A Prior Information Enhanced Extraction Framework for Document-level Financial Event Extraction

Haitao Wang[1], Tong Zhu[1], Xiaoye Qu[2], Mingtao Wang[1], Guoliang Zhang[1], Wenliang Chen[1], Zhefeng Wang[2]

[1] School of Computer Science and Technology, Soochow University, China
{htwang2019,tzhu7,mtwang96,glzhang}@stu.suda.edu.cn,
wlchen@suda.edu.cn
[2] Huawei Technologies Co., Ltd., China
{quxiaoye,wangzhefeng}@huawei.com

Abstract. Document-level financial event extraction (DFEE) is the task of detecting event and extracting the corresponding event arguments in financial documents, which plays an important role in information extraction in the financial domain. This task is challenging as the financial documents are generally long text and event arguments of one event may be scattered in different sentences. To address this issue, we propose a novel **P**rior **I**nformation **E**nhanced **E**xtraction framework (PIEE) for DFEE, leveraging prior information from both event types and pre-trained language models. Specifically, PIEE consists of three components: event detection, event arguments extraction, and event table filling. In event detection, we identify the event type. Then, the event type is explicitly used for event argument extraction. Meanwhile, the implicit information within language models also provides considerable cues for event arguments localization. Finally, all the event arguments are filled in a event table by a set of predefined heuristic rules. To demonstrate the effectiveness of our proposed framework, we participate the share task of CCKS2020 Task5-2 : Document-level Event Arguments Extraction. On both Leaderboard A and Leaderboard B, PIEE takes the first place and significantly outperforms the other systems.

Keywords: Event Extraction · Information Extraction· Financial Event.

1 Introduction

Event Extraction (EE) aims to identify the event type events and their corresponding arguments in text. In the financial domain, EE provides valuable structured information for investment analysis and asset management. To promote financial event extraction, 14th China Conference on Knowledge Graph and Semantic Computing (CCKS2020) sets Task 5-2 for document-level financial event extraction (DFEE). The organizer collects documents from financial news and announcements, and requires the participants to identify the event types and extract event arguments from the documents.

In recent years, event extraction has attracted increasing attention due to its vast application and significant efforts have been devoted to it. However, most existing studies merely extract arguments within the sentence scope [2, 14, 15], dubbed as sentence-level EE (SEE). For document-level EE, these methods provide sub-optimal solutions

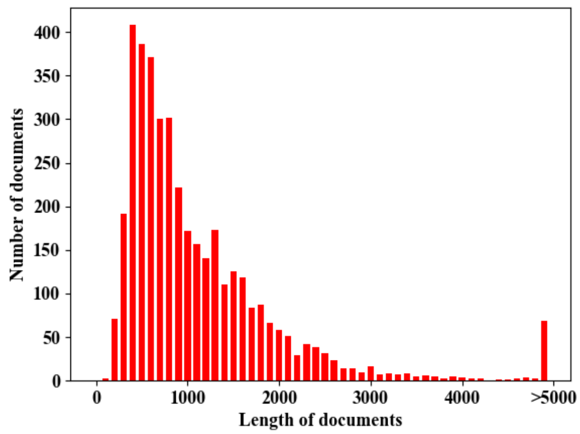

Fig. 1. The text length distribution of data in CCKS2020 Task 5-2.

because the event arguments are often scattered across different sentences in a document. As shown in Fig. 1, most texts contain more than 500 Chinese characters. Under this circumstance, independently processing each sentence in the document destroys the integrity of events. Therefore, a document-level EE framework is vital to extract events from such long documents.

In this paper, we propose **P**rior **I**nformation **E**nhanced **E**xtraction framework (PIEE) for document-level financial event extraction, which can be decomposed into three steps: event detection, event arguments extraction, and event table filling. Specifically, event detection first identifies event type of the document. Then, we utilize the event type as a prior information for sentence-level event arguments extraction. In this paper, we explore three paradigms for event arguments extraction. With prior type information, all the three paradigms obtain consistent performance improvement. Moreover, inspired by the recent success of pre-trained language model (PLM) which is trained on large corpus and provides implicit prior information, we explore different language models for event arguments extraction. Finally, event table filling integrates all event arguments extracted from different sentences by a set of heuristic rules.

In summary, our contributions of this paper are as follows:

- We propose a novel prior information enhanced extraction framework (PIEE) for document-level financial event extraction, which is comprised of three steps: event detection, event arguments extraction and event table filling.
- We utilize event type as explicit prior information for sentence-level event arguments extraction. Meanwhile, we explore the implicit prior information in different language models for event arguments extraction.
- In CCKS2020 Task 5-2, our system achieves 0.83007 F1-score on the Leaderboard A and 0.66996 F1-score on the Leaderboard B, both ranking the first place.

2 Related Work

Event extraction has achieved great progress in recent years. However, most research [11, 26, 23] focus on sentence-level event extraction (SEE), and document-level event extraction (DEE) is less concerned. Yang et al. [22] and Zheng et al. [27] propose two different frameworks for DEE. The former method extracts event arguments in the form of SEE and combines the results of SEE into DEE by a key event detection and arguments-completion strategy. The latter one establishes an end-to-end framework Doc2EDAG based on multiple transformer models and exploits an entity-based directed acyclic graph to implement the DEE effectively. In the stage of event arguments extraction, both of them regard it as a sequence labeling problem similar to NER, where BiLSTM-CRF [6] is a classic model to address this issue. Beyond that, with the successful application of machine reading comprehension (MRC) in many NLP problems [7, 9], MRC is also used to NER task with the advantage of significant prior information of the entity category. Recently, Yu et al. [24] apply biaffine model to NER task and achieve the state-of-the-art performance on eight corpora.

In addition, compared to GloVe [16] and ELMo [17], recent language model BERT can capture more contextual and semantic information from texts. To mitigate the drawbacks of masking strategy in BERT, BERT-wwm [3] uses the Whole Word Masking (WWM) and ERNIE [19] designs entity-level strategy and phrase-level strategy to integrate external knowledge. RoBERTa [12] further proposes the dynamic masking strategy and removes the next sentence prediction task. Relative positional encoding is also employed in NEZHA [21] to enhance the encoding ability.

Inspired by the above works, we propose a prior information enhanced extraction framework for document-level financial event extraction. In contrast to DCFEE and Doc2EDAG, we first discover events in texts, which helps identify the event arguments in subsequent stages. To improve the event argument extraction performance, advanced technologies in NER and recent language models are also introduced in our model.

3 Data

This section presents data analysis and describes how to preprocess data.

3.1 Data Analysis

In order to have a comprehensive understanding of the data in the shared task, we list statistical information. Fig. 2 presents the co-occurrence distribution of different event types in the training data, including Bankruptcy Liquidation (BL), Equity Freeze (EF), Equity Underweight (EU), Equity Overweight (EO), Equity Pledge (EP), Asset Loss (AL), Accident (AC), Leader Death (LD), and External Indemnity (EI). We can conclude that all the events in one document share the same even type. This observation greatly simplifies the process of event type identification.

Fig. 3 further shows the distribution histogram of the number of events and instances in each event type. It can be observed that the event types are divided into two

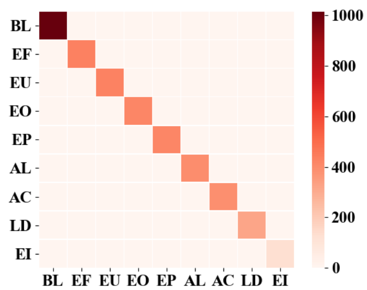

Fig. 2. Co-occurrence distribution of event types in training data.

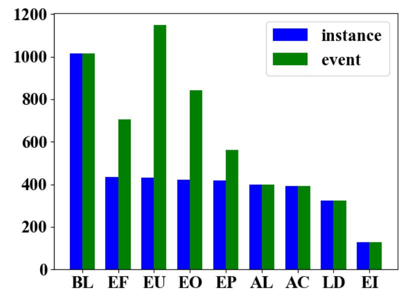

Fig. 3. Number of events and instances in each event type.

categories: one is that the event occurs only once in the document like *Bankruptcy Liquidation*, and the other is that the event can occur more than once in the same document such as *Equity Pledge*. This fact also contributes to subsequent event table filling.

In summary, we can conclude the following two corollaries:

Corollary 1. *Each document contains only one type of event.*

Corollary 2. *There is only one event in the document which describes BL, AL, AC, LD and EI, and documents introducing EU, EO, EF and EP usually contain more than one event.*

3.2 Data Preprocessing

The data of this evaluation task mainly comes from financial announcements and news on the Internet. Inevitably, there are noises in the crawled texts. Thus, it is necessary to clean the data for better building system.

Table 1. Escape symbols and tags of HTML in the evaluation data.

	&quot	&apos	&amp	&gt	&lt	
\s	″	′	&	>	<	\n

As shown in Table 1, the original data contains the escape symbols and tags of HTML, which hinders the system's semantic understanding of texts. We restore them except $
$, which is specially replaced with a single space considering that $\backslash n$ is a special flag when splitting the document.

Moreover, in order to minimize the length of the text as possible, the continuous repeated punctuation, extra spaces and web links are removed. We also convert traditional texts into simplified texts, and convert punctuation from SBC case to DBC case to

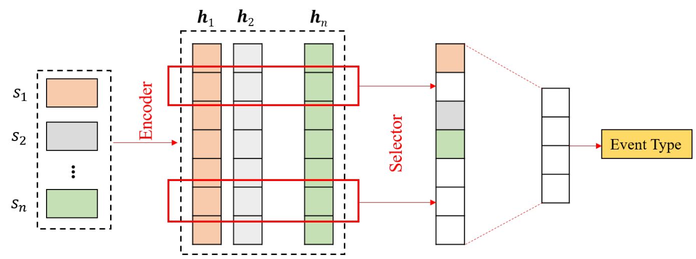

Fig. 4. The architecture of event detection.

construct more standardized data. Finally, all documents are divided into multiple sentences with a maximum length of 500 and event arguments in the sentence are tagged with BIO (Begin, Inside, Other) scheme in the training data.

4 Methodology

In this section, we will introduce the details in our proposed framework. First of all, we need detect which event types are described in the documents. Then, we treat event arguments extraction as a sequence labeling problem. At last, some heuristic strategies are applied to fill in the event tables.

4.1 Event Detection

Inspired by the assumption of at-least-one-sentence [18] in distantly supervised relation extraction, we also assume: if a document contains some type of event type, there is at least one sentence from this document can fully describe that event type. Thus, each document can be considered as a sentence bag.

Fig. 4 shows the architecture of event detection. Sentences from the same document $\{s_1, s_2, ..., s_n\}$ are first transformed into distributed representations by looking up the pre-trained char embeddings [3]. Then, sentence encoder such as CNN and LSTM is applied to extract deep semantic features $\{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_n\}$ for text classification. Similar to the research in relation extraction, sentences from the same document are regarded as one bag, and we can use the following strategies to represent a document $\boldsymbol{d}$:

ONE Zeng et al. [25] select the most valuable sentence to represent the whole sentence bag $\boldsymbol{d}$ and the highest probability sentence is defined as follows:

$$\begin{aligned} \boldsymbol{o}_i &= \boldsymbol{W}\boldsymbol{h}_i + \boldsymbol{b} \\ j^* &= \underset{j}{\operatorname{argmax}} \frac{\exp(\boldsymbol{o}_i)}{\sum_k \exp(\boldsymbol{o}_k)} \\ \boldsymbol{d} &= \boldsymbol{h}_{j^*} \end{aligned} \tag{1}$$

where $\boldsymbol{W} \in \mathbb{R}^{n \times h}$, n is the number of event types and h is the size of hidden units.

[3] https://github.com/Embedding/Chinese-Word-Vectors

ATT Following Lin et al. [10], to exploit the information of all available sentences, we can use attention mechanism to aggregate sentence-level features. The score a_i measuring how well the input sentence s_i and the target event type e matches can be obtained by the following formula:

$$a_i = \boldsymbol{h}_i \boldsymbol{A} \boldsymbol{r}_e \tag{2}$$

where $\boldsymbol{A}$ is a weighted diagonal matrix, and $\boldsymbol{r}_e$ is the representation of event type e.

Then, the representation of the document $\boldsymbol{d}$ is computed as a weighted sum of sentence-level features:

$$\boldsymbol{d} = \sum_i \frac{\exp(a_i)}{\sum_k \exp(a_k)} \boldsymbol{h}_i \tag{3}$$

MAX Jiang et al. [5] claim that critical information can be also inferred implicitly from all sentences, so a max pooling operation is employed to capture the most valuable features in various aspects from all sentences. Formally, the document-level feature $\boldsymbol{d}$ is computed as follows:

$$\boldsymbol{d} = \max(\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_n) \tag{4}$$

Finally, event type is predicted by the representation of document $\boldsymbol{d}$ and cross-entropy is used as the objective function to optimize the models.

4.2 Event Arguments Extraction

For event arguments extraction, many classic methods of sequence labelling task can be used to extract event arguments in texts. In order to make full use of prior information of event type, we concatenate sentences and the representation of the corresponding event type before encoding. Thus, all sentences from the same document share the same event type predicted by event detection. Based on such input representation, we propose three PLM-based architectures for sentence-level event arguments extraction: PLM-CRF, PLM-MRC, PLM-Biaffine.

PLM-CRF BiLSTM-CRF is a classic model to address the NER task and has once achieved the state-of-the-art result in accuracy. Since pre-trained language models like BERT can capture deeper semantic and contextual information, in our PLM-CRF, the input sequence of PLM consists of event type and sentence. With the help of multiple layers of transformers in PLM, sentence can make full interaction with prior information.

Given the output of PLM $\{\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_m, \boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$, where $\boldsymbol{r}_i$ is the output of event type and $\boldsymbol{x}_i$ is the output of sentence, $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$ is then used as the input of CRF layer. For a sequence of predictions $\boldsymbol{y} = \{y_1, y_2, ..., y_n\}$, we define the score of it as:

$$s(\boldsymbol{X}, \boldsymbol{y}) = \sum_{i=0}^{n} \boldsymbol{A}_{y_i, y_{i+1}} + \sum_{i=1}^{n} (\boldsymbol{W}\boldsymbol{X})^{\top}_{i, y_i} \tag{5}$$

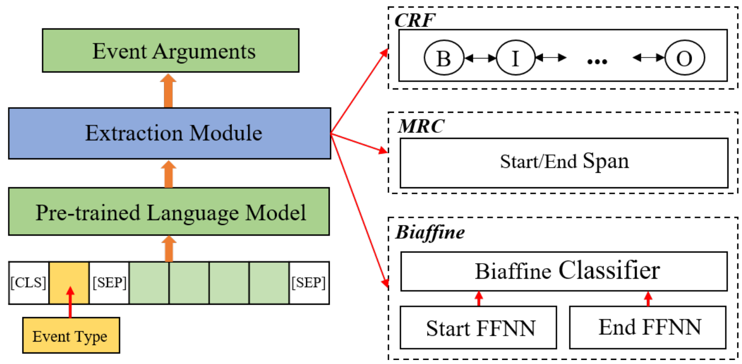

Fig. 5. The architecture of event arguments extraction.

where $\boldsymbol{A} \in \mathbb{R}^{(n_t+2)\times(n_t+2)}$ is a matrix of transition scores and $\boldsymbol{W} \in \mathbb{R}^{n_t \times h}$ is used to calculate the scores of each label for each token, n_t is the number of BIO tags and h is the hidden size of PLM.

During training, we maximize the log-probability of the correct tag sequence. In the testing stage, we use Viterbi algorithm to decode the sequence.

PLM-MRC At present, many NLP tasks can be converted into machine reading comprehension (MRC) problems, and inspired by Li et al. [8], we propose a simplified version of MRC to address event arguments extraction.

First of all, we manually construct some queries for event roles in different event types. For example, for *Pledgor* in *Equity Pledge*, the corresponding query is "*who is the pledgor in equity pledge*". Similar to the operation in PLM-CRF, we also concatenate the query and sentence before PLM encoding.

Then, given the representation of sentence $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$ output from the BERT, we can compute the probabilities of each token being a start index and an end index respectively as follows:

$$\begin{aligned} P_s &= softmax(\boldsymbol{W}_s \boldsymbol{X} + \boldsymbol{b}_s) \\ P_e &= softmax(\boldsymbol{W}_e \boldsymbol{X} + \boldsymbol{b}_e) \end{aligned} \tag{6}$$

where $\boldsymbol{W}_s \in \mathbb{R}^{h\times 2}$ and $\boldsymbol{W}_e \in \mathbb{R}^{h\times 2}$, h is the hidden size of PLM.

In the prediction stage, all valid combinations for a start index and an end index are regarded as the span of event arguments, where there are no other start/end indices between them.

PLM-Biaffine The biaffine model is widely used in dependency parsing [4] and Yu et al. [24] first applies this architecture to address the NER task. Following their work, we also use biaffine model to extract event arguments in texts.

Same as the operation in PLM-CRF, we first obtain the sentence representation $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$ from PLM. After that, two feedforward neural networks (FFNN) are used to generate the representations for the start/end of the spans. Then a biaffine model is applied to predict possible event roles for each span, including a special role named as NA, which means that the current span is not an valid event argument. Specifically, the score of event role for span $<i, j>$ is computed as follows:

$$\begin{aligned}
\boldsymbol{h}_s^i &= \boldsymbol{W}_s \boldsymbol{x}_i + \boldsymbol{b}_s \\
\boldsymbol{h}_e^j &= \boldsymbol{W}_e \boldsymbol{x}_j + \boldsymbol{b}_e \\
s(i,j) &= {\boldsymbol{h}_s^i}^\top \boldsymbol{U} \boldsymbol{h}_e^j + \boldsymbol{W}(\boldsymbol{h}_s^i \oplus \boldsymbol{h}_e^j) + \boldsymbol{b}
\end{aligned} \tag{7}$$

where $\boldsymbol{h}_s^i$ and $\boldsymbol{h}_e^j$ are the start/end representation of token i and j, $s(i,j)$ is the score distribution for span $<i, j>$ among n_r event roles. $\boldsymbol{W}_s \in \mathbb{R}^{h \times d}$, $\boldsymbol{W}_e \in \mathbb{R}^{h \times d}$, $\boldsymbol{U} \in \mathbb{R}^{d \times n_r \times d}$, $\boldsymbol{W} \in \mathbb{R}^{2d \times n_r}$ are trainable parameters in the biaffine model.

When decoding, the event role of each span is the one of highest score and we rank all non-NA spans by their category scores in a descending order. Entities in the sentence are regarded as event arguments only if its span does not clash the boundaries of higher ranked entities, or there is no inclusive relation between higher ranked entities and it.

4.3 Event Table Filling

After obtaining the event types and event arguments in the document, we design some heuristic strategies to convert the results of SEE to DEE. According to Corollary 2 in Section 3.1, all event types can be divided into two categories: one type one event (OTOE) and one type multiple events (OTME).

In the training data, events in OTOE always appear in the plain texts. The combination of valid event arguments with minimum internal distance [4] is selected as the event in document. *Leader Death* is a special event type in OTOE since it is obvious to find event triggers in the sentences, such as “去世”, “逝世”, “辞世” (*all mean pass away*) and so on. The distance between triggers and event arguments is also considered while computing the internal distance.

In OTME scenario, events mainly appear in the table. Thus, we first tend to use keywords, such as “本次增持股票数量(万股)” (*number of equity overweight*), to locate the table, and parse table content with the help of regular expressions and event arguments extracted by models. If no event is found by table parsing, events are generated by the same methods in OTOE.

Additionally, there are some universal strategies. For example, we compare the longest common sequence (LCS) to determine whether a company name is a full name or an abbreviation. To reserve the special token (is mostly $
$) in the final answer, we check all answers which contain space and do not appear in the original text, and restore them to their original form.

5 Experiments

In this section, we present the experimental results on the evaluation data, and make detailed analysis. We compare different variants in event detection and event arguments extraction mentioned in Section 4.

[4] We define the internal distance as sum of distances between all event arguments.

5.1 Dataset and Experimental Setup

Experiments are conducted on CCKS2020 Task 5-2 dataset. This dataset contains 9 event types. In the training data, there are 3,956 documents containing 5,521 events, which are annotated by distant supervision [13, 1]. Validation data and testing data are used for online evaluation on the Leaderboard A and Leaderboard B. They contain 750 documents and 28,096 documents respectively. In order to achieve better robustness and anti-noise capability, we use a 5-fold cross-validation to train each model.

In the experiments of event detection, we use Adam to optimize parameters with a learning rate of 0.001 and a minibatch size of 32. The hidden size of BiLSTM and CNN are both 256. While extracting event arguments, the learning rate is set to 2e-5 in BERT layers and 2e-4 in other layers. The maximum epoch of PLM-CRF, PLM-MRC and PLM-Biaffine is separately 5, 3 and 5. In particular, the output size of FFNNs are both 256 in PLM-Biaffine.

5.2 Experimental Results of Event Detection

Table 2 shows the results of different models mentioned in Section 4.1. It is obvious that MAX-based models achieve the highest accuracy as MAX can capture the most valuable information from all sentences in the document. On the other hand, since predictive features could be diluted by noises in the document, ATT is not as good as MAX. Among three strategies, ONE shows the worst performance both in CNN-based models and BiLSTM-based models, which means that it is not enough to use the information of a single sentence to represent the full text in text classification.

Table 2. Different models for event detection.

	CNN	BiLSTM
ONE	0.97524	0.94045
ATT	0.98233	0.97251
MAX	**0.98560**	**0.98988**

5.3 Experimental Results of Event Arguments Extraction

For three paradigms of event arguments extraction, we all use BERT-wwm-chinese as pre-trained language model. As shown in Table 3, it is obvious that models using prior information of event types always perform better, which shows it is necessary to detect event type before event arguments extraction.

Among all models, although PLM-MRC yields the best performance, PLM-Biaffine still achieves similar results, and has enormous advantage of training speed. Thus, we select PLM-Biaffine as the basic model and further explore different PLMs in order to make full use of implicitly prior information within PLMs. From Table 4, we can observe NEZHA-large performs best, which directly leads to that we just use the combination of NEZHA-large and PLM-Biaffine (NEZHA-Biaffine) in the final competition.

Table 3. Different model variants for event arguments extraction.

Models	F1-score	Training Time/Epoch
PLM-CRF †	0.82503	31 min
PLM-CRF	0.84033	31 min
PLM-MRC †	0.00000	63 min
PLM-MRC	**0.84777**	63 min
PLM-Biaffine †	0.82691	**18 min**
PLM-Biaffine	0.84772	**18 min**

† means no prior event type information is utilized.

Table 4. Different PLMs for PLM-Biaffine.

PLM	F1-score
BERT-base	0.84615
BERT-wwm	0.84772
BERT-wwm-ext	0.84977
ERNIE	0.84298
RoBERTa-wwm-ext	0.85546
RoBERTa-wwm-ext-large	0.86533
NEZHA-large	**0.86693**

5.4 Online Results

According to the above experimental results, BiLSTM+MAX and NEZHA-Biaffine are selected as our final models. The detailed results are listed in Table 5, and it shows that our model is extremely effective. Moreover, since the online result of *Bankruptcy Liquidation*, *Asset Loss*, *Accident*, *Leader Death* and *External Indemnity* are always 0 on the final testing data, we train the new model on the data of rest event types again, which increases the results from 0.66247 to 0.66996.

Table 5. Top 5 Teams on the Leaderborad A and Leaderborad B.

Leaderborad A		Leaderborad B	
Teams	F1-score	Teams	F1-score
SUDA-HUAWEI	**0.83007**	**SUDA-HUAWEI**	**0.66996**
同花顺	0.81411	mulan	0.65043
ztjerry	0.80578	uloveqian	0.63469
mulan	0.78422	同花顺	0.61530
FreeWings	0.78359	LTF_	0.60464

6 Conclusion

In this paper, we propose a prior information enhanced extraction framework for document-level financial event extraction, which consists of three components: event detection, event arguments extraction and event table filling. In our solution, we show that it is necessary to detect event types first in DEE, which is helpful to extract event arguments as an explicit prior information. Moreover, we explore the implicit prior information of different PLMs in event arguments extraction. For Document-level Event Arguments Extraction in CCKS2020 Task 5-2, our system achieves 0.83007 F1-score and 0.66996 F1-score on the leaderboard A and leaderboard B respectively, which are both the highest scores, showing the advantages of our framework.

References

1. Chen, Y., Liu, S., Zhang, X., Liu, K., Zhao, J.: Automatically labeled data generation for large scale event extraction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 409–419 (2017)
2. 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 (Volume 1: Long Papers). pp. 167–176 (2015)
3. Cui, Y., Che, W., Liu, T., Qin, B., Yang, Z., Wang, S., Hu, G.: Pre-training with whole word masking for chinese bert. arXiv preprint arXiv:1906.08101 (2019)
4. Dozat, T., Manning, C.D.: Deep biaffine attention for neural dependency parsing. arXiv preprint arXiv:1611.01734 (2016)
5. Jiang, X., Wang, Q., Li, P., Wang, B.: Relation extraction with multi-instance multi-label convolutional neural networks. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 1471–1480 (2016)
6. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 260–270 (2016)
7. Levy, O., Seo, M., Choi, E., Zettlemoyer, L.: Zero-shot relation extraction via reading comprehension. In: Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017). pp. 333–342 (2017)
8. Li, X., Feng, J., Meng, Y., Han, Q., Wu, F., Li, J.: A unified MRC framework for named entity recognition. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 5849–5859 (2020)
9. Li, X., Yin, F., Sun, Z., Li, X., Yuan, A., Chai, D., Zhou, M., Li, J.: Entity-relation extraction as multi-turn question answering. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 1340–1350 (2019)
10. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2124–2133 (2016)
11. Liu, X., Luo, Z., Huang, H.Y.: Jointly multiple events extraction via attention-based graph information aggregation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 1247–1256 (2018)

12. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
13. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2. pp. 1003–1011. Association for Computational Linguistics (2009)
14. Nguyen, T.H., Cho, K., Grishman, R.: Joint event extraction via recurrent neural networks. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 300–309 (2016)
15. Nguyen, T.M., Nguyen, T.H.: One for all: Neural joint modeling of entities and events. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 6851–6858 (2019)
16. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
17. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 2227–2237 (2018)
18. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 148–163. Springer (2010)
19. Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., Wu, H.: Ernie: Enhanced representation through knowledge integration. arXiv preprint arXiv:1904.09223 (2019)
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
21. Wei, J., Ren, X., Li, X., Huang, W., Liao, Y., Wang, Y., Lin, J., Jiang, X., Chen, X., Liu, Q.: Nezha: Neural contextualized representation for chinese language understanding. arXiv preprint arXiv:1909.00204 (2019)
22. 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. pp. 50–55 (2018)
23. Yang, S., Feng, D., Qiao, L., Kan, Z., Li, D.: Exploring pre-trained language models for event extraction and generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 5284–5294 (2019)
24. Yu, J., Bohnet, B., Poesio, M.: Named entity recognition as dependency parsing. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 6470–6476 (2020)
25. Zeng, D., Liu, K., Chen, Y., Zhao, J.: Distant supervision for relaftion extraction via piecewise convolutional neural networks. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1753–1762 (2015)
26. Zhang, T., Ji, H.: Event extraction with generative adversarial imitation learning. arXiv preprint arXiv:1804.07881 (2018)
27. 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). pp. 337–346 (2019)