# Chinese Event Extraction Based on Hierarchical Attention Mechanism

Qingmeng Hu and Hongbin Wang(✉)

Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China
whbin2007@126.com

**Abstract.** Event extraction try to extract structured event information from unstructured text. Its major researches can be summarized into three categories: classification methods based on feature learning, methods based on question answering and methods based on Seq2seq. All of these methods are difficult to deal with multi-event sentence. In order to solve this problem, we use hierarchical attention mechanism to treat event extraction as a relation classification task. The BERT and CRF are first used to identify candidate triggers and arguments, and then the hierarchical attention mechanism is used to identify the relationship between trigger and argument. Experiments on both ACE2005 and CEC show that our method outperforms in both trigger classification and argument classification.

**Keywords:** Event Extraction · Relation Classification · Hierarchical Attention Mechanism

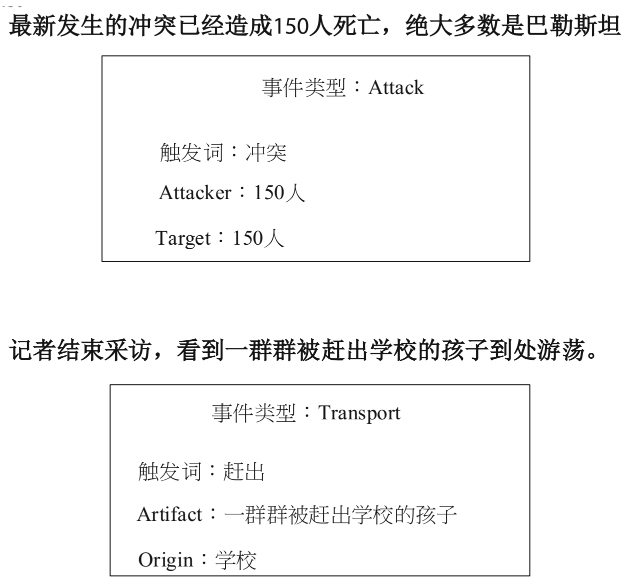
## 1 Introduction

Event extraction is one of the information extraction tasks. It mainly task is to identify event trigger word and arguments from unstructured texts. Ahn [1] first divided event extraction into four tasks: trigger word detection, trigger classification, argument detection, and argument classification. Such as, the sentence “巴沙尔就任总统之职仅两个半月就走出国门展开外交活动”, which contains a “Start-Position” event. The event extraction first needs to identify whether the sentence contains an event. After confirming the inclusion of the event, continue to identify the “就任” as an event trigger word, and classify its type as “Start-Position”. Finally, system should identify the arguments “巴沙尔” and “总统” in the sentence, and classify their role as “Person” and “Position” respectively.

We divide most of the current event extraction models into three categories: classification methods based, question and answer based, Seq2seq based. Classification methods [2–7] usually treat event extraction as a two-stage task. In the first stage, these methods identify trigger words and event types. The second stage is to identify arguments and their roles. These models can be devided into two modules: feature learning module and classification module. With the development of research, more and more features contained by the models are more and more complex. Most of the models ignore the

Y. Sun et al. (Eds.): ChineseCSCW 2022, CCIS 1682, pp. 401–411, 2023.
https://doi.org/10.1007/978-981-99-2385-4_30

overlap and multi-roles problem of argument. Argument overlap and multi-roles argument problem are shown as Fig. 1. Where the roles of argument “150人” are “Attacker” and “Target”. About overlap, “学校” is a part of the argument “一群群被赶出学校的孩子”.



最新发生的冲突已经造成150人死亡，绝大多数是巴勒斯坦人。

事件类型：Attack

触发词：冲突

Attacker：150人

Target：150人

记者结束采访，看到一群群被赶出学校的孩子到处游荡。

事件类型：Transport

触发词：赶出

Artifact：一群群被赶出学校的孩子

Origin：学校

**Fig. 1.** Argument overlap and multi-roles argument

The methods based on question answering [8–11], can design questions for event types and argument roles. These methods extract event information by answer multi-round question for specific event type. Comparing with classification methods, the question answer-based models can solve the problems of argument overlap and multi-roles argument. However, this kind of models is pipelined, which is prone to error propagation. If the trigger word is identified incorrectly, the trigger classification, the argument identify and argument classification will be wrong.

Models based on Seq2seq [12–16], such as [15] proposed a text-to-event (Text2Event) event extraction method, which extracts event information end-to-end. Although the event extraction method based on Seq2seq looks perfect, it eliminates the error propagation and also deal argument overlap and multi-roles argument. However, when the Seq2seq based model generates the target sequence, there is a large amount of information unconcerned needs to be cut, and the generation process is difficult to control. In addition, models based on Seq2seq usually require a large amount of training data, yet there are a few well-labeled event data.

To sum up the above problems, we propose a relatively simple model that treat event extraction as relation classification task, merge four tasks in to one. Our Model classify relation between trigger and argument. As shown in Fig. 2, the method first identifies “出任” as a candidate trigger word, identifies “陈德良” and “国家主席” as candidate arguments, and then classifies the relationship between the trigger words and arguments in the sentence for event extraction. Taking the candidate pair $<$ 出任, 陈德良 $>$ as an example, model identify the relationship between them as “start-position/person”, in which “start-position” is the event type, “person” is the argument role of “陈德良” in the event “start-position”.

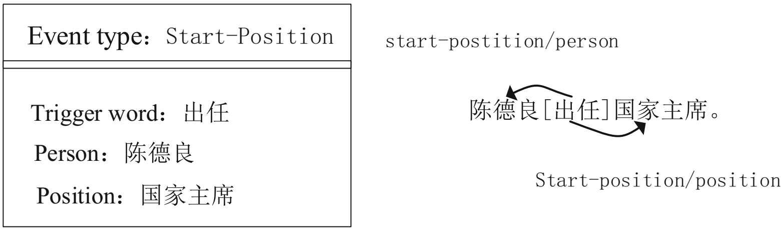


**Fig. 2.** Trigger and argument relationship classification

The advantage of our methods is merging four event extraction tasks into one classification task by a relatively simple way. The model structure is simple and easy to train, and at the same time, it solves the problems of argument overlap and multi-roles argument.

## 2 Model

Our model treats event extraction as a relation classification task. This method classifies the relationship between trigger word and argument, and the relationship type is “event type + argument role”. The method utilizes the information between candidate trigger words and candidate arguments to achieve event extraction, and also tackle multiple events sentence, argument overlap and multi-roles argument.

The idea of our method is shown in Fig. 2. First, identifying candidate trigger words and candidate arguments, and then classifying the relationship between candidate trigger words and candidate arguments. The incorrectly identified candidate trigger words and candidate arguments will be further processed in the following relation classification. If one of trigger word or argument is wrong, and the relationship will be “none”.

Main structure of the model as shown in Fig. 3 includes two parts: bidirectional GRU and hierarchical attention. The hierarchical attention in our model is different to word-level attention and sentence-level attention in HAN [17]. We study in sentence-level event extraction, and trigger words such as “受了伤” and “杀青”, where each character has different importance for event type detection. Therefore, we modified the hierarchical attention mechanism in HAN to word-level attention and character-level attention.

### 2.1 Candidate Trigger Word and Candidate Argument Detection

To identify candidate trigger words and arguments, it is necessary to identify as many trigger words and arguments in the sentence as possible, wrongly identified trigger words

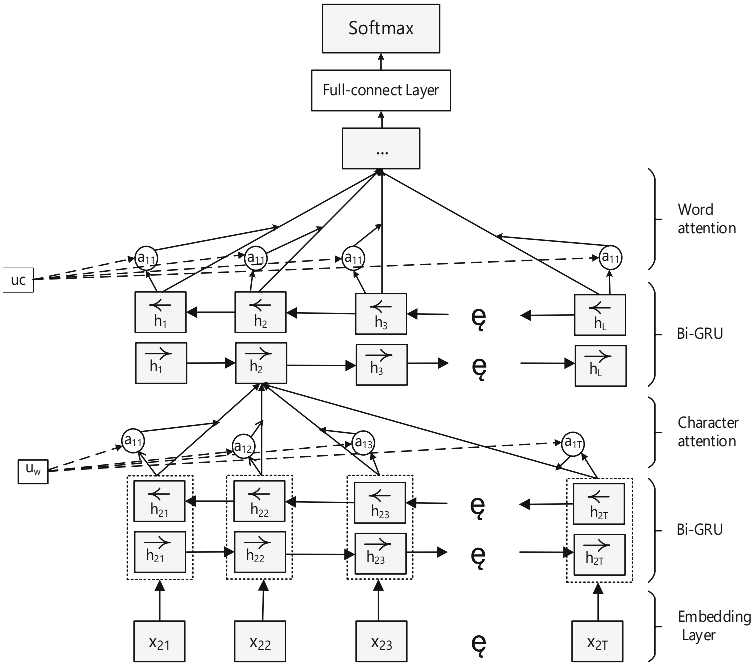


**Fig. 3.** Model structure of this paper

and arguments while be handled during relation classification. We use the BERT and CRF model to treat candidate triggers and event arguments detection as a word-level sequence labeling task.

Assuming that the sentence contains n words, input these n words into the BERT model to learn its features. The features of these words are then fed into the CRF to identify then most likely labeled sequences for the words. When identifying trigger words, we just need to identify whether they are arguments. Therefore, we only use three labels (*B*, *I* and *O*) for the CRF. "*B*" means that the current word is at the begin character of argument, "*I*" means that it is in the middle of argument, and "*O*" means that it is not in argument. The same method is used for candidate trigger word detection.

## 2.2 Model Input

In the hierarchical attention mechanism, we suppose that there are $T$ characters and $L$ words in the input sentence of the model. First input $T$ words, use Embedding Layer to convert the words into embedding vectors. The model obtains the model input vector $X$ by concatenating word embedding, candidate argument and trigger word position embedding. In order to make the model fully learn the forward and backward semantic information in the sentence, compared with the bidirectional LSTM model, we choose the bidirectional GRU as base model which is easier to train.

### 2.3 Hierarchical Attention

Considering that not all character in the trigger word have the same importance to the word, such as the trigger word “受了伤”, it is obvious that the importance of “伤” is much higher than “了”, so in this article we refer to Hierarchical attention mechanism uses character-level attention mechanism. The attention score comes from character hidden and word embedding, as shown in Eq. (2). The representing of word is get by weighted sum the hidden states of these words, as shown in Eq. (3).

$$u_{it} = \tanh(W_w h_{it} + b_w) \tag{1}$$

$$a_{it} = \frac{\exp(u_{it}^T u_w)}{\sum_t \exp(u_{it}^T u_w)} \tag{2}$$

$$w_i = \sum_t a_{it} h_{it} \tag{3}$$

where $W_w$ is the weight matrix of the fully connected layer, $u_w$ which is the word vector containing the character, which $h_{it}$ represents the feature vector obtained by the character $t$ after bidirectional GRU. The word vector is obtained by summing the word vector weighted by the attention score $w_i$. After that, Bi-GRU of the same structure use for learning the hidden state of the words. Word attention scores compute from context vector $u_c$ and word represent by softmax, as shown in Eq. (6). Finally, the word hidden vector is weighted and summed to obtain the sentence hidden vector $s$. The $s$ as the present of candidate trigger words, arguments, and sentence information feed into fully connected layer to obtain the relation label.

$$h_i = Bi - GRU(w_i), t \in [1, \mathrm{L}] \tag{4}$$

$$u_i = \tanh(W_s h_i + b_s) \tag{5}$$

$$a_i = \frac{\exp(u_i^T u_c)}{\sum_i \exp(u_i^T u_c)} \tag{6}$$

$$s = \sum_i a_i h_i \tag{7}$$

### 2.4 Model Output

We regard event extraction as a relationship prediction between candidate trigger words and candidate arguments. The next step is to map the feature vector $s$ into the classification space using a fully connected layer. We construct the relationship between candidate trigger words and candidate arguments as event types add argument roles. There are 33 types of events, 35 types of argument roles, and 223 types of relationships are finally constructed. In order to deal with wrong arguments, it is also necessary to add a “none” relationship label, indicating that the candidate trigger word or the candidate argument is wrong. Therefore, we regard event extraction as a 224-way classification task. In order prompt model learning more event knowledge, when the trigger words are the same, we want the event types same too, so we continue to do event type classification on the feature vector $s$, which is a 34-way classification task (33 event types add none). The relationship classification output is shown in Eq. (9), and the event type classification is shown in Eq. (11).

$$o = \tanh(W_o s + b_o) \tag{8}$$

$$y = softmax(o) \tag{9}$$

$$t = \tanh(W_p s + b_p) \tag{10}$$

$$p = softmax(t) \tag{11}$$

Output y is the score that classify $s$ into various relations, where $W_o$, $W_p$ is matrix, $b_o$, $b_p$ is the bias term, $o \in \boldsymbol{R}^{224}, p \in \boldsymbol{R}^{34}$ are the relational probability of classification and the probability of event type.

### 2.5 Loss Function

We use the cross entropy loss function as model's loss function. The calculation process is shown in Eq. (12), where $y$ is one-hot relationship label between the candidate trigger word and the candidate argument, and $z$ is the event type label, where N = 224, M = 34.

$$Loss = -\sum_{j=1}^{N} y_j \bullet \log o_j - \sum_{j=1}^{M} z_j \bullet \log t_j \tag{12}$$

## 3 Experiment

We evaluate on the public datasets ACE2005 and CEC (Chinese Emergency Corpus). We divide ACE 2005 into 549 texts as the training dataset, 20 as the validation dataset, and 64 as the test dataset. The CEC dataset contains 333 texts, and we randomly divide it into training set, validation set and test set according to the ratio of 7:2:1. In order to verify our method, we selected the following models for comparison:

(1) DMCNN was proposed by Chen et al. [5] in 2015, for multiple events sentence. On the basis of convolutional neural network, dynamic multi-pooling is designed to extract more important features.

(2) Rich-C was proposed by Chen et al. [6]. Based on Lin et al., they mix trigger word context features, dependency features, semantic features, and nearest entity features for event detection.

(3) C-BiLSTM was proposed by Zeng et al. [18] in 2016, combining CNN and LSTM for feature extraction. This method treats event extraction as a word-level sequence label task.

(4) NPNs was proposed by Lin et al. [15] in 2018, in order to solve the problem of trigger word segmentation errors. The model learns the structural and semantic information of words and characters, and the word and character features are mixed for event detection. This method is a classification method based on character features.

(5) JMCEE was proposed by Xu et al. [19] in 2020. This model regards the task of argument extraction as a word-level binary classification task for the case of multiple events in a sentence. Each word uses binary classification to determine whether it is the start word or the end word of the argument.

### 3.1 Experimental Results and Analysis

Experimental results are shown in Table 1 and Table 2. From the experimental results, we can see that compared with DMCNN and C-BiLSTM, the F1 score of our model is improved by 2.3% and 2.1%, respectively. In Table 2, our method also outperforms JMCEE (BERT-pipeline) on the argument classification task, but it is not satisfactory compared with other models. Although the effect is not so good, our model is relatively simple, and the model can deal the problems of argument overlap and multi-roles argument that are difficult to other models. The model did not achieve good results. The main reason is probably that we use BERT + CRF for trigger word and argument detection. Only 62.7% of the trigger words and 59.3% of the arguments in the test set were identified. One of the reasons may be that the argument is not a simple named entity, but also includes time and value expressions. Another reason may be that the number of features labeled in the ACE dataset is too small to train a sufficiently powerful BERT + CRF model.

**Table 1.** Experiment on event detection

| | Trigger detection | | | Trigger classification | | |
|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 |
| DMCNN | 66.6 | 63.6 | 65.1 | 61.6 | 58.8 | 60.2 |
| Rich-C | 62.2 | 71.9 | 66.9 | 58.9 | 68.1 | 63.2 |
| C- BiLSTM | 65.6 | 66.7 | 66.1 | 60.0 | 60.9 | 60.4 |
| NPNs | 75.9 | 61.2 | 67.8 | 73.8 | 59.6 | 65.9 |
| JMCEE(BERT-Pipeline) | 82.5 | 78.0 | 80.2 | 72.6 | 68.2 | 70.3 |
| JMCEE(BERT-Joint) | 84.3 | 80.4 | 82.3 | 76.4 | 71.7 | 74.0 |
| Ours | 63.6 | 71.5 | 67.3 | 59 | 66.4 | 62.5 |

In order to exclude the influence of the low accuracy of BERT and CRF in identifying arguments, verify whether our method is useful for event classification and argument

**Table 2.** Experiment on argument extraction

| | Argument detection | | | Argument classification | | |
|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 |
| Rich-C | 43.6 | 57.3 | 49.5 | 39.2 | 51.6 | 44.6 |
| C- BiLSTM | 53.0 | 52.2 | 52.6 | 47.3 | 46.6 | 46.9 |
| JMCEE(BERT-Pipeline) | 59.5 | 40.4 | 48.1 | 51.9 | 37.5 | 43.6 |
| JMCEE(BERT-Joint) | 66.3 | 45.2 | 53.7 | 53.7 | 46.7 | 50.0 |
| Ours | 62.3 | 38.3 | 47.4 | 58.8 | 36.2 | 44.8 |

role classification. We continued to do the following experiments, constructing pairs of trigger words and arguments as positive samples, and adding some negative samples with a relationship "none" to simulate the situation where the trigger words or arguments were identified incorrectly. The experimental results are shown in Table 3.

**Table 3.** Trigger classification and argument classification

| | Trigger classification | | | Argument classification | | |
|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 |
| DMCNN | 61.6 | 58.8 | 60.2 | 39.2 | 51.6 | 44.6 |
| Rich-C | 58.9 | 68.1 | 63.2 | 47.3 | 46.6 | 46.9 |
| C-BiLSTM | 60.0 | 60.9 | 60.4 | 51.9 | 37.5 | 43.6 |
| NPNs | 73.8 | 59.6 | 65.9 | 53.7 | 46.7 | 50.0 |
| MCEE(BERT-Pipeline) | 72.6 | 68.2 | 70.3 | 39.2 | 51.6 | 44.6 |
| JMCEE(BERT-Joint) | 76.4 | 71.7 | 74.0 | 47.3 | 46.6 | 46.9 |
| ours | 80.4 | 79.5 | 80.0 | 60.7 | 58.0 | 59.3 |

From the experimental results in Table 3, it can be seen that with the improvement of the accuracy of trigger word detection, our model has great advantages in classification tasks such as event type classification and argument role classification. In order to further verify the performance, we further select the following models for experiments on the CEC dataset.

Yang et al. [20] used Bi-LSTM and CRF for the financial field to regard event extraction as a sequence labeling task. Ma et al. [21] proposed BiGRU (Bi-direction Gate Recurrent Unit), which uses an end-to-end approach to simultaneously perform trigger word detection and event type classification to avoid error propagation. This method aims at the shortcomings of DCFEE, Zheng et al. [22] proposed Doc2EDAG (Document to Event Directed Acyclic Graph), which can effectively convert text into a directed acyclic graph of entities for end-to-end event extraction. Transfer [23], this method was proposed by Huang et al. for new event detection. The LEAM [24] model utilizes the information of labels and treats text classification as a joint label-word embedding problem. Yin et al. [25] introduced a residual network to change the network structure to alleviate the problem of gradient disappearance, and proposed Conv-RDBiGRU. The experimental results are shown in Table 4.

**Table 4.** CEC dataset event detection experiment

| Model | P | R | F1 |
|---|---|---|---|
| DCFEE | 68.7 | 70.9 | 69.4 |
| BiGRU | 71.1 | 69.0 | 70.0 |
| Doc2DEAG | 73.5 | 70.3 | 71.9 |
| Transfer | 74.1 | 70.5 | 72.2 |
| Conv-RDBiGRU | 78.8 | 69.3 | 73.8 |
| LEAM | 71.1 | 79.7 | 75.2 |
| ours | 75.9 | 76.1 | 76.0 |

From Table 4, we can see that compared with other models, our model has achieved better results. Which indicates that incorporating both trigger word and argument information into the model is useful. On the other hand, the knowledge learned by our model is more abundant, and it finishes event extraction in one step, which indicates joint model outperform pipeline model. Compared with models such as Doc2EDAG, Transfer, and LEAM, our model is more simple and easier to train.

## 4 Summarize

In this paper, we investigate event extraction researches based on deep learning, and summarize them into three categories: classification methods based on feature learning, methods based on question answering and methods based on Seq2seq. The classification method based on feature learning is difficult to deal with argument overlap and multi-roles argument, the method based on question answering is prone to error propagation, and the method based on Seq2seq is difficult to train. Since the current methods have their own intractable problems, we propose a method for event extraction based on the idea of relation classification. This method inputs candidate trigger words and candidate argument pairs, and the model classifies their relationships to extract event. After experiments on ACE2005 and CEC data, our method performs well on both event type classification and argument classification tasks.

**Acknowledgments.** This work is supported by the National Natural Science Foundation of China under Grants no. 61966020. Moreover, we sincerely thank all reviewers for their valuable comments.

## References

1. Ahn, D. The stages of event extraction. In: Proceedings of the Workshop on Annotating and Reasoning About Time and Events, pp. 1–8 (2006)
2. Zheng, S., Cao, W., Xu, W., et al.: Doc2edag: an end-to-end document-level framework for Chinese financial event extraction. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 337–346 (2019)
3. Chen, Z., Ji, H.; Language specific issue and feature exploration in Chinese event extraction. In: Proceedings of Human Language Technologies: the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, pp. 209–212 (2009)
4. Wang, X., Wang, Z., Han, X., et al.: Hmeae: hierarchical modular argument extraction. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (emnlp-ijcnlp), pp. 5781–5787 (2019)
5. Chen, Y., Xu, L., Liu, K., et al.: 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, pp. 167–176 (2015)
6. Chen, C., Ng, V.: Joint modeling for Chinese event extraction with rich linguistic features. In: Proceedings of Coling 2012, pp. 529–544 (2012)
7. Zhang, J., Qin, Y., Zhang, Y., et al.: Extracting entities and events as a single task using a transition-based neural model. In: IJCAI, pp. 5422–5428 (2019)
8. Chen, Y., Chen, T., Ebner, S., et al.; Reading the manual: event extraction as definition comprehension. In: Proceedings of the Fourth Workshop on Structured Prediction for Nlp, p. 783 (2020)
9. Du, X., Cardie, C.: Event extraction by answering (almost) natural questions. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (emnlp), pp. 671–683 (2020)
10. Liu, J., Chen, Y., Liu, K., et al.: Event extraction as machine reading comprehension. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (emnlp), pp. 1641–1651 (2020)
11. Li, F., Peng. W., Chen, Y., et al.: Event extraction as multi-turn question answering. In: Findings of the Association for Computational Linguistics, pp. 829–838 (2020)
12. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 6000–6010 (2017)
13. Devlin, J., Chang, M., Lee, K., et al.: Bert: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, pp. 4171–4186 (2019)
14. Li, S., Ji, H., Han, J.: Document-level argument extraction by conditional generation. In: Proceedings of the 2020 Conference on Empirical. Methods in Natural Language Processing (emnlp): Association for Computational Linguistics, pp. 671–683 (2021)
15. Lin, H., Lu, Y., Han, X., et al.: Nugget proposal networks for Chinese event detection. arXiv preprint arXiv:1805.00249 (2018)
16. Lin, J., Jian, J., Chen, Q.: Eliciting Knowledge From Language Models for Event Extraction. Arxiv Preprint arxiv:2109.05190 (2021)
17. Paolini, G., Athiwaratkun, B., Krone, J., et al.: Structured prediction as translation between augmented natural languages. In: International Conference on Learning Representations, pp. 1–26 (2021)

18. Zeng, Y., Yang, H., Feng, Y., Wang, Z., Zhao, D.: A convolution BiLSTM neural network model for Chinese event extraction. In: Lin, C.Y., Xue, N., Zhao, D., Huang, X., Feng, Y. (eds) Natural Language Understanding and Intelligent Applications. ICCPOL NLPCC 2016 2016. Lecture Notes in Computer Science, vol. 10102, pp. 275–287. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50496-4_23
19. Xu, N., Xie, H., Zhao, D.: A novel joint framework for multiple Chinese events extraction. In: Sun, M., Li, S., Zhang, Y., Liu, Y., He, S., Rao, G. (eds.) CCL 2020. LNCS (LNAI), vol. 12522, pp. 174–183. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63031-7_13
20. Yang, H., Chen, Y., Liu, K., Xiao, Y., Zhao, J.: DCFEE: a document-level Chinese financial event extraction system based on automatically labeled training data. In: Proceedings of ACL 2018, System Demonstrations, vol. 4, pp. 50–55, Melbourne, Australia (2018)
21. Ma, C., Chen, X., Wang, W.: Chinese event detection based on recurrent neural network. Netinfo Secur. **5**, 75–81 (2018)
22. Zheng, S., Cao, W., Xu, W., Bian, J.: Doc2EDAG: an end-to-end document-level framework for Chinese financial event extraction. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing EMNLP /IJCNLP, vol. 1, pp. 337–346, Hong Kong, China (2019)
23. Huang, L., Ji, H., Cho, K., Dagan, I., Riedel, S., Voss, C.: Zero -shot transfer learning for event extraction. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 2160–2170, Melbourne, Australia (2018)
24. Wang, G., Li, C., Wang, W., et al.: Joint embedding of words and labels for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 2321–2331, Melbourne, Australia (2018)
25. Yin, H., Cao, J., Cao, L., et al.: Chinese emergency event recognition using conv- RDBiGRU model. Comput. Intell. Neurosci. 2020 (2020)