter | values | Hyperparameter | values |
|----------------|--------|----------------|--------|
| Kernel_size    | [2, 3, 4] | Activation    | ReLu   |
| Dropout        | 0.5    | num_filters   | 256    |

Table 6. Network parameters of BiLSTM model

| Hyperparameter | values | Hyperparameter | values |
|----------------|--------|----------------|--------|
| hidden_size    | 128    | num_layers     | 2      |
| Dropout        | 0.5    | loss_fun       | CrossEntropyLoss |

- CNN: Constantly Basic convolutional neural network, using characters as the smallest processing unit.
- BiGRU: Combine the hidden states of the before and back directions as text features.
- BiGRU_Attention: The attention mechanism can focus on the more important information for classification from the information, and assign more weights[18].
- BERT: Use the "BERT-Base, Chinese" pre-training model to complete text vectorization. Pre-trained model character input, character Mask.
- BERT_BiGRU: Get text word vectors by fine-tuning the pre-trained BERT, and use BiLSTM to learn contextual semantic information.
- ERNIE: ERNIE is continuously improved on the basis of the BERT, and text knowledge can be obtained through training. In this experiment, the "ERNIE 1.0" pre-training model was used, and the pre-training model was fine-tuned during training.
- ERNIE_BiGRU: GRU is an improvement of the LSTM, with fewer model parameters and faster training speed.

4.2.2. Results analysis. Figure 8 shows the impact of static and dynamic word vectors on downstream model performance. The performance of classification models using CNN, BERT and ERNIE as word embedding layer is compared.

![Figure 8. The impact of text semantic representation on model performance](image)

It can be seen from Figure 8 that although the accuracy of CNN is rapidly improved at the beginning of training, the performance of the final model is still inferior to BERT and ERNIE, and ERNIE has the best performance. Static word vectors have great limitations and lack global contextual semantic information. Dynamic word vectors can well retain contextual semantic information and effectively make up for the shortcomings of static word vectors, but it hardly pays attention to semantic knowledge
and cannot solve the problem of polysemy of a word. ERNIE introduces knowledge information through learning, solves the problem of polysemy of a word, and enhances the semantic expression ability of the model. It can be seen that for text classification tasks, the text vectorization part of the data has a great impact on downstream model tasks.

Table 7 shows the results of comparing the final $P$, $R$, $F_{1}$,score and $acc$ of all models. Figure 9 shows the performance trend of all comparison models.

| Comparative model   | $P$   | $R$   | $F_{1}$,score | $acc$ |
|---------------------|-------|-------|---------------|-------|
| CNN                 | 0.9125| 0.9123| 0.9122        | 0.9123|
| BiGRU               | 0.9078| 0.9074| 0.9071        | 0.9074|
| BiGRU_Attention     | 0.9095| 0.9092| 0.9090        | 0.9092|
| BERT                | 0.9393| 0.9388| 0.9388        | 0.9388|
| BERT_BiGRU          | 0.9440| 0.9439| 0.9439        | 0.9439|
| ERNIE               | 0.9421| 0.9421| 0.9420        | 0.9421|
| ERNIE_BiGRU         | 0.9453| 0.9448| 0.9449        | 0.9448|
| EBGA                | 0.9478| 0.9476| 0.9476        | 0.9476|
| ECBA                | 0.9486| 0.9480| 0.9481        | 0.9480|

Figure 9. Model performance comparison

It can be seen from Table 7 that the EBGA and ECBA used in this paper are better than other comparison models in the four evaluation indicators of $P$, $R$, $F_{1}$,score, and $acc$. EBGA and ECBA are denoted as models 8 and 9 respectively. It can be seen from Figure 9 that the model 9 used in this paper has the highest accuracy among all models. On the whole, the accuracy of all models shows an upward trend with the increase of epochs. Models 4 and 6 reached the optimal performance after the third round, and models 5, 7, 8 and 9 reached the optimal performance after the fourth round. There is no significant difference in the performance of models 1-3, among which CNN model is the best, and its performance is lower than that of models 4-9, which indicates that the embedding part affects the performance of downstream models, and CNN has some advantages over RNN in short text classification. Comparing models 4-9, it can be seen that the performance of the ERNIE model that introduces knowledge...
information is higher than that of the BERT model. Model 8 not only uses ERNE to enhance the semantic representation of the text, but also uses the attention mechanism to focus the weight on more useful features, which effectively improves the performance of the model. Model 9 adds a CNN model to extract local features on this basis, which further improves the performance of the model. This proves that by introducing knowledge information to enhance the semantic representation of the text, the classification performance of the model can be effectively improved. At the same time, the EBGA and ECBA proposed in this paper combine different model advantages, making the model achieve better results in short text classification tasks.

5. Conclusions
This paper proposes a knowledge-enhanced ERNIE hybrid model EBGA and ECBA. Firstly, the ERNIE1.0 pre-training model is used as the word embedding layer. This text representation with semantic knowledge greatly improves the accuracy of text classification. CNN extracts the local features of the text and combines BiGRU to make up the shortage of CNN's inability to learn grammatical and semantic structure information. The attention model is used to assign weights to each feature word, and the fully connected layer serves as the final classifier. The comparison with other models fully proves the advantages of ECBA in the classification of Chinese short texts. Since the ERNIE1.0 pre-training model is trained for Chinese short texts, the next stage of research is how to apply ERNIE in long text classification. At this stage, you can consider splitting long sentences into ERNIE separately, and then splitting joint the output result. Subsequent work can also train a pre-training model suitable for long texts use a large number of Chinese long text corpora.

6. References
[1] PANG B, Lee L and Vaithyanathan S 2002 Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing vol 10 (USA: Association for Computational Linguistics) pp 79-86
[2] Chen K, Liang B, Ke WD, Xu B and Zeng GC 2018 Sentiment analysis of Chinese weibo cased on multi-channel convolutional neural network IBM J RES DEV. 5 945-57
[3] Bengio Y, Ducharme R and Vincent P 2003 A Neural Probabilistic Language Model J. Mach. Learn. Res. 3 1137-55
[4] Mikolov T, Sutskever I, Chen K and Corrado G 2013 Proceedings of the 26th International Conference on Neural Information Processing Systems (Lake Tahoe, Nevada) vol 2 (Red Hook, NY, USA: Association for Computational Linguistics) pp 3111-19
[5] Wu XH, Chen L, Wei TT and Fan TT 2019 Sentiment analysis of Chinese short text based on Self-Attention and Bi-LSTM J Chin Inf Proc. 6 100-7
[6] Peters M, Neumann M, Iyyer M, Gardner N, Clark C, Lee K and Zettlemoyer L 2018 Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies vol 1 (New Orleans, Louisiana: Association for Computational Linguistics) pp 2227-37
[7] Devlin J, Chang MW, Lee K and Toutanova K 2019 Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) vol 1 (Minneapolis, Minnesota: Association for Computational Linguistics) pp 4171-86
[8] Kim Y 2014 Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (Doha, Qatar: Association for Computational Linguistics) pp 1746-51
[9] Zhang X, Zhao J and Lecun Y 2015 Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1 (Montreal, Canada) vol 1 (Cambridge, MA, USA: MIT Press) pp 649-57
[10] Lai SW, Xu LH, Liu K and Zhao J 2015 Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (Austin, Texas : AAAI Press) pp 2267-73
[11] Hochreiter S and Schmidhuber J 1997 Long Short-Term Memory Neural Comput.9 1735-80
[12] Li Y and Dong HB 2018 Text sentiment analysis based on CNN and BiLSTM network feature fusion Journal of Computer Applications 11 3075-80
[13] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł and Polosukhin I 2017 Proceedings of the 31st International Conference on Neural Information Processing Systems (Long Beach, California, USA) (Red Hook, NY, USA: Curran Associates Inc.) pp 6000-10
[14] Benchaji I, Douzi S, and Ouahidi BE 2021 Credit Card Fraud Detection Model Based on LSTM Recurrent Neural Networks Journal of Advances in Information Technology 12 113-8
[15] Cho K, Merrienboer BV, Gulcehre C, Bahdanau D, Bougares F, Schwenk H and Bengio Y 2014 Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (Doha, Qatar: Association for Computational Linguistics) pp 1724-34
[16] Wang LY, Liu CH, Cai DB and Lu T 2019 Sentiment analysis of Chinese text with attention mechanism introduced in CNN-BiGRU network Journal of Computer Applications 10 2841-46
[17] Wang N, Li SL, Liu TL ang Zhao W 2019 Analysis of BiGRU Judgment Tendency Based on Attention Mechanism Comput Syst Appl. 3 191-195
[18] Zhou P, Shi W, Tian J, Qi ZY, Li BC, Hao HW and Xu B 2016 Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) vol 2 (Berlin, Germany: Association for Computational Linguistics) pp 207-212

Acknowledgments
Thanks to the National Social Science Foundation of China (16XXW006) for supporting this paper!