



Research Article

Hierarchical Annotation Event Extraction Method in Multiple Scenarios

Shi Wang ,¹ Zhujun Wang ,^{1,2} Yi Jiang ,^{2,3} and Huayu Wang⁴

¹*Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*

²*School of Information Engineering, Yangzhou University, Yangzhou, Jiangsu 225127, China*

³*State Key Laboratory of Ocean Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*

⁴*Information Engineering College, Graduate School of Army Engineering University, Nanjing 210007, China*

Correspondence should be addressed to Zhujun Wang; mx120180424@yzu.edu.cn

Received 28 September 2020; Revised 4 December 2020; Accepted 27 February 2021; Published 19 March 2021

Academic Editor: Paul Honeine

Copyright © 2021 Shi Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the event extraction task, considering that there may be multiple scenarios in the corpus and an argument may play different roles under different triggers, the traditional tagging scheme can only tag each word once, which cannot solve the problem of argument overlap. A hierarchical tagging pipeline model for Chinese corpus based on the pretrained model Bert was proposed, which can obtain the relevant arguments of each event in a hierarchical way. The pipeline structure is selected in the model, and the event extraction task is divided into event trigger classification and argument recognition. Firstly, the pretrained model Bert is used to generate the feature vector and transfer it to bidirectional gated recurrent unit+conditional random field (BiGRU+CRF) model for trigger classification; then, the marked event type features are spliced into the corpus as known features and then passed into BiGRU+CRF for argument recognition. We evaluated our method on DUEE, combined with data enhancement and mask operation. Experimental results show that our method is improved compared with other baselines, which prove the effectiveness of the model in Chinese corpus.

1. Introduction

As a form of information, event is defined as the fact that some people or things interact at a certain time and place. As an important task in information extraction, event extraction is aimed at identifying triggers and arguments from unstructured text and expressing them in a structured way, which is the basic work of constructing knowledge graph. Natural language text contains many events and related arguments, as shown in Figure 1. Two events appear in ex1, and it describes two volleyball matches. “Eliminated” and “defeated” are the triggers of the event type of “competition behavior—win or lose.” “The National Women’s Volleyball Championship” plays the role of “competition name” in both events, “Tianjin Women’s volleyball team” plays the role of “loser” in the first event, and “Jiangsu women’s volleyball team” and “Shandong women’s volleyball team” played the role of “loser” and “winner,” respectively, in the second event.

The traditional event extraction model cannot reasonably distinguish the arguments under multiple scenario events. If there is more than one event in the sentence, the model cannot be extracted accurately.

There are two ways to implement event extraction task; one is based on pattern matching, and the other is based on machine learning. The method based on pattern matching mainly uses lexical symbol features, semantic features, and self-organizing constraints to extract events, and the key lies in the construction of event templates. However, the method based on pattern matching has poor cross domain and needs a lot of manual operation. With the continuous enrichment of corpus in recent years, researchers use machine learning method to extract events, especially deep learning and neural network. The traditional machine learning method uses the classified thought to transform the task of trigger classification and argument recognition into classification problem. The key point is the construction of classifier and the selection of fea-

Ex1 : The National Women's Volleyball Championship held in Jiangmen, Guangdong Province, has entered the final climax stage. All the top four teams have been produced this afternoon. After Tianjin Women's volleyball team was eliminated, there was a big surprise today. Jiangsu women's volleyball team was defeated 0-3 by Shandong women's volleyball team, and was not qualified for the top four.

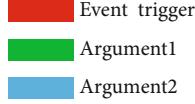


FIGURE 1: Trigger classification and argument recognition in event extraction.

tures. Chieu [1] introduced the maximum entropy model into event extraction task for the first time and realized the extraction of seminar notice and personnel management event. Llorens et al. [2] used conditional random field (CRF) to tag semantic roles, which improved the performance of the system. Support vector machine (SVM) and hidden Markov model (HMM) are also commonly used classification models.

Aiming at the problems of existing models, this paper proposes a pipeline model based on the pretrained model Bert. The main contributions of this paper are as follows:

- (i) An event extraction model based on the pretrained model Bert was designed. First, the event triggers in the corpus are labeled by pipeline, and then, the relevant arguments of each trigger are extracted
- (ii) Pipeline model extracts event triggers by hierarchical way and improves recall and accuracy of event recognition by data enhancement
- (iii) The model extracts the arguments from each identified event, respectively, which successfully solves the problem of argument overlapping. Through the use of window, the performance of the model is improved successfully, and the argument recognition errors in multiple scenarios are reduced

The rest of this paper is organized as follows. Section 2 reviews the related work on path event extraction, introduces a hierarchical annotation model for event trigger extraction and related argument recognition, and introduces each module of the model. Section 4 analyzes the experimental results of the current model through comparative experiments to verify the reliability of the model we proposed. Section 5 summarizes the paper and plans the direction of future work.

2. Related Work

Compared with traditional way of event extraction, more and more event extraction models based on neural network appear. Zheng et al. [3] proposed a tagging scheme, which transforms the task of information extraction into a tagging problem. However, one word can only be tagged once, which makes it difficult to extract from multievent sentences. According to the overlap of triples, Zeng et al. [4] divide sentences into normal, entity pair overlap, and single entity overlap and proposes an end-to-end model based on replication mechanism to jointly extract information, which solves the problem of entity overlap. However, this model can only

be used for a single word. If an entity has more than one word, it cannot extract the relationship accurately.

Traditional machine learning methods for event extraction need to design a large number of features manually and also need the support of external Natural Language Processing (NLP) tools. Based on neural network method, event extraction is modeled as an end-to-end model, which gets rid of the dependence on external NLP tools and uses word vectors with rich features as input, thus avoiding complex manual work. Nguyen and Grishman [5] studied the problem of event triggered word extraction in unbalanced corpus and used convolutional neural network to capture important feature information in sentences. Chen et al. [6] proposed a dynamic multipool convolutional neural network (DMCNN) to extract sentence level features. In Ghaeini et al.'s [7] paper, bidirectional recurrent neural network (Bi-RNN) is first used to detect events that can be words or phrases, which is the first attempt to extract multitoken events. Feng et al. [8] combined bidirectional long short-term memory (LSTM) and convolutional neural network to learn word representation and predict event triggers.

Although the neural network-based event extraction method has achieved good performance, due to the fact that there may be multiple scenario events in a sentence, the confusion of argument annotation and role overlap are also important problems in the current event extraction, which makes event extraction still a difficult NLP problem. Therefore, this paper focused on the research of hierarchical annotation event extraction method in multiple scenarios, which improves the recognition effect of event triggers and alleviates the confusion of argument.

3. Methodology

3.1. Model Flow Chart. In the traditional event extraction research, scenario switching under multiple scenario events leads to argument extraction confusion, and the traditional annotation scheme cannot solve the problem of argument overlap. In this paper, a pipeline model of event extraction based on pretrained model Bert is proposed to solve the chaotic problem of event argument extraction in multiple scenarios. The flow chart is shown in Figure 2, which is divided into three parts: pretrained model Bert, event trigger classification, and argument recognition.

3.2. Model Architecture. The architecture of the model in this paper is shown in Figure 3. The model represents the event extraction task as a pipeline model based on hierarchical tagging, which solves the problem that a word can only be

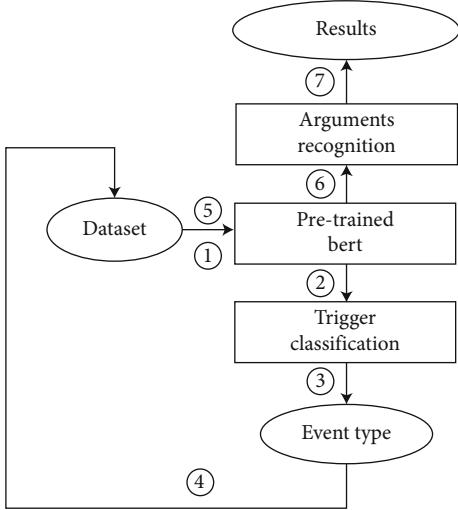


FIGURE 2: Flow chart of the model.

tagged once, resulting in argument overlap. The first stage of the model is trigger classification using a model based on pre-trained model Bert. If the trigger is identified in the sentence, the second stage is carried out. The extracted event types are placed in the front of the sentence as features and input into the argument extraction model to extract relevant arguments and identify roles.

3.3. Pretrained Model Bert. The word2vec [9] model considers that the meaning of a word is associated with the meaning of the word that appears around it, thus mapping each word into a vector. However, in natural language, a word may have different meanings, but the traditional word2vec model does not generate static vectors and ignores the context, which cannot solve the problem of polysemy. The pretrained model Bert [10] makes full use of context and solves the problem of polysemy.

The Bert model is shown in Figure 4(a). The model selects the editing module of transformer model [11] as the model of feature extraction for bidirectional coding. The module structure is shown in Figure 4(b). By adding attention mechanism to replace the traditional convolutional neural network and cyclic neural network, the coding features of each word can obtain the information of all words.

The most important content of transformer encoding module is self-attention mechanism, which inputs the encoded vector and calculates the relationship between the current token and the context and obtains the weighted sum as the output of the current word. This makes the output vector contain not only the meaning of the word itself but also the relationship with other words. The weight calculation formula is as follows:

$$\text{attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V, \quad (1)$$

where Q , K , and V represent query vector, key vector, and value vector, respectively, and d_k is the dimension of the

input vector. After the inner product sum of current token's Q and each token's K , the weight is obtained by softmax, and then, V is weighted and summed by using the weight obtained in the previous step to get the output coding vector of the current token. However, the use of self-attention mechanism can only obtain one feature expression; transformer module uses multihead attention mechanism to map Q , K , and V to QW_i^Q , KW_i^K , and VW_i^V with different projections, and the specific formula is as follows:

$$\begin{aligned} \text{attention}(Q, K, V) &= \text{concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_n), \\ \text{head}_i &= \text{attention}\left(QW_i^Q, KW_i^K, VW_i^V\right). \end{aligned} \quad (2)$$

In the self-attention mechanism, the position of each word and the position between the included segments have an impact on the representation of the current word. Therefore, the position vector and segment vector are added to Bert. In addition, a normalization and residual link are added after each self-attention module and feedforward neural network module, which solves the problem of gradient dispersion and improves the training efficiency of the model.

3.4. GRU Layer. In recurrent neural networks, hidden state is always transmitted from front to back. However, in the event extraction, the hidden state of the current time step is associated with the previous time and the next moment, so the bidirectional recurrent neural network is needed to establish the correlation. The bidirectional GRU [12] model selected in this paper solves the problems of long-term memory and gradient in backpropagation, as shown in Figure 5. Compared with LSTM [13], GRU can achieve similar results, and it is easier to train and improve training efficiency.

The input and output structure of GRU is consistent with the traditional RNN [14], including the current input x_t and the hidden state h_{t-1} passed in the previous time step $t-1$. But different from RNN, GRU uses gating mechanism to control the hidden state of the previous moment, instead of receiving all the features of h_{t-1} like RNN. The two gates built into GRU are reset gate r and update gate z , and the formulas are as follows: $r = \sigma(W^r[h_{t-1}, x_t])$ and $z = \sigma(W^z[h_{t-1}, x_t])$. After getting the gating signal, the reset gate r is used to reset the hidden state of the previous time, i.e., $h'_{t-1} = h_{t-1} \odot r$. Then, the data range is reduced to $(-1, 1)$ by activating function tanh, and the formula is expressed as $h'_t = \tanh(Wh'_{t-1})$. Finally, according to the calculated update gate, two operations, forgetting and memorizing, are carried out at the same time. The specific formula is

$$h_t = (1 - z) \odot h_{t-1} + z \odot h'_t \quad (3)$$

3.5. Conditional Random Field. CRF [14] is a sequence tagging algorithm, which outputs the target sequence after inputting a segment of sequence. In NLP annotation task, the input sequence is a piece of text, and the output sequence is the corresponding tag. Considering the correlation between adjacent tags, CRF obtains a global optimal tagging chain.

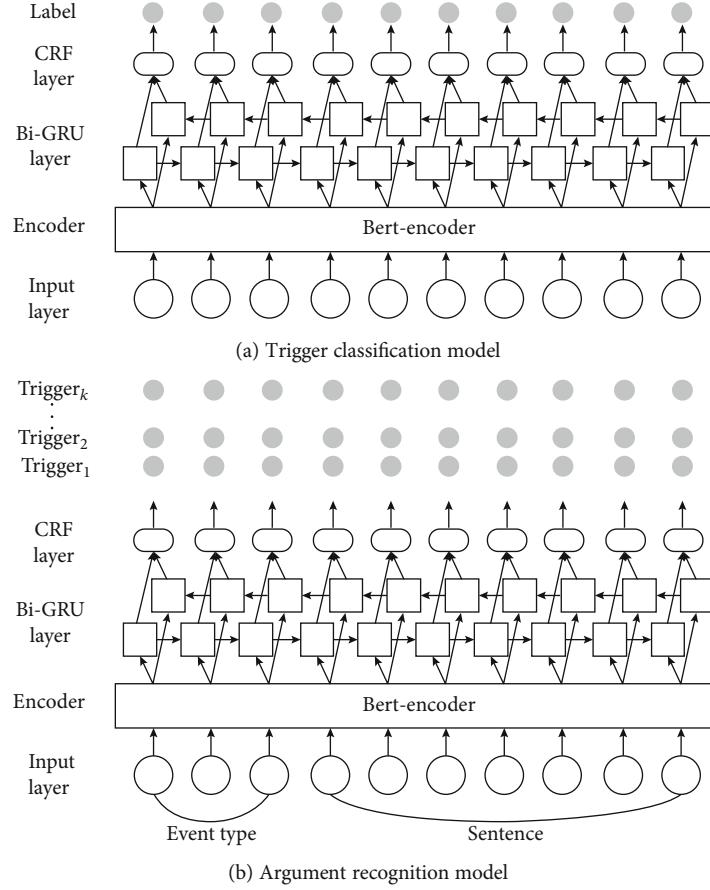


FIGURE 3: Event extraction model architecture.

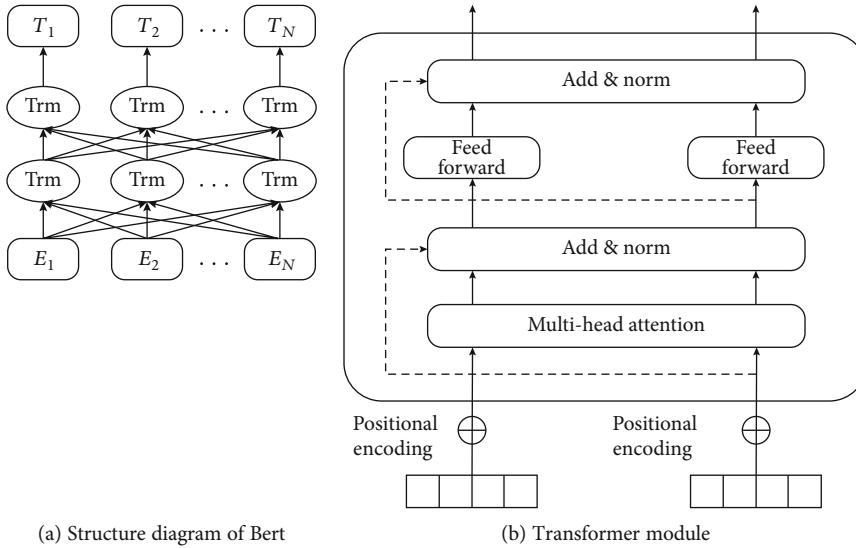


FIGURE 4: Pretrained model Bert.

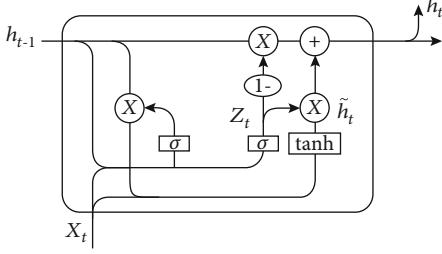


FIGURE 5: GRU module.

Set the matrix $P \in \mathcal{R}^{n \times N_t}$ as the fractional matrix output through the linear layer, where p_{ij} represents the probability that the i th word in the sentence is marked as the j th label. For the sentence $S = \{x_1, x_2, \dots, x_n\}$ and the corresponding tag $y = \{y_1, y_2, \dots, y_n\}$, CRF will give a specific score, the formula is as follows:

$$\text{score}(S, y) = \sum_{i=0}^n T_{y_{i-1}y_i} + \sum_{i=1}^n p_{i,y_i}, \quad (4)$$

where T is the transfer matrix, $T_{i,j}$ is a transfer probability from tag i to tag j , due to the special markers at the beginning and end of a sentence, and T is a square matrix with dimension $N_t + 2$. Then, the probability that the tag sequence of sentence S is y is

$$P(y | S) = \frac{\exp \{ \text{score}(S, y) \}}{\sum_{y' \in Y_s} \exp \{ \text{score}(S, y') \}}, \quad (5)$$

where Y_s represents all the tag sequences that sentence S can give. The logarithm of both sides of the formula is transformed into

$$\log(p(y | S)) = \text{score}(S, y) - \log \left(\sum_{y' \in Y_s} \exp \{ \text{score}(S, y') \} \right). \quad (6)$$

The loss function is defined as $\text{loss} = -\text{score}(S, y)$, and then, the decoded tag sequence is obtained by formula $y^* = \text{argmax}_{y' \in Y_s} \text{score}(S, y')$.

3.6. Trigger Classification. The model uses the BiGRU-CRF model based on pretrained model Bert to identify and classify the triggers. The input of the model is the vector pretrained by Bert, and the encoding features of each word contain the information of all words. The structure of BiGRU-CRF model is shown in Figure 6(a), which consists of three parts: encoding layer, BiGRU layer, and CRF layer.

Let $S = \{x_1, x_2, \dots, x_n\}$ be a sample input, where x_i is the i th word in the sentence, and sentence S is mapped to the matrix $W = \{e_1, e_2, \dots, e_n\}$ after random initialization and passed into the pretrained model Bert. The vector generated by Bert is mapped to the feature matrix $V \in \mathcal{R}^{n \times d}$, where n is the length of the sentence and D is the dimension of the

			B-victory	O	O	O	O	O	O
Trigger	O								
Sentence		Gasley won the British Grand Prix.							
Argument	B-champion	O	O	B-game	I-game	I-game			

FIGURE 6: Tagging example.

word vector. Next, the feature matrix V is transferred to the BiGRU layer for further feature extraction, where the hidden state at step i is $h_i = (\overrightarrow{h_i}, \overleftarrow{h_i})$. The calculation formula of hidden state vector is as follows:

$$\begin{aligned} \overrightarrow{\text{GRU}}(v_1, v_2, \dots, v_n) &= (\overrightarrow{h_1}, \overrightarrow{h_2}, \dots, \overrightarrow{h_n}), \\ \overleftarrow{\text{GRU}}(v_1, v_2, \dots, v_n) &= (\overleftarrow{h_1}, \overleftarrow{h_2}, \dots, \overleftarrow{h_n}). \end{aligned} \quad (7)$$

After the hidden state h_i of each word is obtained, it is passed as input to the CRF layer for final label classification, and the final score matrix $P \in \mathcal{R}^{N_t \times n}$ is obtained. The k th column of the matrix P represents the fraction of each corresponding tag for input x_1 . According to the matrix P , the tag sequence y is obtained, and then, the trigger in the text is extracted and its type is determined.

3.7. Argument Recognition. The identification of argument is based on the extraction of triggers, and the extracted event types are spliced into the text as features for the next step. As shown in Figure 6(b), this model adopts hierarchical tagging scheme to label multiple events separately, avoiding the defect that one word can only be tagged once at a time. The overall structure of the model is similar to the extraction of triggers. The BiGRU-CRF model based on pretrained model Bert is selected as the extraction model. The difference is that the hierarchical extraction method is selected for the argument extraction. In order to avoid errors in multiscenario event corpus extraction, mask preprocessing is carried out before argument extraction.

The event type and text are spliced in the model, and the feature matrix V is generated in the pretrained model Bert. However, due to the existence of multiscenario event corpus, it is necessary to mask the short sentences unrelated to the current event before it is passed into Bert. The calculation process of mask vector is shown in Algorithm 1.

In this paper, the hierarchical argument extraction model extracts arguments from the related trigger words extracted in the previous step and identifies their roles in the trigger. It is known that sentence $S = \{\text{event type}, x_1, x_2, \dots, x_n\}$ is randomly initialized to matrix $W = \{e_0, e_1, \dots, e_n\}$, and the goal is to extract relevant arguments for the current trigger. According to the mask M , the input of the pretrained model is input = $\{\text{event type}, W \odot M\}$. The vector input is mapped to the feature matrix V based on the pretrained model Bert. Next, the feature matrix is transferred to BiGRU layer for further feature extraction, where the hidden state at step i is h_i

```

Input: Sentence:  $S = \{x_1, x_2, \dots, x_n\}$ ; Event type: list; event trigger; triggeri
Output: mask vector:  $M_i$ 
 $M_i = [1] * \text{len}(S);$ 
 $i, j = 0, 0;$ 
if length of event type is 1 then
    return the vector  $M_i$ 
end
other_trigger = list.pop(triggeri)
for triggerj in other-trigger do
    find  $i, j$  which is the index of  $S$  where  $s[i : j + 1]$  is the minimal clause containing triggerj;
    if triggeri is not in the clause  $s[i : j + 1]$  then
        change 1 of  $M[i : j + 1]$  to 0;
    end
end
return the vector  $M_i$ 

```

ALGORITHM 1: Calculation of mask vector for specified event trigger.

$= (\vec{h}_i, \vec{h}_i)$. Continue to pass in the CRF layer to calculate the score of each word corresponding to each tag, so as to further extract the relevant arguments of the current event.

3.8. Training and Optimization. This summary mainly introduces the learning and optimization details of the framework model. It is known that trigger classification model takes text S as input; the network with parameters θ outputs the event category vector O , where O_i is the value of i th position and indicates that the trigger is a fraction of type I. The optimization of argument recognition model is consistent with the trigger classification model, but the difference is that the event types identified in the previous step are added to the data set. The model in this paper maximizes the log likelihood of the data, and the optimization method used in the model is Adam proposed by Kingma and Ba [15]. The objective function is defined as

$$J(\theta) = \max \sum_{i=1}^{|D|} \sum_{t=1}^{S_i} \left(\log \left(p_t^{(i)} = y_t^{(i)} | S_j, \theta \right) \right) \bullet I + \alpha \left(\log \left(p_t^{(i)} = y_t^{(i)} | S_j, \theta \right) \right) \bullet (1 - I), \quad (8)$$

where $|D|$ represents the size of training data set, S_i represents the length of the i th sentence, $y_t^{(i)}$ represents the actual tag of the sentence, and $p_t^{(i)}$ is the tag based on the CRF score. α is the bias weight, and the larger the value, the greater the influence of relation label on the model. In addition, $I(O)$ is a conversion function used to distinguish the loss of the mark ' O ' and the relation marker, which is defined as follows:

$$I(O) = \begin{cases} 1, & \text{if tag} = 'O', \\ 0, & \text{if tag} \neq 'O'. \end{cases} \quad (9)$$

4. Experiment

4.1. Experiment Settings

4.1.1. Data Set. In order to evaluate the performance of our method, the corpus of event extraction task in 2020 language and intelligent technology competition is selected as our data set. The data set DUEE contains 65 defined event type and 17000 Chinese sentences, including 12000 training sets, 1500 verification sets, and 3500 test sets.

4.1.2. Evaluation. Similar to previous work [16], we use the following criteria to determine the correctness of each prediction event trigger and argument extraction:

- (i) A trigger is correct if its event type and offsets match those of a reference trigger
- (ii) An argument is correctly identified if its event type and offsets match those of any of the reference argument mentions
- (iii) An argument is correctly identified and classified if its event subtype, offsets, and argument role match those of any of the reference argument mentions

Finally, we use Precision (P), Recall (R), and F-measure ($F1$) to evaluate the overall performance.

4.1.3. Hyperparameters. Our model consists of the pretraining layer, BiGRU layer, and CRF layer. The word embedding before the pretraining is generated by random initialization, and the dimension of word embedding is set to $D = 300$. The maximum length of a single sentence is limited to 300 words, dropout is 0.1, and Adam optimizer selects training learning rate of $1e-3$ and batch size of data set of 32.

4.2. Tagging Scheme. In this paper, we use the “BIO” tagging scheme [17], where “B” (begin) represents the first word of the trigger or argument, I (inside) represents the subsequent word, and O (other) represents the unrelated word. Taking Figure 6 as an example, event type and argument roles have been predefined, and the extracted results are saved in a structured manner. In the case of triggers, the number of tags

for the event type is $N_t = 2 * |R| + 1$, where $|R|$ is the number of predefined event types, and Figure 3 shows an example of the marking method.

4.3. Experimental Results and Analysis

4.3.1. Experiment 1: Event Trigger Classification and Argument Recognition Based on Tagging Scheme. The model is based on the pretrained model Bert, and BiGRU layer and CRF layer are added for trigger classification and argument recognition. Four related neural network models are selected as baseline models to compare with the model proposed in this paper. The four control neural network models are as follows:

- (i) Bert: the data set is used to fine-tune the parameters of the Bert model, and finally, the sequence tag is obtained
- (ii) Bert-CRF: after fine-tuning the parameters of the Bert model using the data set, the conditional random field is added to constrain the related tags
- (iii) Bert-BiLSTM: it consists of Bert and a long short-term memory network layer
- (iv) Bert-BiLSTM-CRF: on the basis of Bert-BiLSTM, conditional random field is added for training

The experimental results of trigger recognition and argument extraction on DUEE by five models are shown in Tables 1 and 2.

Through the comparison of the results between the models in Tables 1 and 2, it can be seen that the addition of the pretrained model Bert improves the event extraction and argument recognition classification, and the average $F1$ score increased by 6.65%.

In Table 3, error analysis is conducted for all trigger classification and argument recognition results. The main causes of errors can be found as follows:

- (i) Trigger classification error: due to the fuzzy vocabulary of event triggers and inconsistent annotation in the data, classification errors occur
- (ii) Missing of trigger word recognition: similar to the multiple scenario event corpus, there are more than one event trigger. However, the model can only identify one or part of the triggers, but not all the triggers
- (iii) Argument classification errors: the model successfully extracted and marked arguments, but the classification was wrong
- (iv) Missing of argument recognition: there will be arguments with multiple roles under an event trigger, and the model will miss some argument when identifying the argument
- (v) Argument boundary segmentation error: event argument extraction is realized by tagging, and the tag will have boundary error

TABLE 1: Trigger classification results on DUEE.

Model	P (%)	R (%)	F (%)
BiLSTM-CRF	87.76	83.26	85.45
Bert	93.17	84.40	88.57
Bert-CRF	94.28	87.99	91.02
Bert-BiLSTM-CRF	94.86	88.03	91.31
Bert-BiGRU-CRF	94.62	88.69	91.56

TABLE 2: Argument recognition results on DUEE.

Model	P (%)	R (%)	F (%)
BiLSTM-CRF	75.27	76.18	75.72
Bert	86.29	81.96	84.07
Bert-CRF	86.86	79.14	82.82
Bert-BiLSTM-CRF	86.36	82.03	84.13
Bert-BiGRU-CRF	86.48	82.37	84.37

4.3.2. Experiment 2: The Influence of Corpus Distribution on Trigger Classification. The distribution of various types of events in the competition data set is not balanced, and the triggers in the same type of event corpus are unbalanced. For example, there are 605 articles on “organizational relationship resignation” and only 74 articles on “organizational behavior parade.” The F score of the two is 97.70% and 61.54%, respectively. Therefore, this experiment studies the influence of corpus distribution on the performance of trigger extraction. By analyzing the relation between corpus distribution and extraction performance, the training data set is enriched manually. The experimental results are shown in Table 4.

The results in Table 4 show that the distribution of corpus has an important impact on trigger recognition. The performance of event extraction is improved by knowledge enhancement of training set by manual supplement of data set. It also lays a better foundation for further argument recognition.

4.3.3. Experiment 3: The Influence of Adding Mask on Argument Recognition. Due to the frequent occurrence of argument recognition confusion in multiscene corpus, this paper chooses to add mask operation before argument extraction to reduce the confusion. In this experiment, we also choose the Bert-BiGRU-CRF model to extract and classify the argument and mask the operation before extracting the argument. The experimental results are shown in Table 5. The change of argument extraction results brought by whether or not to add mask operation is compared.

The results in Table 5 show that the mask operation improves the accuracy of argument extraction to some extent and alleviates the confusion in argument extraction. And the effect was improved most obviously in the Bert-BiGRU-CRF.

5. Conclusion

In this paper, a trigger classification and argument extraction model based on hierarchical annotation scheme is proposed.

TABLE 3: Event extraction error instance.

Type	Error instance
Trigger classification error	Pick up orders at 4 am! 51-year-old takeaway rider killed in a car accident Corpus label: disaster/accident-traffic accident Model tag: life-death
Missing of trigger word recognition	Three people have been killed and five injured in the collapse of Shenzhen sports center Corpus label: disaster/accident-collapse life-death Model tag: life-death
Argument classification errors	Lack of major general Braves hard won the G5, even if the public opinion is more optimistic about the rackets into the west Argument role: {winner: Braves} Role tag: {winner: Braves} {winner: rocket}
Missing of argument recognition	A 5.4 earthquake occurred in Gongxian country. Argument role: {magnitude: 5.4} {epicenter: Gongxian country} Role tag: {magnitude: 5.4}
Argument boundary segmentation error	The movie “Cold Pursuit” will be released nationwide on September 6! Argument role: {time: September 6} {movie: “Cold Pursuit”} Role tag: {movie: ““Cold Pursuit” will be released nationwide on September 6”}

TABLE 4: Data augmentation performance of trigger classification.

Model	P (%)	R (%)	F (%)
Bert-BiLSTM-CRF (none)	94.86	88.03	91.31
Bert-BiLSTM-CRF (DA)	95.25	88.44	91.72
Bert-BiGRU-CRF (none)	94.62	88.69	91.56
Bert-BiGRU-CRF (DA)	95.47	88.74	91.98

TABLE 5: The effect of adding mask or not on argument recognition.

Model	P (%)	R (%)	F (%)
Bert-BiLSTM-CRF (none)	86.36	82.03	84.13
Bert-BiLSTM-CRF (mask)	86.73	82.85	84.74
Bert-BiGRU-CRF (none)	86.48	82.37	84.37
Bert-BiGRU-CRF (mask)	86.92	83.14	84.99

The event extraction task is completed by pipeline. Without complex NLP pretreatment, lexical features are extracted, and hierarchical tagging effectively alleviates the problem of argument overlap. The operation of adding mask before argument extraction reduces the confusion of argument extraction and proves the effectiveness of mask operation, which provides an effective event extraction model for multiscene event corpus.

Compared with the traditional model, Bert-BiGRU can extract more than one expected event at the same time. For each event, different roles of the same argument can be distinguished accurately. However, the error in the process of event discrimination in pipeline model will lead to the error of argument extraction in the later stage, which leads to a wrong transmission. Therefore, the future work will focus on the model construction of joint extraction. For multiscene event extraction, a more reasonable segmentation method

can be used to improve the extraction performance of multievent corpus. Combining knowledge enhancement [18, 19] is also a major research focus in the future.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61872313, in part by National Natural Science Foundation of China (61702234), and in part by the National Key Research and Development Program Grants 2017YFB1002300 and 2018YFC1700302.

References

- [1] H. L. Chieu and H. T. Ng, “A maximum entropy approach to information extraction from semi-structured and free text,” *National Conference On Artificial Intelligence*, vol. 1, pp. 786–791, 2002.
- [2] H. Llorens, E. Saquete, and B. Navarrocolorado, “TimeML events recognition and classification: learning CRF models with semantic roles,” in *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 725–733, Beijing, 2010.
- [3] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, “Joint extraction of entities and relations based on a novel tagging scheme,” in *Proceedings of the 55th Annual Meeting of the*

- Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 1227–1236, 2017.
- [4] X. Zeng, D. Zeng, S. He, K. Liu, and J. Zhao, “Extracting relational facts by an end-to-end neural model with copy mechanism,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 506–514, Melbourne, Australia, 2018.
 - [5] T. H. Nguyen and R. Grishman, “Relation extraction: perspective from convolutional neural networks,” in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 39–48, Denver, Colorado, 2015.
 - [6] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, “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 (Volume 1: Long Papers)*, vol. 1, pp. 167–176, Beijing, China, 2015.
 - [7] R. Ghaeini, X. Z. Fern, L. Huang, and P. Tadepalli, “Event nugget detection with forward-backward recurrent neural networks,” 2018, <https://arxiv.org/abs/1802.05672>.
 - [8] X. Feng, B. Qin, T. Liu et al., “A language-independent neural network for event detection,” *Science China Information Sciences*, vol. 61, no. 9, pp. 92–106, 2018.
 - [9] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, *Efficient Estimation of Word Representations in Vector Space*, International Conference On Learning Representations, 2013.
 - [10] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” 2018, <https://arxiv.org/abs/1810.04805>.
 - [11] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” *Neural Information Processing Systems*, vol. 1, pp. 5998–6008, 2017.
 - [12] K. Cho, B. Van Merriënboer, C. Gulcehre et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” 2014, <https://arxiv.org/abs/1406.1078>.
 - [13] R. C. Staudemeyer and E. R. Morris, “Understanding LSTM – a tutorial into long short-term memory recurrent neural networks,” 2019, <https://arxiv.org/abs/1909.09586>.
 - [14] J. Lafferty, A. McCallum, F. Pereira et al., “Conditional random fields: probabilistic models for segmenting and labeling sequence data,” *International Conference On Machine Learning*, pp. 282–289, 2001.
 - [15] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” 2014, <https://arxiv.org/abs/1412.6980>.
 - [16] Q. Li, H. Ji, and L. Huang, “Joint event extraction via structured prediction with global features,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 73–82, Sofia, Bulgaria, 2013.
 - [17] Z. Li, Q. Li, X. Zou, and J. Ren, “Causality extraction based on self-attentive bilstm-crf with transferred embeddings,” *Neurocomputing*, vol. 423, pp. 207–219, 2021.
 - [18] S. Bhatt, A. Sheth, V. Shalin, J. Zhao, and A. Sheth, “Knowledge graph semantic enhancement of input data for improving AI,” *IEEE Internet Computing*, vol. 24, no. 2, pp. 66–72, 2020.
 - [19] J. Liu, Y. H. Yang, and H. He, “Multi-level semantic representation enhancement network for relationship extraction,” *Neurocomputing*, vol. 403, pp. 282–293, 2020.