Research Article

BERT-BiGRU Intelligent Classification of Metro On-Board Equipment Faults Based on Key Layer Fusion

Endong Liu and Junting Lin

School of Automation and Electrical Engineering, Lanzhou Jiaotong University, 730000 Lanzhou, China

Correspondence should be addressed to Junting Lin; linjt@lzjtu.edu.cn

Received 25 April 2022; Revised 30 May 2022; Accepted 6 June 2022; Published 25 June 2022

Academic Editor: Kuruva Lakshman

Copyright © 2022 Endong Liu and Junting Lin. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Metro train operation may result in massive and complex unstructured fault text data. To solve the problem of low classification accuracy and incomplete classification effect of unstructured fault data automatic classification task, a BERT-BiGRU fault text classification model based on key layer fusion is proposed. Firstly, the unstructured text data is processed into word vectors with location information in the word embedding layer and then input into the BERT layer. Based on the traditional 12-layer BERT model, the semantic information is fully obtained by encoding the two-way transformer encoder in layers 2, 4, 6, 8, and 12 for fusion and dimensionality reduction, which is then input into the BiGRU layer to extract the context information to obtain the high-level feature representation of the text. After that, the final classification results are output in the output layer through the full connection layer FC and softmax functions. This model is tested with other mainstream models on the fault text data of metro on-board equipment. The experiment results show that on the same data set, the $F_1$-score of this model is 7%–8% higher than that of the current mainstream classification model, that is, about 1–2% higher than that of other PLMs, and the $F_1$-score of this model is higher than that of BERT model with different transformer layers, reaching the highest value of 0.9272, and the convergence speed in the training process is fast.

1. Introduction

Metro on-board equipment, as an important link in the field of rail transportation, plays an important role in the safe operation of metro trains. The accumulation of metro train mileage may result in a large number of on-board equipment fault data. And most of these data are stored in the form of unstructured text data, recording the fault phenomenon in detail. If the diagnosis, location, and treatment of metro on-board equipment fault are based on the experience of on-site maintenance personnel and expert knowledge only, it may easily let to fault judgment error and prolonged maintenance time. Driven by the construction of metro big data and smart railway, text mining technology and relevant algorithms of in-depth learning are used to intelligently classify the faults of metro on-board equipment and help on-site maintenance personnel to quickly locate the fault according to the fault phenomenon and give corresponding maintenance decisions in time, which is of far-reaching significance for improving the safety guarantee level of metro operation [1].

TC (text categorization), as an important task in NLP (natural language processing) [2], refers to the automatic recognition of text categories based on text content by text classifier. In recent years, Chinese text classification has been studied and put into practical applications in various fields in China, such as sentiment analysis, movie reviews, stock market, and classification tasks for fault-related descriptions [3]. With the development of database in the field of rail transportation in China, TC is also applied in this field as well. As an important part of the rail transportation field, metro on-board equipment generates a large amount of fault data consisting of short Chinese texts during operation, recording fault profiles, fault cause analysis, fault rectification measures, etc., in the form of large text descriptions [4]. To make full use of these fault text data, it is necessary to realize automatic classification of equipment faults by text classification model, so as to improve the processing ability
of metro on-board equipment fault data and avoid the arbitrariness and untruthfulness of manual classification of equipment fault data. So far, there is little research on the fault text mining of metro on-board equipment at home and abroad. Therefore, this paper constructs a relevant data set based on the statistical data of metro on-board equipment faults in a metro depot in the past five years and then turns to the research on the text classification of metro on-board equipment faults.

PLMs (pretrained language models) are neural network models obtained by pretraining on large amounts of text data using unsupervised learning. The PLMs are fine-tuned by inputting relevant domain datasets to generate feature representations of the datasets, which in turn can improve the classification performance of downstream classifiers. BERT (Bidirectional Encoder Representations from Transformers) [5, 6], as typical PLMs, have achieved good results in short TC tasks, and it can effectively extract global features of sentences through the self-attention mechanism in the bidirectional transformers module. Later on, derivative models of BERT emerged, such as follows: the RoBERTa (Robustly Optimized BERT Pretraining Approach) [7] has a larger batch size compared to BERT, the ALBERT model [8] uses a parameter sharing mechanism and thus achieves a reduction in the overall number of parameters, the Distill-BERT [9] improves efficiency through distillation operations, and the BERT-wwm was combined with WWM (Whole Word Masking) mechanism [10]. While the BERT model is composed of 12-layer coding network, the performance of different network layers varies, so it is necessary to go deeper into the characteristics of the information extracted by each transformer layer and think about how to flexibly utilize the 12-layer transformer coding network, which makes it possible to reduce the redundancy and repetition of information while ensuring the comprehensiveness of the information extracted by BERT and thus improve the training accuracy and efficiency of BERT. Finally, the word vectors generated by PLMs are fed to the deep learning-based classifier to complete the text classification task.

The paper further deepens the structured processing of metro on-board equipment fault text data. Besides, combined with the data characteristics, it also puts forward the intelligent classification model of metro on-board equipment fault based on the dynamic fusion of BERT and BiGRU in the key layers; the most critical five layers are selected from the original 12-layer transformer of the upstream model BERT, and the dimension of the input features is fused and reduced before sending it to BiGRU; on the one hand, the bidirectional gating unit BiGRU can be used to ensure the sequence of feature vectors, which can fuse the context information extracted from the forward and backward GRU sequences, so as to obtain the feature representation with richer semantic information. Finally, the generated feature vector is sent to the full connection layer FC (full connection), and the classification result is output through the softmax function connected with it.

The main contributions of this paper are as follows:

(1) In this paper, for the Chinese short text fault data generated in the depot of metro company within 5 years, we divide them into 10 types of faults and count the number of each type of fault data based on the existing knowledge to build a metro on-board fault data set.

(2) For the BERT model with 12 layers of transformer, the focus of each coding layer on text information extraction is explored. After that, the information extracted from the most critical 5 coding layers in the BERT model is fused and fed to the classifier for experimental validation, and the experimental results show that the performance of BERT based on fusion of critical layers is improved compared with other PLMs.

(3) In this paper, the BiGRU (Bidirectional Gate Recurrent Unit) model is used as the downstream model for the text classification task, which overcomes the limitation that CNNs (convolutional neural networks) cannot extract long sequence information and also solves the drawback that RNN and LSTM models cannot capture both forward and reverse sequence information and reduces the complexity compared with BiLSTM (Bidirectional Long Short-Term Memory).

2. Related Work

In order to make full use of the fault data generated during rail transit operations, it is necessary to propose a model that can automatically complete the text classification task; thereupon, many scholars have conducted research in the field of rail transit text classification. In literature [11], for fault data generated by high-speed rail on-board equipment, a topic model is used to extract the feature information in the text, and the classification is completed by the Bayesian structure learning algorithm HDBN_SL. In literature [12], for unbalanced fault text data, the electrical service signal equipment fault text data was transformed into feature vectors by TF-IDF (Term Frequency-Inverse Document Frequency), and a voting-based approach was used to achieve integrated multiclassifier learning classification. Literature [13] used TF-IDF to transform on-board fault logs into feature vectors, built an ensemble classifier under the bagging framework, and used KELM (Kernel-Based Extreme Learning Machine) as the basic classifier to complete the text classification task; with the rise of deep learning in the field of NLP, it has also been applied to the field of text processing for rail transportation. Literature [14] converts fault text into a word vector matrix by word2vec and uses multipooling CNN for signal equipment fault classification. In literature [15], the text classification of high-speed railroad equipment faults was done by BiGRU and BiLSTM integrated models. In the literature [16], the BERT model was used to vectorize the text representation of CBTC (Communication-Based Train Control System) fault data, and the text classification task was completed by BiGRU-attention network. The above experimental results show that the deep learning-based model
can further improve the text classification performance in the rail transportation domain. For the fault data generated during the operation of metro on-board equipment, the same way as above, the unstructured fault text data are firstly transformed into feature vectors by the word vector training model, and then the downstream classifier model is used to further extract the hidden information in the feature vectors and complete the classification of the fault text.

2.1. Word Vector Training Models. The role of the word vector training model is to extract the features of the faulty text data and obtain its feature vector representation. To make the computer better understand human language [17], Uriarte-Arcia et al. [18] proposed one hot coding to number each subword in order (represented by 0 and 1). However, this numbering method is easy to generate vectors with too high dimension and sparse representation, which cannot integrate the characteristics of the previous and subsequent articles. With the rapid development of NLP field, the simple one hot coding method is gradually replaced by representing text with word vector [19, 20]. That is, words are mapped into a real number vector with small dimension and dense value, which can describe the semantic and grammatical relationship of words [21]. At the same time, it shows good computability, avoiding the difficulty of manual feature selection and the problem of sparse vector representation.

Common word vector transformation methods mainly include TF-IDF [12, 22], word2vec algorithm [13, 23], and ELMO (Embeddings from Language Models) algorithm [24]. With various models of PLMs being proposed, for word vector transformation tasks, word vector training can be done with simple fine-tuning on top of PLMs. Common PLM models include ELMO, XLNet [25], and BERT; among them Google released BERT model as one of the most popular PLMs currently, and BERT uses MLM (Masked Language Model) task for training, i.e., masking some words randomly and then predicting the masked words to achieve the training effect. Then, Facebook released an improved version of BERT, RoBERTa [7], and Cui et al. [10] combined BERT with WWM mechanism and proposed BERT-wwm, BERT-wwm-ext, and RoBERTa-wwm-ext.

2.2. Classifier Models. The development of classifier models is divided into two main phases: machine learning and deep learning. The commonly used text classification models based on machine learning mainly include DT (Decision Tree) [26], SVM (Support Vector Machine) [27], and integrated classification models bagging [28] and boosting [29]. However, machine learning cannot meet the needs of increasing data processing, nor can it learn deeper features from the data. In that case, text classification methods based on deep learning have gradually become a research focus in the field of NLP [30, 31]. The commonly used text classification models based on deep learning mainly include convolutional neural network (CNN) [32], cyclic neural network RNN (recurrent neural network), and their improved models [33]. Document proposed CNN as a text classification model which obtains local sensitive information from the text represented by word vector through convolution and pooling operation [34, 35] and extracts high-level text features; Zhou et al. and Xie et al. used BiLSTM neural network as the downstream model in the composition classification task, further integrating the context information into it based on the generation of feature vector and achieving good classification results [36, 37]. Wang et al. proposed to take BiGRU as the classifier model [38]. The neural network integrates the forgetting gate and input gate into a single update gate based on BiLSTM, which can still achieve the training effect of BiLSTM while reducing the structural complexity [39]. They proposed BERT model as a pretraining model to solve the joint learning problem of event text classification and event allocation [40].

3. Design of Intelligent Fault Classification Model of BERT-BiGRU Metro On-Board Equipment Based on Key Layer Fusion

The structure of BERT-BiGRU metro on-board equipment fault intelligent classification model based on key layer fusion proposed in this paper is shown in Figure 1.

The model is mainly composed of word embedding layer, BERT layer, BiGRU layer, and output layer. Firstly, each word in the fault text is transformed into word embedding through word embedding layer and added with location-based word embedding to generate token representation with location information; the transformer of layers 2, 4, 6, 8, and 12 in the original 12-layer BERT model is taken...
out, the context features in the 5-layer transformer are comprehensively encoded, and the dimension of the coding results is spliced and reduced. The obtained features are added to the representation of token to generate character level semantic information $t_c$. After that, the acquired sequence $t_c$ is input into the BiGRU layer, and the bidirectional GRU module in this layer can capture the front and rear sequence information at the same time, so as to realize the high-dimensional feature extraction of the front and rear sequence information. Finally, in the output layer, the feature vector outputs the probability distribution composed of the probability values corresponding to different categories through the full-connection layer FC and the softmax function. And the category corresponding to the maximum probability value in the probability distribution is the final classification result.

3.1. Word Embedding Layer. The word embedding layer is the embedding part of BERT, which represents a word in the form of the addition of three layers of embedding vectors. The fault text input to the word embedding layer is preprocessed by using the tokenization model in the transformer library, and the generated word vector is converted into token embedding, as shown in Figure 2. And the other two are segment embedding and position embedding, respectively. The segment embeddings can be implemented, and the encoding of the sentence is used to demarcate the contextual relationship between the two clauses. The position embeddings encode the position information and record the word order features in the sentence, so that the word order in the input sequence can be explained. These three vectors are added to get the final feature vector as the input of the encoder in the subsequent transformer.

3.2. BERT Layer. The main structure of BERT and its diffraction model form the deep bidirectional transformer encoder. The transformer introduces the self-attention mechanism and draws lessons from the residual connection of convolutional neural network. The model is characterized by fast training speed and strong expression ability. It is confirmed by Jawahar et al. [41] that each layer of BERT has different representation of text information through the detection tasks in 12 NLP fields. Layer 1–4 transformers complete the embedding of features and learn the surface features of natural language text at the same time; layers 4–8 learn and encode syntactic information; layers 8–12 mainly encode semantic features. Literature [42] studies the transformation of the internally hidden state when the BERT model completes the reading comprehension task and makes a visual and qualitative analysis of the hidden state vector of the transformer at each layer. It is found that both the middle layer and the top layer can detect the professional semantic information of the text comprehension task earlier, which is beneficial to the correct prediction. This conclusion also shows that different layers of BERT encode different levels of information of the text. As shown in Figure 3, this paper extracts layers 2, 4, 6, 8, and 12 of the BERT models at the BERT layer, splices the features extracted by using the 5-layer transformer, and then reduces the linear layer to 512 dimensions to obtain the feature of $t_c$, which can not only reduce the complexity of the model but also fully capture the information at all levels.
3.2.1. Pretraining Method of BERT Model.

The BERT pretraining task is divided into two parts: masking language model MLM and next-sentence prediction (NSP). The basic idea of MLM is to block 15% of the words in the sentence randomly and let the encoder in each layer of transformer predict the original value of the masked words according to the context information, so as to achieve the learning effect of the model, while the basic idea of NSP is to put two sentences together, and the encoder part judges whether the two sentences are adjacent and in normal order in the original text through learning. Specifically, input sentence A into a BERT, and then, select the next continuous sentence as sentence B with 50% probability, and the other 50% randomly extract sentences from the corpus, so as to judge whether sentence A is continuous with the randomly selected sentence in model learning.

The pretraining of BERT is shown in Figure 4, where sentence A and sentence B are randomly selected as the input of the BERT pretraining model, in which a part of the characters in sentence A and sentence B are MASK, and the corresponding feature vector sequences are generated by embedding layer and transformer encoder, respectively, and the feature vectors are input to three classifier 1 outputs $P_1$ to determine the association between A and B, classifier 2 outputs $P_2$ to predict the part of sentence A that is MASK, and classifier 3 outputs $P_3$ to predict the part of sentence B that is MASK. Meanwhile, the three classifiers contain three loss functions, and the gradient of the loss function about the model is calculated by back propagation in the pretraining process, and then, gradient descent is performed to update the model parameters.

3.2.2. Transformer Encoder. BERT uses the encoder part of the bidirectional multilayer encoder transformer to encode the input data, and the encoder network is composed of self-attention layer, feedforward neural network, and add and normalize layer. Its structure is shown in Figure 5. Firstly, the sequence $[E_1, E_2, \ldots, E_n]$ obtained from the word embedding layer is input to the self-attention layer. By defining $Q, K, V$ characteristic matrices for each word in the sentence and making the characteristic matrices of different words cross multiply, the characteristic vector is calculated by using the following formula:

$$Z = \text{softmax} \left( \frac{Q \times K^T}{\sqrt{d_k}} \right) \times V,$$  \hspace{1cm} \text{(1)}$$

where $Q$ represents the index, $K$ stands for the key, and $V$ refers to the value, which represents the dimension of $K$. The value is introduced as the denominator to prevent the internal product of the molecule from being too large, so as to make the gradient of training more stable. Self-attention adopts MHA (Multiheaded Attention) mechanism with its working principle shown in Figure 6. Taking the input value as an example, firstly, the input vector is initialized into multiple groups of matrices, and multiple groups of characteristic matrices are generated by using point multiplication. After that, the multiple groups are spliced and multiplied with the weight matrix to reduce the dimension, in this manner to obtain the vector containing multiple features, thereby expanding the ability of the model to pay attention to different positions. Input the obtained eigenvector to the add and normalize layer, and add the different outputs in

\[ \text{Figure 4: The process of BERT pretraining.} \]

\[ \text{Figure 5: Encoder structure.} \]
controls how much information has to be forgotten from the feedforward neural network layer, and the way of addition of residual, then input the results to the feedforward neural network layer, and finally pass through the add and normalize layer again to obtain the final output of encoder.

With the bidirectional transformer encoder training word vector in BERT, the semantic information of metro on-board equipment fault text is completely saved, thereby improving the context bidirectional feature capture ability of the model, solving the problem of one-word polysemy, and theoretically improving the accuracy of subsequent classification models.

3.3. BiGRU Layer. Since the encoder part of the transformer adopts the self-attention structure, the output features are lack of order. To obtain the sequence characteristics of the fault text of metro on-board equipment, the BiGRU model is used to model the information below the upper BiGRU layer of the fault text of metro on-board equipment, and the GRU structure is adopted in the hidden layer unit with its unit structure shown in Figure 7.

GRU is a variant of LSTM, which effectively calculates and controls the input and output of information by designing gating units in neurons. The design of this gating unit solves the problem of text sequence length dependence. Compared with LSTM network, GRU combines the forgetting gate and input gate into an update gate \( z_t \), which controls how much information has to be forgotten from the previous hidden layer \( h_{t-1} \) through the sigmoid function with the calculation shown in Formula (2). In Formula (3), the information is output to the current hidden layer \( \tilde{h}_t \) through the tanh function, and then, \( h_t \) is obtained by using Formula (4); the reset gate \( r_t \) controls how much previous information has to be retained through sigmoid function, as shown in Formula (5). Compared with LSTM, GRU model is simpler, which requires fewer parameters, and less tensor computation. Besides, it has faster training speed in theory and is not easy to produce overfitting.

\[
\begin{align*}
z_t &= \sigma(W_z \cdot [h_{t-1}, x_t] + b_z), \\
\tilde{h}_t &= \tanh(W_h \cdot [r_t \ast h_{t-1}, x_t] + b_h), \\
h_t &= (1 - z_t) \ast h_{t-1} + z_t \ast \tilde{h}_t, \\
r_t &= \sigma(W_r \cdot [h_{t-1}, x_t] + b_r).
\end{align*}
\]

In the BiGRU layer shown in Figure 1, the two-way GRU network forward propagation and back propagation in this layer are carried out together. At the same time, the forward and reverse hidden layer state sequences are captured and combined, solving the problem of the inability of the traditional one-way GRU model to capture context information. BiGRU layer gives full play to the advantages of two-way communication of GRU. It can not only fully combine the context but also extract features through the overall environment, thereby reducing the loss of features.

3.4. Output Layer. The task of the output layer is mainly accomplished by the full-connection layer FC and softmax functions, and its working process is shown in Figure 8. Among them, for the full-connection layer, each unit of the previous layer is connected with the latter layer. The weight matrix \( w_j \) is multiplied by the input vector \([h_1, h_2, h_3, \cdots, h_n]\), and bias is added, as shown in Formula (6), to obtain the output \( C_j \) of FC, so as to map \( n \) real numbers to \( j \) real numbers (there are 10 fault categories in this paper; therefore, \( j = 10 \)), which is more conducive to the classification task of softmax function.

\[
C_j = w_j \cdot h + \text{bias} = \omega_{j1}h_1 + \omega_{j2}h_2 + \cdots + \omega_{jn}h_n + \text{bias}.
\]
The softmax function maps the input values from the FC layer to (0,1). All input values are calculated to the $n$th power of $E$, and then, these results are summed before calculating the proportion of each value in the sum value. The probability value $P_j$ output as category $j$ is calculated by Formula (7). Through the calculation of the probability value $P_j$ of each category, a probability distribution $P_1, P_2, \ldots, P_{10}$ is obtained, in which the subscript number corresponding to the maximum probability value is the classification result of softmax function.

4. Evaluation Index Modelling

In this paper, the precision, recall, and $F_1$-score are used as the evaluation indexes of this experiment, in which TP represents the number of samples classified and divided correctly; FP refers to the number of samples classified and divided incorrectly; FN stands for the number of samples that are not classified; therefore, they must be wrong.

(1) Precision

The accuracy rate is only for the positive samples with correct prediction, instead of all the samples with correct prediction. It is calculated by dividing the number of positive samples with correct prediction by the ratio of the number of positive samples predicted by the model. It shows that the predicted positive samples are really positive, as shown in the following formula:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%.$$  \hspace{1cm} (7)

(1) Recall

Calculated by dividing the predicted correct number of positive samples by the actual number of positive samples in the test set, it shows how many samples that are really positive can be recalled by using the classifier, as shown in the following formula:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%.$$  \hspace{1cm} (8)

(2) $F_1$-score

$F_1$-score is the harmonic average of accuracy rate and recall rate, and both precision and recall are expected to be higher, but these two indicators are contradictory and cannot be both high. Therefore, $F_1$-score should be introduced as an appropriate threshold point to maximize the ability of the classifier, as shown in the following formula:

$$F_1\text{-score} = \frac{2}{1/\text{precision} + 1/\text{recall}} \times 100\%.$$  \hspace{1cm} (9)

5. Experiment

5.1. Experimental Environment and Data Set. In terms of experimental hardware, the CPU is i7-6700HQ, the graphics card is GTX960M, the video memory is 8 G, the operating system is Win10 64 bit, the python version is 3.60, the development tool is Spider 5.0.5, and the pytorch version is 1.11.1GPU.

According to the functions and fault characteristics of each equipment, this paper selected the fault text data recorded in the depot of a metro company from 2016 to 2021, counted the total data as 7418, and summarized 10 fault categories according to the data, including transponder receiving module fault, door state loss fault, movement blocked fault, alignment fault, vehicle on-board computer loss of communication fault, input failure fault, overspeed fault, communication timeout fault external emergency braking fault, and other on-board faults represented by F1–F10 in the following experiments. The fault sample data of metro on-board equipment is shown in Table 1, which is

<table>
<thead>
<tr>
<th>Serial number</th>
<th>Description of fault occurrence</th>
<th>Fault classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>During the train operation, the communication between on-board equipment and TCM equipment is abnormal, and MMI displays TMS cross. Then, the train emergency braking exits AM mode, and ATO is unavailable at this time, reporting ATO alarm abnormal fault. When the train passes through two continuous balise during operation, the on-board BTM does not receive the balise information. The train outputs emergency braking and prompts the driver to switch to RM mode after parking. The train is mainly in ATO mode with 3 ends. When running to the platform, SMC displays and reports abnormal alignment and 2 recovers. CCOT displays and reports fc04, the train opens and closes normally, and the PSD is linked normally. After the operation dispatching discharges the train route and activates the train, the train starts emergency braking. CCOT displays that the train reports fc12 for 25 seconds and then recovers. The operation dispatching releases EB command for the train, and the release is successful. Vehicle on-board computer gave 81.5% traction, and the vehicle equipment did not respond in time.</td>
<td>Communication interruption fault Transponder receiving module failure Alignment fault Movement blocked fault</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
composed of the description of fault phenomenon and the corresponding type of fault.

The fault data set with a total amount of 9418 is divided into training set, development set, and test set, and 6891 pieces of data are selected as training set data for model fitting data samples; 1416 pieces of data are development set data, which are used to adjust parameters, select features, and make other decisions on learning algorithms; 1411 pieces of data are test set data used to evaluate the model. The quantity distribution of train set, test set, and dev set corresponding to various faults is shown in Figure 9.

5.2. Experiment and Results

5.2.1. Experimental Super Parameter Optimization. It is found from the experiment that the changes of some parameters also have a significant impact on the results of the experimental model. In the experiment of this paper, the influence of the key parameters of the model on the experimental results is first evaluated. Learning rate is one of the most important parameters affecting the performance of deep learning model. In the case that the learning rate is too large, it may lead to nonconvergent model; in contrast, if the learning rate is too small, it may lead to long convergence time and even model training failure. To study the influence of learning rate on the performance of the model, based on the previous experimental experience, this paper selected the learning rate of 5e-4, 8e-4, 1e-3, 3e-3, 5e-3, and 8e-3 for the experiment. As shown in Figure 10, at the learning rate value of 8e-4, the performance of the model is the best.

The number of epochs in deep neural network has to be adjusted manually. In general, different tasks have different settings of the number of epochs. In the case that the epoch number is too small, the model may not be able to converge to the local minimum, and the fitting is insufficient; on the contrary, if the epoch number is too large, it may increase the training time of the model or even make the model over fit. Therefore, to explore the impact of epoch number on model performance, Figure 11 in this paper shows the relationship between the model evaluation index and epoch number.

As can be seen from Figure 11, the model in this paper experienced rapid convergence before 15 iterations, and after that, the performance of the model improved slowly with the increase of the number of iterations. After about 20 iterations, the model performance will change within a small range with the number of iterations. Therefore, the number of epochs selected in this paper is 30.
The parameter setting is as shown in Table 2. Among them, BERT model adopts BERT released by using Google_Base model and the downloaded BERT with transformer layers of 2, 4, 6, 8, and 12 and hidden layers of 768 Base model. The embedded layer size is 128, the number of attention heads is 12, and Gelu is used as the activation function.

### Table 2: Parameter setting.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch_size</td>
<td>128</td>
</tr>
<tr>
<td>Learning rate</td>
<td>8e-4</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
<tr>
<td>Hidden layer dimension</td>
<td>768</td>
</tr>
<tr>
<td>Epoch</td>
<td>30</td>
</tr>
<tr>
<td>LSTM_dim</td>
<td>256</td>
</tr>
<tr>
<td>GRU_dim</td>
<td>256</td>
</tr>
<tr>
<td>Activation function</td>
<td>Relu</td>
</tr>
<tr>
<td>Max_seq_len</td>
<td>128</td>
</tr>
<tr>
<td>Dropout</td>
<td>0.5</td>
</tr>
</tbody>
</table>

### Table 3: The recognition effect of the model for each fault category studied.

<table>
<thead>
<tr>
<th>Type</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_1$-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>0.9213</td>
<td>0.9400</td>
<td>0.9306</td>
</tr>
<tr>
<td>F2</td>
<td>0.9166</td>
<td>0.9218</td>
<td>0.9192</td>
</tr>
<tr>
<td>F3</td>
<td>0.9363</td>
<td>0.9277</td>
<td>0.9320</td>
</tr>
<tr>
<td>F4</td>
<td>0.9299</td>
<td>0.9397</td>
<td>0.9348</td>
</tr>
<tr>
<td>F5</td>
<td>0.9091</td>
<td>0.9267</td>
<td>0.9178</td>
</tr>
<tr>
<td>F6</td>
<td>0.9217</td>
<td>0.9137</td>
<td>0.9177</td>
</tr>
<tr>
<td>F7</td>
<td>0.9140</td>
<td>0.9294</td>
<td>0.9216</td>
</tr>
<tr>
<td>F8</td>
<td>0.9334</td>
<td>0.9457</td>
<td>0.9395</td>
</tr>
<tr>
<td>F9</td>
<td>0.9447</td>
<td>0.9361</td>
<td>0.9404</td>
</tr>
<tr>
<td>F10</td>
<td>0.9292</td>
<td>0.9079</td>
<td>0.9184</td>
</tr>
</tbody>
</table>

The recognition effect of the model in this paper on F1–F10 fault types. In terms of the performance of accuracy rate, F3 accuracy rate reached the highest of 0.9447 which is only 3.56% higher than F5 with the lowest accuracy rate; similarly, F8 with the highest recall rate can reach 0.9457 which is only 3.78% higher than F10 with the lowest recall rate. In the performance of $F_1$-score, the maximum value is only 2.27% higher than the minimum value, showing that the recognition effect of this model with the fusion calculation characteristics of five key layers is comprehen-

sive, with excellent and balanced classification effect on each fault category.

### 5.2.3. The Classification Effect of Different Classifiers on the Data Set.

Several mainstream text classification models in NLP field are used to train the data set in this paper. After 30 epochs, the experimental results shown in Table 4 are obtained (2-BERT, 4-BERT, 6-BERT, 8-BERT, and 12-BERT, respectively, represent the BERT model with different transformer layers). The following comparative analysis is carried out:

Fasttext, CNN, BiLSTM, and BiGRU classification models all take the word vector generated by embedding as the upstream task and then send it into the model itself for the completion of text classification. The three evaluation indexes of BiGRU and BiLSTM models are close, and the difference in $F_1$-score is only 0.1%, which is because that BiGRU, as a variant of BiLSTM, has a similar structure, and the $F_1$-score of CNN model is about 2% higher than that of CNN model. Due to the fact that CNN has strong ability to extract local features of text, however, which cannot capture text information with longer distance between front and back, the effect of slightly longer text classification task will be discounted. Compared with the fasttext model, the $F_1$-score is increased by about 1%. As an improved version of word2vec model, the fasttext has simple classifier and fast classification speed, but its simple structure has general effect on the classification task of large-scale data. Both BiGRU and BiLSTM can effectively capture context information and generate high-level text semantic representation. Therefore, BiGRU or BiLSTM can be used as fault text classifier to achieve better classification effect.

As shown in Table 4, the comparison between 3 and 6 shows that the $F_1$-score of ALBERT-BiLSTM is 6.66% higher than BiLSTM. Compared with 4 and 7, the $F_1$-score of ALBERT-BiGRU is increased by 7.14% based on BiGRU. Then, by comparing the models after 4 and 8, the BiGRU with different layers of BERT model can increase the $F_1$-score of the original model by more than 7.5% based on the original model. Therefore, through the comparison with the feature vector generated by embedding, the feature vector obtained by using the BERT series model as the upstream task contains richer semantic information, which promotes the classification effect. ALBERT and BERT have the same model structure and learning style. The comparison between 7 and 8 shows that although ALBERT adopts the parameter-sharing mechanism to greatly reduce the number of parameters of the model and make the model structure simpler, the classification effect of large-scale data is not as good as BERT model, and the $F_1$-score is reduced by about 0.4%–1.7% compared with BERT model. Compared with 5 and 8, the $F_1$-score of 2-BERT with BiGRU as the downstream model is 2.94% higher than that of 2-BERT model alone, which is because the features classified by BERT model alone lack sequencing, and the addition of BiGRU can make up for this disadvantage. By comparing 6 and 7, it can be seen that although the structure of BiGRU and BiLSTM models is similar, the effect of BiGRU equipped with BERT system model is slightly better than BiLSTM. Therefore, BiGRU
can be chose as the downstream task of BERT model to achieve better classification effect.

The comparison of the experiments 8~11 shows that the more layers of BERT, the higher the $F_1$-score, which is because the information extracted from each part of the layer in BERT is different, and the more layers, the more comprehensive the extracted features, so as to improve the classification effect of the subsequent BiGRU model. Through the comparison between the experiments of 8 and 9 with model 11 in this paper, it can be seen that the $F_1$-score of this model is 0.16% and 1.08% higher than that of 2-layer and 4-layer BERT models, respectively, the reason is that this model covers the information encoded by 5-layer transformer, and the extracted features are richer. Compared with the experiments of 10 and 11, the $F_1$-score of this model is 0.59% and 0.24% higher than that of 8-layer and 12-layer BERT models, respectively, although the layers of this model only include the layers of 2, 4, 6, 8, and 12 of the original 12-layer BERT, and the number of layers is less than that of the corresponding models of the experiments of 10 and 11, covering the same comprehensive feature extraction range as the 12-layer BERT model, with fewer layers and lighter structure, which can reduce redundant information. Further, it can promote the classification effect and improve the accuracy and $F_1$-score of subsequent classification.

Finally, comparing this paper’s model with other PLM models, as shown in Figure 12, with all downstream classifiers being BiGRU, the model in this paper outperforms Distill-BERT, RoBERT, ALBERT, and BERT-wwm mentioned
in the review in the three metrics of precision, recall, and $F_1$-score. ALBERT and Distill-BERT have reduced the number of parameters based on the original BERT, but the comprehensiveness of extracted text information is not as good as other BERT models; RoBERTa improves the pretraining strategy of traditional BERT and regenerates MASK after every cycle, so the learning content is richer, so the performance is improved compared with ALBERT and Distill-BERT. BERT-wwm adds WWM mechanism to the original BERT, replacing single character MASK with full word MASK, which helps to capture semantic features at the Chinese word level, thus improving the overall performance of the model. The size is limited, and the complexity of RoBERTa and BERT-wwm models is greater, while the training process of the model proposed in the paper is simpler and can reduce the redundancy of information while ensuring the comprehensiveness of extracted information, so the model proposed in paper performs better on the metro on-board equipment fault dataset.

5.2.4. Influence of Different Layers of BERT on Convergence Rate. In this experiment, the $F_1$-score of five transformer layers with different BERT models is monitored with the change of epoch number. The iterative results are shown in Figure 13, and it can be seen that before 10 epochs, the model proposed in this paper and the 2-layer BERT model obtained higher $F_1$-score at the beginning of training, and their $F_1$-score is higher and faster than that of other models. After about 13 epochs, the $F_1$-score of this model exceeds that of the 2-layer BERT model and is still rising, while the $F_1$-score of the 2-layer BERT model tends to be stabilized slowly. Between 10 and 20 epochs, the $F_1$-score of the other three models rises, and the 2-layer BERT model tends to be stabilized at about 13 epochs. The $F_1$-score of this model exceeds the 2-layer BERT model after 13 epochs and tends to be stabilized at about 16 epochs. After about 20 epochs, the performance of all models fluctuates within a small range with the epoch value, and the $F_1$-score converges completely between 25 and 30 epochs. In the entire training process, the $F_1$-score of this model is higher than the 4, 8, and 12-layer BERT model in the whole process. After 13 epochs, the $F_1$-score is higher than the 2-layer BERT model. Compared with the 4, 8, and 12-layer BERT model, the convergence speed is faster, and the $F_1$-score reaches the highest value of 0.9272 at 30 epochs.

5.2.5. Influence of Different Layers of BERT on the Classification Effect of Key Fault Categories. For further comparison, this paper tested the BERT model with different layers and the model in this paper on six fault categories with the largest amount of data, namely, F1, F3, F4, F7, F8, and F9. Figures 14(a)–14(f) show the classification effects of five different layer BERT models on F1, F3, F4, F7, F8, and F9, respectively. In the classification of fault categories of F1, F8, and F9, the $F_1$-score of the model in this paper is higher than that of the other four models, which is because the amount of text data corresponding to these four types of faults is too large and covers comprehensive semantic information, including surface features, grammatical features, and semantic features. For the fault category F3, the classification effect of this model is close to and higher than that of 8-layer BERT, but the recall rate is lower than that of 8-layer and 12-layer BERT models. For the fault category F4, the $F_1$-score of the model in this paper is slightly lower than that of the 8-layer BERT model, the accuracy is lower than that of the 4-layer and 8-layer BERT models, and the recall rate is the highest; the reason is that the deep semantics in the fault text of F4 category is less, and most of them are presented in the form of surface features, while the low-level transformer in the BERT model is better in learning this kind of text. For the fault category F7, the accuracy, recall, and $F_1$-score of this model are slightly lower than those of the 12-layer BERT model.
<table>
<thead>
<tr>
<th>Value</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 layers</td>
<td>0.86</td>
<td>0.88</td>
<td>0.90</td>
</tr>
<tr>
<td>4 layers</td>
<td>0.92</td>
<td>0.92</td>
<td>0.94</td>
</tr>
<tr>
<td>8 layers</td>
<td>0.94</td>
<td>0.94</td>
<td>0.96</td>
</tr>
<tr>
<td>12 layers</td>
<td>0.96</td>
<td>0.96</td>
<td>0.98</td>
</tr>
<tr>
<td>Paper model</td>
<td>0.98</td>
<td>0.98</td>
<td>1.00</td>
</tr>
</tbody>
</table>

**Figure 14:** Influence of different layers of BERT on the classification effect of key fault categories: (a) classification effect of F1; (b) classification effect of F3; (c) classification effect of F4; (d) classification effect of F7; (e) classification effect of F8; (f) classification effect of F9.
model, which is because the fault text corresponding to the F7 fault has deeper semantics and more domain proper nouns. The high-level transformer in the BERT model shows strong ability to learn deep-seated text information, and the 12-layer BERT has more transformers located above 8 layers; therefore, the effect is slightly better than that of this model. However, in general, the model in this paper is more comprehensive for the identification of key fault categories, and the classification effect of each fault category is considerable in terms of value. The total three evaluation indexes are the highest compared with the other four BERT models.

6. Conclusion

To solve the problem of low classification accuracy and incomplete classification effect of unstructured fault data automatic classification task, in this paper, a BERT-BiGRU fault text classification model was proposed based on key layer fusion. Based on the experiment of metro on-board equipment fault text data with other common text classification models and BERT-BiGRU models with different transformer layers, the experimental results are shown as follows:

Compared with Fasttext, CNN, BiLSTM, BiGRU, ALBERT-BiLSTM, and ALBERT-BiGRU, the classification model with BERT as the upstream task has obvious advantages in classification precision, recall, and F1-score. The F1-score of the model in this paper is higher than that of the BERT-BiGRU model with 2, 4, 6, 8, and 12-layer transformers, and the convergence speed is second only to the 4-layer BERT-BiGRU model; compared with other PLMs such as RoBERTa, Distill-BERT, and BERT-wwm, the paper model performs better in three indicators precision, recall, and F1-score.

Compared with other BERT-BiGRU models with different layers, this model has a more comprehensive identification of key fault categories, which can achieve excellent classification results for each fault category.

In this paper, the model was used to extract the most critical 5-layer transformer module in the conventional 12-layer BERT, which can not only extract more comprehensive semantic features but also reduce the complexity and redundancy of multidimensional semantic information. In the meantime, BiGRU is used as the downstream task to extract more comprehensive context information. The final F1-score of the model reaches the highest value of 0.9272 in the experiment with high convergence speed. It can meet the performance requirements of high accuracy and rapidity of metro on-board equipment fault text classification and provide theoretical basis and application value for metro on-board equipment fault text classification.

However, the data set in this paper is mainly short text, and the experiment can only prove that the model in this paper works well for short text classification, because short text is short and the coverage rate of important information in short text is higher, and it is easier to learn important information in the text by this model. As the fault data generated during the operation of metro trains increases, long text fault data may appear in the database, so it is necessary to explore models suitable for long text classification, which could work together with the model proposed in paper in the field of TC of metro on-board equipment to solve the classification problems of long and short text, respectively.

Data Availability

The dataset can be accessed upon request.

Conflicts of Interest

The authors declare that they have no competing interest.

Acknowledgments

We gratefully acknowledge the valuable cooperation of Dr. Lin and other members of the laboratory in collecting data, debugging programs, and writing papers, as well as the National Natural Science Foundation of China (Grant No. 52162050) and the Natural Science Foundation of Gansu Province (Grant No. 20JR5RA375).

References


