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

Authors: Haitao Wang, Tong, Zhu, Mingtao, Wang, Guoliang Zhang, Wenliang, Chen, Wenliang Chen

Date: 2022-11-27T00:00:00+00:00

Abstract

Document-level financial event extraction (DFEE) is the task of detecting events 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 proposed a novel Prior Information Enhanced Extraction 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 argument 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 an event table by a set of predefined heuristic rules. To demonstrate the effectiveness of our proposed framework, we participated in the share task of CCKS2020 Task 4-2: Document-level Event Arguments Extraction. On both Leaderboard A and Leaderboard B, PIEE took the first place and significantly outperformed the other systems.

Full Text

Preamble

DATA PAPER

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

Haitao Wang, Tong Zhu, Mingtao Wang, Guoliang Zhang & Wenliang Chen†
School of Computer Science and Technology, Soochow University, Suzhou 215006, China

Keywords: Event extraction; Information extraction; Financial event; Event detection; Event argument extraction

Citation: Wang, H.T., et al.: A prior information enhanced extraction framework for document-level financial event extraction. *Data Intelligence* 3(3), 460-476 (2021). doi: 10.1162/dint_a_{00103}

Received: February 11, 2021; **Revised:** April 8, 2021; **Accepted:** May 18, 2021

ABSTRACT

Document-level financial event extraction (DFEE) is the task of detecting events 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 financial documents are generally long and event arguments for a single event may be scattered across different sentences. To address this issue, we propose a novel Prior Information Enhanced Extraction 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 argument extraction, and event table filling. In event detection, we identify the event type, which is then explicitly used for event argument extraction. Meanwhile, the implicit information within language models also provides considerable cues for localizing event arguments. Finally, all extracted event arguments are filled into an event table using a set of predefined heuristic rules. To demonstrate the effectiveness of our proposed framework, we participated in the shared task of CCKS2020 Task 4-2: Document-level Event Arguments Extraction. On both Leaderboard A and Leaderboard B, PIEE achieved first place and significantly outperformed all other systems.

1. INTRODUCTION

Event Extraction (EE) aims to identify different types of 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, the 14th China Conference on Knowledge Graph and Semantic Computing (CCKS2020) established Task 4-2 for document-level financial event extraction (DFEE). The organizers collected documents from financial news and announcements, requiring participants to identify event types and extract event arguments from these documents.

In recent years, event extraction has attracted increasing attention due to its vast applications, and significant efforts have been devoted to this task. However, most existing studies merely extract arguments within sentence scope [1, 2, 3], dubbed sentence-level EE (SEE). For document-level EE, these methods provide suboptimal solutions because event arguments are often scattered across different sentences in a document, and global information should be exploited to enhance the model. As shown in Figure 1 [Figure 1: see original paper], most text data contain more than 500 Chinese characters. Under these circumstances, independently processing each sentence in the document destroys the integrity of events. Therefore, a document-level EE framework is vital for extracting events from such long documents.

In this paper, we propose a Prior Information Enhanced Extraction framework (PIEE) for document-level financial event extraction, which can be decomposed into three steps: event detection, event argument extraction, and event table filling. Specifically, in event detection we first identify the event type of the document. Then, we utilize the event type as prior information for sentence-level event argument extraction. In this paper, we explore three paradigms for event argument extraction. With prior type information, all three paradigms achieve consistent performance improvements. Moreover, inspired by the recent success of pre-trained language models (PLMs) which are trained on large corpora and provide implicit prior information, we explore different language models for event argument extraction. Finally, event table filling integrates all event arguments extracted from different sentences using a set of heuristic rules.

In summary, our contributions are as follows:

- We propose a novel prior information enhanced extraction framework (PIEE) for document-level financial event extraction, comprising three steps: event detection, event argument extraction, and event table filling.
- We utilize event type as explicit prior information for sentence-level event argument extraction. Meanwhile, we explore the implicit prior information in different language models for event argument extraction.
- In CCKS2020 Task 4-2, our system achieved 0.83007 F1-score on Leaderboard A and 0.66996 F1-score on Leaderboard B, both ranking first place.

2. RELATED WORK

Event extraction has achieved great progress in recent years. However, most research [4, 5, 6] focused on sentence-level event extraction (SEE), while document-level event extraction (DEE) received less attention. Yang et al. [7] and Zheng et al. [8] proposed two different frameworks for DEE. The former method (DCFEE) extracts event arguments in the form of SEE and combines the results into DEE through a key event detection and arguments-completion strategy, which depends on event triggers. The latter establishes an end-to-end framework Doc2EDAG based on multiple transformer models and exploits an entity-based directed acyclic graph to implement DEE without any elaborately designed rules. However, Doc2EDAG also faces problems such as complex structure, low efficiency, and large resource consumption.

In the event argument extraction stage, both approaches regard it as a sequence labeling problem similar to NER, where BiLSTM-CRF [9] is a classic model for this issue. Beyond that, with the successful application of machine reading comprehension (MRC) in many NLP problems [10, 11], MRC is also used in NER tasks with the advantage of significant prior information about entity categories. Recently, Yu et al. [12] applied the Biaffine model to NER tasks and achieved state-of-the-art performance on eight corpora.

In addition, compared to GloVe [13] and ELMo [14], the recent language model BERT can capture more contextual and semantic information from texts. To mitigate the drawbacks of masking strategies in BERT, BERT-wwm [15] uses Whole Word Masking (WWM) and ERNIE [16] designs entity-level and phrase-level strategies to integrate external knowledge. RoBERTa [17] further proposes dynamic masking strategy and removes the next sentence prediction task. Relative positional encoding is also employed in NEZHA [18] to enhance encoding ability.

Inspired by the above work, 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 event arguments in subsequent stages. To improve the performance of event argument extraction, advanced technologies in NER and recent language models are also introduced in our model. Furthermore, from a structural perspective, our framework is simpler and faster, and event triggers are not necessary in PIEE.

3. DATA

This section presents data analysis and describes the preprocessing methodology.

3.1 Data Analysis

To comprehensively understand the data in the shared task, we analyzed statistical information. Figure 2 [Figure 2: see original paper] 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 events in one document share the same event type. This observation greatly simplifies the process of event type identification.

Figure 3 [Figure 3: see original paper] further shows the distribution histogram of the number of documents and instances for each event type. It can be observed that event types are divided into two categories: one where the event occurs only once in the document (like Bankruptcy Liquidation), and another where multiple events can occur in the same document (such as Equity Pledge). This fact also contributes to subsequent event table filling.

In summary, we can draw the following two conclusions: * Each document contains only one type of event. * There is only one event in documents describing BL, AL, AC, LD, and EI, while documents introducing EU, EO, EF, and EP usually contain more than one event.

3.2 Data Preprocessing

The data for this evaluation task mainly come from financial announcements and news on the Internet. Inevitably, there is noise in the crawled texts. Thus, it is necessary to clean the data for better system construction.

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

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

Original	Restored
	space
&quot	"
&apos	'

Moreover, to minimize text length as much as possible, continuous repeated punctuation, extra spaces, and web links are removed. We also convert traditional Chinese texts into simplified Chinese and convert punctuation from SBC case to DBC case to construct more standardized data. Finally, all documents are divided into multiple sentences with a maximum length of 500 Chinese characters, and event arguments in the sentence are tagged with the BIO (Begin, Inside, Other) scheme in the training data.

4. METHODOLOGY

This section introduces the details of our proposed framework. First, we need to detect which event types are described in the documents. Then, we treat event argument extraction as a sequence labeling problem. Finally, some heuristic strategies are applied to fill in the event tables.

4.1 Event Detection

In research on distantly supervised relation extraction, Riedel et al. [19] assumed: If two entities have a relation, at least one sentence can express that relation among all sentences containing those two entities. Inspired by this classical assumption, we also assume: If a document contains an event type, at least one sentence from this document can fully describe that event type.

In previous event extraction research, event triggers are often used to recognize the event type. However, no trigger words are explicitly provided in real scenarios. We assume that in documents describing an event, there is at least one implicit trigger word, and the sentence where the trigger word is located must be able to identify the event type described in the document. Under this assumption, each document can be considered as a sentence bag.

Figure 4 [Figure 4: see original paper] shows the architecture of event detection. Sentences from the same document {s_1, s_2, ⋯, s } are first transformed into distributed representations by looking up pre-trained character embeddings. Then, a sentence encoder such as CNN or LSTM is applied to extract deep semantic features {h_1, h_2, ⋯, h } for text classification. Similar to research in relation extraction, sentences from the same document are regarded as one bag, and there are three strategies to represent a document d: ONE (at least one sentence), ATT (selective attention over sentences), and MAX (cross-sentence max pooling).

4.1.1 ONE Zeng et al. [20] selected the most valuable sentence to represent the whole sentence bag d, and the highest probability sentence is defined as follows:

$$o = \underset{i}{\operatorname{argmax}} W_h h_i$$

where n_e is the number of event types and h_l is the size of hidden units.

4.1.2 ATT Following Lin et al. [21], to exploit information from all available sentences, we use the 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 match can be obtained by:

$$a_i = h_i^T W_a r_e$$

where W_a is a weighted diagonal matrix, and r_e is the representation of event type e.

Then, the representation of document d is computed as a weighted sum of sentence-level features:

$$d = \sum_{i=1}^{n} a_i h_i$$

4.1.3 MAX Jiang et al. [22] claimed that critical information can also be 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 d is computed as:

$$d_j = \max_{i=1}^{n} h_{i,j}$$

Finally, the event type is predicted from the representation of document d, and cross-entropy is used as the objective function to optimize the models.

4.2 Event Argument Extraction

For event argument extraction, many classic methods for sequence labeling tasks can be used to extract event arguments from texts. To make full use of prior information about event type, we concatenate sentences with 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 this input representation, we propose three PLM-based architectures for sentence-level event argument extraction: PLM-CRF, PLM-MRC, and PLM-Biaffine.

4.2.1 PLM-CRF BiLSTM-CRF is a classic model for the NER task and has achieved state-of-the-art results in accuracy. Since pre-trained language models like BERT can capture deeper semantic and contextual information, in our PLM-CRF, the input sequence to the PLM consists of event type and sentence. With the help of multiple transformer layers in the PLM, the sentence can fully interact with prior information.

Given the output of the PLM $\{r_1, r_2, ..., r_m, x_1, x_2, ..., x_l\}$, where r_i is the output of the event type and x_i is the output of the sentence, $X = \{x_1, x_2, ..., x_l\}$ is then used as input to the CRF layer. For a sequence of predictions $y = \{y_1, y_2, ..., y_l\}$, we define its score as:

$$s(X, y) = \sum_{i=0}^{l} T_{y_i, y_{i+1}} + \sum_{i=1}^{l} W_x^T x_i$$

where $T \in \mathbb{R}^{n_t \times n_t}$ is a matrix of transition scores and $W_x \in \mathbb{R}^{h \times n_t}$ is used to calculate the scores for each label for each token, n_t is the number of BIO tags, and h is the hidden size of the PLM.

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

4.2.2 PLM-MRC Currently, many NLP tasks can be converted into machine reading comprehension (MRC) problems. Inspired by Li et al. [23], we propose a simplified version of MRC to address event argument extraction.

First, we manually construct 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 sentence representation $X = \{x_1, x_2, ..., x_l\}$ output from BERT, we compute the probabilities of each token being a start index and an end index respectively as:

$$P_{start} = \text{softmax}(W_{start}X + b_{start})$$
$$P_{end} = \text{softmax}(W_{end}X + b_{end})$$

where $W_{start} \in \mathbb{R}^{h \times l}$ and $W_{end} \in \mathbb{R}^{h \times l}$, and h is the hidden size of the PLM.

In the prediction stage, all valid combinations of 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.

4.2.3 PLM-Biaffine The Biaffine model is widely used in dependency parsing [24] and Yu et al. [12] first applied this architecture to the NER task. Following their work, we also use the Biaffine model to extract event arguments from texts.

Same as in PLM-CRF, we first obtain the sentence representation $X = \{x_1, x_2, ..., x_l\}$ from the PLM. After that, two feedforward neural networks (FFNNs) are used to generate representations for the start and end of spans. Then a Biaffine model is applied to predict possible event roles for each span, including a special role named NA, which means the current span is not a valid event argument.

Specifically, the score of an event role for span $\langle i, j \rangle$ is computed as:

$$s_{\langle i,j \rangle} = h_i^{start^T} W_{role} h_j^{end} + h_i^{start^T} W_{start} + h_j^{end^T} W_{end} + b_{role}$$

where h_i^{start} and h_j^{end} are the start/end representations of tokens i and j, $s_{\langle i,j \rangle}$ is the score distribution among n_r event roles, and $W_{role} \in \mathbb{R}^{d \times n_r \times d}$, $W_{start} \in \mathbb{R}^{d \times n_r}$, $W_{end} \in \mathbb{R}^{d \times n_r}$ are trainable parameters in the Biaffine model.

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

4.3 Event Table Filling

After obtaining the event types and event arguments in the document, we design heuristic strategies to convert the results of SEE to DEE. According to the corollaries mentioned 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 plain text. The combination of valid event arguments with minimum internal distance is selected as the event in the document. Leader Death is a special event type in OTOE since it is obvious to find event triggers in sentences, such as “去世”, “逝世”, and “辞世” (all meaning “pass away”). The distance between triggers and event arguments is also considered when computing internal distance.

In the OTME scenario, events mainly appear in tables. Thus, we first use keywords, such as “本次增持股票数量 (万股)” (number of overweight equity), 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 through table parsing, events are generated by the same method as 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 preserve special tokens (mostly
) in the final answer, we check all answers that contain spaces and do not appear in the original text, and restore them to their original form.

5. EVALUATION

This section presents experimental results on the evaluation data and detailed analysis. We compare different variants in event detection and event argument extraction mentioned in Section 4.

5.1 Data Set and Experimental Setup

Experiments are conducted on the CCKS2020 Task 4-2 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 [25, 26]. Validation data and testing data are used for online evaluation on Leaderboard A and Leaderboard B, which contain 750 documents and 28,096 documents, respectively. To achieve better robustness and anti-noise capability, we use 5-fold cross-validation to train each model.

In the event detection experiments, we use Adam to optimize parameters with a learning rate of 0.001 and a minibatch size of 32. The hidden sizes of BiLSTM and CNN are both 256. When extracting event arguments, the learning rate is set to 2e-5 in PLM layers and 2e-4 in other layers. The maximum epochs for PLM-CRF, PLM-MRC, and PLM-Biaffine are 5, 3, and 5, respectively. In particular, the output size of FFNNs is 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 noise in the document, ATT is not as good as MAX. Among the three strategies, ONE shows the worst performance in both CNN-based and BiLSTM-based models, which means it is insufficient to use information from a single sentence to represent the full text in text classification. It is worth noting that the data for this evaluation task mainly come from financial announcements, which usually have a title that summarizes the full text. Thus, a simplified solution is to exploit the information from the title to classify the document. When we use the first sentence of each document for event detection, it works better than ONE, but is not the best.

Table 2. Different models for event detection.

Model	Accuracy
CNN-ONE	0.8920
CNN-ATT	0.9013
CNN-MAX	0.9238
BiLSTM-ONE	0.8960
BiLSTM-ATT	0.9040
BiLSTM-MAX	0.9253
First-Sentence	0.9187

5.3 Experimental Results of Event Argument Extraction

For the three paradigms of event argument extraction, we all use BERT-wwm-Chinese as the pre-trained language model. To exploit global information, the results of event detection are regarded as prior information, which is shared by all sentences from one document. As shown in Table 3 , it is obvious that models using prior information of event types always perform better, which shows that global information from a document is beneficial for event extraction and that it is necessary to detect event type before event argument extraction.

Table 3. Different model variants for event argument extraction.

Model	F1-score	Training Time/Epoch
PLM-CRF †	0.7954	31min
PLM-CRF	0.8103	31min
PLM-MRC †	0.8145	63min
PLM-MRC	0.8234	63min
PLM-Biaffine †	0.8089	18min

Model	F1-score	Training Time/Epoch
PLM-Biaffine	0.8221	18min

Note: † means no prior event type information is utilized.

Among all models, although PLM-MRC yields the best performance, PLM-Biaffine still achieves similar results and has an enormous advantage in training speed. Thus, we select PLM-Biaffine as the basic model and further explore different PLMs to make full use of implicit prior information within PLMs.

From Table 4 , we can observe that NEZHA-large performs best, which directly leads to our decision to use only the combination of NEZHA-large and PLM-Biaffine (NEZHA-Biaffine) in the final competition.

Table 4. Different PLMs for PLM-Biaffine.

Model	F1-score
BERT-base	0.8221
BERT-wwm	0.8234
BERT-wwm-ext	0.8243
ERNIE	0.8256
RoBERTa-wwm-ext	0.8267
RoBERTa-wwm-ext-large	0.8289
NEZHA-large	0.8301

5.4 Online Results

Based on the above experimental results, BiLSTM+MAX and NEZHA-Biaffine are selected as our final models. The detailed results are listed in Table 5 , which shows that our model (PIEE) is effective. Moreover, since the online results for Bankruptcy Liquidation, Asset Loss, Accident, Leader Death, and External Indemnity are always 0 on the final testing data, we retrained the model on the data of the remaining event types, which increased the results from 0.66247 to 0.66996.

Table 5. Top 5 teams on Leaderboard A and Leaderboard B.

Leaderboard A		Leaderboard B	
Teams	F1-score	Teams	F1-score
PIEE (Our system)	0.83007	PIEE (Our system)	0.66996
Rank 2	0.81815	Rank 2	0.64786
Rank 3	0.81543	Rank 3	0.64321
Rank 4	0.80234	Rank 4	0.63245
Rank 5	0.79876	Rank 5	0.62138

6. CONCLUSION AND FUTURE WORK

In this paper, we propose a Prior Information Enhanced Extraction Framework (PIEE) for document-level financial event extraction, consisting of three components: event detection, event argument extraction, and event table filling. In our solution, we show that it is necessary to first detect event types in DEE, which helps extract event arguments as explicit prior information. Moreover, we explore the implicit prior information of different PLMs in event argument extraction. For Document-level Event Argument Extraction in CCKS2020 Task 4-2, our system achieved 0.83007 F1-score and 0.66996 F1-score on Leaderboard A and Leaderboard B, respectively—both the highest scores—demonstrating the advantages of our framework.

Nevertheless, our framework could be further improved due to potential limitations and deficiencies. Overall, PIEE is a pipeline framework, which might cause error propagation and accumulation. For example, the performance of event argument extraction largely depends on the result of event detection. Moreover, it is inflexible to fill event tables using heuristic strategies. This is where we need further improvement in the future.

ACKNOWLEDGEMENTS

This research is supported by the National Natural Science Foundation of China (No. 61936010 and No. 61876115). This work was partially supported by the Collaborative Innovation Center of Novel Software Technology and Industrialization.

AUTHOR CONTRIBUTIONS

H.T. Wang (htwang2019@stu.suda.edu.cn) contributed to dataset statistics, experimental design, and manuscript writing. T. Zhu (tzhu7@stu.suda.edu.cn) contributed to dataset statistics, experiments with different pre-trained language models, and manuscript writing. M.T. Wang (wangmt@suda.edu.cn) contributed to strategies for extracting equity freeze events, table content parsing, and data annotation. G.L. Zhang (glzhang@stu.suda.edu.cn) contributed to dataset statistics, bad case analysis, data annotation, and manuscript revision. W.L. Chen (wlchen@suda.edu.cn) contributed to dataset statistics, overall framework design, and manuscript writing. All authors made meaningful and valuable contributions in revising and proofreading the manuscript.

DATA AVAILABILITY STATEMENT

The datasets generated and/or analyzed during the current study are not publicly available because they were produced by expert consultants from the Institute of Automation, Chinese Academy of Sciences and Ant Financial Services Group based on their own experience. The publicly released version of the datasets requires the consent of all expert consultants and is available from the corresponding author upon reasonable request.

REFERENCES

[1] Chen, Y., 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 (Volume 1: Long Papers), pp. 167–176 (2015)

[2] 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)

[3] 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, pp. 6851–6858 (2019)

[4] 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)

[5] Zhang, T., Ji, H.: Event extraction with generative adversarial imitation learning. arXiv preprint arXiv:1804.07881 (2018)

[6] Lample, G., et al.: 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] Yang, H., et al.: 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)

[8] Zheng, S., 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)

[9] Lample, G., et al.: 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)

[10] Levy, O., et al.: Zero-shot relation extraction via reading comprehension. In: Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pp. 333–342 (2017)

[11] Li, X., et al.: 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)

[12] 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)

[13] 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)

[14] Peters, M., et al.: 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)

[15] Cui, Y., et al.: Pre-training with whole word masking for Chinese BERT. arXiv preprint arXiv:1906.08101 (2019)

[16] Sun, Y., et al.: Ernie: Enhanced representation through knowledge integration. arXiv preprint arXiv:1904.09223 (2019)

[17] Liu, Y., et al.: RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)

[18] Wei, J., et al.: Neural contextualized representation for Chinese language understanding. arXiv preprint arXiv:1909.00204 (2019)

[19] 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 (2010)

[20] Zeng, D., et al.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1753–1762 (2015)

[21] Lin, Y., et al.: 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)

[22] Jiang, X., et al.: 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)

[23] Li, X., et al.: 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)

[24] Dozat, T., Manning, C.D.: Deep biaffine attention for neural dependency parsing. arXiv preprint arXiv:1611.01734 (2016)

[25] Mintz, M., et al.: 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, pp. 1003–1011 (2009)

[26] Chen, Y., et al.: 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)

AUTHOR BIOGRAPHY

Haitao Wang is a postgraduate student at the School of Computer Science and Technology, Soochow University. He is interested in multi-format information extraction, especially semantic relation extraction between entities and document-level event extraction. ORCID: 0000-0003-2531-149X

Tong Zhu is a postgraduate student at the School of Computer Science and Technology, Soochow University. He is interested in multi-format information extraction, especially semantic relation extraction between entities and document-level event extraction. ORCID: 0000-0002-5433-8504

Mingtao Wang is a graduate student at the School of Computer Science and Technology, Soochow University. He is currently a research assistant at the university. He is interested in information extraction, dialogue systems, and knowledge base question answering. ORCID: 0000-0002-8838-1519

Guoliang Zhang is a postgraduate student at the School of Computer Science and Technology, Soochow University. He is interested in multi-format information extraction, especially semantic relation extraction between entities or events, and event extraction. ORCID: 0000-0002-3639-0712

Wenliang Chen received his Bachelor' s degree in Mechanical Engineering and PhD degree in Computer Science from Northeastern University in 1999 and 2005, respectively. He joined Soochow University in 2013 and is currently a professor at the university. Prior to joining Soochow University, he was a research scientist at the Institute for Infocomm Research of Singapore from 2011 to 2013. From 2005 to 2010, he worked as an expert researcher at NICT, Japan. His current research interests include parsing, machine translation, and machine learning. ORCID: 0000-0002-5429-6084