A Trigger-Free Method Enhanced by Coreference Information for Document-Level Event Extraction

Fumin Chen[1], Xu Wang[1,*], Xiaohui Liu[2], Dezhong Peng[1,3,4]
[1]*College of Computer Science, Sichuan University, Chengdu, China*
[2]*School of Cyber Science and Engineering, Sichuan University, China*
[3]*Chengdu Ruibei Yingte Information Technology Co., Ltd, Chengdu, China*
[4]*Sichuan Zhiqian Technology Co., Ltd, Chengdu, China*
Email:2428896862@qq.com, wangxu.scu@gmail.com, liuxhuih@163.com, pengdz@scu.edu.cn

2023 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-8867-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/IJCNN54540.2023.10192046

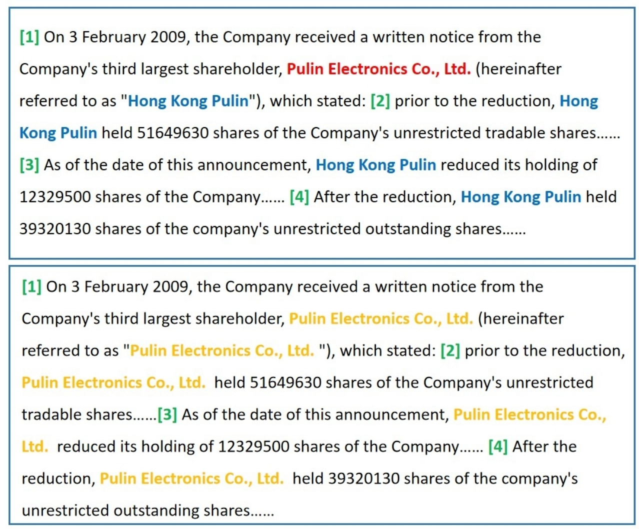
***Abstract*—Document-level event extraction (DEE) aims to extract structured event information from a document. Previous document-level event extraction methods relied on trigger annotation, which was very expensive and time-consuming. In addition, entity representation plays an important role in the overall event extraction task, but we found that the previous work could not effectively use entity abbreviations and coreference information, which limited the representation ability of event-related entities in documents. Based on the above two aspects, we propose a new model named TFECI. In our model, we propose an efficient and effective method to make full use of abbreviations and coreference information to better exert the representational ability of event-related entities. Besides, triggers play an extremely central role in an event, but trigger annotation is often very difficult, so we propose a new strategy to select pseudo-trigger automatically. Experiments show that, compared with the previous system, our system can extract events without trigger annotation and achieve competitive results.**

***Index Terms*—event extraction, entity abbreviation, coreference information, pseudo-trigger, TFECI**

I. Introduction

Event extraction is an important and challenging task in the field of Information Extraction (IE), which aims to detect events from the target text and extract the corresponding arguments. Most of the previous work has focused on event extraction at the sentence level [1], [2], [3], [4], extracting events from individual sentences. However, in a real-world scenario, arguments are usually scattered among different sentences in the document, and the sentence-level event extraction model cannot resolve such a case. Therefore, extracting events from documents has recently attracted more and more attention.

Even so, several challenges remain with document-level event extraction. First, arguments in an event record may be scattered among different sentences in the document, which requires the model to have a holistic understanding of cross-sentence context. To solve this challenge, the recently proposed Doc2EDAG [5] model converts the event table into a directed acyclic graph (DAG) and iteratively extracts each event role. Although this model performs well, building DAG requires significant computational resources to store previous paths and is particularly time-consuming during the training and reasoning phases. Focusing on single event extraction, Du and Cardie [6] and Du et al. [7] connect multiple sentences and only consider a single event record, lacking the ability to model the relation between multiple event records in the long document. Second, there may be abbreviations and coreference to entities that have the same meaning, for example, the example in Fig. 1 is a document from the Chinese dataset in the financial domain proposed by Zheng et al. [5], we translate it into English for illustration. The document above in the figure is the original document, where "Hong Kong Pulin" is the abbreviation form of "Pulin Electronics Co., Ltd.", although they have different forms on the surface. The bottom of the figure shows the document with fused coreference information obtained after processing by our method so that the same entity can converge more information. However, previous work in entity feature aggregation, only mention with identical expressions can participate, resulting in the loss of entity representation information. Besides, there are still many challenges in document-level event extraction without trigger annotation. Currently, the document-level event extraction datasets usually use distantly supervised and knowledge base [8], [9] to annotate triggers, which are missing and the


[1] On 3 February 2009, the Company received a written notice from the Company's third largest shareholder, Pulin Electronics Co., Ltd. (hereinafter referred to as "Hong Kong Pulin"), which stated: [2] prior to the reduction, Hong Kong Pulin held 51649630 shares of the Company's unrestricted tradable shares...... [3] As of the date of this announcement, Hong Kong Pulin reduced its holding of 12329500 shares of the Company...... [4] After the reduction, Hong Kong Pulin held 39320130 shares of the company's unrestricted outstanding shares......

[1] On 3 February 2009, the Company received a written notice from the Company's third largest shareholder, Pulin Electronics Co., Ltd. (hereinafter referred to as "Pulin Electronics Co., Ltd. "), which stated: [2] prior to the reduction, Pulin Electronics Co., Ltd. held 51649630 shares of the Company's unrestricted tradable shares......[3] As of the date of this announcement, Pulin Electronics Co., Ltd. reduced its holding of 12329500 shares of the Company...... [4] After the reduction, Pulin Electronics Co., Ltd. held 39320130 shares of the company's unrestricted outstanding shares......

Fig. 1. Overall architecture of our coreference information enhancement.

* Corresponding Author.

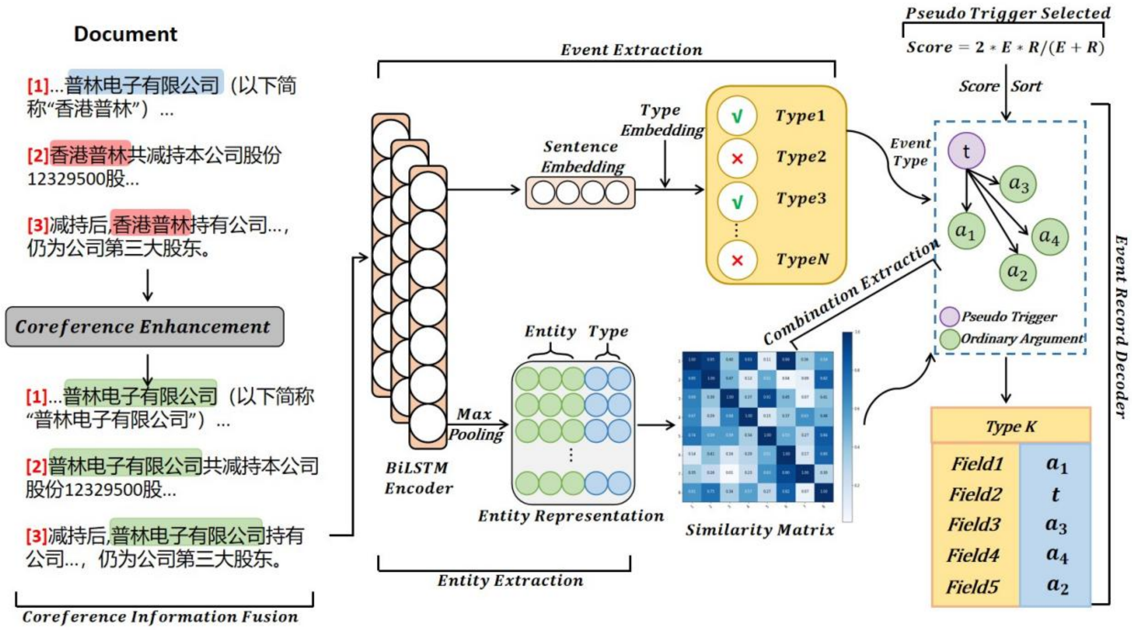


Fig. 2. The general framework of TFECI.

quality of annotation is low. Based on this, several methods of event extraction without triggers are proposed. Yang et al. [10] first identify a key sentence in the document and then fill the event table by looking for arguments close to the key sentence, but this approach misses the context and global arguments information. Zheng et al. [5] make full use of the global context information and achieve significant improvement by establishing directed acyclic graphs (DAGs). However, this method relies on a large number of computing resources. It takes almost one week to train the DAG-based model on four 32GB GPUs, and the inference speed is also extremely slow. Zhu et al. [11] use the pruned complete graph non-autoregressive decoding algorithm to conduct event extraction guided by selected triggers automatically, greatly shortening the time of training and reasoning and achieving remarkable results, but there is no better use of abbreviations and coreference information.

Considering the above problems with the DEE task, our contributions include:

- We propose a concise and efficient method to take full advantage of entity coreference and abbreviation information and we conduct extensive experiments to show the effectiveness.
- Facing the situation without trigger annotation, we propose a more comprehensive and transferable pseudo-trigger selection strategy.
- We propose TFCEI, a novel end-to-end joint learning framework for document-level event extraction.

In summary, we propose a novel approach named **T**rigger-**F**ree Method **E**nhanced by **C**oreference **I**nformation Model (TFECI). Compared with the previous work, we combine entity coreference and abbreviation information, formulate a pseudo-trigger selection strategy to select important arguments as the event's core, and apply pruned complete graphs to decode the event record, achieving a competitive result.

II. Related Work

A. Sentence-level Event Extraction

Sentence-level Event Extraction is one of the important tasks in the field of natural language processing. A lot of work has obtained good performance by using machine learning algorithm[12], [13], [14], [15], [16], [17] for feature extraction and information mining. However, most of the previous work has focused on sentence-level event extraction task. Chen et al. [1] automatically extract lexical level and sentence level features without using the tools of complex natural language processing. This model is a pipeline model that first identifies triggers and then extracts argument roles from events. Nguyen et al. [2] use two different recurrent neural networks to learn sentence representation and conduct joint event extraction, that is, extract triggers and argument roles simultaneously. Chen et al. [18] give full play to the potential of the pre-training model for event extraction, and take advantage of external knowledge [19], [20]. Du et al. [21] are the first to apply the Question-Answer method to solve the problem of event extraction. Liu et al. [22] integrates syntactic dependency information and uses graph convolution neural network to learn syntactic context representation for better event extraction. However, in real-world scenarios where extracting multiple events from a document containing multiple sentences is more common, sentence-level event extraction methods can not handle this problem very well.

B. Document-level Event Extraction

Document-level event extraction has been studied by more and more researchers recently. Yang et al. [23] use well-defined features to model and learn event-argument relations between different sentences in a document. Although Zheng et al. [5] use a Transformer to fuse the features of entities and sentences, the interdependencies between events are ignored. In addition, the training and reasoning process of the model consumes a lot of memory and time. Moreover, entity abbreviations and coreference are often found in event extraction datasets, these abbreviations usually contain contextual relations between entities. Besides, in document-level event extraction task, fully annotated datasets only have a small scale, and most of them are large-scale annotated in the way of DS-constructed. However, it is difficult to match triggers in this way, so triggers are usually absent. Different from the trigger-based methods [24], [25]. Liu et al. [21] discuss how to do event types detection without triggers. Doc2EDAG [5] and PTPCG [11] can also complete the document-level event extraction well without triggers, but they ignore the use of coreference and abbreviations information. However, unlike previous approaches [5], [11], they just aggregate features from mentions with the same surface name into entity representation, ignoring abbreviations and coreference information, we use entity coreference and abbreviations to obtain a better representation of the entity.

III. Method

In this section, we describe the details of TFECI. We first present the general framework of our model, then we talk about approaches to event extraction.

To better illustrate our model, we first need to clarify the meanings of several terms: 1) **mention** is a text span that refers to an entity object in a document; 2) **role** is predefined for each event type; 3) **argument** is an entity that plays the role of a specific event; 4) **event record** refers to a collection of arguments for different roles in the event for a specific event type.

As shown in Fig. 2, our model can be divided into six components: 1) **Coreference information fusion** uses rule-based coreference feature pattern to obtain entity abbreviations and coreference information and then integrates them into the subsequent model to improve performance. 2) **Pseudo-trigger selection strategy**. The mixed metrics of existence and representation proposed by us select a pseudo-trigger that can represent events based on event role. 3) **Event detection** uses an attention mechanism [26] to define it as a multi-label task. 4) **Entity extraction representation module** will extract all entities from the document and represent them as dense feature vectors. 5) **Combination extraction** coincides with PTPCG model [11], by constructing pruned complete graph and decoding the corresponding arguments from the graph. 6) **Event record generation**. Combine event types and extracted arguments to generate the final event record.

A. Coreference Information Fusion

In an event document, an entity may be scattered across multiple sentences in the document, and there may be instances of referring to abbreviations. Previous methods only regard mentions with the same surface name as one entity. It is hard to get a more appropriate representation of the entity, as shown at the top of Fig. 1, although the entity "Pulin Electronics Co., Ltd." has a different name from the entity "Hong Kong Pulin", the meaning is the same. Based on the above problems, more coreference and abbreviations information are integrated by our method, and coreference recovery is carried out in the data. The result is shown below in Fig. 1, this allows for a better representation of the entity even if there are abbreviations and coreference words scattered across different sentences in the document. Since we explore coreference and abbreviations information at the data level, the results of processing can be easily migrated to any document-level event extraction models using entity representation.

B. Pseudo-trigger Selection Strategy

In real scenarios, it is very difficult to annotate triggers in large-scale document data. Facing this challenge, the PTPCG model [11] proposes the concept of a pseudo-trigger to replace triggers annotating for the first time. Based on this approach, we propose a novel pseudo-trigger selection strategy. In the previous event extraction method, the triggers as the core of the event mainly act as two important symbols: 1) triggers can distinguish subtle differences between different events; 2) triggers can be used to extract all arguments throughout the event. Based on the two symbols that trigger serve as, for each event type, we designed a selection metric to evaluate the metric score of a role as a pseudo-trigger for an event, then select the argument corresponding to the role with the highest metric score in the current event type as the pseudo-trigger.

The metric score has two main components: 1) **Representation** refers to the ability of a role to represent the current event, that is, the ability to distinguish the subtle differences between different events. Let C_u^R represent the number of times that arguments corresponding to a role appear in the current event record but not in other event records; 2) **Existence** refers to the ability of the current role to identify arguments in an event. If a certain role is used as the trigger, many arguments corresponding to the role of the current event type in the data should not be empty. So for all the roles contained in the event type t_i, let C_i^R represent the number of elements in the current role that are not Null, and C_i represent the number of documents corresponding to the current event type in the entire dataset. *Representation* and *Existence* are calculated as follows.

$$Representation = \frac{C_u^R}{C_i}, Existence = \frac{C_i^R}{C_i} \tag{1}$$

After calculating the *Representation* and *Existence*, to further balance the two indicators, we further process to obtain the final pseudo-trigger selection score, such as the *Pseudo Trigger Selected* module in Fig. 2.

$$Score = \frac{2 \times Representation \times Existence}{Representation + Existence} \quad (2)$$

C. Event Detection

Given a document *D*, for each sentence s_i in the document, we use BiLSTM to encode s_i into a sequence vector. Then, according to the practice of using Doc2EDAG [5], for each event type t_i, the randomly initialized event representation vector is used as the Query of attention mechanism [26], and the document sentence representation vector is used as the Key and Value of multi-head attention for binary classification to identify whether the current document is of type t_i. The loss function of event detection module $\mathcal{L}_{detection}$ is defined as a cross-entropy loss.

D. Entity Extraction Representation

To achieve a fair comparison, we formulate the entity extraction task as a sequence tagging task in BIO scheme [5], [11]. For the document-level entity extraction task, we divide the document into multiple sentences and performed entity extraction in each sentence. We use the BiLSTM module, which is the same as in event classification detection, to obtain the token representation of sentences and extract entities combined with CRF [27]. The training objective of entity extraction is to minimize the negative log-likelihood loss $\mathcal{L}_{entity}$ of CRF for each sentence.

To obtain entity representation, we first apply the extracted mention to the token level by a max-pooling operation to form a representation m_i'. Because mention type features have been shown to improve the performance on downstream tasks [5], [28], we convert mention type features into vectors by looking up the embedding table. By concatenating m_i' and its type features f_j, we get the final mention representation $m_j = m_i' \bigoplus f_j$, $m_j \in R^{d_m}$, where $d_m = d_h + d_f$, d_h and d_f refers to the dimension size of the representation and dimension size of the type, respectively. Finally, all mentions of an entity are aggregated by max-pooling to form the final entity representation.

E. Combination Extraction

In the process of combination extraction, the pseudo-trigger selection strategy is first used to automatically select a pseudo-trigger for each event type. According to the method in PTPCG [11], we form an event arguments combination with the pseudo-trigger as the core, and a pruned completed graph is built for each combination. For arguments in a combination, if an argument a_t^i is identified as a pseudo-trigger and the other argument a_o^j is identified as an ordinary entity, they are connected with a directed link, and the adjacency matrix $y_A^{(i,j)} = 1$. Besides, every argument $a^{(i)}$ will have a self-loop connection and the adjacency matrix $y_A^{(i,i)} = 1$. The other entities that do not participate in any combination construction are isolated nodes with a value of 0 in the adjacency matrix. After the entity representation is obtained, the method of calculating similarity through a dot scale is used to evaluate the semantic distances between different entities, as in:

$$\begin{aligned} e_i' &= e_i \times W_i^{\intercal} + b_i, e_j' = e_j \times W_j^{\intercal} + b_j \\ A_{i,j} &= \sigma(e_i'^{\intercal} e_j' / \sqrt{d_h}) \end{aligned} \quad (3)$$

where *A* represents the similarity matrix between entities, W_i, $W_j \in R^{d_m \times d_m}$ and b_i, $b_j \in R^{d_f}$ are trainable parameters for semantic space linear projection. In the training phase, the binary cross entropy loss function is used to calculate the combination loss $\mathcal{L}_{comb}$. A threshold $\lambda = 0.5$ is used in the decoding phase to predict the value of the adjacency matrix *A*.

F. Event Record Decoding

For decoding the event record in the document, we follow the formula in PTPCG [11] model. After our predicted adjacency matrix *A* based on a non-autoregressive decoding algorithm to extract the arguments combinations of the event, the next step is to fill these combinations into event tables. Because event detection is a multi-label classification task, it is possible to predict more than one type of result. For all predicted types $T_p = \{t_j\}_{j=1}^{|T_p|}$ and combination *C*, form a Cartesian product and get all type-combination pairs.

For each pair $< t_j, c_k >$, a feed-forward neural network of event-specific is used as the classifier to get all the corresponding role results of the current event type, and the sigmoid activation function is used to get the corresponding probability. The loss $\mathcal{L}_{role}$ of this part is calculated according to the binary cross entropy loss function.

G. Model Optimization

Our TFECI model is an end-to-end joint training process, and the overall loss calculation can be obtained by assigning different weights, as in:

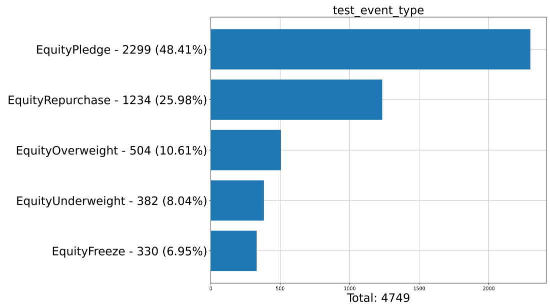
$$\mathcal{L} = \lambda_1 \mathcal{L}_{detect} + \lambda_2 \mathcal{L}_{entity} + \lambda_3 \mathcal{L}_{comb} + \lambda_4 \mathcal{L}_{role} \quad (4)$$

where λ_1, λ_2, λ_3, λ_4 are hyper-parameters to optimize the training of the model. Through parameter experiment analysis, when they are set as 1.0, 0.05, 1.0, and 1.0, the model performance reaches a better result. In our implementation of TFECI, we use the same 2 layers of BiLSTM for entity extraction and event detection. Being consistent with PTPCG [11], we use the same vocabulary and initialize embedding randomly where d_h = 768 and d_f = 32. We use the Adam optimizer [29] to optimize the model while using the learning rate of 5e-4 and set the batch size to 64. Following the work of Doc2EDAG [5] and PTPCG [11], we train our model after 100 epochs to select the model with the highest F1 score on the development set to evaluate the test set.

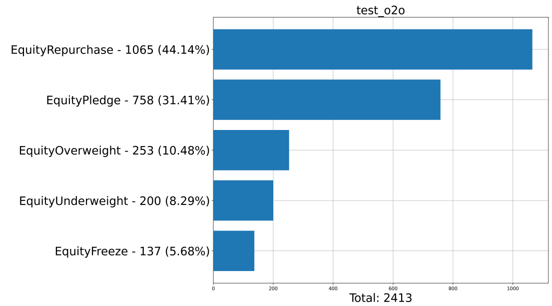
IV. Experiments

A. Experiments Setup

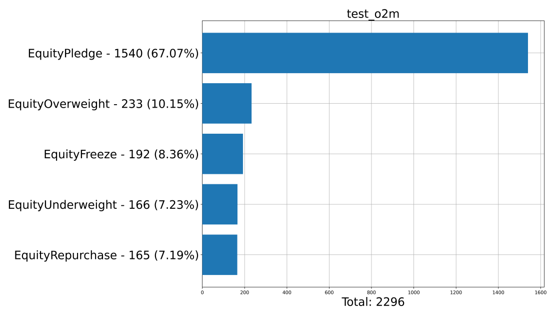
Dataset. We evaluate our model on **ChFinAnn**, a public dataset proposed by Zheng et al. [5], which is a large-scale event extraction dataset constructed using remote supervision methods in Chinese financial scenarios and widely used in



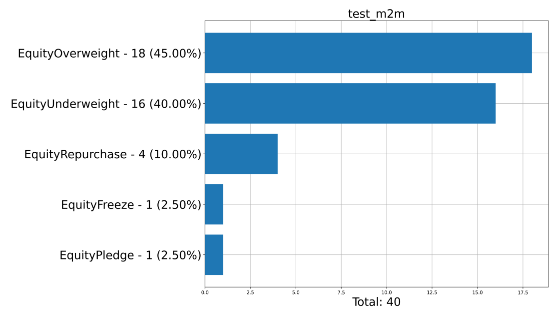
(a) Data distribution of all documents in the test set



(b) Data distribution of o2o documents in the test set



(c) Data distribution of o2m documents in the test set



(d) Data distribution of m2m documents in the test set

Fig. 3. Visualization of data distribution of different document types. o2o represents one event type in a document and only corresponds to one event. o2m indicates that one event type exists in a document and corresponds to multiple events. m2m indicates that a document has multiple event types and corresponds to multiple events

TABLE I
THE MAIN EXPERIMENTAL RESULTS OF BASELINES AND THE TFECI EVALUATED BY F1 SCORE ON THE TEST SET.∗WE REPRODUCE THE RESULTS USING THEIR OPEN-SOURCE CODES.

Model	P	R	F1	Params
DCFEE-O*	69.7	57.8	63.19	32M
DCFEE-M*	60.1	61.3	60.69	32M
GreedyDec*	81.9	51.2	63.01	64M
Doc2EDAG*	81.1	**77.0**	78.99	64M
PTPCG*	**83.4**	75.1	79.03	32M
TFECI(ours)	83.3	76.0	**79.48**	32M

previous work. It contains 32040 documents and focuses on five event types: *Equity Freeze* (EF), *Equity Repurchase* (ER), *Equity Underweight* (EU), *Equity Overweight* (EO) and *Equity Pledge* (EP). There are 35 different types of event roles, 29% of the documents have more than one event record, and 98% of the event record arguments are scattered among different sentences in the document. **Metrics.** The ultimate goal of the document-level event extraction is to properly fill every role argument in the event table. We use the same evaluation metrics as the Doc2EDAG [5] method to evaluate them by directly comparing the predicted event table to the golden table for each event type, and F1 scores are calculated by comparing arguments.

Baselines. 1) **DCFEE** [10] has two variants: **DCFEE-O** extracts only one event record from the document and **DCFEE-M** extracts as many event records as possible. 2) **Doc2EDAG** [5] uses a transformer encoder to obtain sentence and entity embeddings, followed by another transformer to fuse cross-sentence context. The model constructs records as argument chains (DAG) and uses an auto-regressive way to extract final results. 3) **GreedyDec** is the baseline contrast model adopted by Zheng et al. [5], using a greedy pattern to fill the event table. 4) **PTPCG** [11] designs a novel strategy for DEE together with a non-autoregressive decoding algorithm.

B. Main Results and Ablation Analysis

Overall performance. The overall experimental results on the **ChFinAnn** dataset are illustrated in Table I. Thanks to better use of coreference information and pseudo-triggers selection strategy, our TFECI model surpasses other baselines. Specifically, compared with Doc2EDAG [5], TFECI uses fewer model parameters but improves 0.49 micro F1 score. Besides, compared with the previous state-of-the-art model PTPCG [11] in the event extraction task, TFECI takes full advantage of abbreviations and coreference of an entity and improves the 0.45 micro F1 score.

Among all the models, TFECI is the lightest one and is on the same scale with DCFEE [10], while TFECI surpasses DCFEE-O with about 15.9 absolute gain in F1 score on the ChFinAnn dataset.

Ablation Analysis. We design further experiments to analyze the effect of key modules in the event extraction task.

TABLE II
ABLATION TEST RESULTS OF TFECI REPORTING F1 SCORE ON TEST SET.

Model	EF			ER			EU			EO			EP			ALL		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
TFECI	83.2	66.7	**74.1**	92.7	90.4	**91.6**	77.2	65.5	70.9	75.8	67.6	71.4	80.6	73.1	**76.7**	83.3	76.0	**79.48**
-Coref	81.2	54.2	65.0	92.8	89.8	91.3	80.4	65.5	72.2	78.5	68.1	72.9	80.1	73.4	76.6	83.4	75.4	79.20
-Pseudo	82.8	57.6	68.0	92.6	89.9	91.2	80.5	65.5	**72.3**	77.0	69.7	**73.1**	80.9	72.0	76.2	83.7	74.9	79.06

TABLE III
PERFORMANCE ANALYSIS RESULTS OF MODELS IN DIFFERENT DOCUMENT TYPES

Data Type	Event Detection			Entity Extraction			Combination			ALL		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
o2o	99.38	99.50	99.44	97.45	99.51	98.47	54.28	60.67	57.23	86.69	90.33	88.47
o2m	98.33	98.58	98.45	96.47	99.43	97.93	23.54	21.44	22.44	78.79	61.86	69.31
m2m	100.00	64.87	78.69	94.70	98.08	96.36	36.67	27.50	31.43	75.42	49.17	59.53
overall	99.13	98.88	99.01	97.12	99.47	98.28	40.92	41.55	41.23	83.28	76.02	79.48

TABLE IV
THE RESULTS OF SEMANTIC SIMILARITY METHODS APPLIED TO COMBINATION EXTRACTION.

Method	EF			ER			EU			EO			EP			ALL		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Cosine	85.5	50.4	63.4	95.8	85.8	90.5	81.8	55.3	66.0	78.3	54.2	64.0	90.6	49.1	63.7	90.5	58.7	71.21
Dot Scaled	83.2	66.7	**74.1**	92.7	90.4	**91.6**	77.2	65.5	**70.9**	75.8	67.6	**71.4**	80.6	73.1	**76.7**	83.3	76.0	**79.48**

In Table II, Coref and Pseudo represent the Coreference Enhancement Module and our Pseudo-trigger Selection Strategy, respectively.

On the effect of coreference information fusion. This module is used to fuse more useful global information for event-related entities, taking advantage of more features by restoring the same coreference. To explore how it works, we temporarily remove this module from the overall model. In addition to calculating the F1 score, we separately calculate the model's F1 score for each event type. Results are shown in Table II, without coreference information enhancement to obtain entity representation, the total micro F1 score decreases by 0.28. Performance in Equity Freeze (EF), Equity Repurchase (ER) and Equity Pledge (EP) decreases, especially the F1 score in Equity Freeze (EF) event type decreases by 9.1. It demonstrates that coreference information can improve event extraction performance by helping entities obtain better representation.

On the effect of pseudo-trigger selection strategy. This module conducts event extraction without trigger annotation by designing generalization and targeted selection metric and using the mixed method of representation and existence to specify pseudo-trigger. To explore its effect, we remove the pseudo-trigger selection module in addition to the coreference information module. As shown in Table II, compared with the complete TFECI model, the total micro F1 decreases by 0.42. Compared to the performance of removing the coreference enhancement module from the TFECI model, the total micro F1 decreases by 0.14. It demonstrates that the pseudo-trigger selection strategy helps conduct the event extraction task, which validates the effectiveness of this module.

TABLE V
HYPER PARAMETERS SENSITIVITY ANALYSIS RESULTS.

λ_1	λ_2	λ_3	λ_4	EF	ER	EU	EO	EP	ALL
0.5	0.05	1.0	1.0	66.3	90.9	71.4	74.0	76.0	78.82
1.0	1.0	1.0	1.0	68.4	91.3	70.5	72.1	75.9	78.74
1.0	0.05	1.0	0.5	68.4	91.0	72.0	74.1	75.6	78.72
1.0	0.05	1.0	0.5	68.5	91.2	**73.5**	**74.3**	76.4	79.34
1.0	0.05	1.0	1.0	**74.1**	**91.6**	70.9	71.4	**76.7**	**79.48**

C. *Performance analysis of models in different document types*

To explore the performance of our model components under different document types, we divide the test set into o2o, o2m, and m2m. o2o represents only one event type in the document and corresponds to one event record. o2m indicates that one event type exists in a document and corresponds to multiple event records. m2m indicates that a document has multiple event types and corresponds to multiple event records. We count the distribution of different document types in the test set, as shown in Fig. 3. In ChFinAnn's test set, the number of o2o documents is more than half the number of the test set. Through experiments on different document types, as shown in Table III, we find that the performance of event detection and entity extraction is relatively good. However, combination extraction is not satisfactory because it is difficult to select event-related arguments from the redundant entity list and classify them correctly. Besides, the F1 score on the m2m type is much lower than on the o2o type, indicating it is challenging to extract multiple event records from different event types.

D. Analysis of semantic similarity methods applied in Combination Extraction

In the process of combination extraction, we need to calculate the semantic distances between entities to form a similarity matrix for the event record. we try a dot-scaled similarity function (3) and a cosine similarity function to calculate the semantic distance between two entities, respectively. To further explore which method is more suitable for our model, we further design experiments for verification. As shown in Table IV, compared with a cosine similarity function, when a dot-scaled similarity function is adopted for semantic distance calculation, the model performance improves the F1 score of the model is increased by 8.27. A dot-scaled similarity function described above is more suitable for our model.

E. Sensitivity analysis of hyper parameters

In this subsection, in order to explore the influence of the loss weight of every module on model performance, we further adopt several sets of different weight parameter schemes for experiments. The experimental results are shown in Table V. Compared with the final parameter scheme adopted for the last row in the table, when the weight λ_1 of the event detection task module is reduced, the F1 score decreases, indicating that the result of event detection is of great significance in guiding the subsequent model steps. For the weight λ_2 of the entity extraction task module, the performance is better when we set a small value, and the performance decreases when we increase its value, indicating that the model is relatively easy to learn to extract entities from sentences in a document. Besides, for λ_3 and λ_4, reducing their weight values will also degrade model performance.

V. CONCLUSION

In this paper, we exploit the DEE model in two aspects: 1) incorporate more coreference information for a better representation of the entity; 2) in the scenario without triggers annotations, we propose a selection metric to select the pseudo-trigger as the core of an event for event extraction automatically. Experiments on the ChFinAnn dataset show TFECI outperforms previous competitive methods. In the future, we should incorporate a more intelligent and complex approach, although we have incorporated some coreference information, some hidden coreference such as it, you, he, she, etc., are not well handled. If we can further integrate this part of coreference information, perhaps the performance of event extraction will be further improved.

ACKNOWLEDGMENT

The work is supported by Sichuan Science and Technology Planning Project (Grant No. 2021YFS0390), Sichuan University Postdoctoral Interdisciplinary Innovation Fund (Grant No. JCXK2234).

REFERENCES

[1] Yubo Chen 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*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 167–176.

[2] Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. "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*. San Diego, California: Association for Computational Linguistics, 2016, pp. 300–309.

[3] David Wadden et al. "Entity, Relation, and Event Extraction with Contextualized Span Representations". In: (2019), pp. 5784–5789.

[4] Jiawei Sheng et al. *CasEE: A Joint Learning Framework with Cascade Decoding for Overlapping Event Extraction*. Online, 2021.

[5] Shun Zheng et al. "Doc2EDAG: An End-to-End Document-level Framework for Chinese Financial Event Extraction". In: Hong Kong, China, 2019, pp. 337–346.

[6] Xinya Du and Claire Cardie. "Document-Level Event Role Filler Extraction using Multi-Granularity Contextualized Encoding". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 8010–8020.

[7] Xinya Du, Alexander M Rush, and Claire Cardie. "Document-level event-based extraction using generative template-filling transformers". In: *arXiv preprint arXiv:2008.09249* (2020).

[8] Mike Mintz 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*. Suntec, Singapore: Association for Computational Linguistics, 2009, pp. 1003–1011.

[9] Yubo Chen 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)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 409–419.

[10] H. Yang et al. "DCFEE: A Document-level Chinese Financial Event Extraction System based on Automatically Labeled Training Data". In: *Meeting of the Association for Computational Linguistics*. Melbourne, Australia, 2018, pp. 50–55.

[11] T. Zhu et al. "Efficient Document-level Event Extraction via Pseudo-Trigger-aware Pruned Complete Graph". In: (2022), pp. 4552–4558.

[12] Qian Wang et al. "CLSEP: Contrastive Learning of Sentence Embedding with Prompt". In: *Know.-Based Syst.* 266.C (2023).

[13] Jie Chen et al. "Two-Stage Sparse Representation Clustering for Dynamic Data Streams". In: *IEEE Transactions on Cybernetics* (2022), pp. 1–13.

[14] Jie Chen, Shengxiang Yang, and Zhu Wang. "Multi-View Representation Learning for Data Stream Clustering". In: *Inf. Sci.* 613.C (2022), pp. 731–746.

[15] Fu Zhao et al. "Document Summarization using Word and Part-of-speech based on Attention Mechanism". In: *Journal of Physics: Conference Series* 1168.3 (2019), p. 032008.

[16] Yunfei Ji et al. "A deep learning method for named entity recognition in bidding document". In: *Journal of Physics: Conference Series* 1168.3 (2019), p. 032076.

[17] Tianyi Lei et al. "Adaptive Meta-learner via Gradient Similarity for Few-shot Text Classification". In: *Proceedings of the 29th International Conference on Computational Linguistics*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, 2022, pp. 4873–4882.

[18] Y. Chen et al. "Collective Event Detection via a Hierarchical and Bias Tagging Networks with Gated Multilevel Attention Mechanisms". In: *Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 1267–1276.

[19] J. Liu, Y. Chen, and K. Liu. "Exploiting the Ground-Truth: An Adversarial Imitation Based Knowledge Distillation Approach for Event Detection". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 6754–6761.

[20] M. Tong et al. "Improving Event Detection via Open-domain Trigger Knowledge". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, pp. 5887–5897.

[21] Xinya Du and Claire Cardie. "Event Extraction by Answering (Almost) Natural Questions". In: (2020), pp. 671–683.

[22] Xiao Liu, Zhunchen Luo, and Heyan Huang. "Jointly multiple events extraction via attention-based graph information aggregation". In: *arXiv preprint arXiv:1809.09078* (2018).

[23] Bishan Yang and Tom M. Mitchell. "Joint Extraction of Events and Entities within a Document Context". In: (2016), pp. 289–299.

[24] Amir Pouran Ben Veyseh et al. "Modeling Document-Level Context for Event Detection via Important Context Selection". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 5403–5413.

[25] Shulin Liu et al. "Event Detection without Triggers". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 735–744.

[26] Ashish Vaswani et al. "Attention is All You Need". In: (2017), pp. 6000–6010.

[27] Guillaume Lample et al. "Neural architectures for named entity recognition". In: *arXiv preprint arXiv:1603.01360* (2016).

[28] Runxin Xu et al. "Document-level Event Extraction via Heterogeneous Graph-based Interaction Model with a Tracker". In: (2021), pp. 3533–3546.

[29] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).