applied sciences

MDPI

Article

Chinese Fine-Grained Named Entity Recognition Based on BILTAR and GlobalPointer Modules

Weijun Li [1,2,*], Jintong Liu [1], Yuxiao Gao [1], Xinyong Zhang [1] and Jianlai Gu [1]

[1] School of Computer Science and Engineering, North Minzu University, Yinchuan 750021, China; 15632418501@163.com (J.L.)

[2] State Ethnic Affairs Commission Key Laboratory of Graphic Image Intelligent Processing, North Minzu University, Yinchuan 750021, China

* Correspondence: lwj@nmu.edu.cn

Abstract: The task of fine-grained named entity recognition is to locate entities in text and classify them into predefined fine-grained categories. At present, Chinese fine-grained NER only uses the pretrained language model to encode the characters in the sentence and lacks the ability to extract the deep semantic, sequence, and position information. The sequence annotation method is character-based and lacks the processing of entity boundaries. Fine-grained entity categories have a high degree of similarity, which makes it difficult to distinguish similar categories. To solve the above problems, this paper constructs the BILTAR deep semantic extraction module and adds the GlobalPointer module to improve the accuracy of Chinese fine-grained named entity recognition. The BILTAR module is used to extract deep semantic features from the coding information of pretrained language models and use higher-quality features to improve the model performance. In the GlobalPointer module, the model first adds the rotation position encoding information to the feature vector, using the position information to achieve data enhancement. Finally, the model considers all possible entity boundaries through the GlobalPointer module and calculates the scores for all possible entity boundaries in each category. In this paper, all possible entity boundaries in the text are considered by the above method, and the accuracy of entity recognition is improved. In this paper, the corresponding experiments were carried out on CLUENER 2020 and the micro Chinese fine-grained NER dataset, and the $F1$ scores of the model in this paper reached 80.848% and 75.751%, respectively. In ablation experiments, the proposed method outperforms the most advanced baseline model and improves the performance of the basic model.

Keywords: named entity recognition; natural language processing; deep neural networks; feature extraction; knowledge graph

check for updates

Citation: Li, W.; Liu, J.; Gao, Y.; Zhang, X.; Gu, J. Chinese Fine-Grained Named Entity Recognition Based on BILTAR and GlobalPointer Modules. *Appl. Sci.* **2023**, *13*, 12845. https://doi.org/10.3390/app132312845

Academic Editor: Douglas O'Shaughnessy

Received: 23 September 2023
Revised: 13 November 2023
Accepted: 27 November 2023
Published: 30 November 2023

1. Introduction

Named entity recognition (NER) is a critical task in natural language processing and an essential component in various NLP technologies, such as information extraction, information retrieval [1], machine translation, and question-answering systems [2]. The task of NER is to locate entities in text and predict their categories, providing downstream tasks with rich entity information. There have been many research achievements in coarse-grained entity recognition, and the $F1$ values of various models have far-exceeded 90%. However, in the real world, it is not enough to use only coarse-grained category information, and in most cases, fine-grained categories are needed to obtain deeper semantic information from the text. For example, sometimes it may be necessary to predict whether an organization belongs to the government or a company. However, fine-grained categories have a downside. Fine-grained categories often have similar categories that are difficult to distinguish, which presents a challenge in predicting fine-grained entity types. For example, when the coarse-grained category "organization" is split into "government" and "company", the two categories are not as easy to distinguish as "organization" and "person name". Therefore, fine-grained entity recognition is more challenging and valuable for research. The use of simple neural networks such as BERT-CRF [3] may not extract enough semantic information to distinguish between similar fine-grained categories. Entity recognition models need a more powerful feature extraction layer to improve the semantic extraction ability.

In the early days, named entity recognition methods often used the English language, and there was little research on Chinese entity recognition. With the widespread application of Chinese named entity recognition (CNER), many studies have been carried out on NER in Chinese, and Chinese NER faces two challenges. The first challenge is that classical sequence labeling methods are based on character-level tagging. The method marks the characters at the specific positions of the entity. One example is the "B-I-O" encoding method, which represents "B" as the first character of the entity, "I" as the internal character of the entity, and "O" as the non-entity character. Due to the problem of vanishing gradients over long distances in RNN networks, many models are unable to extract long-distance semantic information and mainly rely on local context for predictions. The second challenge is that in Chinese, there is a lack of boundary information for each character. There are no separators between Chinese characters. But in English, each word is separated by a space. In general, sequential labeling methods do not carry out special labeling and processing of entity boundaries. Chinese entity boundaries are affected by their contextual features and semantic information, resulting in a certain degree of ambiguity. This ambiguity reduces the predictive effectiveness of entity boundaries. At present, most of the classical entity recognition models use CRF to predict entity labels. Conditional random fields (CRF) is a conditional probability distribution model of another set of output sequences given a set of input sequences, which is often used in problems such as annotation and entity recognition. However, this method belongs to the sequence labeling method, and the research based on this method generally conducts corresponding labeling based on characters and lacks the processing of entity boundaries. Therefore, in order to solve the above problems, the GlobalPointer module came into being. The module uses the upper triangular matrix to consider all possible entity boundaries and eliminate the fuzziness of entity boundaries. The GlobalPointer module takes a multi-step matrix product of two eigenvectors and then performs a dimensional transformation of the product result to obtain an entity boundary score matrix. The entity boundary score matrix contains the confidence degree of each boundary for each category, and the module obtains the corresponding category of each entity boundary through the confidence degree, thus identifying entities and entity categories.

To address these issues, this paper proposes a Chinese fine-grained entity recognition model, Bert-ATT-BILTAR-GlobalPointer. In this model, a multi-position attention mechanism is added to denoise the data. The model in this paper extracts deep semantic information from the text to compensate for the shortcomings of BERT performance by the BILTAR module. The model adds rotation position encoding in the GlobalPointer module to increase the position information, and the GlobalPointer module of the model uses an upper triangular matrix to calculate the scores of all possible entity boundaries in each category. In comparative experiments, the performance of the proposed model is higher than that of recent baseline models. In ablation experiments, it is demonstrated that each submodule of the proposed model can improve the model's performance among various module combinations.

The innovation points of this paper are described as follows: (1) Replacing the CRF in the classical entity recognition model with the GlobalPointer module and adding the rotation position encoding (RoPE); (2) The multi-head self-attention mechanism (ATT) with multiple positions; (3) The application of a positive–negative direction module (PN) and time-step module (TIME) in BILSTM; (4) We added the corresponding innovation modules to different module combinations, so as to carry out ablation experiments on each innovation point of the model. The results of the ablation experiments demonstrate the effectiveness of all innovations in this paper. There are 18 groups of ablation experiments in the ablation experiment chapter.

In real life, text content generally covers a wide range. Each piece of text may contain several different types of content. In order to improve the recognition ability of the entity recognition model for multiple fine-grained categories, the CLUENER2020 dataset was selected as the first experimental dataset. The CLUENER2020 dataset contains 10 entity categories, covering the vast majority of real-life text content. On the Internet, the general communication information is network language. Such text has the characteristics of being short, random, fuzzy, having a complex context, and so on. In order to investigate the ability of processing network language, we chose a microblog dataset as the second experimental dataset. The Weibo dataset contains four categories and marks whether the entity is explicitly or generally referred to.

This paper is divided into five chapters, namely, introduction, related work, model, experiment, conclusion, and future work. Structurally, this paper first introduces the current status and significance of entity recognition research in the introduction and related work sections. Then, we describe the specific details of the model in the model section. The effectiveness of the model is analyzed through the experimental section. Finally, the main work of this paper and future prospects of entity recognition research are summarized in the conclusion and future work section.

2. Related Work

Generally, there are two types of methods for entity recognition: sequence-based methods and span-based methods. The sequence-based approach defines entity recognition as a sequence labeling problem. In this method, each character is labeled with one or more labels for entity recognition. Classical sequence labeling methods include the neural network model [4] (NNS), the hidden Markov model [5] (HMM) and conditional random fields [6] (CRF). With the wide application of deep learning in natural language processing, deep learning techniques have been applied to entity recognition tasks. The classic modules of deep learning include the convolutional neural network [7] (CNN), fully connected layer [8] (Linear), recurrent neural network [9] (RNN), long short-term memory network [10] (LSTM), and gated recurrent neural network [11] (GRU). For example, Huang et al. [12] proposed the BILSTM-CRF model, which combines bidirectional LSTM and CRF to achieve entity recognition. The CNN lacks the extraction of sequence features, so Chiu et al. [13] combined the cyclic neural network LSTM and the convolutional neural network CNN to construct a LSTM-CNNs neural network structure. The model uses the CNN to extract local features from word-level and character-level features and uses BILSTM to annotate each element. Ma et al. [14] proposed the BILSTMCNNs-CRF model, which combines a CNN, bidirectional LSTM, and CRF to construct a complete end-to-end neural network model for various sequence labeling tasks. For Chinese NER, Zhang et al. [15] proposed the Lattice-LSTM model, which incorporates Chinese word features into the model to enhance entity information. However, the Lattice-LSTM model is unable to determine the length and number of Chinese words, which may lead to problems such as difficult batch training and poor generalization ability. To solve these problems, Ma et al. [16] proposed the Soft-Lexicon model based on Lattice-LSTM. This new model improves the running speed and generalization ability of the Lattice-LSTM model. Zhu et al. [17] proposed a convolutional attention network (CAN) model. The model adds an attention mechanism on CNNS and a global attention mechanism on GRNN to extract semantic and contextual information between characters. Gui et al. [18] proposed a rethinking mechanism and integrated this mechanism into LR-CNN (lexiconrethinkingCNN). In this study, encoded information of binary and ternary words was added to achieve data enhancement.

The span-based approach defines entity recognition as the classification problem of entity boundaries. Firstly, all possible entity boundaries are listed, and the semantic features of entity boundaries are extracted by the feature extraction layer. Finally, entities and entity categories are predicted by the entity prediction module. Compared with sequence-based methods, span-based methods do not require labeling of individual elements, making them suitable for nested entity recognition tasks. For example, Sohrab et al. [19] proposed an exhaustive model that enumerates all possible spans and uses LSTM to predict the categories of generated spans. Xu and Jiang [20] proposed a local detection method that encodes the span and its context into a fixed-size feature vector and uses the feature vector to predict the category of the corresponding entity. Xia et al. [21] separated entity span extraction and entity type prediction. The study used the overall detector to obtain the possible entity span and then used the classifier to predict the class of the corresponding entity. Li et al. [22] proposed a unified NER framework based on word pair relationships. The model constructs a table of all possible entity spans and predicts the categories of these entity spans.

A span-based approach can also be implemented by adding a network of pointers to the network model. Pointer networks can generate output sequences of variable lengths. They break the limitation of fixed sequence length in the general sequence-to-sequence model. The entity recognition model based on a pointer network takes text as input data, first predicts the beginning and end boundaries of the entity to obtain the entity boundary, and then calculates the representation of the entity boundary to predict the entity type corresponding to the entity boundary. Zhai et al. [23] added a pointer network on the model to implement sequence partitioning and annotation. Li et al. [24] used a GRU neural network on the feature extraction layer to extract the semantic features of the data and used a pointer network to eliminate the fuzziness of the entity boundary. The pointer network has an obvious effect in dealing with nested entities, but the pointer network generally transforms the multi-entity extraction into multiple binary classification problems, so the model may converge too slowly when the sequence is too long.

In the traditional entity recognition or reading comprehension pointer network, the head and tail boundaries of the entity are extracted with two different modules, which brings about inconsistencies between training and prediction. To solve the problem of the pointer network, Su et al. [25] proposed the GlobalPointer module. The GlobalPointer module enumerates all entity boundaries and computes the confidence levels of the entity boundaries on each class through the upper trigonometric matrix. This confidence level is used to predict entities and entity classes. GlobalPointer is designed to address inconsistencies in common pointer networks. It takes the beginning and end of the physical boundary as a whole, so it is more “global”. Zhang et al. [26] used the GlobalPointer module to predict entities and add a RoBERT pretrained language model on the feature extraction layer to extract character-level features. Sun et al. [27] proposed the nezhan-cnn-globalpointer architecture, which adds an annotation semantic network and uses multi-granularity context semantic information to improve the semantic extraction ability of the entity recognition model.

Through the analysis of the two main methods in entity recognition, it is concluded that the span-based method can avoid the element labeling error of the sequence method, and the powerful feature extraction layer is suitable for the fine-grained entity recognition task. Therefore, this paper adopts the GlobalPointer module to implement the span-based method and adds the rotation position encoding to increase the feature’s positional information in the GlobalPointer module. This paper proposes the BILTAR module to extract deep semantic information from data. In this paper, the attention mechanism is added in several positions of the model to reduce the noise of the data and improve the data quality. Finally, this paper proposes the Chinese fine-grained entity recognition model Bert-ATT-BILTAR-GlobalPointer.

3. Model

This paper proposes the Chinese fine-grained entity recognition model Bert-ATT-BILTAR-GlobalPointer. The model first uses the BERT layer to encode the text into word vectors. BILTAR is used to extract the deep semantic features of the text, and the position information is added to the features by rotation position encoding. Finally, the model uses the GlobalPointer module to compute each possible entity boundary and extract all entities. Figure 1 shows the overall framework of the model proposed in this paper. In Figure 1, circular blocks represent vectors, rectangles represent input text, and rounded rectangles represent modules. In Figure 1, "北京大学" is a Chinese text that represents the input instance of the model.

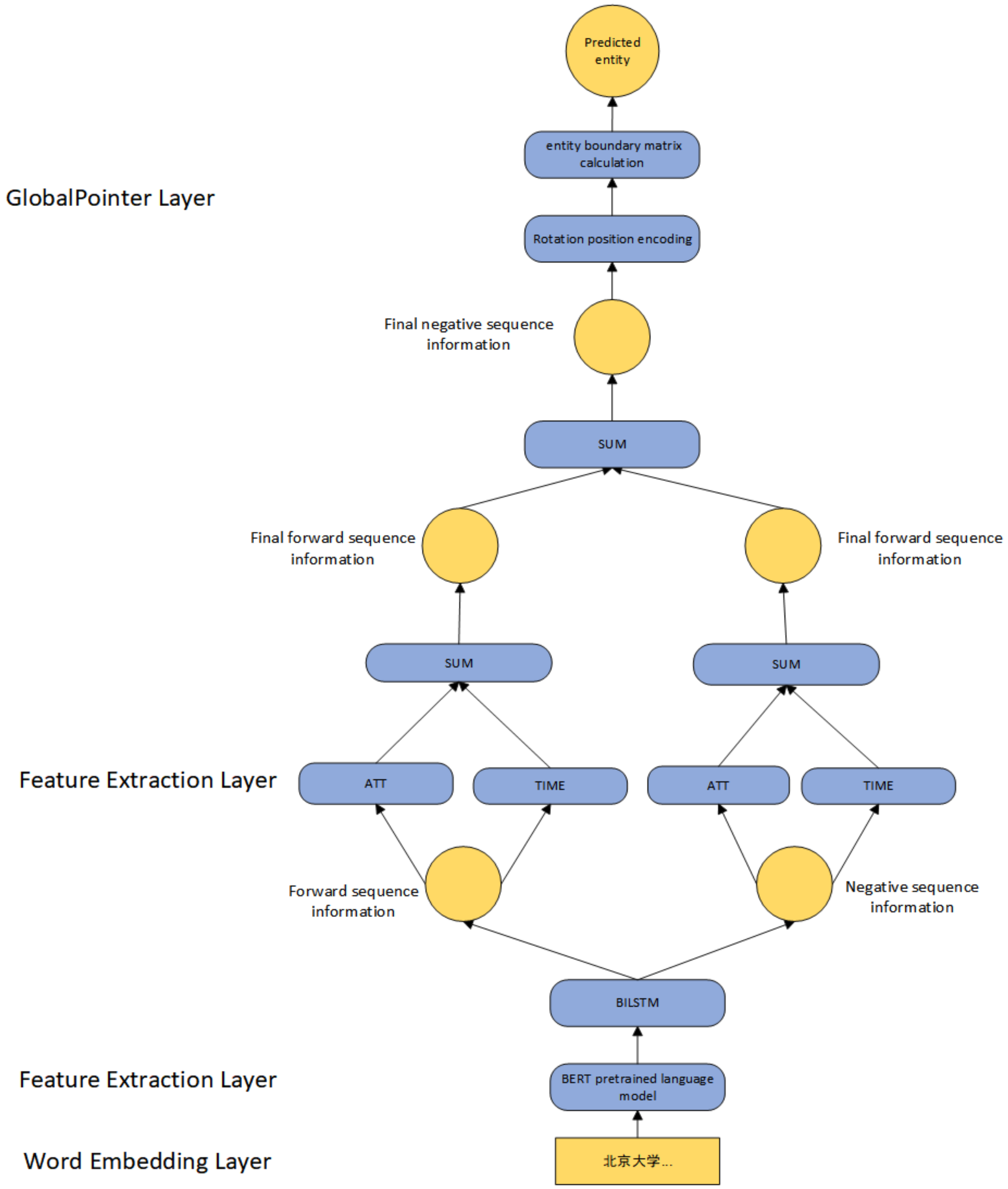
Predicted entity
entity boundary matrix calculation
GlobalPointer Layer
Rotation position encoding
Final negative sequence information
SUM
Final forward sequence information
Final forward sequence information
SUM
SUM
Feature Extraction Layer
ATT
TIME
ATT
TIME
Forward sequence information
Negative sequence information
BILSTM
Feature Extraction Layer
BERT pretrained language model
Word Embedding Layer
北京大学...

Figure 1. The overall framework of the model Bert-ATT-BILTAR-GlobalPointer in this paper.

3.1. Word Embedding Layer

In order to encode text into word vectors and extract certain semantic information, the pretrained language model BERT is used as the word embedding layer to extract semantic information from the text.

The pretrained language model BERT [28] is composed of a word embedding layer and multiple transformer modules. Its word embedding layer uses a variety of text information as input to calculate the word vector representation of the text. Its multiple stacked transformer modules extract the deep semantic features of the text vectors. The BERT model adopts two pretraining methods, the Masked Language Model (MLM) and Next Sentence Prediction (NSP). The MLM tasks tend to extract markup-level representations and implement gestalt tasks by masking random positions in the sequence. This task desensitizes the BERT model to specific masking locations in order to obtain semantic information for each location in the sequence.

The NSP task first selects a sentence as the previous sentence, selects the next sentence as the next sentence with a 50% probability, and, finally, determines whether the two sentences are upper and lower sentences. This task is able to improve the BERT model's perception of the relationships between sentences. Therefore, the BERT model is used as the word embedding layer of the model, and the BERT model provides more semantic information for the feature-extraction module, BILTAR.

3.2. Multi-Head Self-Attention Mechanism

The entity recognition dataset may contain some useless information or noise. BERT models may struggle to represent completely clean semantic information. In order to solve the above problems, an attention mechanism [29] is used to highlight the important information of feature vectors and reduce the noise, thus improving the performance of the model. In order to extract semantic information from more dimensions, we use a multi-head self-attention mechanism to denoise data at multiple locations in the model. Figure 2 shows the structural diagram of the multi-head self-attention mechanism. In Figure 2, the rounded rectangle represents a module, and the circle represents a vector.

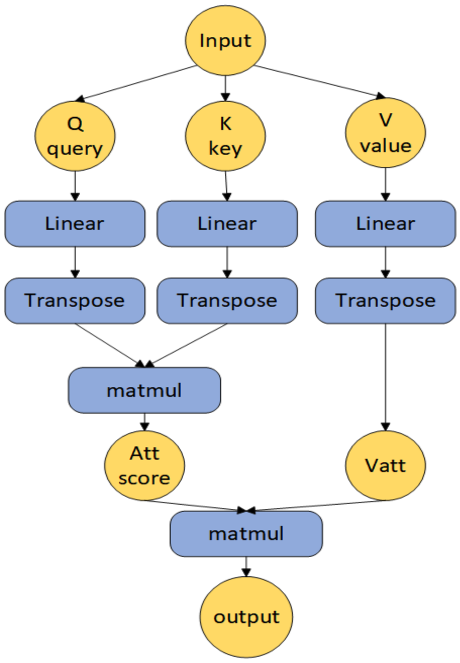

Figure 2. Structural diagram of the multi-head self-attention mechanism.

Taking the eight-head self-attention mechanism module as an example, this paper introduces the working principle of the multi-head self-attention mechanism. $Input$ is the input vector of the attention module, and the dimension of it is (batch_size, seq_size, word_embedding_size). The input is passed through three different fully connected layers to obtain the encoded data Q_{all}, K_{all}, and V_{all}. The calculation formulas for Q_{all}, K_{all}, and V_{all} are shown in Equations (1)–(3), respectively. The dimensions of $W_{q,all}$, $W_{k,all}$, and $W_{v,all}$ are (word_embedding_size, word_embedding_size); batch_size is the batch size for model training and validation; seq_size is the sequenced length of the text; and the value called word_embedding_size is the dimension of the word vector.

$$Q_{all} = W_{q,all} \times Input + bias_{q,all} \tag{1}$$

$$K_{all} = W_{k,all} \times Input + bias_{k,all} \tag{2}$$

$$V_{all} = W_{v,all} \times Input + bias_{v,all} \tag{3}$$

The keys K_{all}, values V_{all}, and query Q_{all} vectors are split into 8 parts along the last dimension, and the last dimension d_last is split into two dimensions (8, d_last/8). The second and third dimensions of the vectors are swapped, resulting in the query, key, and value vectors Q_{att}, K_{att}, and V_{att} of the 8-head self-attention mechanism. Compared with the single-head attention mechanism, the multi-head attention mechanism extracts important information from more dimensions from the same data, resulting in higher-quality data. Formula (4) shows the dimension changes of the key, value, and query vectors. *batch_size* is the batch size when the model is trained, *seq_size* is the length of each text sequence, and *word_embedding_size* is the dimension of the word vector.

The key and value vectors are multiplied to obtain the attention score, and the attention score is divided by the length of the last dimension d_last/8 in the key vector to prevent the attention score from being too large.

$$\begin{aligned}
&Q_{\mathrm{all}}(batch\_size, seq\_size, word\_embedding\_size)\\
&\to Q_{att}(batch\_size, 8, seq\_size, word\_embedding\_size/8)\\
&K_{\mathrm{all}}(batch\_size, seq\_size, word\_embedding\_size)\\
&\to K_{att}(batch\_size, 8, seq\_size, word\_embedding\_size/8)\\
&V_{\mathrm{all}}(batch\_size, seq\_size, word\_embedding\_size)\\
&\to V_{att}(batch\_size, 8, seq\_size, word\_embedding\_size/8)
\end{aligned} \tag{4}$$

The attention score and the value vector are multiplied to obtain the output vector $Output_{first}$. The model combines the last two dimensions (8, d_last/8) of the output vector $Output_{first}$ into one dimension d_last to obtain the intermediate output vector $Output_{middle}$. The calculation formula for this step is shown in Formula (5).

$$\begin{aligned}
&Att\_score = (Q_{att} * K_{att})/(d\_last/8)\\
&Output_{first} = Att\_score * V_{all}\\
&Output_{first}(batch\_size, 8, seq\_size, word\_embedding\_size/8)\\
&\to Output_{middle}(batch\_size, seq\_size, word\_embedding\_size)
\end{aligned} \tag{5}$$

The output vector $Output_{middle}$ is passed through a fully connected layer and a dropout layer. The dropout layer is used to prevent the model from over-fitting. The output

$$\begin{aligned}
&Output_{last} = Laynorm(Dropout(Output_{middle}\\
&*W_{last} + bias_{last}) + Input)
\end{aligned} \tag{6}$$

is summed with the original input data to retain the necessary information of the original input vector. The layer-normalization layer Laynorm is used to adjust the value of the output vector to a suitable range. The dimension of W_{last} is (word_embedding_size, word_embedding_size). The calculation formula of the final output feature is shown in Equation (6).

3.3. BILTAR Module

The BILTAR module is composed of a BILSTM, TIME layer, attention mechanism, PN forward–backward module, and residual structure. The model first puts the feature vector into the BILSTM to obtain the forward and backward temporal information. Then, the PN module separates the forward and backward temporal information, and each piece of temporal information is processed separately through its own attention module and TIME layer to obtain two feature vectors, ATT and TIME. These two feature vectors, ATT and TIME, are summed to obtain the deep semantic feature vectors L_post and L_neg for the forward and backward sequences, respectively; and these two vectors, L_post and L_neg, are summed to obtain the final deep semantic information.

3.3.1. BILSTM Module

The BILTAR module adjusts the dimension of the output vector to make the feature vector size suitable for the GlobalPointer module. The method of changing the dimension is as follows:

Set the output vector dimension of the LSTM to change the last dimension of the feature vector to ent_type_size*inner_dim*2 and obtain a feature vector with a dimension of (batch_size, seq_len, ent_type_size*inner_dim*2).

Split the last dimension of the eigenvector to obtain a feature vector with a dimension of (batch_size, seq_len, ent_type_size, inner_dim*2).

In the last dimension of the feature vector, take the first inner_dim values as the feature vector q, and the last inner_dim values as the feature vector k. The dimensions of q and k are (batch_size, seq_len, ent_type_size, inner_dim). The feature vectors q and k are the inputs of the GlobalPointer module.

The BILSTM is used to extract sequence information from data, the TIME layer is used to encode features at each position of the sequence, the attention mechanism is used to denoise the data, the PN forward–backward module is used to refine the forward and backward sequence information, and the residual structure is used to fuse different features. The following describes each structure block of the BILTAR module in detail.

The BILSTM is a classic neural network structure that processes sequence data. Text data are also sequence data, so the entity recognition task is very suitable for using BILSTM to obtain the sequence information of the text. RNN has a problem of long-term dependency and struggles to remember long sequence information. Therefore, the LSTM structure is used to increase the model's memory capacity for long-term sequences, so that the model can extract more long-time sequence information. There is an inevitable connection between the entity and the entity context, so BILSTM is used to obtain the sequence information of the entity context.

At the same time, the forward and backward sequence information may not be the same. Therefore, LSTM is upgraded to BILSTM in order to obtain the forward and backward sequence information, so as to obtain the semantic information of the sequence from more dimensions. Figure 3 shows the structure diagram of the bidirectional recurrent neural network BILSTM.

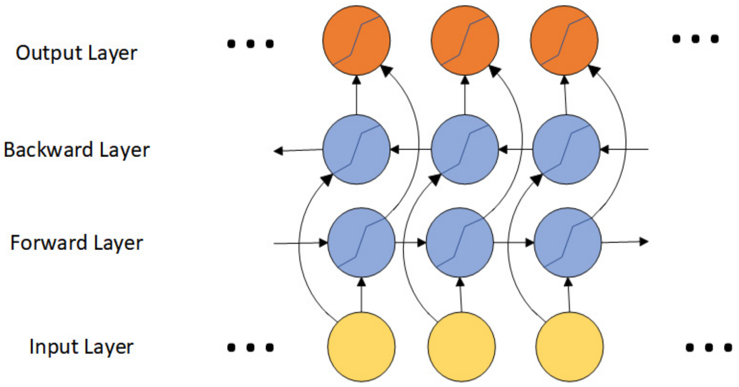

Figure 3. Bidirectional recurrent neural network BILSTM.

In this paper's model, the BILSTM is a one-layer structure. Taking the internal structure of a one-layer BILSTM as an example, this paper introduces the implementation principle of BILSTM. The single-layer BILSTM consists of two different LSTMs [30]. One layer of LSTMs is responsible for the forward propagation of the data to obtain the forward information of the sequence. Another layer of LSTM is responsible for the backward propagation of the data to obtain the backward information of the sequence. The forward and backward information are concatenated to obtain the output vector of the BILSTM.

The LSTM unit consists of an input gate i_t, a forget gate f_t, an output gate o_t, and a candidate memory cell g_t. Based on the input data x_t and the previous hidden state h_{t-1}, the values of the three gates and the candidate memory cell g_t are calculated. The forget gate f_t and the candidate memory cell g_t are multiplied to delete some useless memory information. The input gate i_t and the candidate memory cell g_t are multiplied to obtain the necessary memory information. The sum of the two inner products is the memory cell c_t at this time step. After passing through the tanh activation function, the memory cell c_t is multiplied by the value of the output gate to obtain the hidden state h_t at this time step. The calculation formula for the LSTM unit is shown in Equation (7).

$$\begin{aligned}
i_t &= \sigma(W_{ii} * x_t + b_{ii} + W_{hi} * h_{(t-1)} + b_{hi}) \\
f_t &= \sigma(W_{if} * x_t + b_{if} + W_{hf} * h_{(t-1)} + b_{hf}) \\
g_t &= \tanh(W_{ig} * x_t + b_{ig} + W_{hg} * h_{(t-1)} + b_{ho}) \\
o_t &= \sigma(W_{io} * x_t + b_{io} + W_{ho} * h_{(t-1)} + b_{ho}) \\
c_t &= f_t * c_{(t-1)} + i_t * g_t \\
h_t &= o_t * \tanh(c_t)
\end{aligned} \tag{7}$$

3.3.2. TIME Module

In the TIME layer, each data point in the sequence is processed separately without data interaction between different time points. This layer requires a sequence as input, so it is generally added after the LSTM network. This layer separately calculates the data for each time point on the LSTM output sequence. Therefore, this paper places this layer behind BILSTM to process each data point, that is, each word. The TIME module implements separate processing of the data at each point in time.

In the TIME layer of this paper's model, the internal implementation network is an encoder consisting of four fully connected layers. The first two fully connected layers serve as encoders and reduce the feature dimension from lstm_hidden_size to the specified value of 10. The last two fully connected layers serve as decoders and increase the feature dimension from 10 back to lstm_hidden_size. Therefore, the TIME layer can encode each data point and extract deep semantic information. lstm_hidden_size is the output dimension of the BILSTM, that is, ent_type_size*inner_dim*2. The process of forward propagation at the TIME layer is described as follows. The feature is firstly reduced by the first encoder to obtain the high-density encoding feature. The encoding feature is then raised by the second encoder to obtain the feature vector of the same size as the original feature. The second encoder decodes the encoding feature. The TIME layer finally outputs the encoded feature vector.

In the following chapters, this paper conducts ablation experiments on fully connected encoders with different numbers of layers and finds that the encoder has the best performance under four fully connected layers. The structure of TIME is shown in Figure 4.

3.3.3. PN Module

The PN module separates the forward and backward temporal information in the output vector of the BILSTM. Each piece of temporal information passes through its own attention module and TIME layer to obtain two eigenvectors, ATT and Time. ATT and Time have stronger temporal information and time unit semantic information, respectively. These two feature vectors are fused through residual summation to obtain forward and backward information, and the forward and backward information is similarly fused through residual summation to obtain the final deep semantic vector.

The later comparative experiments show that the fusion method of summing the forward and backward semantic information is more effective than the fusion method of concatenation.

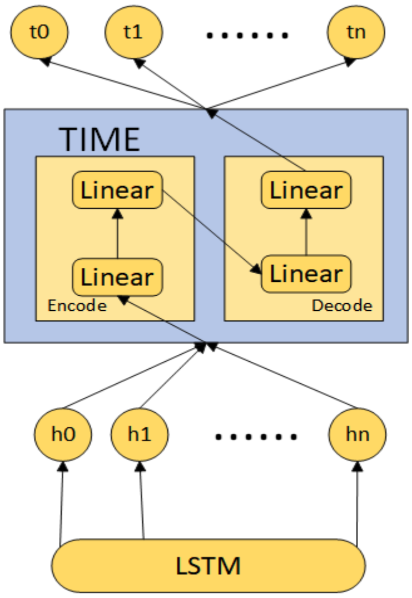

Figure 4. TIME structure diagram.

3.3.4. GlobalPointer

The GlobalPointer module is a nested entity recognition model proposed by Su et al. The module consists of two parts: rotation position encoding and entity boundary score calculation. Su. [31] pointed out that experimental results show that adding rotation position encoding to the GlobalPointer module can significantly improve the performance of the model. Ablation experiments in later chapters demonstrate the effectiveness of rotation position encoding and solid boundary calculations.

Firstly, the GlobalPointer module first uses rotation position encoding (RoPE) to extract the position information for the sequence, and then injects the position information into the feature vector. If the model is not sensitive enough to entity boundaries, the model might predict the start boundary of the previous entity and the end boundary of the next entity as one entity boundary. To address these issues, the GlobalPointer module uses entity boundary score calculations to make the model more sensitive to entity boundaries, which prevents this from happening.

The other part of the GlobalPointer module is entity boundary score calculations. The BILTAR module extracts deep semantic feature vectors q and k. The model adds rotation position encoding information to the two feature vectors, and finally obtains the input data of the entity boundary score calculations part. The inner product of vectors q and k is the score matrix of the entity boundaries corresponding to each class. Since q and k have a category dimension, each value on this dimension corresponds to a category, and the resulting score vector has a dimension of (batch_size, ent_type_size, seq_len, seq_len). The model excludes invalid entity boundaries for each categorical matrix. The final entity boundary score matrixes contain the scores for all possible entity boundaries in each category.

The entity boundary represented by the element (a, b, c, d) of the score matrix is the entity boundary T from index c to index d on the text sequence in training batch a, and the element value is the score of the entity boundary T in category (b + 1). Figure 5 shows the corresponding matrix for the score vector for the example text “北京大学”.

Figure 5 shows the scores of the “北京大学” sequence in the person name, place name and organization categories. In Figure 5, “北京大学” is a Chinese text, and every two successive characters build a Chinese entity boundary in the matrix. Each upper triangular matrix represents the score of each subsequence in the corresponding category. For example, in the first figure, the scores of all subsequences in the person name category are 0, indicating that there is no person name entity in the predicted result for this sequence.

In the second figure, the subsequence (0, 1) has a score of 1 in the place name category, indicating that the entity fragment “北京” is a place name entity, and so on.

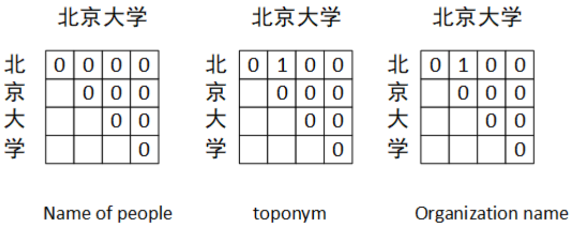

Figure 5. The core matrix of GlobalPointer module.

4. Experiment

4.1. Datasets

This paper uses two fine-grained Chinese entity recognition datasets, namely, the CLUENER2020 and Weibo datasets.

CLUENER2020 [32] is a Chinese fine-grained named entity recognition dataset based on the THUCTC open-source text classification dataset from Tsinghua University. CLUENER2020 contains 10 tag types, including organization, name, address, company, government, book title, game, movie, organizational structure, and attractions. Compared with other available Chinese datasets, CLUENER2020 is annotated with more categories and details, making it more challenging and difficult. Table 1 lists the information statistics of the CLUENER2020 dataset.

Table 1. Statistics of the datasets.

Dataset	Train	Dev	Test	Classes	Total Number
CLUENER 2020	10,748	1343	Null	10	12,091
Weibo	1350	270	270	8	1890

The Weibo dataset [33] is a Chinese fine-grained named entity recognition dataset from the social media field. This dataset consists of four entity types: personal names (PRE), place names (LOCs), organization names (ORG), and geopolitical entities (GEPs). Each type is further divided into two categories: explicit entities (NAM) and generic entities (NOM). The Weibo dataset can test the model’s ability to distinguish between explicit and generic entities. Table 1 lists the information statistics of the Weibo dataset.

4.2. Experimental Metrics

To evaluate the performance of the model and compare the effects of different models, this paper uses three metrics: *P* (precision), *R* (recall), and *F* (*F*1 score). Among them, *P* represents the ratio of correctly identified entities to all identified entities, *R* represents the ratio of correctly identified entities to all entities that should be identified, and *F* is a comprehensive evaluation index that combines *P* and *R*. The formulas for calculating the three metrics are shown in Equations (8)–(10).

$$P = TP/(TP + FP) \times 100\% \tag{8}$$

$$R = TF/(TP + FN) \times 100\% \tag{9}$$

$$F = 2 \times P \times R/(P + R) \times 100\% \tag{10}$$

In the above equations, *TP* represents the number of samples that are actually positive and predicted as positive, *FP* represents the number of samples that are actually negative but predicted as positive, and *FN* represents the number of samples that are actually positive but predicted as negative.

4.3. Parameter Settings

This paper uses Python 3.8.16 and PyTorch 1.12.0 as the configuration environment for the experiments. The BERT pretrained language model is used to generate word vectors. Table 2 shows the settings of the hyperparameters used in the experiments. The value d_classes represents the number of categories for every dataset.

Table 2. Setting of the hyperparameters used in the experiment.

Parameter	Value
BILSTM hidden size	128*d_classes
Number of BILSTM layers	1
Dropout rate	0.5
Optimizer	Adam
Learning rate	2×10^{-5}
Epoch	20 (CLUENER2020)/50 (Weibo)
Batch size	64
Bert output dimension	768
Learning rate adjustment strategy	CosineAnnealingWarmRestarts (CLUENER2020)/MultiStepLR (Weibo)
BERT	Bert-base-chinese

4.4. Comparative Experiment

4.4.1. Baseline Models

To prove the effectiveness of the model, comparison experiments were conducted to compare the proposed model with baseline models and classic entity recognition neural network models proposed in recent years. The comparison results are shown in Tables 3 and 4. Bold data indicates the highest value of the corresponding indicator in the table.

Table 3. Contrast experiments of CLUENER2020 dataset.

Compare Models	*P*%	*R*%	*F*1%
BERT-ATT-BILTAR-GLOBALPOINTER	**80.485**	**81.270**	**80.848**
BERT-GLOBALPOINTER	78.683	80.839	79.697
BERT-CRF	72.337	80.273	76.099
BERT-IDCNN-CRF	74.244	78.352	76.243
BERT-BILSTM-CRF	73.396	80.468	76.770
BERT-BILSTM-IDCNN-CRF	73.689	78.678	76.102

Table 4. Contrast experiments of Weibo dataset.

Compare Models	*P*%	*R*%	*F*1%
BERT-ATT-BILTAR-GLOBALPOINTER	**79.227**	**72.720**	**75.751**
BERT-GLOBALPOINTER	71.691	67.838	69.644
BERT-CRF	62.831	69.607	66.046
LLPA	57.440	67.010	61.860
MFT	63.720	65.030	64.380
BERT + para-lattice + CRF	68.920	69.740	70.570
AT-CBGP	74.110	68.700	71.190
BERT + BILSTM + CRF	55.701	62.254	58.796

BERT-CRF encodes the semantic features of the text with the BERT model and applies the CRF layer for sequence decoding to obtain the prediction results. BERT-GlobalPointer uses rotation position encoding (RoPE) on the output features of the BERT model to add position information and extracts entities by calculating the score of the entity boundary in the diagonal matrix. AT-CBGP [34] improves the robustness and generalization of the model by adding an adversarial neural network on the GlobalPointer module. LLPA [35] adds relative position encoding on the bidirectional Lattice-LSTM module. MFT [36] adds Chinese word root information and improves the structure of the transformer. BERT + paralattice + CRF [37] adds Chinese character vectors as features to enhance performance.

This paper proves the effectiveness of the proposed model on the CLUENER2020 dataset by comparing its performance with that of classic models in comparative experiments. The following models are classic entity recognition network structures that were reimplemented in this paper. BERT-CRF extracts the semantic information of the text with the BERT layer and learns the constraint information between entity tags with the CRF conditional random field. On the basis of the BERT-CRF module, three complex entity recognition models are created by adding the feature extraction layer IDCNN module, the BILSTM module, and the IDCNN-BILSTM module, respectively, to obtain four classic entity recognition models.

The experimental results of AT-CBGP, LLPA, MFT, and BERT + para-lattice + CRF are the experimental results of other papers, and the remaining six baseline models are the experimental results obtained by reproducing other papers' models. The Weibo dataset is divided into training, testing, and validation sets. The training method on the Weibo dataset is to verify the validation set after each round of training, and after 50 rounds of training, the model selects the model parameters with the highest performance on the validation set, tests the performance on the test set with the best model parameters, and the experimental results on the test set are the final results of the model. The CLUENER 2020 dataset is divided into training and validation sets. The training method on the CLUENER2020 dataset is to validate the validation set after each round of training, and after 20 rounds of training, the model selects the best performance on the validation set as the final result of the model.

4.4.2. Comparative Experiment

Tables 3 and 4 show the comparative experimental results of the proposed model on the CLUENER2020 dataset and the Weibo dataset. The analysis of the comparative experiments is as follows.

According to the comparison between BERT-ATT-BILTAR-GlobalPointer and BERT-GlobalPointer, the $F1$ value of the model increased by 1.151% on the CLUENER2020 dataset and increased by 6.107% on the Weibo dataset. BERT-ATT-BILTAR-GlobalPointer adds the BILTAR module and attention mechanism to BERT-GlobalPointer. The improvement in performance shows that the BITAR module can extract deeper semantic information, and the attention mechanism can highlight important information, thereby improving the performance of the model.

In the comparative experiments on the Weibo dataset, compared with the LLPA model, the $F1$ value of the proposed model was increased by 13.89%. Compared with the MFT model, the $F1$ value of the proposed model was increased by 11.37%. Compared with BERT + para-lattice + CRF, the $F1$ value of the proposed model was increased by 5.18%. Compared with AT-CBGP, the $F1$ value of the proposed model was increased by 4.56%. The experimental results prove that the combination of the GlobalPointer module and the deep semantic feature extraction layer BILTAR can achieve good performance of the model.

Comparative experiments on the two datasets show that the entity boundary score calculation method of GlobalPointer is significantly better than the sequence decoding method of CRF. The BILTAR feature extraction layer is significantly better than the BILSTM and BILSTM-IDCNN modules. The BILTAR module is superior to the IDCNN module in terms of both model parameter volume and performance. Splitting the forward and backward information separately in the BILSTM module can extract more sequence information, adding attention mechanisms at multiple positions can denoise the features, and adding the TIME module on the output of BILSTM can extract more semantic information based on each word.

4.5. Ablation Experiments

The BILTAR module of this study is divided into six ablation experiment modules. The six modules are the attention mechanism module ATT, the positive and negative direction module PN, the time module TIME, the rotation position encoding RoPE, the entity boundary score calculation of GlobalPointer, and the overall BILTAR module. This study adopted the method of controlling variables for the experimental design. In order to prove the effectiveness of the six modules, this study used uniform data processing methods, the same operating environment, and training parameter settings. The only difference between the models existed in the different structural compositions, and the common parts were consistent. The hyperparameters of the models are the same when the models are trained on the same dataset. A total of 18 module combinations were tested in the ablation experiment. Table 5 shows the ablation experiment results of the modules. Bold data indicates the highest value of the corresponding indicator in the table.

Table 5. Eighteen ablation experiments of the present model.

	CLUENER2020			**Weibo**		
Compare Models	***P*%**	***R*%**	***F*1%**	***P*%**	***R*%**	***F*1%**
BERT-ATT-BILTAR-GlobalPointer	**80.485**	81.270	**80.848**	**79.227**	72.720	**75.751**
BERT-BILTAR-GlobalPointer	80.203	79.611	79.845	70.034	60.886	65.019
BERT-GlobalPointer	78.683	80.839	79.697	71.691	67.838	69.644
BERT-ATT-GlobalPointer	76.746	81.047	78.801	76.533	67.004	71.116
BERT-BiLSTM-GlobalPointer	76.638	80.233	78.370	71.625	68.167	69.767
BERT-ATT-BiLSTM-ATT-GlobalPointer	78.689	81.778	80.162	67.894	70.224	68.882
BERT-ATT-BiLSTM-PNATT-GlobalPointer	79.858	79.775	79.773	75.468	68.006	71.451
BERT-BiLSTM-PNTIMER-GlobalPointer	75.394	81.580	78.337	72.701	70.223	71.306
BERT-BiLSTM-PN-GlobalPointer	75.602	81.716	78.515	69.242	68.885	69.012
BERT-BiLSTM-TIMER-GlobalPointer	76.715	79.956	78.274	70.780	64.475	67.352
BERT-BiLSTM-ATT-TIMER-GlobalPointer	78.752	80.751	79.731	72.603	**73.487**	72.749
BERT-ATT-BiLSTM-ATT-TIMER-GlobalPointer	79.608	**81.856**	80.683	76.927	67.653	71.939
BERT-BILTAR-CRF	73.064	80.175	76.455	62.272	67.156	64.622
BERT-CRF	72.337	80.273	76.099	65.446	70.098	67.692
BERT-ATT-BILTAR-CRF	74.182	79.785	76.882	62.962	66.666	64.761
BERT-ATT-CRF	74.133	67.653	71.939	62.831	69.607	66.046
BERT-ATT-BILTAR-GlobalPointer(without RoPE)	79.628	80.330	79.948	76.342	65.979	70.778
BERT-GlobalPointer(without RoPE)	77.421	80.850	79.237	66.719	68.568	67.306

This study conducted five ablation experiments on the encoder of the TIME module by changing the number of fully connected layers and found that the encoder composed of two fully connected layers and the decoder composed of two fully connected layers had the best performance as the encoder of the TIME module. Below are the introductions and analyses of each ablation experiment:

4.5.1. Ablation Experiments of Rotation Position Encoding RoPE

To prove the effectiveness of the rotation position encoding RoPE, this study removed the RoPE from the GlobalPointer module and conducted two comparative experiments.

In the comparison between BERT-ATT-BILTAR-GlobalPointer and BERT-ATT-BILTAR-GlobalPointer (without RoPE), the *F*1 value of the model decreased by 0.841% on the CLUENER2020 dataset, and the *F*1 value of the model decreased by 4.972% on the Weibo dataset.

In the comparison between BERT-GlobalPointer and BERT-GlobalPointer (without RoPE), the *F*1 value of the model decreased by 0.460% on the CLUENER2020 dataset, and the *F*1 value of the model decreased by 2.338% on the Weibo dataset.

Therefore, it is shown that the rotational position encoding RoPE can extract the position information of the sequence, thereby improving the predictive ability of the model.

Putting rotation position encoding into the GlobalPointer module has significant performance on Twitter datasets.

4.5.2. Ablation Experiments of Entity Boundary Score Calculation of GlobalPointer

To prove that the entity boundary score calculation method of GlobalPointer was better than the conditional random field (CRF), this study replaced the CRF module with the GlobalPointer module and conducted four comparative experiments.

In the comparison between BERT-GlobalPointer and BERT-CRF, the *F*1 value of the model increased by 3.597% on the CLUENER2020 dataset, and the *F*1 value of the model increased by 1.952% on the Weibo dataset.

In the comparison between BERT-ATT-BILTAR-GlobalPointer and BERT-ATT-BILTAR-CRF, the *F*1 value of the model increased by 3.965% on the CLUENER2020 dataset, and the *F*1 value of the model increased by 10.990% on the Weibo dataset.

In the comparison between BERT-BILTAR-GlobalPointer and BERT-BILTAR-CRF, the *F*1 value of the model increased by 3.390% on the CLUENER2020 dataset, and the *F*1 value of the model increased by 0.397% on the Weibo dataset.

In the comparison between BERT-ATT-GlobalPointer and BERT-ATT-CRF, the *F*1 value of the model increased by 6.862% on the CLUENER2020 dataset, and the *F*1 value of the model increased by 5.069% on the Weibo dataset.

Through the four experiments, it is found that the *F*1 value of the model can increase significantly by replacing the CRF layer with the GlobalPointer module in various module combinations. Therefore, it is concluded that the performance of the GlobalPointer module is superior to that of the CRF layer. The span-based GlobalPointer module avoids the CRF problem of predicting invalid element labels with the sequence tagging method.

4.5.3. Ablation Experiments of Attention Mechanism

To demonstrate the effectiveness of the attention mechanism, this study conducted five comparative experiments by adding attention mechanisms to multiple locations in the model.

In the comparison between BERT-ATT-BILTAR-GlobalPointer and BERT-BILTAR-GlobalPointer, the *F*1 value of the model increased by 0.944% on the CLUENER2020 dataset and by 10.732% on the Weibo dataset.

In the comparison between BERT-ATT-GlobalPointer and BERT-GlobalPointer, the *F*1 value of the model did not improve on the CLUENER2020 dataset but increased by 1.471% on the Weibo dataset.

In the comparison between BERT-ATT-BiLSTM-ATT-GlobalPointer and BERT-BiLSTM-GlobalPointer, the *F*1 value of the model increased by 1.792% on the CLUENER2020 dataset but did not improve on the Weibo dataset.

In the comparison between BERT-ATT-BiLSTM-PNATT-GlobalPointer and BERT-BiLSTM-PN-GlobalPointer, the *F*1 value of the model increased by 1.257% on the CLUENER2020, and the *F*1 value of the model increased by 2.438% on the Weibo dataset.

In the comparison between BERT-ATT-BILTAR-GlobalPointer and BERT-BiLSTM-PNTIMER-GlobalPointer, the *F*1 value of the model increased by 2.451% on the CLUENER2020 dataset, and the *F*1 value of the model increased by 4.445% on the Weibo dataset.

The experiments show that adding the attention mechanism to most module combinations can improve the *F*1 value of the models. It can be concluded that processing features with an attention mechanism in most positions can achieve a denoising effect and improve the performance of the model.

4.5.4. Ablation Experiments of the Time-Step Function

In the comparison between BERT-ATT-BILTAR-GlobalPointer and BERT-ATT-BiLSTM-PNATT-GlobalPointer, the time-step module TIME behind PN was added. On the CLUENER2020 dataset, the *F*1 value of the model increased by 1.016%, and on the Weibo dataset, the *F*1 value of the model increased by 4.300%. This indicates that the time-step module TIME can mine deep semantic information of words and improve the performance of the model. It should be noted that the model generally does not converge if the output vector of the time-step function is not residual with the output vector of the attention vector or the output vector of the BERT layer.

4.5.5. Ablation Experiments of the Positive–Negative Module PN

To demonstrate the effectiveness of the positive–negative module PN, this study conducted five comparative experiments by adding PN behind the BILSTM module.

In the comparison between BERT-BILTAR-GlobalPointer and BERT-BiLSTM-ATT-TIMER-GlobalPointer, the *F*1 value of the model increased by 0.114% on the CLUENER2020 dataset but did not improve on the Weibo dataset.

In the comparison between BERT-BiLSTM-PN-GlobalPointer and BERT-BiLSTM-GlobalPointer, the *F*1 value of the model increased by 0.145% on the CLUENER2020 dataset but did not improve on the Weibo dataset.

In the comparison between BERT-BiLSTM-PNTIMER-GlobalPointer and BERT-BiLSTM-TIMER-GlobalPointer, the *F*1 value of the model increased by 0.063% on the CLUENER2020 dataset and by 3.954% on the Weibo dataset.

In the comparison between BERT-ATT-BILTAR-GlobalPointer and BERT-ATT-BiLSTM-ATT-TIMER-GlobalPointer, the *F*1 value of the model increased by 0.106% on the CLUENER2020 dataset and by 3.812% on the Weibo dataset.

This study conducted four comparative experiments on the PN module and found that adding the PN module to most module combinations can improve the *F*1 value of the models. It can be concluded that separately processing the forward and backward outputs of the BILSTM module can extract more sequence information.

4.5.6. Ablation Experiments of the BLITAR Module

To demonstrate the effectiveness of the BLITAR module, this study conducted five comparative experiments by adding the BLITAR module to the experimental model.

In the comparison between BERT-ATT-BILTAR-GlobalPointer and BERT-ATT-GlobalPointer, the *F*1 value of the model increased by 1.987% on the CLUENER2020 dataset and by 4.635% on the Weibo dataset.

In the comparison between BERT-BILTAR-GlobalPointer and BERT-GlobalPointer, the *F*1 value of the model increased by 0.148% on the CLUENER2020 dataset but did not improve on the Weibo dataset.

In the comparison between BERT-BILTAR-CRF and BERT-CRF, the *F*1 value of the model increased by 0.678% on the CLUENER2020 dataset but did not improve on the Weibo dataset.

In the comparison between BERT-ATT-BILTAR-CRF and BERT-ATT-CRF, the *F*1 value of the model increased by 0.530% on the CLUENER2020 dataset but did not improve on the Weibo dataset.

Adding the BLITAR module to the model combinations did not have a significant effect on the model performance with CRF on the Weibo dataset. However, adding the BLITAR module to the BERT-ATT-GlobalPointer model improved the *F*1 value of the model. Therefore, the BLITAR module can extract deeper semantic information from text sequences. Compared with the CRF, the BILTAR module is more suitable for the GlobalPointer module.

Compared to CRF modules, it is better to add a GlobalPointer module after the BILTAR module under the same conditions.

Through these four module combination experiments, it is shown that the BILTAR module can extract the deep features of the text sequence. In the BILTAR module, BILSTM can extract the connections between words, the PN module processes the forward and backward information of the sequence, respectively, the attention mechanism denoises the data, the TIME module encoder extracts the deep semantic features of the words, the residual summation method efficiently fuses the forward and backward information of the sequence, and the fused information is the deep semantic information of the sequence. Finally, the effect of the model is improved by the various submodules in the experimental results.

4.5.7. Ablation Experiment on the Number of Fully Connected Layers in TIME Module

The encoder of the TIME module changes the number of fully connected layers to do five groups of ablation experiments and compare the experimental effect. The experimental results show that the model performs best when the encoder is set with two fully connected layers and the decoder also with two fully connected layers. It is concluded that the TIME module can extract deeper semantic information by feature dimension reduction and dimension increase. (A,b) indicate that the encoder of the module has A fully connected layers and the decoder has B fully connected layers. (A) indicates that the module consists of A fully connected layers. The experimental results are shown in Table 6. Bold data indicates the highest value of the corresponding indicator in the table.

Table 6. Ablation experiments of summing and splicing.

	CLUENER2020			**Weibo**		
Compare Models	***P*%**	***R*%**	***F*1%**	***P*%**	***R*%**	***F*1%**
BERT-ATT-BILTAR-GlobalPointer	**80.485**	81.270	**80.848**	**79.227**	**72.720**	**75.751**
BERT-ATT-BILTAR-GlobalPointer (Concatenate)	79.000	**81.729**	80.317	76.921	68.068	72.169

4.5.8. Ablation Experiment of Information Fusion Method of Summation and Concatenation

In the fusion of forward and backward information, two sets of ablation experiments were performed by summation and cascade, respectively. The results shows that the residual summation method is superior to the concatenation method. On the CLUENER2020 dataset, the $F1$ value of the model increased by 0.531%, while on the Weibo dataset, the $F1$ value of the model increased by 3.582%. It is concluded that the summation method for fusing forward and backward information can retain more effective information compared to the concatenation method. The experimental results are shown in Table 7. Bold data indicates the highest value of the corresponding indicator in the table.

Table 7. Ablation experiments where the encoder of the TIME module changes the number of fully connected layers.

	CLUENER2020			**Weibo**		
Compare Models	***P*%**	***R*%**	***F*1%**	***P*%**	***R*%**	***F*1%**
(2,2)	**80.485**	81.270	**80.848**	**79.227**	**72.720**	**75.751**
(1,1)	77.796	**83.386**	80.449	74.571	67.582	70.797
(1)	79.958	81.264	80.575	74.966	68.155	71.368
(3,3)	79.710	80.856	80.232	74.155	68.776	71.348
(4,4)	79.180	81.819	80.447	68.813	72.035	70.206

4.5.9. Summary and Analysis of Ablation Experiment

In short, the performance of the model is improved when the submodules of the model are added to various combined models. Eighteen sets of ablation experiments show the validity of each submodule of the model. The CLUENER2020 dataset contains ten categories, and each category represents a domain. The Weibo dataset contains four categories, and each category is labeled with two markers, explicit entities (NAM), and generic entities (NOM). The CLUENER2020 dataset focuses on distinguishing between multiple categories, and the Weibo dataset has a lower requirement for distinguishing between categories. But the Weibo dataset requires the model to recognize whether entities are explicitly or generically referred to. The difficulties of the two datasets are different, so in the same ablation experiment, the model has different enhancement effects on the two datasets.

5. Conclusions and Future Work

In fine-grained entity recognition tasks, it is difficult to distinguish between similar categories, and simple neural network layers cannot extract enough semantic features to distinguish between similar categories. This paper proposes the BILTAR deep semantic extraction module to extract more semantic information from the data, thereby improving the model's ability to distinguish between fine-grained categories. Sequence labeling tasks are insensitive to entity boundaries. Therefore, this paper uses the GlobalPointer module instead of the conditional random field to calculate the score of each possible entity boundary in each category, thereby fully considering all entity boundaries. The experiments show that the BILTAR module can improve the model's feature extraction ability, each sub-module of the BILTAR module can also individually improve the model's performance, the model's performance is slightly improved after the GlobalPointer module replaces the CRF layer, and the multi-head self-attention mechanism added to multiple positions can also improve the model's performance. Therefore, the experimental results fully prove the effectiveness of the multi-head self-attention mechanism, the BILTAR module, and the GlobalPointer module.

The performance improvement of entity recognition model is of great significance to the development of the knowledge graph. The identified entities and entity categories can serve as knowledge for knowledge fusion, knowledge reasoning, and other downstream tasks of the knowledge graph. In each knowledge graph task, improving the entity recognition ability of the model can reduce the error propagation in the entity recognition stage and improve the realization effect of each downstream task. In view of the above situation, the entity recognition task is prospected as follows.

In the future, improvements can be made to the BILTAR module by incorporating more complex network structures. The stronger semantic extraction module enables the model to extract deeper semantic features and improve the model performance. It is possible to optimize the GlobalPointer module to improve the model's ability to compute entity boundaries.

Author Contributions: Conceptualization, J.L.; Methods, J.L. and W.L.; Verification, J.L., Y.G. and J.G.; Formal analysis, X.Z.; Resources, J.L.; Data management, Y.G.; Writing—Original manuscript preparation, J.L.; Writing—Review and editing, W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Basic Scientific Research in Central Universities of North Minzu University (2021JCYJ12), the Ningxia Natural Science Foundation Project (2021AAC03215), the National Natural Science Foundation of China (62066038, 61962001).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. Because the data sets in this article were developed by individual institutions, the data are not publicly available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, Y.; Xu, L.; Liu, K.; Zeng, D.; Zhao, J. Event extraction via dynamic multi-pooling convolutional neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; Volume 1, pp. 167–176.
2. Diefenbach, D.; Lopez, V.; Singh, K.; Maret, P. Core techniques of question answering systems over knowledge bases: A survey. *Knowl. Inf. Syst.* **2018**, *55*, 529–569. [CrossRef]
3. Li, P. Prosodic unit boundary prediction of Myanmar based on BERT-CRF model. *Comput. Sci. Appl.* **2021**, *11*, 505–514.
4. Liu, L.; Shang, J.; Ren, X.; Xu, F.; Gui, H.; Peng, J.; Han, J. Empower Sequence Labeling with Task-Aware Neural Language Model. *Proc. AAAI Conf. Artif. Intell.* **2018**, *32*, 5253–5260. [CrossRef]

5. Morwal, S.; Jahan, N.; Chopra, D. Named entity recognition using hidden Markov model (HMM). *Int. J. Nat. Lang. Comput.* **2012**, *1*, 581–586. [CrossRef]
6. Zheng, Q.S.; Liu, S.X. Research of web text named entity recognition based on CRF. *J. Zhongyuan Univ. Techol.* **2016**, *27*, 70–73.
7. Cucerzan, S.; Yarowsky, D. Language independent named entity recognition combining morphological and contextual evi-dence. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*; Johns Hopkins University: Baltimore, MD, USA, 1999.
8. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
9. Li, S.; Li, W.; Cook, C.; Zhu, C.; Gao, Y. Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 5457–5466.
10. Tai, K.S.; Socher, R.; Manning, C.D. Improved Semantic Representations from Tree-Structured Long Short-Term Memory Networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 26–31 July 2015; pp. 1556–1566.
11. Chung, J.; Gulcehre, C.; Cho, K.H.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
12. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
13. Chiu JP, C.; Nichols, E. Named entity recognition with bidirectional LSTM-CNNs. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 357–370. [CrossRef]
14. Ma, X.; Hovy, E. *End-to-End Sequence Labeling via Bi-Directional LSTM-CNNs-CRF*; Association for Computational Linguistics: Berlin, Germany, 2016. [CrossRef]
15. Zhang, Y.; Yang, J. Chinese NER Using Lattice LSTM. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Association for Computational Linguistics: Melbourne, Australia, 2018; Volume 1, pp. 1554–1564.
16. Ma, R.; Peng, M.; Zhang, Q.; Huang, X. Simplify the usage of lexicon in Chinese NER. *arXiv* **2019**, arXiv:1908.05969.
17. Zhu, Y.; Wang, G.; Karlsson, B.F. CAN-NER: Convolutional attention network for Chinese named entity recognition. *arXiv* **2019**, arXiv:1904.02141.
18. Gui, T.; Ma, R.; Zhang, Q.; Zhao, L.; Jiang, Y.-G.; Huang, X. CNN-Based Chinese NER with Lexicon Rethinking. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019.
19. Sohrab, M.G.; Miwa, M. Deep exhaustive model for nested named entity recognition. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2843–2849.
20. Xu, M.; Jiang, H. A FOFE-based local detection approach for named entity recognition and mention detection. *arXiv* **2016**, arXiv:1611.00801.
21. Xia, C.; Zhang, C.; Yang, T.; Li, Y.; Du, N.; Wu, X.; Fan, W.; Ma, F.; Yu, P. Multi-grained Named Entity Recognition. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019.
22. Li, J.; Fei, H.; Liu, J.; Wu, S.; Zhang, M.; Teng, C.; Ji, D.; Li, F. Unified Named Entity Recognition as Word-Word Relation Classification. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 10965–10973. [CrossRef]
23. Zhai, F.; Potdar, S.; Xiang, B.; Zhou, B. Neural models for sequence chunking. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 1 January 2017; pp. 3365–3371.
24. Li, J.; Ye, D.; Shang, S. Adversarial transfer for named entity boundary detection with pointer networks. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 5053–5059.
25. Su, J.; Murtadha, A.; Pan, S.; Hou, J.; Sun, J.; Huang, W.; Wen, B.; Liu, Y. Global Pointer: Novel Efficient Span-based Approach for Named Entity Recognition. *arXiv* **2022**, arXiv:2208.03054.
26. Zhang, X.; Luo, X.; Wu, J. A RoBERTa-GlobalPointer-Based Method for Named Entity Recognition of Legal Documents. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 18–23 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–8.
27. Sun, S.; Zhang, K.; Li, J.; Cen, J.; Wang, Y. NCG-LS: Named Entity Recognition Model Specializing for Analyzing Product Ti-tles. In Proceedings of the 2022 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), Niagara Falls, ON, Canada, 17–20 November 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 582–587.
28. Mengge, X.; Yu, B.; Liu, T.; Zhang, Y.; Meng, E.; Wang, B. Porous lattice transformer encoder for Chinese NER. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain (Online), 8–13 December 2020; pp. 3831–3841.
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
30. Graves, A. *Long Short-Term Memory. Supervised Sequence Labelling with Recurrent Neural Networks*; Springer Science and Business Media LLC: Dordrecht, The Netherlands, 2012; ISBN 3540306765.
31. Su, J.; Lu, Y.; Pan, S.; Murtadha, A.; Wen, B.; Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *arXiv* **2021**, arXiv:2104.09864. [CrossRef]
32. Xu, L.; Dong, Q.; Liao, Y.; Yu, C.; Tian, Y.; Liu, W.; Li, L.; Liu, C.; Zhang, X. CLUENER2020: Fine-grained named entity recognition dataset and benchmark for Chinese. *arXiv* **2020**, arXiv:2001.04351.

33. Peng, N.; Dredze, M. Named entity recognition for Chinese social media with jointly trained embeddings. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 548–554.
34. Li, H.; Cheng, M.; Yang, Z.; Yang, L.; Chua, Y. Named entity recognition for Chinese based on global pointer and adversarial training. *Sci. Rep.* **2023**, *13*, 3242. [CrossRef] [PubMed]
35. Gan, L.; Huang, C. A Chinese named entity recognition method combined with relative position information. In Proceedings of the 2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS), Shenyang, China, 22–24 January 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 250–254.
36. Han, X.; Yue, Q.; Chu, J.; Han, Z.; Shi, Y.; Wang, C. Multi-Feature Fusion Transformer for Chinese Named Entity Recognition. In Proceedings of the 2022 41st Chinese Control Conference (CCC), Heifei, China, 25–27 July 2022.
37. Li, D.; Zhang, H.; Wang, J.; Li, S. A Chinese NER Method Based on Chinese Characters' Multiple Information. In Proceedings of the 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 24–26 February 2023; pp. 1048–1052.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.