Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 6374988, 13 pages
https://doi.org/10.1155/2022/6374988

Research Article

Research on Named Entity Recognition Method of Metro On-Board Equipment Based on Multiheaded Self-Attention Mechanism and CNN-BiLSTM-CRF

Junting Lin and Endong Liu

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

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

Received 8 May 2022; Accepted 12 June 2022; Published 6 July 2022

Academic Editor: Yaxiang Fan

Massive and complex unstructured fault text data will be generated during the operation of subway trains. A named entity recognition model of subway on-board equipment based on Multiheaded Self-attention mechanism and CNN-BiLSTM-CRF is proposed to address the issue of low recognition accuracy and incomplete recognition features of unstructured fault data named entity recognition task of subway on-board equipment: BiLSTM-CNN parallel network extracts context feature information and local attention information, respectively; In the MHA layer, the features learned from different dimensions are fused through the Multiheaded Self-attention mechanism, and the dependencies of various ranges in the sequence are captured to yield the internal structure information of the features. The conditional random field CRF is used to learn the internal relationship between tags to ensure their sequence. This model is tested with other named entity recognition models on the marked subway on-board fault data. The experimental results demonstrate that this model is able to recognize 10 kinds of labels in the dataset. Moreover, the recognition effect of each label has a good performance in the three evaluation indexes of P, R, and $F1$ score. Moreover, the weighted average evaluation indexes Avg $-$ P, Avg $-$ R, and Avg $-$ F_1 of 10 labels in this model reach the highest 95.39%, 95.48%, and 95.37%, which has high evaluation indexes and can be applied to the named entity recognition of Metro on-board equipment.

1. Introduction

Subway on-board equipment is the basic piece of equipment guaranteeing the safe operation of the subway train. On-board equipment is also constantly upgraded owing to the rapid development of China's urban rail transit. With the accumulation of subway operation mileage and operation time, a consequential amount of fault data about on-board equipment has been generated. These data record the detailed fault information in the form of text, containing useful knowledge of fault diagnosis and processing. However, given that it is stored in the form of unstructured text, it is not conducive for computer processing and understanding. It has long been delved into by field engineers and technicians who suggest that the fault knowledge cannot be reused efficiently. Therefore, for these large amounts of unstructured subway fault knowledge, knowledge entities should be efficiently identified and integrated, the fault cases and treatment methods in the fault knowledge should be identified, the subway knowledge map should be built, and field personnel should be provided with accurate subway fault information. The human-computer interaction platform provides field personnel with three kinds of information: subway fault information, fault causes, and other knowledge information. Moreover, the named entity recognition task related to subway on-board equipment also establishes a knowledge base to serve subway fault diagnosis, subway train information service, and subway information intelligent recommendation [1].

Named entity recognition (NER) [2] is an essential component of natural language processing (NLP) [3]. It aims to identify various named entities from the original text, such as name, location, and organization. It can subsequently extract the concerned information in the fault text data as named entities [4]. The extracted entities can subsequently pave the way for other NLP tasks. The methods of named entity recognition mainly include rule-based methods, statistics-based methods, and deep learning-based methods: Pan [5, 6] constructed the rule base of named entity recognition and used the method of rule matching to identify named entities. However, the rule writing based on rules and dictionary methods requires the involvement of domain experts, thereby requiring high language knowledge and poor portability. Therefore, statistical machine learning was employed to deal with the NER problem. In statistical machine learning, the main algorithms suitable for sequence annotation tasks are: Hidden Markov models (HMM) [7, 8], Maximum Entropy Markov models (MEMM) [9], conditional random field (CRF) [10], etc. However, the method based on machine learning requires a substantial amount of labeled data to train the model, requiring significant manpower, thereby leaving much room for improvement in recognition accuracy. In recent years, deep neural network has been used to realize the key tasks in the knowledge map due to the advent of deep learning technology [11], garnering extensive attention. The use of named entity recognition technology to identify entities in subway operation and maintenance logs is a basic step in the conversion of subway fault text into structured data, thereby laying a foundation for mining and developing the rich knowledge contained in a large amount of fault data recorded during subway operation [12]. The current mainstream deep learning solutions tend to embed layer and Bidirectional long short-term memory (BiLSTM) layer, allowing the machine to directly learn the features. It subsequently directly inputs the learned features into CRF, thereby circumventing the tedious task of manually formulating the feature function [13]. Literature [14, 15] uses the neural network model to learn the internal representation of text on a large number of unmarked datasets, which does not require the setting of artificial features. Literature [16] adopts the long short-term memory (LSTM) neural network model, boosting the performance of word segmentation. However, this method cannot yield the semantic information behind the sentence. Literature [17, 18] proposes that CRF is used as the processing mode of output processing layer on the basis of bidirectional LSTM, effectively improving the performance of the model. Furthermore, convolutional neural network (CNN) [19] has also achieved desirable results in solving NER problems; literature [20] uses CNN to obtain multilevel features, thereby yielding local attention information and improving the sensitivity of entity boundary information; literature [21] adopts the serial strategy of CNN and LSTM-CRF to recognize the named entity of the conll2003 English dataset, and obtains a higher F1 value. However, LSTM network cannot capture text information in both directions. Document [22] uses the Bidirectional gate recurrent unit (BiGRU) and CRF combined with CNN for named entity recognition, and uses the connection vector including affix vector, part of speech vector, and word vector as input. It ultimately outputs through the CRF layer, which can address the issue of automatic named entity recognition and exert a desirable effect on entity recognition. Document [18] proposed a method to fuse character and word vectors. It adopted the Chinese named entity recognition method of BiLSTM-CRF to effectively extract two features at character and word level, thereby effectively improving the accuracy of named entity recognition. Literature [23] adopts the BERT-CRF model, extracts the global features of the input sequence through the Bert pretraining model, adds the CRF layer at the end of the model, introduces hard constraints, and constructs the model framework of named entity recognition. However, the Bert model has a lengthy pretraining time, and it is only used as a transfer learning model, which is hindered by insufficient information recognition ability for small areas. Much research has been conducted in the field of railway text data analysis. In terms of named entity recognition, Yang [24] used word2vec to represent the characteristics of railway accident faults, and used BiLSTM-CRF to realize the named entity recognition of railway electrical service accident faults. Literature [17] uses BiLSTM-CRF to realize the named entity recognition of high-speed railway signal equipment and puts forward the entity relationship representation method of multidimensional word segmentation features, thereby achieving high evaluation indexes for the task of named entity recognition of high-speed railway signal equipment.

Based on the above literature research, this paper proposes a named entity recognition method for Metro on-board equipment based on multiheaded self-attention (MHA) and BiLSTM-CNN- CRF. The core idea of the method is as follows:

(1) YMDAA is used to complete the sequence annotation [25], and the location, phenomenon, and measure of the fault in the subway on-board fault text are marked and exported in an Ann format file. The file is subsequently read through Python and added to the BMEO label to complete the preannotation of fault text data.

(2) The tag and word sequence of the prelabeled fault text are input into the word2vec model and transformed into feature vectors. The strategy of CNN and BiLSTM working in parallel is adopted, whereby CNN and BiLSTM work simultaneously, extract the context and local attention features in the fault text, respectively, and ultimately fuse the two kinds of information.

(3) The multihead self-attention mechanism is adopted to give higher weight to the more important information in the input word sequence and label sequence. This mechanism can boost the sensitivity of the machine towards important information, mining the association between different input features to extract the feature vector containing other word information. The recognition ability of the machine to feature information can be more comprehensive by defining the number of heads of multiple groups of attention mechanisms, extracting important features from different dimensions, and splicing and linear processing these features [26].

2. Design of Named Entity Recognition Model Based on Multihead Self-Attention Mechanism and BiLSTM-CNN-CRF

Based on the multiheaded self-attention mechanism and BiLSTM-CNN-CRF, the named entity recognition model architecture of Metro on-board equipment is illustrated in Figure 1. It includes four main layers: word embedding layer, BiLSTM-CNN layer, MHA layer, and CRF layer.

In the word embedding layer, the subway on-board fault database is first loaded, the fault text records in the database are marked with BMEO through YMMDA, the word vector of large-scale marked text is subsequently trained in the same field as word2vec. The generated word vector is then input into the BiLSTM module and CNN module in the BiLSTM-CNN layer, respectively. The BiLSTM module is used to learn the time characteristics and context information of the text sequence, The CNN module is used to extract the local features in the text. The outputs of BiLSTM and CNN are then spliced and fed to the MHA layer to yield the global features of the text sequence and the correlation strength between words. Finally, the CRF layer marks the output sequence from the MHA layer according to the importance of the features and outputs the entity prediction label.

2.1. Word Embedding Layer. Data preprocessing is first performed on the prelabeled subway on-board fault text, which is subsequently segmented. Stop words and low-frequency words are then discarded. The accuracy of word segmentation exerts a direct impact on the training effect of the model, while the Jieba word segmentation tool may fail to identify some proper nouns in this field. Therefore, a dictionary of proper nouns in the subway on-board field should be defined according to relevant data and existing knowledge, to improve the reliability of word segmentation task and lay a foundation for the vectorization of text [27].

In this paper, word2vec model is used to train word vectors, transforming large-scale subway vehicle fault text and label data into low dimensional and dense word vectors. This model can reflect the relationship between words but does not necessarily ensure sufficient training of proper nouns. To address this issue, word2vec is used to train word vectors on the training set data and other corpora in the field. Word2vec trains the word vector through the skip gram model, whereby the central word predicts the words around it and solves the context word vector through the conditional probability value of the intermediate word vector, to fully learn the semantic vector representation [28]. Suppose the sample S is composed of n sentences, input the text sequence $S = [s_1, s_2, \cdots, s_n]$, the i sentence in the text sequence is represented as $s_i = [w_{i1}, w_{i2}, \cdots, w_{ik}]$, whereby k represents the number of words contained in the sentence s_i, and $\mathrm{w_{ik}}$ represents the k word in the i sentence. The skip gram model converts the input text sequence into word vector, and further generates the corresponding word vector matrix. $\mathbf{W}_{ij}$ represents the j word vector in the i sentence, and the word vector matrix of the sentence s_i with the length of k is represented as $\mathbf{W}_{i1:\,im} = [\mathbf{W}_{i1}, \mathbf{W}_{i2}, \mathbf{W}_{i3}, \cdots, \mathbf{W}_{ik}]$. Finally, $\mathbf{E} = [\mathbf{W}_{11:\,2k}, \mathbf{W}_{21:\,2k}, \mathbf{W}_{31:\,3k}, \cdots, \mathbf{W}_{n1:\,nk}]$, the word vector matrix spliced by n sentences in the sample S, is used as the output of the word embedding layer.

2.2. BiLSTM-CNN Layer. This layer adopts the strategy of BiLSTM and CNN working in parallel, whereby the feature vectors generated by word2vec are input to BiLSTM and CNN networks, respectively; the context features and local attention features are extracted respectively; and the two fusion features are subsequently input to the MHA layer.

2.2.1. BiLSTM. LSTM effectively calculates and controls the input and output of information by designing gating units in neurons. The design of this gating unit addresses the problem of text sequence length dependence. Its structure is illustrated in Figure 2.

The information of cell state C_{t-1} is transmitted through the top straight line. The hidden layer state h_t and input x_t at t time will modify C_t appropriately and then output to the next time. Moreover, C_{t-1} will participate in the calculation of h_t output at t time, and alter the cell state through the gate structure of LSTM. After connecting h_{t-1} and x_t, calculate with different weight matrices $(\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_i)$ and offset (b_f, b_i, b_o) through the sigmoid function, and output f_t, i_t, and o_t respectively. The calculation formula is shown in (1)–(3). The amount of information needed to be forgotten from the previous hidden layer h_{t-1} is controlled by multiplying f_t and C_{t-1}; the content is planned to (–1, 1) through the function, so that the updated cell $\widetilde{C}_t$ is multiplied with i_t to control which information needs to be retained. The calculation is shown in formula (4), where $\mathbf{W}_C$ denotes the weight matrix. When the information in the cell state C_t is completely updated, as shown in formula (5), it is scaled by tanh and multiplied by o_t to output h_t as the next LSTM hidden layer state. The calculation is shown in formula (6).

$$f_t = \sigma\left(\mathrm{W}_f \cdot [h_{t-1}, x_t] + b_f\right), \quad (1)$$

$$i_t = \sigma\left(\mathrm{W}_i \cdot [h_{t-1}, x_t] + b_i\right), \quad (2)$$

$$o_t = \sigma\left(\mathrm{W}_o \cdot [h_{t-1}, x_t] + b_o\right), \quad (3)$$

$$\widetilde{C}_t = \tanh\left(\mathrm{W}_C \cdot [h_{t-1}, x_t] + b_C\right), \quad (4)$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t, \quad (5)$$

$$h_t = o_t * \tanh(C_t). \quad (6)$$

However, given that the unidirectional LSTM model can only capture the information before the sequence and cannot capture the context semantics, Li et al. [27] improved the RNN model to yield the LSTM, which can solve the problems of gradient disappearance and gradient explosion that could occur in the process of long sequence training. BiLSTM is composed of forward propagating LSTM and back propagating LSTM. It captures the above and below

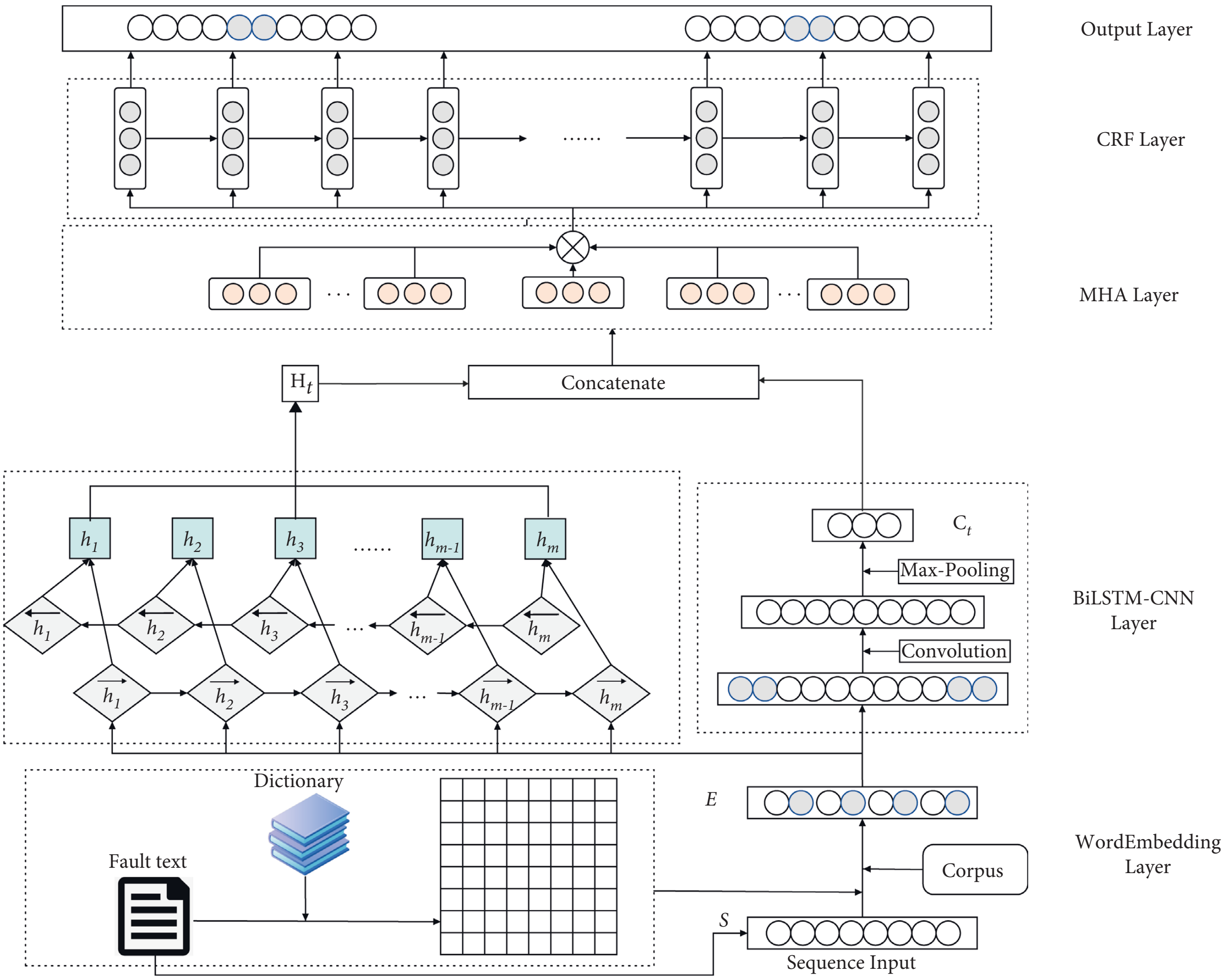


Figure 1: The overall model architecture of NER.

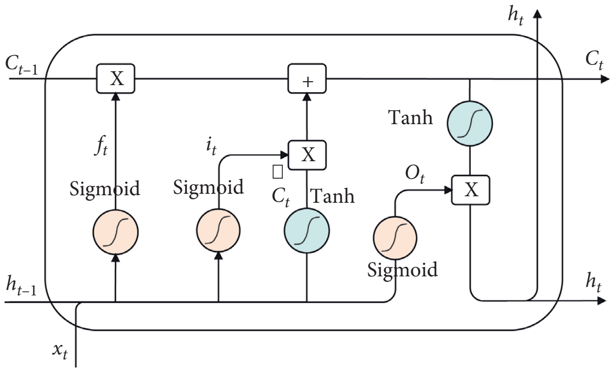


Figure 2: The structure of LSTM.

information of the current text, respectively, and then combines the feature information extracted from the two directions to yield the text features of remembering the past and the future. The word vector matrix E obtained through the word embedding layer is input to the BiLSTM part in the BiLSTM CNN layer as illustrated in Figure 1. The LSTM forward propagation generates the forward hidden layer state sequence: $\mathbf{H}_1 = [\overrightarrow{h_1}, \overrightarrow{h_2}, \ldots, \overrightarrow{h_m}]$, and the reverse hidden layer state sequence: $\mathbf{H}_2 = [\overleftarrow{h_1}, \overleftarrow{h_2}, \ldots, \overleftarrow{h_m}]$. The forward hidden layer state sequence H_1 is spliced with the reverse hidden layer state sequence H_2 to obtain the complete hidden layer state sequence $\mathbf{H}_t = [\overrightarrow{h_m}, \overleftarrow{h_m}]$, where m represents the dimension of the BiLSTM input word vector. This combination of forward and reverse states gives full play to the advantages of BiLSTM and addresses the issue whereby the traditional one-way LSTM model fails to capture the context information. It fully combines the context and extracts the features through the overall environment, which can substantially mitigate feature loss. The hidden layer state sequence $\mathbf{H}_t = [h_1, h_2, \ldots, h_m]$ is the final output of the BiLSTM layer and is input to the MHA layer.

2.2.2. CNN. The word vector matrix set generated by the word embedding layer is input to the CNN layer. The CNN layer includes two steps: convolution and max pooling. Its working process is depicted in Figure 3. Convolution is the

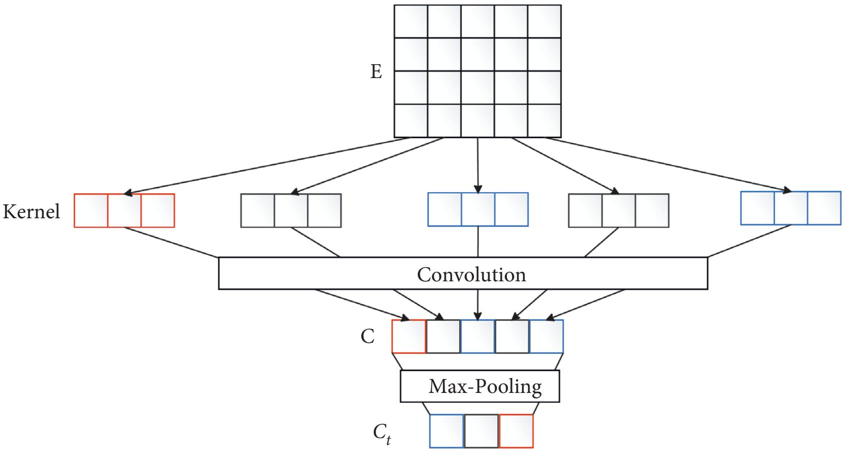


Figure 3: Working principle of the CNN layer.

use of different sizes of convolution to check the input eigenvector matrix for feature calculation, and then extracting the local feature information of the text. The operation process of convolution can be expressed as follows:

$$c_i = f\left(\mathbf{F} \cdot \mathbf{W}_{\mathrm{i:j}} + b\right), \tag{7}$$

where c_i denotes the ith eigenvalue of the text output through convolution operation; F represents the matrix corresponding to the convolution kernel; f is a nonlinear activation function; · indicates that the two matrices are multiplied by points; $\mathbf{W}_{i:\,j}$ represents the word vector matrix from i word to j word; and b is the offset term. Convolution operation on the characteristic matrix of each word vector is carried out in the input, and the characteristic graph $\mathbf{c}$ is calculated through formula (8).

$$\mathbf{c} = \left(c_1, c_2, \cdots, c_n\right). \tag{8}$$

The pooling layer samples the text features by setting a fixed step stripe. In this paper, the maximum pooling strategy max pooling is used for pooling processing. This process aims to effectively extract the local key information in the sequence, compress the input feature map, reduce the size of the feature map $\mathbf{c}$, to simplify the network calculation, and finally calculate the output fixed length vector $\mathbf{C}_{\mathrm{t}}$ through formula (9).

$$\mathbf{C}_{\mathbf{t}} = \mathrm{Max} - \mathrm{pooling}\left(\mathbf{c}\right). \tag{9}$$

2.3. MHA Layer. The output $\mathbf{H}_t$ of BiLSTM network and the output $\mathbf{C}_t$ of CNN network are spliced into a feature vector $\mathbf{X}_t$ with a dimension of 320 (the dimension of feature vector $\mathbf{H}_t$ is 256 and the dimension of feature vector $\mathbf{C}_t$ is 64). However, this feature vector cannot display the importance of key information in the context, which could entail the loss of important information in the named entity recognition task. Therefore, the introduction of the multiheaded self-attention mechanism is essential to learn the dependence between any two words in the sentence, obtain the internal structure information, and distinguish the significance of each word. The calculation principle of self-attention mechanism is illustrated in Figure 4.

Taking the feature x_1 in $\mathbf{X}_t$ as an instance, the self-attention mechanism initializes the $\mathbf{W}^{Q}$, $\mathbf{W}^{K}$, and $\mathbf{W}^{V}$ matrices, and obtains $\mathbf{Q}_n$, $\mathbf{K}_n$, and $\mathbf{V}_n$ matrices, respectively, by multiplying with the input feature x_1 points, as shown in formula (10). It then calculates the attention a_{1n} from formula (12) to represent the correlation degree between the feature x_1 and the feature x_n.

$$x_1 \cdot \mathbf{W}^{Q} = \mathbf{Q}_n x_1 \cdot \mathbf{W}^{K} = \mathbf{K}_n x_1 \cdot \mathbf{W}^{V} = \mathbf{V}_n, \tag{10}$$

$$a_{1n} = \mathrm{softmax}\left(\frac{\mathbf{Q}_1 \mathbf{K}_n^{T}}{\sqrt{d_k}}\right). \tag{11}$$

Among them, $\mathbf{Q}_n$ represents the query matrix, $\mathbf{K}_n$ represents the key value matrix, $\mathbf{V}_n$ represents the score matrix, and n is the serial number corresponding to other input features. Through the combination of a_{1n} and $\mathbf{V}_n$, the association $\mathbf{head}_{1n}$ between x_1 and other different features is obtained. The calculation is shown in formula (12). All the feature vectors $\mathbf{head}_{1n}$ are added and the Z_{11} vector is calculated and can represent the connection between the first word and other words through formula (12).

$$\mathbf{head}_{1n} = a_{1n}\mathbf{V}_n, \tag{12}$$

$$\mathbf{Z}_{11} = \mathbf{Concat}\left(\mathbf{head}_{11}, \mathbf{head}_{12}, \ldots, \mathbf{head}_{1n}\right). \tag{13}$$

The MHA calculation principle is depicted in Figure 5, whereby the multiple groups of $\mathbf{W}^{Q}$, $\mathbf{W}^{K}$, and $\mathbf{W}^{V}$ matrices are initialized, multiple groups of $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ characteristic matrices are generated through point multiplication, thereby yielding multiple groups of Z_{1t}. After completing the splicing of multiple groups of Z_{1t}, the dimension is reduced through linear transformation, whereby t denotes the number of self-attention heads, to obtain $\mathbf{Z}_1$ containing other feature information. MHA linearly maps the input features to different information subspaces through different weight matrices, and calculates the same attention function

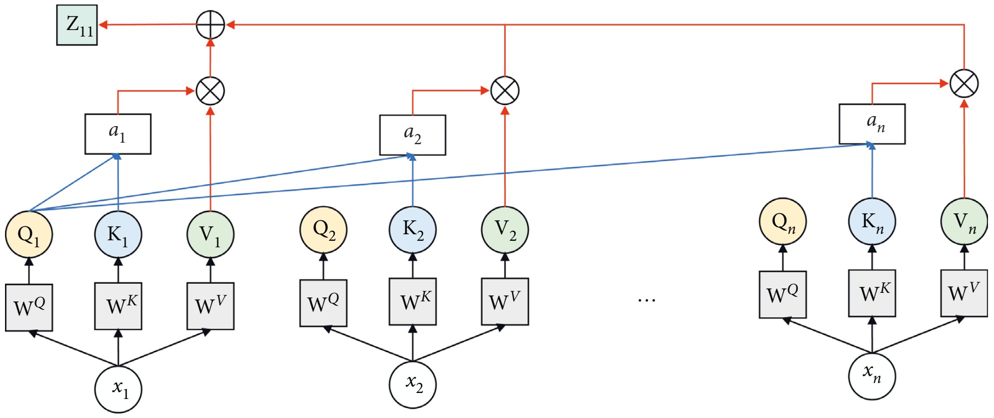


Figure 4: Calculation schematic diagram of self-attention mechanism.

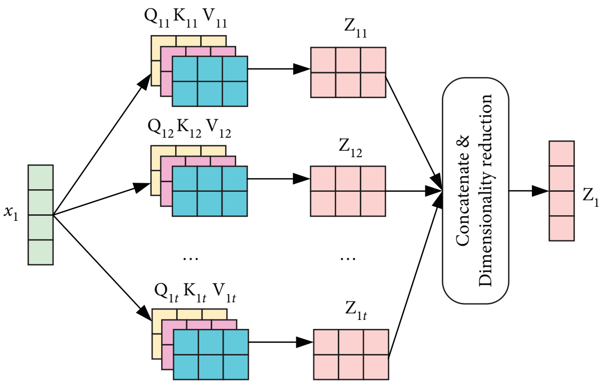


Figure 5: MHA calculation schematic diagram.

in each subspace, thereby expanding the ability of the model to consider different positions, to fully understand the structure and semantics of sentences. The output x_1' of the final MHA is calculated by the tanh function from $\mathbf{Z}_1$ and the input characteristic x_1, as shown in formula (14). The MHA value of other features x_n in X_t is calculated as above.

$$\mathbf{Z}_1 = \tanh\left(x_1 \oplus \mathbf{Z}_1\right). \tag{14}$$

2.4. CRF Layer. BiLSTM only considers the long-term dependency information of sentences but overlooks the dependency between tags. For instance, in the entity tags defined in this paper, b-phenomenon cannot appear after the m-phenomenon. Therefore, CRF needs to be introduced to learn the internal relationship between tags to ensure the sequence of tags. The conceptual diagram of CRF conditional random field is depicted in Figure 6.

The conditional random field model, CRF, is based on the calculation of a given random variable sequence $\mathbf{X} = (x_1, x_2, \cdots, x_n)$. The conditional probability distribution of the random variable sequence $\mathbf{Y} = (y_1, y_2, \cdots, y_n)$ is $P(\mathbf{X}|\mathbf{Y})$, and n denotes the sequence length. The model assumes that the random variable sequence satisfies the Markov property:

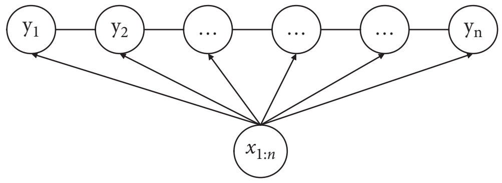


Figure 6: Conceptual diagram of CRF.

$$P\left(y_i|\mathbf{X}, y_i, \cdots, y_n\right) = P\left(y_i|\mathbf{X}, y_{i-1}, y_{i+1}\right). \tag{15}$$

$P(\mathbf{X}|\mathbf{Y})$ can subsequently represent the linear chain conditional random field. In the labeling problem, $\mathbf{X}$ represents the input observation sequence, $\mathbf{Y}$ represents the corresponding output mark sequence or state sequence, and the evaluation score $\mathrm{Score}(\mathbf{X}, \mathbf{Y})$ can be obtained through formula (16).

$$\mathrm{Score}(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{n} \mathbf{P}_{i,y_i} + \sum_{i=0}^{n} \mathbf{W}_{y_i, y_{i+1}}, \tag{16}$$

whereby $\mathbf{W}$ represents the transition matrix, and $\mathbf{W}_{i,j}$ represents the state transition score from the i character to the j character. $\mathbf{P}$ denotes the weight matrix output by the decoding layer, $\mathbf{P}_{i,y_i}$ represents the probability that the ith word is marked as y_i, and exp represents the exponential function of the natural constant e. Assuming that the input sentence feature is $\mathbf{X}$, the probability distribution of the output sequence y' is $P(y'|\mathbf{X})$. Finally, the maximum probability is yielded by the maximum likelihood estimation in the process of fitting the model. The calculation process is shown in formula (17).

$$\begin{aligned} \log P\left(y'|\mathbf{X}\right) &= \log \frac{\exp\left(\mathrm{score}\left(\mathbf{X}, y'\right)\right)}{\sum_{i=0}^{n} \exp\left(\mathit{score}\left(\mathbf{X}, y\right)\right)} \\ &= \mathrm{score}\left(\mathbf{X}, y'\right) - \log\left(\sum_{i=0}^{n} \exp\left(\mathit{score}\left(\mathbf{X}, y\right)\right)\right). \end{aligned} \tag{17}$$

3. Data Set and Experimental Evaluation Index

3.1. CRF Layer. The named entity recognition method of subway on-board equipment requires the deep learning method of supervised learning. Therefore, the sample data labeling is required before training. According to the fault knowledge structure, the fault text data of each Metro on-board equipment define three types of named entities: fault location, fault phenomenon, and fault solution. The named entity identification sequence is represented by BMEO, where B (begin) represents the starting position of the entity, M (middle) represents the middle part of the entity, and E (end) represents the end character of the entity, O (other) represents a nonentity character, and "–" is used to connect the sequence annotation symbol with the defined entity type. Therefore, this paper selects the fault text data recorded in the depot of a subway company from 2016 to 2021 according to the functions and fault characteristics of each equipment. After preprocessing these fault text data, the total amount of data is 51652 marked data, divided into the training set, development set, and test set; 41526 pieces of data are selected as the training set data as the data samples of model fitting; 5035 pieces of data are development set data, which are used to adjust parameters, select features, and make other decisions on learning algorithms; 5091 pieces of data are test set data, used in model evaluation. The knowledge annotation of some data in this paper and the process of input to the model in this paper are illustrated in Figure 7.

3.2. Experimental Evaluation Index. In this paper, the Precision, Recall, and $F_1 -$ Score are used as the evaluation indexes of this experiment, whereby TP represents the number of samples classified and divided correctly; FP represents the number of samples classified and divided incorrectly; FN indicates the number of unclassified samples, which are wrong.

3.2.1. Precision. The accuracy rate is only for the positive samples with correct prediction, as opposed to all 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 formula (8):

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\%. \tag{18}$$

3.2.2. Recall. It is calculated by dividing the predicted correct number of positive samples by the actual number of positive samples in the test set; it shows that the number of samples that are really positive can be recalled by using the classifier, as shown in formula (9):

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\%. \tag{19}$$

3.2.3. F1-Score. $F_1 -$ Score is the harmonic average of accuracy rate and recall rate. Both Precision and Recall are expected to be higher; however, both these indicators are contradictory and cannot both be high. Therefore, $F_1 -$ Score should be introduced as an appropriate threshold point to maximize the ability of the classifier, as shown in formula (10):

$$\mathrm{F}_1 - \text{Score} = \frac{2}{1/\text{Precision} + 1/\text{Recall}} \times 100\%. \tag{20}$$

3.2.4. Weighted Average. In this paper, the evaluation indexes $\mathrm{Avg} - P$, $\mathrm{Avg} - R$, and $\mathrm{Avg} - F_1$ are defined as the weighted average values of 10 entity labels' Precision, Recall, and ,$F_1 -$ Score respectively. The calculation process is shown in formula (21).

$$Avg - P = \frac{\sum_{i=1}^{10} N_i \cdot P_i}{\sum_{i=1}^{10} N_i}\ Avg - R = \frac{\sum_{i=1}^{10} N_i \cdot R_i}{\sum_{i=1}^{10} N_i}\ Avg - F_1 = \frac{\sum_{i=1}^{10} N_i \cdot F_i}{\sum_{i=1}^{10} N_i}. \tag{21}$$

Where i denotes the value corresponding to the entity category (there are 10 named entity categories in this paper), N_i represents the number of entities in this category, and P_i, R_i, and F_i represent Precision, Recall, and $F_1 -$ Score corresponding to class i entities, respectively.

4. Experimental Verification

4.1. Experimental Environment. The experimental hardware includes a i7-6700HQ CPU, a GTX960M graphics card, a video memory of 8G, a Win10 64bit operating system, a 3.60 python version, a Spider 5.0.5 development tool, and a 1.11.1GPU Pytorch version.

4.2. Experimental Super Parameter Optimization

4.2.1. MHA Attention Heads Number Selection and Other Parameter Settings. The number of attention heads is set to t in the MHA layer. During the operation of the MHA layer, the input features need to be divided into t parts, and the dimension of the feature vector X_t input to the MHA layer is 320. It is necessary to ensure that the set number of attention heads' t value is divisible by 320. Therefore, this paper selects MHA with the number of attention heads of 2, 4, 5, 8, and 10 to test the model in this paper, and the experiment uses all the parameters in Table 1 except attention_heads, and the optimizer chosen is Adam.

It can be inferred from Figure 8 that after the addition of the multihead self-attention mechanism, the prediction results of the model gradually improves with the increase of the number of self-attention heads; when the head is 8, the $\mathrm{Avg} - P$, $\mathrm{Avg} - R$, and $\mathrm{Avg} - F_1$ of the model reach the optimum, and the $Avg - F_1$ of the model is increased by 0.52%, 0.46%, and 0.34%, respectively, compared to the number of self-attention heads of 2, 4, and 5. By further increasing the number of heads, the accuracy of the model decreases. This finding arises because an excessive number of attention heads will lead to overfitting of the model. The

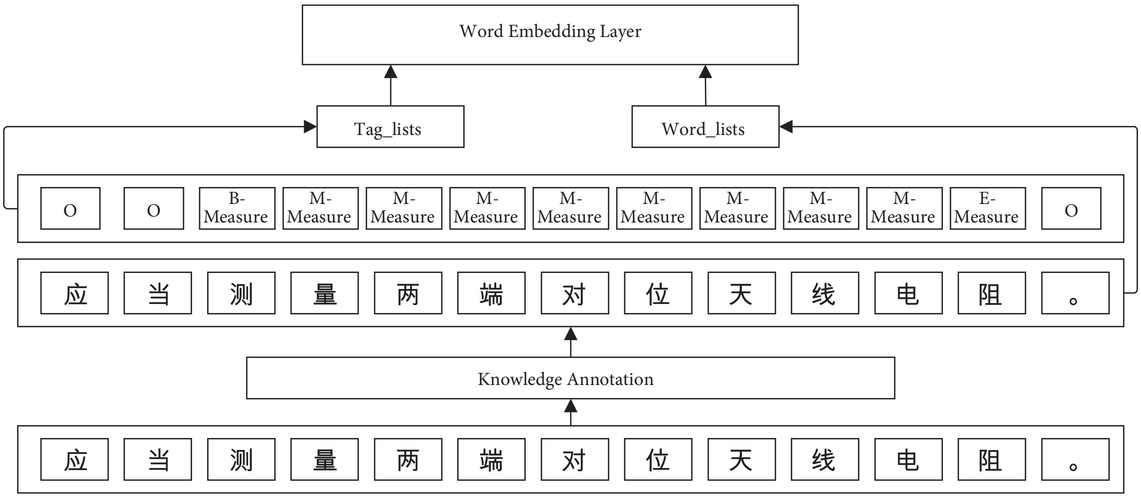


Figure 7: Knowledge annotation and input process of fault text of metro on-board equipment.

Table 1: Parameter setting.

Parameter	Value
Batch_size	64
Learning rate	$8e-4$
Hidden layer dimension	128
Epoch	25
LSTM_dim	128
cnn_size	64
attention_heads	8
kernel_size	(3, 4, 5)
Activation function	Relu
Max_seq_len	128
Dropout	0.5
Loss function	Cross entropy

number of MHA attention heads is set to 8, and the parameter settings are shown in Table 1.

4.2.2. Optimizer Selection. In the deep learning task, the optimizer is used to update and calculate the network parameters affecting the model training and model output, to approximate or reach the optimal value, and to minimize (or maximize) the loss function. In this experiment, five commonly used optimizers are selected: SGD (stochastic gradient descent), momentum optimization method, adaptive learning rate optimization algorithm Adagrad, RMSprop, and Adam. SGD selects a mini batch each time and uses the gradient descent to update the model parameters; Momentum optimization method adds the momentum optimization mechanism based on SGD. The Adagrad algorithm automatically attenuates the learning rate by using the number of iterations and cumulative gradient; RMSprop adds iterative attenuation; The Adam optimizer dynamically adjusts the learning rate of each parameter by using the first-order moment estimation and second-order moment estimation of the gradient [28]. In this experiment, the five optimizers are applied to the named entity recognition training task of this model. The parameter values set in Table 1 are used as the parameter values of this experiment, and The alpha of RMSprop is 0.99 and eps is $1e-08$; the beta1 of Adam is 0.9, beta2 is 0.999, and eps is $1e-08$, and the rest of the optimizers are the system default parameters. In the process of model training, the variation of loss function value loss with iteration step in the 5th, 10th, 15th, 20th, and 25th epoch rounds is shown in Figures 9(a) to 9(e), respectively.

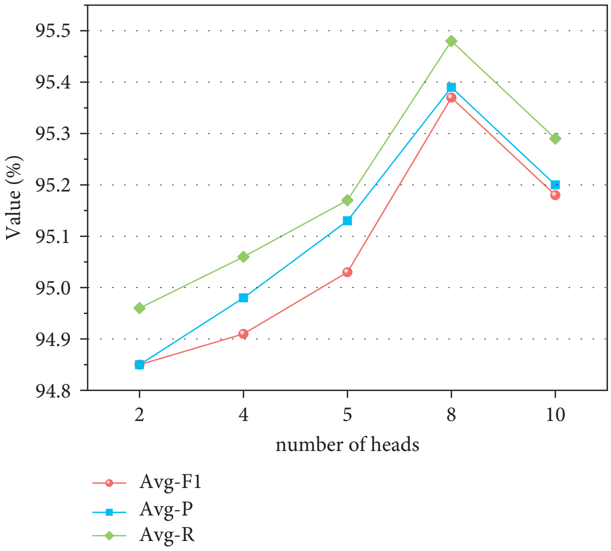
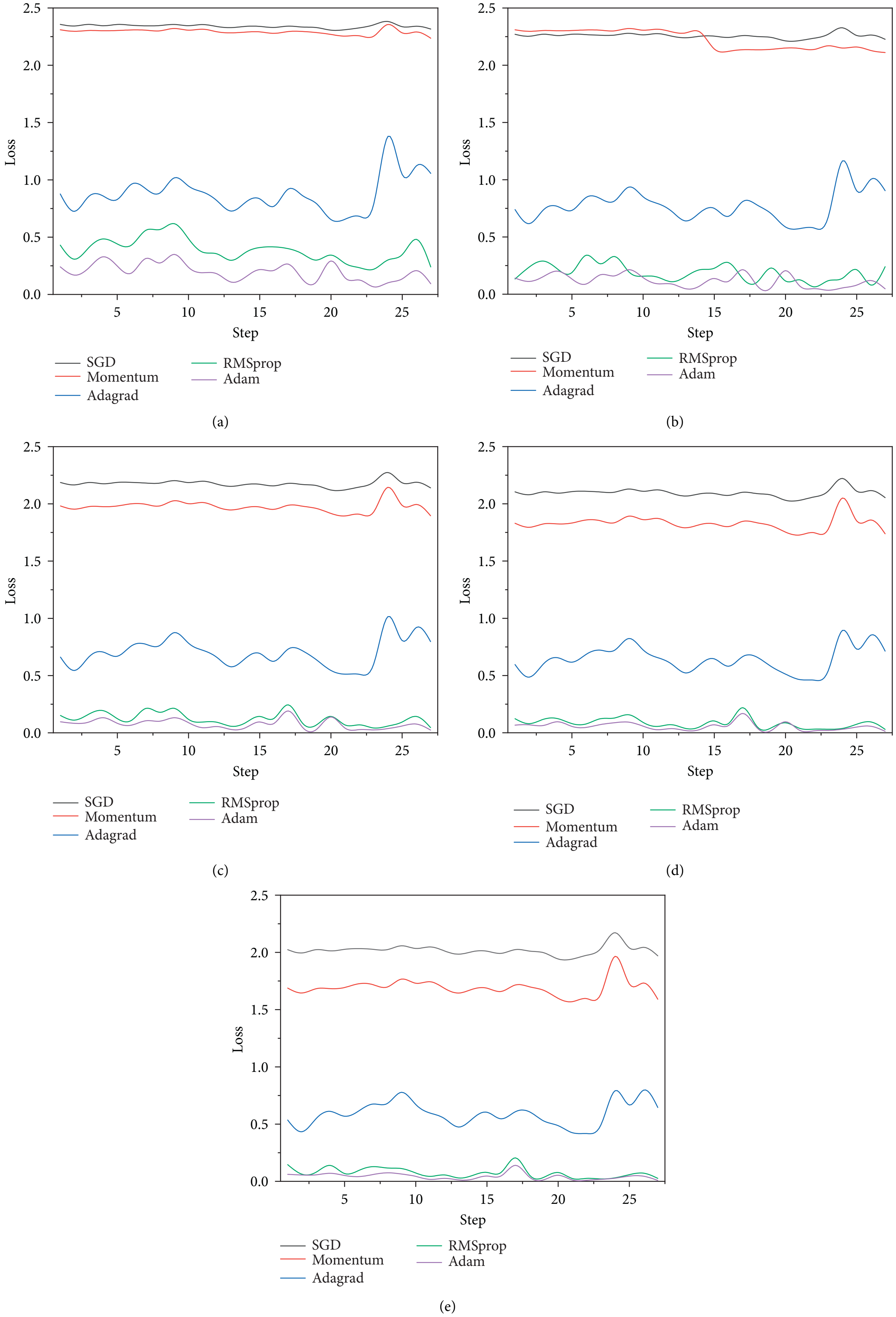


Figure 8: Effects of different attention heads on model performance.

Adam and RMSprop have the smallest loss function value and stable iterative waveform, while Adam showcases better performance in these two aspects. Therefore, Adam is selected for subsequent experiments in this paper. Moreover, it can be inferred that with the increase of training rounds epoch and training samples in each round, the loss function value loss constantly declines, and finally tends towards



(a)

(b)

(c)

(d)

(e)

Figure 9: The loss function value of different optimizers varies with the number of iteration steps in different epochs: (a) 5th Epoch; (b) 10th Epoch; (c) 15th Epoch; (d) 20th Epoch; (e) 25th Epoch.

stability and 0, indicating that the parameter values in Table 1 optimizes the network training of this model.

4.3. Overall Comparison

4.3.1. Experimental Super Parameter Optimization. Several common named entity recognition models in NLP field are used to train the dataset in this paper. After 25 rounds of epoch, the recognition results of 10 entity labels in this paper are generated, as listed in Table 2 respectively. The following comparative analysis is carried out:

The recognition results of phenomenon are shown in Table 2. The *F*1 values of HMM model for BME of phenomenon are 65.52%, 80.33%, and 71.93%, respectively. Compared with (Table 3) CRF model, the F1 value of HMM model for entity (Table 4) label b-phenomenon is 4.81% lower, the F1 value of m-phenomenon is 2.01% lower, and the F1 value of e-phenomenon label is 4.99% lower. These findings arise as the limitation of HMM model is that it uses the trained local model to make a global prediction. The independent BiLSTM model is slightly better than the CRF model for the recognition of these three named entities, and the *F*1 value of Table 5 phenomenon's BME is increased by 2.61%, 2.87%, and 0.97%, respectively, because the single CRF model can only capture the internal relationship between named entities within a certain range. It fails to capture long-distance previous and subsequent information. However, the disadvantage of BiLSTM is that it has a general learning effect on the internal relationship between tags, which can be compensated by combining it with the CRF model. Therefore, the *F*1 value of BiLSTM CRF for entity tag b-phenomenon and e-phenomenon is increased by 0.33% and 1.87%, respectively, compared with BiLSTM, and the *p* value of entity tag m-phenomenon is increased by 3.75%. BiLSTM-CNN-CRF adds a CNN network based on BiLSTM-CRF. The output results of CNN and BiLSTM are fused into the CRF network to complete the named entity work and boost the extraction ability of local features. Therefore, the *F*1 value of BME named entities of phenomenon is increased by 1.4%, 1.63%, and 3.45%, respectively, compared with BiLSTM-CRF, and 1.73%, 0.9%, and 5.23%, respectively, compared with BiLSTM. Moreover, the *P* and *R* values are also improved to varying degrees. BiLSTM-CNN-MHA-CRF adds MHA based on BiLSTM-CNN-CRF to learn the dependency between any two words in the sentence and yields the internal structure information. The *F*1 value of BME three named entities has increased by 0.33%, 1.46%, and 0.83%, respectively. Although the *p* value has decreased slightly, the *R* value has increased by about 4–5 percentage points compared with BiLSTM-CNN-CRF, with a more balanced recognition effect of phenomenon entities.

Most location type named entities correspond to station names, place names, or section names, with relatively fixed names. Therefore, the six models have good recognition effects on location type named entities. The *P*, *R*, and *F*1 of BiLSTM-CNN-MHA-CRF have reached over 90%. Compared with the other five models, the *p* values for BME tag recognition of location are 97.37%, 94.27%, and 94.87%, respectively, which are the highest values. Also, the *F*1 values have reached the highest at 93.67%, 92.79%, and 92.50%, respectively. The recognition effect on the next three BME entity tags is relatively balanced.

The recognition effect of three named entity labels of BME of measure is shown in Table 4. The *F*1 value of b-measure in this paper is only second to BiLSTM-CNN-CRF and BiLSTM-CRF, at 86.73%. The recognition effect of E-MEASURE is general, with *P*, *R*, and *F*1 values of 89.58, 81.69, and 85.45%, respectively. The reason is that the number of these two entity labels is the least, resulting in insufficient learning of these two types of labels in the main model and the inability to play the role in MHA mechanism. However, for the largest number of entity tags with a complex structure, the *P* recognized by the m-measure is 96.23%, which is 4–19 percentage points higher than other models. The *F*1 value reached 91.07%, which is also the highest value among the six models, which is 5–13 percentage points higher than other models.

The recognition effect of other nonentity tags *o* is shown in Table 5. Tag O indicates a type of the label o, as the largest number of 10 tag types, the P, R and F1 values of the proposed six models exceed 94%, while the recognition effect of this model is slightly better. The *P*, *R*, and *F*1 values reach 96.37%, 98.87%, and 97.60%, respectively, and the *F*1 value is 0.6–2 percentage points higher than that of other models.

To sum up, for the identification of the above 10 Tags, the other five models except the model in this paper have relatively good recognition effects on BME entity tag and nonentity tag *o* in location. Also, the model in this paper has better recognition effects on these types of tags. The recognition effect of phenomenon's BME entity label is relatively poor, as the description of fault phenomenon will be detailed to each component. Given the numerous components of subway on-board equipment, the description of fault phenomenon is relatively complex, and the model fitting is more complex. Using the model proposed in this paper, the *F*1 value recognized by BME tag and m-measure tag of phenomenon is substantially improved, and the *F*1 value recognized by b-phenomenon is about 0.4–10 percentage points higher than that of other models. The *F*1 value of m-phenomenon increased by about 1.4–7 percentage points; the *F*1 value of e-phenomenon increased by about 2–4 percentage points; the *F*1 value of m-measure increased by about 5–13 percentage points. Due to the insufficient number of b-measure and E-MEASURE entity labels, the recognition effect of this model on b-measure and E-MEASURE entity labels is general; however, this model improves the recognition effect of 8 entity labels except for b-measure and E-MEASURE.

4.3.2. Overall Recognition Effect. Further comparison of the weighted average evaluation indexes Avg $-$ *P*, Avg $-$ *R*, and Avg $- F_1$ of the six models on the recognition results of 10 labels. As depicted in Figure 10 and Table 6, in terms of Avg $-$ *P*, Avg $-$ *R*, and Avg $- F_1$, the combined model IV–VIII outperforms I, II, and III. Compared with IV, V, and VI, Avg $-$ *P*, Avg $-$ *R*, and Avg $- F_1$ of V are increased

Table 2: Recognition effect of different named entity recognition models on phenomenon.

Models	B-phenomenon			M-phenomenon			E-phenomenon		
	P (%)	*R* (%)	*F*1 (%)	*P* (%)	*R* (%)	*F*1 (%)	*P* (%)	*R* (%)	*F*1 (%)
HMM	55.58	79.17	65.52	74.90	86.61	80.33	62.12	85.42	71.93
CRF	74.42	66.67	70.33	86.49	78.57	82.34	81.40	72.92	76.92
BiLSTM	83.78	64.58	72.94	84.92	85.49	85.21	78.72	77.08	77.89
BiLSTM-CRF	81.25	66.72	73.27	88.67	80.67	84.48	84.85	75.09	79.67
BiLSTM-CNN-CRF	84.85	66.67	74.67	95.59	78.33	86.11	91.43	76.19	83.12
BiLSTM-CNN-MHA-CRF	78.95	71.43	75.00	93.67	82.22	87.57	87.18	80.95	83.95

Table 3: Recognition effect of different named entity recognition models on location.

Models	B-location			M-location			E-location		
	P (%)	*R* (%)	*F*1 (%)	*P* (%)	*R* (%)	*F*1 (%)	*P* (%)	*R* (%)	*F*1 (%)
HMM	92.92	91.67	92.29	91.26	92.57	91.91	91.18	86.11	88.57
CRF	96.88	86.11	91.18	95.28	86.43	90.64	93.75	83.33	88.24
BiLSTM	96.43	75.00	84.37	92.81	92.14	92.47	91.43	88.89	90.14
BiLSTM-CRF	94.87	90.24	92.50	91.30	90.74	91.02	92.31	87.80	90.00
BiLSTM-CNN-CRF	97.30	87.80	92.31	92.31	87.80	90.00	93.67	91.36	92.50
BiLSTM-CNN-MHA-CRF	97.37	90.24	93.67	94.27	91.36	92.79	94.87	90.24	92.50

Table 4: Recognition effect of different named entity recognition models on measure.

Models	B-measure			M-measure			E-measure		
	P (%)	*R* (%)	*F*1 (%)	*P* (%)	*R* (%)	*F*1 (%)	*P* (%)	*R* (%)	*F*1 (%)
HMM	67.14	85.42	75.20	77.99	94.88	85.61	75.71	96.36	84.80
CRF	84.62	80.00	82.24	89.40	76.38	82.38	90.38	85.45	87.85
BiLSTM	86.00	78.18	81.90	92.11	68.90	78.83	91.67	80.00	85.44
BiLSTM-CRF	92.59	84.75	88.50	91.37	82.04	86.46	89.29	84.75	86.96
BiLSTM-CNN-CRF	92.41	84.94	88.52	90.76	79.58	84.80	94.34	84.75	89.29
BiLSTM-CNN-MHA-CRF	90.74	83.50	86.73	96.23	86.44	91.07	89.58	81.69	85.45

Table 5: Recognition effect of different named entity recognition models on other nonentity tags *O*.

Models	*O*		
	P (%)	*R* (%)	*F*1 (%)
HMM	97.70	92.76	95.17
CRF	94.99	97.91	96.43
BiLSTM	95.20	97.63	96.40
BiLSTM-CRF	95.26	98.64	96.92
BiLSTM-CNN-CRF	96.59	98.79	97.01
BiLSTM-CNN-MHA-CRF	96.37	98.87	97.60

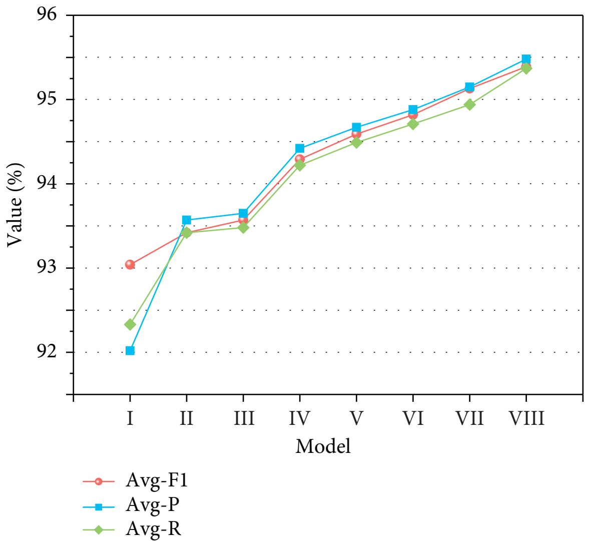


Figure 10: Weighted average of different named entity recognition models.

by 0.3%, 0.25%, and 0.27%, respectively, compared with and those of VI is increased by 0.53%, 0.46%, and 0.49%, respectively, compared with IV, indicating that adding CNN can improve the ability of extracting local features. The introduction of MHA largely makes up for the lack of BiLSTM's ability to capture the association relationship between words when processing long sequences and can capture various semantic features and highlight the key information of characters, the level of words and sentences. Therefore, the $\mathrm{Avg}-P$, $\mathrm{Avg}-R$, and $\mathrm{Avg}-F_1$ of VIII are 0.57%, 0.6%, and 0.66% higher than those of VI, respectively, and the $\mathrm{Avg}-P$, $\mathrm{Avg}-R$, and $\mathrm{Avg}-F_1$ of VIII are the highest among the six models, 95.39%, 95.48%, and 95.37%, respectively. Both V and VII models connect BiLSTM and CNN in series, whereby the word vector generated by

Table 6: Weighted average of different named entity recognition models.

No	Models	Avg − P (%)	Avg − R (%)	Avg − F_1 (%)
I	HMM	93.04	92.02	92.33
II	CRF	93.42	93.57	93.42
III	BiLSTM	93.57	93.65	93.48
IV	BiLSTM-CRF	94.29	94.42	94.22
V	(BiLSTM-CNN)$_{\text{SPI}}$-CRF	94.59	94.67	94.49
VI	BiLSTM-CNN-CRF	94.82	94.88	94.71
VII	(BiLSTM-CNN)$_{\text{SPI}}$-MHA-CRF	95.13	95.15	94.94
VIII	BiLSTM-CNN-MHA-CRF	95.39	95.48	95.37

Word2Vec is first input to BiLSTM. The output of BiLSTM is subsequently used as the input of CNN. Compared with V, VI, VII, and VIII, VI has increased by 0.23%, 0.21%, and 0.22%, respectively, in three indexes compared with V; VIII compared with VII, the three indexes are increased by 0.26%, 0.33%, and 0.43%, respectively, indicating that the effect of parallel work of BiLSTM and CNN outperforms serial work.

Therefore, adopting the strategy of BiLSTM and CNN working in parallel and effectively combining MHA and BiLSTM-CNN-CRF can improve the recognition effect compared with other named entity recognition models, which is of great significance to improve the overall performance of the named entity recognition model of Metro on-board equipment.

5. Conclusion

Based on unified labeling of Metro on-board equipment fault text data, aimed at solving the problem of low accuracy of naming entity recognition task of unstructured Metro on-board fault data, this paper proposes a Metro on-board equipment naming entity recognition model based on multihead self-attention mechanism and CNN BiLSTM CRF. Compared with the traditional naming entity recognition model BiLSTM CRF, this model adds a CNN network with parallel processing characteristics with BiLSTM. The two extracted features are combined and sent to MHA, which extract the context information and local feature information, and mine the internal relationship between different features through MHA. This paper defines the entity tag BME and other nonentity tags *o* corresponding to the three types, through the named entity recognition experiment of Metro on-board equipment fault text data with this model and other common named entity recognition models, and the experiment results show that:

(1) The proposed named entity recognition model has conspicuous advantages in the three indexes of P, R, and $F1$ for the recognition results of all tags except entity tag b-measure and E-MEASURE, which is higher than HMM, CRF, BiLSTM, BiLSTM-CRF, and BiLSTM-CNN-CRF.

(2) The model in this paper has a good performance in the weighted average evaluation indexes Avg − P, Avg − R, and Avg − F_1, reaching 95.39%, 95.48%, and 95.37%, respectively. It is the highest value when compared with the other five named entity recognition models. Moreover, the strategy of parallel work of BiLSTM and CNN outperforms serial work.

Therefore, it can meet the performance requirements of high accuracy of subway on-board equipment fault text named entity recognition, provide theoretical basis and application value for subway on-board equipment fault named entity recognition, and establish a good foundation for the subsequent establishment of subway on-board knowledge map and the subway on-board knowledge base.

Data Availability

The dataset can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors 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

[1] C.-H. Gao, *A Communication Based on Train Control system*, China Railway Publishing House, Beijing.

[2] J. Lei, B. Tang, X. Lu, K. Gao, M. Jiang, and H. Xu, “A comprehensive study of named entity recognition in Chinese clinical text,” *Journal of the American Medical Informatics Association*, vol. 21, no. 5, pp. 808–814, 2014.

[3] R. C. Wasserman, “Electronic medical records (EMRs), epidemiology, and epistemology: reflections on EMRs and future pediatric clinical research,” *Academic pediatrics*, vol. 11, no. 4, pp. 280–287, 2011.

[4] C.-Q. Zong, X. Rui, and J.-J. Zhang, *Text Data Mining*, Tsinghua University Press, Beijing, 2019.

[5] C.-G. Pan, “Research on Chinese named entity recognition based on rule and statistics,” *Information Science*, vol. 30, no. 5, p. 6, 2012.

[6] J.-H. Zheng, X. Li, and H.-Y. Tan, “Research on Chinese name recognition method based on corpus,” *Chinese Journal of information*, vol. 14, no. 1, pp. 7–12, 2000.

[7] D. Bikel, “Nymble: a high-performance learning name-finder,” in *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pp. 194–201, ACM, Washington, USA, March 1997.

[8] D. M. Bikel, R. Schwartz, and R. M. Weischedel, “An algorithm that learns what’s in a name,” *Machine Learning*, vol. 34, no. 1/3, pp. 211–231, 1999.

[9] Y.-J. Zhang, Z.-T. Xu, and X.-Y. Xue, “Maximum entropy Chinese named entity recognition model based on multi feature fusion,” *J], Computer research and development*, vol. 45, no. 6, 2008.

[10] A. McCallum and W. Li, “Early results for named entity recognition with conditional random fields, feature induction and web—enhanced lexicons [C/proceedings of the seventh conference on natural language learning at HLT — NAACL. Stroudsburg, USA,” *ACLPPinforma*, no. 4, pp. 188–191, 2003.

[11] Z. Y. Ji, D. Y. Kong, L. W. D. W, and Y. J. Sang, “Research on named entity recognition based on deep learning [J/OL],” *Computer integrated manufacturing system*.

[12] K. Chen, *Research on Fault Diagnosis Method of Urban Rail Transit CBTC System Based on Text Mining*, Beijing Jiaotong University, Beijing china, 2020.

[13] P. Liu, Y.-P. Guo, F.-L. Wang, and G.-H. Li, “Chinese Named Entity Recognition: The State of the art,” *Neurocomputing*, p. 473, 2022.

[14] R. Collobert, J. Weston, L. Bottou, L. Karlen, M. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[15] Y. L. Miao, W.-F. Cheng, Y.-C. Ji, S. Zhang, and Y. L. Kong, “Aspect-based sentiment analysis in Chinese based on mobile reviews for BiLSTM-CRF,” *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 7, pp. 1–11, 2021.

[16] X. Chen, X. Qiu, C. Zhu, P. Liu, and X. J. Huang, “Long short-term memory neural networks for Chinese word segmentation,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1197–1206, Beijing china, Oct2015.

[17] Q.-H. Zhou and X.-L. Li, “Research on fault short text classification method of railway signal equipment based on MCNN,” *Journal of Railway Science and Engineering*, vol. 2, no. 16, pp. 2859–2865.

[18] N. Ye, X. Qin, L. Dong, X. Zhang, and K. Sun, “Chinese named entity recognition based on character-word vector fusion,” *Wireless Communications and Mobile Computing*, vol. 2020, no. 3, pp. 1–7, 2020.

[19] Y. Kim, *Convolutional Neural Networks for Sentence Classification*, https://arxiv.org/abs/1408.5882, 2014.

[20] J. Gao, Z.-P. Zhang, P. Cao, W. Huang, and F.-F. Li, “Citation entity recognition method using multi-feature semantic fusion based on deep learning,” *Concurrency and Computation: Practice and Experience*, vol. 34, no. 6, 2021.

[21] X. Ma and E. Hovy, “End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

[22] M. Ayifu, S. Wushouer, and M. Palidan, “Multilingual named entity recognition based on the BiGRU-CNN-CRF hybrid model,” *International Journal of Information and Communication Technology*, vol. 15, no. 3, p. 223, 2019.

[23] X.-L. Liu, M.-Q. Zhang, Q. Gu, Y.-Z. Ren, D.-B. He, and W.-L. Gao, “Named entity recognition of fresh egg supply chain based on BERT-CRF model,” *Journal of agricultural machinery*, vol. 52, no. S1, pp. 519–525, 2021.

[24] L.-B. Yang, *Research and Application of Key Technologies of Railway Accident Fault Text Big Data Analysis*, China Academy of Railway Sciences, Beijing, 2018.

[25] J. Yang, Y. Zhang, L. Li, and X. Li, “YEDDA: A Lightweight Collaborative Text Span Annotation Tool,” in *Proceedings of ACL 2018, System Demonstrations*, 2018.

[26] K. Dheeraj and T. Ramakrishnudu, “Negative emotions detection on online mental-health related patients texts using the deep learning with MHA-BCNN model,” *Expert Systems with Applications*, vol. 182, Article ID 115265, 2021.

[27] C.-F. Li and K. Ma, “Entity recognition of Chinese medical text based on multi-head self-attention combined with BILSTM-CRF,” *Mathematical Biosciences and Engineering*, vol. 19, no. 3, pp. p2206–2218, 2022.

[28] T. P. Adewumi, F. Liwicki, and M. Liwicki, “Word2Vec: Optimal Hyper-Parameters and Their Impact on NLP Downstream Tasks,” *Open Computer Scienc*, vol. 12, pp. 134–141, 2020.

[29] A. Graves, *Supervised Sequence Labelling*, Springer Berlin Heidelberg, Berlin, Germany, 2012.

[30] R. Sun, “Optimization for Deep Learning: Theory and algorithms,” 2019, https://arxiv.org/abs/1912.08957.