RAAT: Relation-Augmented Attention Transformer for Relation Modeling in Document-Level Event Extraction

Yuan Liang*, Zhuoxuan Jiang*, Di Yin and Bo Ren
Tencent Cloud, China
{ericyliang, alexzxjiang, endymecyyin, timren}@tencent.com

Abstract

In document-level event extraction (DEE) task, event arguments always scatter across sentences (across-sentence issue) and multiple events may lie in one document (multi-event issue). In this paper, we argue that the relation information of event arguments is of great significance for addressing the above two issues, and propose a new DEE framework which can model the relation dependencies, called Relation-augmented Document-level Event Extraction (ReDEE). More specifically, this framework features a novel and tailored transformer, named as Relation-augmented Attention Transformer (RAAT). RAAT is scalable to capture multi-scale and multi-amount argument relations. To further leverage relation information, we introduce a separate event relation prediction task and adopt multi-task learning method to explicitly enhance event extraction performance. Extensive experiments demonstrate the effectiveness of the proposed method, which can achieve state-of-the-art performance on two public datasets. Our code is available at `https://github.com/TencentYoutuResearch/RAAT`.

1 Introduction

Event extraction (EE) task aims to detect the event from texts and then extracts corresponding arguments as different roles, so as to provide a structural information for massive downstream applications, such as recommendation (Gao et al., 2016; Liu et al., 2017), knowledge graph construction (Wu et al., 2019; Bosselut et al., 2021) and intelligent question answering (Boyd-Graber and Börschinger, 2020; Cao et al., 2020).

Most of the previous methods focus on sentence-level event extraction (SEE) (Ahn, 2006; Liao and Grishman, 2010; Li et al., 2013; Chen et al., 2015; Nguyen et al., 2016; Zhao et al., 2018; Sha et al., 2018; Yan et al., 2019; Du and Cardie, 2020; Li et al., 2020; Paolini et al., 2021; Lu et al., 2021), extracting events from a single sentence. However, SEE is mostly inconsistent with actual situations. For example, event arguments may scatter across different sentences. As illustrated in Figure 1, the event argument [ORG1] of event role *Pledger* is mentioned in Sentence 4 and the corresponding argument [ORG2] of event role *Pledgee* is in Sentence 5 and 6. We call this **across-sentence issue**. Another situation involves the **multi-event issue**, which means that multiple events may exist in the same document. As seen in the example in Figure 1, where two event records coincide, we should recognize that they may partially share common arguments.

Recently, document-level event extraction (DEE) attracts great attention from both academic and industrial communities, and is regarded as a promising direction to tackle the above issues (Yang et al., 2018; Zheng et al., 2019; Xu et al., 2021b; Yang et al., 2021; Zhu et al., 2021). However, by our observation, we discover that the relations between event arguments have patterns which are an important indicator to guide the event extraction. This information is neglected by existing DEE methods. Intuitively, the relation information could build long-range relationship knowledge of event roles among multiple sentences, which could relieve the across-sentence issue. For multi-event issue, shared arguments within one document could be distinguished to different roles based on the different prior relation knowledge. As illustrated in Figure 1, [ORG1] and [ORG2] have a prior relation pattern of *Pledger* and *Pledgee*, as well as [ORG1] and [SHARE1] for the relation pattern between *Pledger* and its *Pledged Shares*. Therefore, the relation information could increase the DEE accuracy if it is well modeled.

In this paper, we propose a novel DEE framework, called Relation-augmented Document-level Event Extraction (ReDEE), which is able to model

*These authors contributed equally to this work

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4985 - 4997
July 10-15, 2022

Event Records for the Equity Pledge (EP) Event Type (Two Examples)								
Record ID	Pledger	Pledged Shares	Pledgee	Total Holding Shares	Total Holding Ratio	Total Pledged Shares	Start Date	Release Date
1	[ORG1]	[SHARE1]	[ORG2]	[SHARE2]	[RATIO1]	[SHARE2]	[TIME1]	[TIME2]
2	[ORG1]	[SHARE2]	[ORG3]	[SHARE2]	[RATIO1]	[SHARE2]	[TIME2]	-

Mark-Entity Mapping Table	
Mark	Entity
[ORG1]	Shenzhen Nanfang Tongzheng Investment Co., Ltd.
[ORG2]	Chengqing Branch of China Zheshang Bank Co., Ltd.
[ORG3]	Jiatianxia Asset Management Co., Ltd
[TIME1]	September 18, 2017
[TIME2]	September 20, 2018
[SHARE1]	10,000,000
[SHARE2]	10,072,158
[RATIO1]	6.57%

ID	Selected Sentences of a Document
S4	Chongqing Wanli New Energy Co., Ltd. (hereinafter referred to as the "Company" or "the Company") received on September 21, 2018 that [ORG1] …
S5	On [TIME1], Nanfang Tongzheng pledged its [SHARE1] unrestricted tradable shares of the company to [ORG2] for …
S6	Nanfang Tongzheng has released all the above-mentioned [SHARE1] shares pledged to [ORG2], and … on [TIME2] …
S8	According to … [ORG3], Nanfang Tongcheng pledged its [SHARE2] unrestricted tradable shares of the company to [ORG3], …([TIME2]) until …
S9	As … [SHARE2] shares of the company, accounting for [RATIO1] of the company's total share capital, and the cumulative number of ledged shares is [SHARE2] …

Event-related Relations between Entities Table (Two Examples)		
Head Entity	Tail Entity	Relation
[ORG1]	[ORG2]	Pledger and Pledgee
[ORG1]	[SHARE1]	Pledger with Pledged Shares

Figure 1: An example document for the event type of Equity Pledge, including selected sentences that are involved in multiple event records and where the event arguments scatter across sentences. We can observe that the relations between these entity mentions have intuitive patterns that could be leveraged to enhance the event extraction task. More information of entity color and complete event-related relations can be found in Appendix A.2.

the relation information between arguments by designing a tailored transformer structure. This structure can cover multi-scale and multi-amount relations and is general for different relation modeling situations. We name the structure as Relation-augmented Attention Transformer (RAAT). To fully leverage the relation information, we introduce a relation prediction task into the ReDEE framework and adopt multi-task learning method to optimize the event extraction task. We conduct extensive experiments on two public datasets. The results demonstrate the effectiveness of modeling the relation information, as well as our proposed framework and method.

In summary, our contributions are as follows:

- We propose a Relation-augmented Document-level Event Extraction (ReDEE) framework. It is the first time that relation information is implemented in the document-level event extraction field.
- We design a novel Relation-augmented Attention Transformer (RAAT). This network is general to cover multi-scale and multi-amount relations in DEE.
- We conduct extensive experiments and the results demonstrate that our method outperform the baselines and achieve state-of-the-art performance by 1.6% and 2.8% F1 absolute increasing on two datasets respectively.

2 Related Work

2.1 Sentence-level Event Extraction

Previously, most of the related works focus on sentence-level event extraction. For example, a neural pipeline model is proposed to identify triggers first and then extracts roles and arguments (Chen et al., 2015). Then a joint model is created to extract triggers and arguments simultaneously via multi-task learning (Nguyen et al., 2016; Sha et al., 2018). To utilize more knowledge, some studies propose to leverage document contexts (Chen et al., 2018; Zhao et al., 2018), pre-trained language models (Yang et al., 2019), and explicit external knowledge (Liu et al., 2019a; Tong et al., 2020). However, these sentence-level models fail to extract multiple qualified events spanning across sentences, while document-level event extraction is a more common need in real-world scenarios.

2.2 Document-level Event Extraction

Recently, DEE has attracted a great attention from both academic and industrial communities. At first, the event is identified from a central sentence and other arguments are extracted from neighboring sentences separately (Yang et al., 2018). Later, an innovative end-to-end model Doc2EDAG, is proposed (Zheng et al., 2019), which can generate event records via an entity-based directed acyclic graph to fulfill the document-level event extraction effectively. Based on Doc2EDAG, there are some variants appearing. For instance, GIT (Xu et al., 2021b) designs a heterogeneous graph interaction network to capture global interaction information among different sentences and entity mentions. It also introduces a specific Tracker module to memorize the already extracted event arguments for assisting record generation during next iterations. DE-PPN (Yang et al., 2021) is a multi-granularity model that can generate event records via limiting the number of record queries. Not long ago, a pruned complete graph-based non-autoregressive model PTPCG was proposed to speedup the record decoding and get competitive overall evaluation results (Zhu et al., 2021). In summary, although those existing works target for solving across-sentence and multi-event issues of the DEE task from various perspectives, to our best knowledge, we conduct a pioneer investigation on relation modeling towards this research field in this paper.

2.3 Trigger-aware Event Extraction

Previously a lot of works((Ji and Grishman, 2008; Liao and Grishman, 2010; Li et al., 2013; Chen et al., 2015; Nguyen et al., 2016; Liu et al., 2018)) deal with event extraction in two stages: firstly, trigger words are detected, which are usually nouns or verbs that clearly express event occurrences. And secondly, event arguments, the main attributes of events, are extracted by modeling relationships between triggers and themselves. In our work, we unify task as a whole to avoid error propagation between sub-tasks.

3 Preliminaries

Firstly, we clarify several key concepts in event extraction tasks. 1) ***entity***: a real world object, such as person, organization, location, etc.2) ***entity mention***: a text span in document referring to an entity object. 3) ***event role***: an attribute corresponding a pre-defined field in an event. 4) ***event argument***: an entity playing a specific event role. 5) ***event record***: a record expressing an event itself, including a series of event arguments.

In document-level event extraction task, one document can contain multiple event records, and an event record may miss a small set of event arguments. Further more, a entity can have multiple event mentions.

4 Methodology

In this section, we introduce the proposed architecture first and then the key components in detail.

4.1 Architecture Overview

End-to-end training methods for DEE usually involve a pipeline paradigm, including three subtasks: named entity recognition, event role prediction and event argument extraction. In this paper, we propose the Relation-augmented Document-level Event Extraction (ReDEE) framework coordinated with the paradigm. Our framework features leverage the relation dependency information in both encoding and decoding stages. Moreover, a relation prediction task is added into the framework to fully utilize the relation knowledge and enhance the event extraction task.

More specifically, shown in Figure 2, there are four key components in our ReDEE framework: Entity Extraction and Representation(EER), Document Relation Extraction(DRE), Entity and Sentence Encoding(ESE), and Event Record Generation(ERG). In the following, we would introduce the detailed definition of each component.

4.2 Entity Extraction and Representation

We treat the component of entity extraction as a sequence labeling task. Given a document D with multiple sentences $\{s_1, s_2, ..., s_i\}$, we use a native transformer encoder to represent the token sequence. Specifically, we use the BERT (Devlin et al., 2019) encoder pre-trained in Roberta setting (Liu et al., 2019). Then we use the Conditional Random Field(CRF) (Lafferty et al., 2001) to classify token representations into labels of named entities. We adopt the classical BIOSE sequence labeling scheme. The labels are predicted by the following calculation: $\hat{y}_{ne} = CRF(Trans(D))$. Then all the intermediate embeddings of extracted entity mentions and sentences are concatenate into a matrix $M_{ne+s} \in \mathbb{R}^{(j+i)\times d_e}$ by max-pooling operation on each sentence and entity mention span, where j and i are the numbers of entity mentions and sentences, and d_e is the dimension of embeddings. The loss function for named entity recognition is

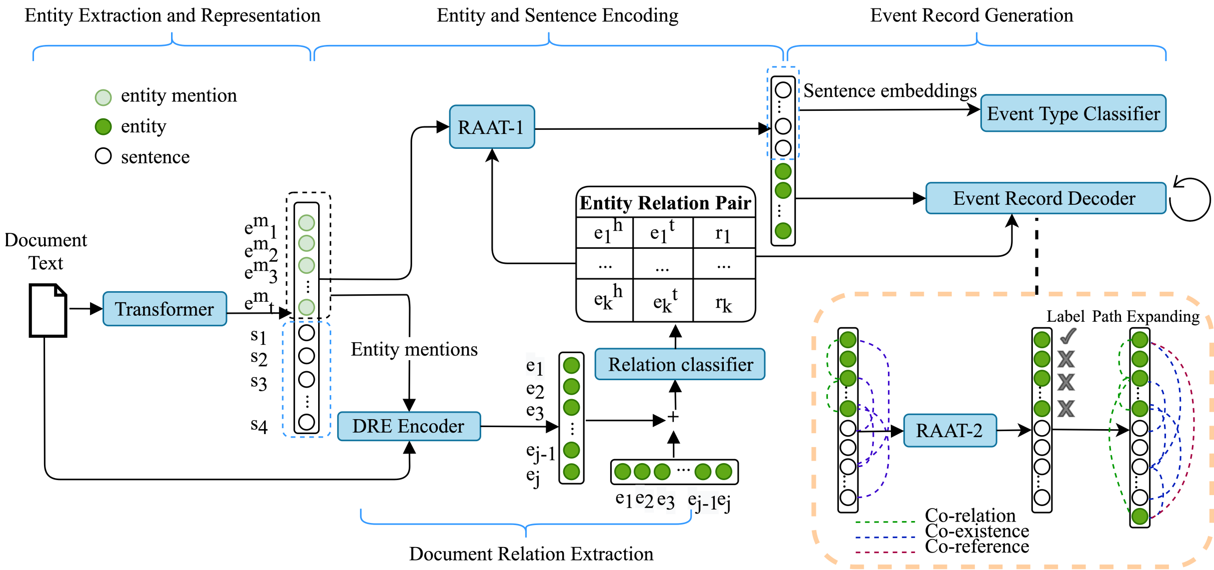

Figure 2: Overall of our proposed ReDEE framework.

denoted:

$$\mathcal{L}_{ne} = - \sum_{s_i \in D} logP(y_i|s_i) \tag{1}$$

where s_i denotes the i^{th} sequence sentence in document, and y_i is the corresponding ground truth label sequence.

4.3 Document Relation Extraction

The DRE component takes the document text (D) and entities ($\{e_1, e_2, ..., e_j\}$) extracted in the previous step as inputs, and outputs the relation pairs among entities, in a form of triples ($\{[e_1^h, e_1^t, r_1], [e_2^h, e_2^t, r_2], ..., [e_k^h, e_k^t, r_k]\}$). $[e_k^h, e_k^t, r_k]$ means the head entity, the tail entity and the relationship of the k^{th} triple respectively.

An important aspect is how to define and collect the relations from data. Here we assume that every two arguments in an event record can be connected by a relation. For example, *Pledger* and *Pledgee* in the *EquityPledge* event could have a relation named as *Pledge2Pledgee*, and the order of head and tail entities is determined by the pre-order of event arguments (Zheng et al., 2019). In this way, every event record with n arguments could create C_n^2 relation samples. Note that this method to build relations is general to event extraction tasks from various domains, and the supervised relation information just comes from event record data itself, without any extra human labeling work. We do statistics for the relation types for ChiFinAnn dataset. Table 1 shows a snippet of statistics and the full edition can be found in Appendix A.3.

Relation Type	#Train	#Dev	#Test
Pledger2PledgedShares	20002	2567	2299
Pledger2Pledgee	20002	2567	2299
PledgedShares2Pledgee	20002	2567	2299
Start2EndDate	19615	2239	1877
Pledger2TotalHoldingShares	18552	2412	2173

Table 1: The example relations with top 5 quantities in the ChiFinAnn dataset. The complete statistic can refer to the Appendix A.3.

To predict the argument relations in this step, we adopt the structured self attention network (Xu et al., 2021a) which is the latest method for document-level relation extraction. However, different from previous work using multi-class binary cross-entropy loss, we use normal cross-entropy loss to predict only one label for each entity pair. The relation type is inferred by this function:

$$\hat{y}_{i,j} = argmax(e_i^T W_r e_j) \tag{2}$$

where $e_i, e_j \in \mathbb{R}^d$ denote entity embedding from encoder module of DRE and d is the dimension of embeddings. $W_r \in \mathbb{R}^{d \times c \times d}$ denotes biaffine matrix trained by DRE task and c is the total number of relations. And the loss function for optimize the relation prediction task is denoted:

$$\mathcal{L}_{dre} = - \sum_{y_{i,j} \in Y} logP(y_{i,j}|D) \tag{3}$$

where $y_{i,j}$ denotes ground truth label between the i^{th} and j^{th} entity, D for document text and Y for set of all relation pairs among entities.

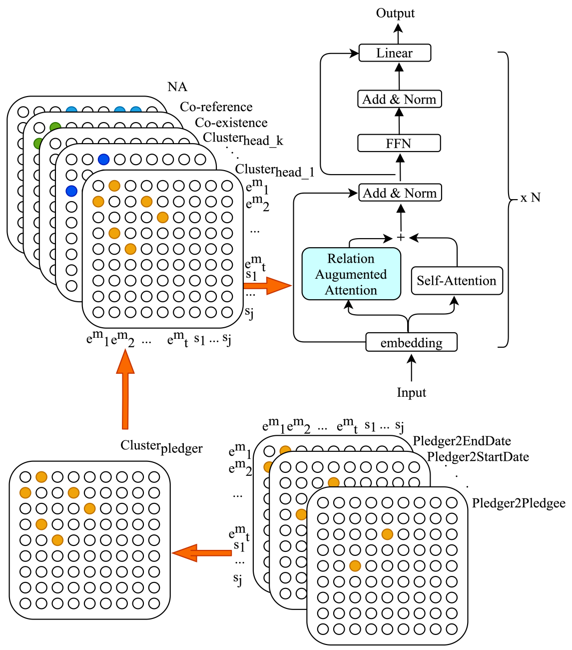

Figure 3: RAAT structure. Firstly each relation between entities and sentences are represented as matrices. Then the matrices are clustered by the head entities. At last the clustered matrices are integrated into the transformer structure for attention calculation.

4.4 Entity and Sentence Encoding

Now we have embeddings of entity mentions and sentences from EER component and a list of predicted triple relations from DRE component. Then this component encodes data mentioned above and output embeddings effectively integrated with relation information. In this subsection, we would introduce the method that translates triple relations to calculable matrices and the novel RAAT structure for encoding all the above data.

4.4.1 Entity and Sentence Dependency

First, we introduce a mechanism: entity and sentence dependency, which not only includes relation triples, but also describes links among sentences and entities beyond triples.

Co-relation and *Co-reference* are defined to represent entity-entity dependency. For the former one, two entities have a *Co-relation* dependency between them if they belong to a predicted relation triple. Entity pairs are considered having different *Co-relation* if their involved triples have different relations. *Co-reference* shows dependency between entity mentions pointing to same entities. That is, if an entity has several mentions existing across document, then each two of them has *Co-reference* dependency. However, in the case that

	sentence	entity
sentence	NA	Co-existence/NA
entity	Co-existence/NA	Co-relation/Co-reference/NA

Table 2: All types of dependency among sentences and entities

head and tail entities in relation triple are the same (i.e. *StartDate* and *EndDate* share same entities in some event records), then *Co-relation* and *Co-reference* are both held between them.

We use *Co-existence* to describe dependency between entities and sentences where entity mentions come from. To be more specific, the entity mention together with its belonged sentence has *Co-existence*. For remaining entity-entity and entity-sentence pairs without any dependency mentioned above, we uniformly treat them as *NA* dependency.

Table 2 shows the complete dependency mechanism. *Co-relation* differs from *NA*, *Co-reference*, and *Co-existence* in that it has several sub-types, with number equaling to that of relation types defined in document relation extraction task.

4.4.2 RAAT

In order to effectively encode entity and sentence dependencies, we design the RAAT which takes advantage of a calculable matrix representing dependencies and integrates it into attention computation. According to the architecture shown in Figure 3, RAAT is inherited from native transformer but has a distinct attention computation module which is made up of two parts: self-attention and relation-augmented attention computation.

Given a document shown as $D = \{s_1, s_2, ...s_j\}$, all entity mentions in this document as $E^m = \{e_1^m, e_2^m, ..., e_t^m\}$, where e_i^m denotes entity mentions with the superscript m denotes mention, and the subscript i denotes index, and a list of triples $\{[e_1^h, e_1^t, r_1], [e_2^h, e_2^t, r_2], ..., [e_k^h, e_k^t, r_k]\}$, we build a matrix $T \in \mathbb{R}^{c \times (t+j) \times (t+j)}$ where c for the number of dependencies, and t and j for the number of sentences and entity mentions respectively. T is comprised of c matrices with same dimensions $R \in \mathbb{R}^{(t+j)\times(t+j)}$, and each R represents one type of dependency $r \in \{Co-relation_k, Co-reference, Co-existence, NA\}, k = 1, 2, ...N$, N as the number of relation types. For element within T, $t_{k,i,j}$ represent the dependency between $node_i$ and $node_j$. Specifically, $t_{k,i,j} = 1$ if they have the k^{th} dependency, otherwise, $t_{k,i,j} = 0$. Here, $node_k \in \{e_1^m, e_2^m, ..., e_t^m, s_1, s_2, ...s_j\}$ can be either entity mention or sentence.

However, T would be giant and sparse if we use the above strategy. To squeeze T and decrease training parameters, we cluster *Co-relation* dependency based on the type of head entity in relation triple. For example, *Pledger2Pledgee* and *Pledger2PledgedShares* are clustered as one Co-relation dependency, and two matrice R_a and R_b corresponding to them are merged into one matrix. As a result, we finally get $T \in \mathbb{R}^{(3+H)\times(t+j)\times(t+j)}$ where H denotes the number of head entity type in *Co-relation*, and 3 covers *NA*, *Co-reference*, and *Co-existence*. Let $X \in \mathbb{R}^{(t+j)\times d}$ as input embeddings of attention module, $W_{rq}, W_{rk}, W_q, W_k, W_v \in \mathbb{R}^{d\times d}$, $M \in \mathbb{R}^{(3+H)\times d\times d}$ as weight matrices, we compute relation-augmented attention in the following steps:

$$Q_r = XW_{rq}, K_r = XW_{rk} \tag{4}$$

$$S_a = \frac{\sum_{i=1}^{3+H} Q_r M[i,:,:] K_r^T \cdot T[i,:,:]}{\sqrt{d}} + bias_i \tag{5}$$

where S_a denotes score matrix of relation-augmented attention, $\cdot$ denotes element-wise multiplication. We compute self attention score and combine it with S_a in the following way:

$$Q = XW_q, K = XW_k, W_v = XW_v \tag{6}$$

$$S_b = \frac{QK^T}{\sqrt{d}} \tag{7}$$

$$O = (S_a + S_b)V \tag{8}$$

where O is the output of attention module. Similar to the structure of native transformer, RAAT has multiple identical blocks stacking up layer by layer. Furthermore, T is extensive since the number of *Co-relation* can be selected. RAAT can be adaptive to the change of input length, which is equivalent to the total number of sentences and entity mentions.

4.5 Event Record Generation

With the outputs from previous component, the embeddings of entities and sentences, this ERG component actually includes two sub-modules: event type classifier and event record decoder.

4.5.1 Event Type Classifier

Given the embeddings of sentences, we adopt several binary classifiers on every event type to predict whether the corresponding event is identified or not. If there is any classifier identifying an event type, the following event record decoder would be activated to iteratively generate every argument for the corresponding event type. The loss function to optimize this classifier is as the following:

$$\mathcal{L}_{pred} = -\sum_i log(P(y_i|S)) \tag{9}$$

where y_i denotes the label of the i^{th} event type, $y_i = 1$ if there exists event record with event type i, otherwise, $y_i = 0$. S denotes input embeddings of sentences.

4.5.2 Event Record Decoder

To iteratively generate every argument for a specific event type, we refer to the entity-based directed acyclic graph (EDAG) method (Zheng et al., 2019). EDAG is a sequence of iterations with the length equaling to number of roles for certain event type. The objective of each iteration is to predict event argument of certain event role. Inputs of each iteration are come up with entities and sentences embeddings. And the predicted arguments of outputs will be a part of inputs for next iteration. However, different from EDAG, we substitute its vanilla transformer part with our proposed RAAT structure (i.e. RAAT-2 as shown in Figure 2). More specifically, the EDAG method uses a memory structure to record extracted arguments and adds role type representation to predict current-iteration arguments. However, this procedure hardly captures dependency between entities both in memory and argument candidates and sentences. In our method, RAAT structure can connect entities in memory and candidate arguments via relation triples extracted by the DRE component, and it can construct a structure to represent dependencies. In detail, before predicting event argument for current iteration, Matrix T is constructed in the way shown above so that dependency is integrated into attention computation. After extracting the argument, it is added into memory, meanwhile, a new T is generated to adapt next iteration prediction.

Therefore, the RAAT can strengthen the relation signal for attention computation. The RAAT-2 has the same structure with RAAT-1 but independent parameters. The formal definition of loss function for event recorder decoder is:

$$\mathcal{L}_a = -\sum_{v\in V_D}\sum_e log(P(y_e|(v,s))) \tag{10}$$

where V_D denotes node set in event records graph, v denotes extracted event arguments of event record by far, s denotes embedding of sentences and event argument candidates, and y_e denotes label of argument candidate e in current step. $y_e = 1$ means e is the ground truth argument corresponding to current step event role, otherwise, $y_e = 0$.

4.6 Model Training

To train the above four components, we leverage the multi-task learning method (Collobert and Weston, 2008) and integrate the four corresponding loss functions together as the following:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{ne} + \lambda_2 \mathcal{L}_{dre} + \lambda_3 \mathcal{L}_{pred} + \lambda_4 \mathcal{L}_a \qquad (11)$$

where the λ_i pre-set to balance the weight among the four components.

5 Experiments

In this section, we report the experimental results to prove the effectiveness of our proposed method. In summary, the experiments could answer the following three questions:

- To what degree does the ReDEE model outperform the baseline DEE methods?
- How well does ReDEE overcome across-sentence and multi-event issues?
- In what level does the each key component of ReDEE contribute to the final performance?

5.1 Datasets

DEE is a relatively new task and there are only a few datasets published. In our experiments we adopt two public Chinese datasets, i.e. **ChiFinAnn** (Zheng et al., 2019) and **DuEE-fin** (Li, 2021).

ChiFinAnn includes 32,040 documents with 5 types of events, involving in equity-related activities for the financial domain. Statistics show that about 30% of the documents contain multiple event records. We randomly split the dataset into train/dev/test sets in the ratio of 8/1/1. Readers can refer to the original paper for details.

DuEE-fin is also from the financial domain with around 11,900 documents in total. The dataset is downloaded from an online competition website*. Since there is no ground truth publicly available for the test set, we can only submit our extracted results to the website as a black-box online evaluation. Compared to ChiFinAnn, there are two differences. The DuEE-fin dataset has 13 different event types and its test set includes a large size of document samples that do not have any event records, which both make it more complicated. We get the distribution information of the dataset from Appendix A.1.

5.2 Baselines and Metrics

Five different baseline models are taken into consideration: 1) **DCFEE** (Yang et al., 2018), the first model proposed to solve DEE task. 2) **Doc2EDAG** (Zheng et al., 2019), proposed an end-to-end model which transforms DEE as directly filling event tables with entity-based path expending. 3) **DE-PPN** (Yang et al., 2021), a pipeline model firstly introducing the non-autoregressive mechanism. 4) **GIT** (Xu et al., 2021b), a model using heterogeneous graph interaction network as encoder and maintaining a global tracker during the decoding process. 5) **PTPCG** (Zhu et al., 2021), a light-weighted and latest DEE model.

For evaluation metrics, we use precision, recall, and F1 score at the entity argument level for fair comparison with baselines. The overall "Avg" in the result tables denotes the micro average value of precision, recall, and F1 score. We conduct several offline evaluations for ChiFinAnn, but only an online test for DuEE-fin.

5.3 Settings

In our implementation, for text processing, we consistently set the maximum sentence number and the maximum sentence length as 128 and 64 separately. We use BERT encoder in the EER component for fine-tuning and Roberta-chinese-wwm (Yiming et al., 2020) as the pre-trained model. Both RAAT-1 and RAAT-2 have four layers of identical blocks. More training details can be found in Appendix A.5.

5.4 Results and Analysis

Overall Performance Table 3 shows the comparison between baselines and our ReDEE model on the ChiFinAnn dataset. The ReDEE can achieve the state-of-the-art performance in terms of micro average recall and F1 scores on almost every type of events (i.e. EF, ER, EU, EO, EP), consistent with the Avg. results increased by 1.5% and 1.6% respectively. Our model also performs competitively well on precision results.

Table 4 shows the comparison results of our model with baselines on the developing set of

*https://aistudio.baidu.com/aistudio/competition/detail/46

Model	EF			ER			EU			EO			EP			Avg		
	P.	R.	F1.	P.	R.	F1.	P.	R.	F1.	P.	R.	F1.	P.	R.	F1.	P.	R.	F1.
DCFEE-S†	61.1	37.8	46.7	84.5	86.0	80.0	60.8	39.0	47.5	46.9	46.5	46.7	64.2	49.8	56.1	67.7	54.4	60.3
DCFEE-M†	44.6	40.9	42.7	75.2	71.5	73.3	51.4	41.4	45.8	42.8	46.7	44.6	55.3	52.4	53.8	58.1	55.2	56.6
Greedy-Dec†	78.5	45.6	57.7	83.9	75.3	79.4	69.0	40.7	51.2	64.8	40.6	50.0	82.1	40.4	54.2	80.4	49.1	61.0
Doc2EDAG†	78.7	64.7	71.0	90.0	86.8	88.4	80.4	61.6	69.8	77.2	70.1	73.5	76.7	73.0	74.8	80.3	75.0	77.5
GIT†	**78.9**	68.5	73.4	**92.3**	89.2	**90.8**	**83.9**	66.6	74.3	80.7	72.3	76.3	78.6	76.9	77.7	82.3	78.4	80.3
DE-PPN♠	78.2	69.4	73.5	89.3	85.6	87.4	69.7	**79.9**	74.4	81.0	71.3	75.8	**83.8**	73.7	78.4	-	-	-
PTPCG♣	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**88.2**	69.1	79.4
ReDEE(ours)	78.0	**70.6**	**74.1**	91.1	**90.3**	90.7	82.5	69.2	**75.3**	**83.7**	**73.1**	**78.1**	81.7	**78.6**	**80.1**	84.0	**79.9**	**81.9**

Table 3: Comparison of event extraction between baselines and our ReDEE model on the ChiFinAnn dataset. The missing parts are caused by the inaccessibility of baseline codes. †: results from (Xu et al., 2021b); ♠: results from (Yang et al., 2021); ♣: results from (Zhu et al., 2021).

Model	Dev			Online test		
	P.	R.	F1.	P.	R.	F1.
Doc2EDAG♣	73.7	59.8	66.0	67.1	51.3	58.1
GIT♣	75.4	61.4	67.7	**70.3**	46.0	55.6
PTPCG♣	71.0	61.7	66.0	66.7	54.6	60.0
ReDEE(ours)	**77.0**	**72.0**	**74.4**	69.2	**57.4**	**62.8**

Table 4: Comparison of event extraction between baselines and our ReDEE model on the DuEE-fin dataset. ♣: results from (Zhu et al., 2021).

Model	I	II	III	IV
DCFEE-S†	64.6	70.0	57.7	52.3
DCFEE-M†	54.8	54.1	51.5	47.1
Greedy-DEC†	67.4	68.0	60.8	50.2
Doc2EDAG†	79.6	82.4	78.4	72.0
GIT†	81.9	85.7	80.0	75.7
ReDEE(ours)	**83.9**	**85.8**	**81.7**	**77.9**

Table 5: F1 scores on four sets growing with average number of sentences involved in event records. †: results from (Xu et al., 2021b).

DuEE-fin and its online testing. Seeing from former results, our model outperforms in a great leap by increasing 6.7% on F1 score. For the online testing evaluation, our model has a distinct growth of 2.8% on F1 score than the baselines. This experiment demonstrates our model could achieve a superior performance than existing methods.

Argument Scattering The across-sentence issue widely exists in datasets. By our statistics, the training sets of ChiFinAnn and DuEE-fin have about 98.0% and 98.9% records that scatter across sentences respectively. To evaluate the performance of our model in different argument scattering degree, we compute the average number of sentences involved in records for each document and sort them in the increasing average number order. Then, all documents for testing are evenly divided into four sets, namely, I, II, III and IV, which means the I set is a cluster of documents that have the smallest average number of involved sentences while the IV set has the largest ones. According to table 5, our model outperforms other baseline models in all settings, and meets the largest growth of 2.2% F1 score in IV, the most challenging set of all. It indicates that our model is capable of capturing longer dependency of records across sentences via relation dependency modeling, thus alleviating the argument scattering challenge.

Single v.s. Multi Events To illustrate how well our model performs in the multi-event aspect, we split the test set of ChiFinAnn into two parts: one for documents with single event record, and the other for documents including multiple events. Table 6 shows the comparison results of all baselines and ReDEE. We find ReDEE performs much better in the multi-event scenario and outperforms baseline models dramatically in all five event types, improving ranging from 1.9% to 3.2% F1 scores. The results suggest that our relation modeling method is more effective to overcome the multi-event issue than existing baseline models.

5.5 Ablation Study

To probe the impact of RAAT structure for different components in ReDEE, we conduct ablation studies on whether to use RAAT or vanilla transformer.

In this experiment, we implement tests on three variants: 1) *-RAAT-1* substitutes the RAAT in the ESE component with vanilla transformer. 2) *-RAAT-2* substitutes the RAAT in the event record generation module with vanilla transformer. 3) *-RAAT-1&2* substitutes the RAATs in both the above places with vanilla transformers, so that our model degrades to only import a relation extraction task via multi-task learning.

The results in Table 7 indicate that both two RAATs have positive influence on our model. Especially on ChiFinAnn, RAAT-2 makes more con-

Model	EF		ER		EU		EO		EP		Avg		
	S.	M.	S.	M.	S.	M.	S.	M.	S.	M.	S.	M.	S.&M.
DCFEE-S†	55.7	38.1	83.0	55.5	52.3	41.4	49.2	43.6	62.4	52.2	69.0	50.3	60.3
DCFEE-M†	45.3	40.5	76.1	50.6	48.3	43.1	45.7	43.3	58.1	51.2	63.2	49.4	56.6
Greedy-Dec†	74.0	40.7	82.2	50.0	61.5	35.6	63.4	29.4	78.6	36.5	77.8	37.0	61.0
Doc2EDAG†	79.7	63.3	90.4	70.7	74.7	63.3	76.1	70.2	84.3	69.3	81.0	67.4	77.5
GIT†	81.9	65.9	**93.0**	71.7	**82.0**	64.1	80.9	70.6	85.0	73.5	87.6	72.3	80.3
DE-PPN♠	**82.1**	63.5	89.1	70.5	79.7	66.7	80.6	69.6	**88.0**	73.2	-	-	-
PTPCG♣	-	-	-	-	-	-	-	-	-	-	**88.2**	69.1	79.4
ReDEE(ours)	79.7	**69.1**	92.7	**73.6**	79.9	**69.2**	**81.6**	**73.7**	86.3	**76.5**	87.9	**75.3**	**81.9**

Table 6: Comparison of event extraction between singular (S.) and multiple (M.) event documents on the ChiFinAnn. †: results from (Xu et al., 2021b); ♠: results from (Yang et al., 2021); ♣: results from (Zhu et al., 2021).

Model	ChiFinAnn			DuEE-fin		
	P.	R.	F1.	P.	R.	F1.
ReDEE	84.0	79.9	81.9	69.2	57.4	62.8
-RAAT-1	+0.4	-1.1	-0.4	+1.5	-1.7	-0.5
-RAAT-2	+1.3	-2.4	-0.7	+0.8	-3.2	-1.7
-RAAT-1&2	-3.1	-0.1	-1.5	-1.3	-5.1	-3.7

Table 7: Ablation studies on ReDEE variants for RAAT.

tribution than RAAT-1, with a decrease of 0.7% versus 0.4% in F1 scores once been substituted. After replacing both two RAATs, the value of relation extraction task becomes more weak and the model encounters a 1.5% drop in F1 score. When it comes to DuEE-fin, a similar phenomenon can be observed that both the RAATs can contribute positively to our model.

6 Conclusion

In this paper, we investigate a challenging task of event extraction at document level, towards the across-sentence and multi-event issues. We propose to model the relation information between event arguments and design a novel framework ReDEE. This framework features a new RAAT structure which can incorporate the relation knowledge. The extensive experimental results can demonstrate the effectiveness of our proposed method which makes the state-of-the-art performance on two benchmark datasets. In the future, we will make more efforts to accelerate training and inference process.

Acknowledgements

We thank the anonymous reviewers for their careful reading of our paper and their many insightful comments and suggestions. This work was supported by Tencent Cloud and Tencent Youtu Lab.

References

David Ahn. 2006. The stages of event extraction. In *In Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8.

Antoine Bosselut, Ronan Le Bras, and Yejin Choi. 2021. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In *AAAI*.

Jordan Boyd-Graber and Benjamin Börschinger. 2020. What question answering can learn from trivia nerds. In *ACL*, pages 7422–7435.

Qingqing Cao, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. 2020. Deformer: Decomposing pre-trained transformers for faster question answering. In *ACL*, pages 4487–4497.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL*, pages 167–176.

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *EMNLP*, pages 1267–1276.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *EMNLP*, pages 671–683.

Li Gao, Jia Wu, Zhi Qiao, Chuan Zhou, Hong Yang, and Yue Hu. 2016. Collaborative social group influence for event recommendation. In *CIKM*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *ACL*, pages 254–262.

John Lafferty, Andrew McCallum, and Fernando C.N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event extraction as multi-turn question answering. In *EMNLP*, pages 829–838.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *ACL*, pages 73–82.

X Li. 2021. Duee-fin: a document-level event extraction dataset in the financial domain released by baidu. https://aistudio.baidu.com/aistudio/competition/detail/46.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *ACL*, pages 789–797.

Chun-Yi Liu, Chuan Zhou, Jia Wu, Hongtao Xie, Yue Hu, and Li Guo. 2017. Cpmf: A collective pairwise matrix factorization model for upcoming event recommendation. In *IJCNN*.

Jian Liu, Yubo Chen, and Kang Liu. 2019a. Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. In *AAAI*, pages 6754–6761.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *EMNLP*, pages 1247–1256.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. https://arxiv.org/pdf/1907.11692.pdf.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2event: Controllable sequence-to-structure generation for end-to-end event extraction. In *ACL*, pages 2795–2806.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *NAACL*, pages 300–309.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, and Alessandro Achille. 2021. Structured prediction as translation between augmented natural languages. In *ICLR*, pages 829–838.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *AAAI*.

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020. Improving event detection via open-domain trigger knowledge. In *ACL*, pages 5887–5897.

Xindong Wu, Jia Wu, Xiaoyi Fu, Jiachen Li, Peng Zhou, and Xu Jiang. 2019. Automatic knowledge graph construction: A report on the 2019 icdm/icbk contest. In *ICDM*.

Benfeng Xu, Quan Wang, Yajuan Lyu, Yong Zhu, and Zhendong Mao. 2021a. Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. In *AAAI*, pages 14149–14157.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021b. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In *ACL*, pages 3533–3546.

Haoran Yan, Xiaolong Jin, Xiangbin Meng, Jiafeng Guo, and Xueqi Cheng. 2019. Event detection with multi-order graph convolution and aggregated attention. In *EMNLP*, pages 1267–1276.

Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018. Dcfee: A document-level chinese financial event extraction system based on automatically labeled training data. In *ACL*, pages 50–55.

Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. 2021. Document-level event extraction via parallel prediction networks. In *ACL*, pages 6298–6308.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pretrained language models for event extraction and generation. In *ACL*, pages 5284–5294.

Cui Yiming, Che Wanxiang, Liu Ting, Qin Bing, Wang Shijin, and Hu Guoping. 2020. Revisiting pre-trained models for chinese natural language processing. In *EMNLP: Findings*, pages 657–668.

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Document embedding enhanced event detection with hierarchical and supervised attention. In *ACL*, pages 414–419.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2edag: An end-to-end document-level framework for chinese financial event extraction. In *EMNLP*, pages 337–346.

Tong Zhu, Xiaoye Qu, Wenliang Chen, Zhefeng Wang, Baoxing Huai, Nicholas Jing Yuan, and Min Zhang. 2021. Efficient document-level event extraction via pseudo-trigger-aware pruned complete graph. https://arxiv.org/pdf/2112.06013.pdf.

Event Type	#Train.	#Dev.
ShareRedemption	1309	243
FinanceDeficit	1062	163
Pledge	1027	160
EnterpriseAcquisition	934	142
BidWin	915	134
ExecutiveChange	901	134
ShareholderHoldingDecrease	876	147
PledgeRelease	728	118
CorporateFinace	535	72
CompanyListing	482	82
ShareholderHoldingIncrease	321	62
CompanyBankruptcy	236	44
Admonition	172	32
Total	**9498**	**1533**

Table 8: Distribution of Duee-fin dataset.

A Appendix

In the appendix, we incorporate the following details that are omitted in the main body due to the space limit.

A.1 Distribution of Event Type DuEE-fin

Table 8 shows the complete event type and corresponding distribution of DuEE-fin dataset. Overall, there are 13 event types in total with uneven distribution. Only train and development sets are shown since test set is not publicly available.

A.2 Complete Relation Triples

Table 9 demonstrates the complete of relation triples of the document event extraction example shown in Figure 1.

Entities in blue are involved in both two event records, while those in green and orange are exclusive to record 1 and 2 respectively. Heavy coupling of arguments among events increases the difficulty of multi-event issue.

A.3 Relation Statistics for ChiFinAnn

Table 10 shows the relation statistics of ChiFinAnn dataset. There are 85 relation types in total, and train, development, and test sets have similar pattern in distribution.

A.4 Case Study

Figure 4 shows the prediction results of our model and the best baseline model GIT on the example in Figure 1. Compared with the ground truth, our model correctly predicts all event arguments except one, while GIT only captures one event, with an argument missed. This example explicitly shows the superiority of our model in dealing with multi-events issue.

Event-related Relations between Entities Table (Two Examples)		
Head Entity	Tail Entity	Relation
[ORG1]	[ORG2]	Pledger2Pledgee
[ORG1]	[SHARE1]	Pledger2PledgedShares
[ORG1]	[SHARE2]	Pledger2TotalHoldingShares
[ORG1]	[RATIO1]	Pledger2TotalHoldingRatio
[ORG1]	[TIME1]	Pledger2StartDate
[ORG1]	[TIME2]	Pledger2ReleaseDate
[SHARE1]	[ORG2]	PledgedShares2Pledgee
[SHARE1]	[SHARE2]	PledgedShares2TotalHoldingShares
[SHARE1]	[RATIO1]	PledgedShares2TotalHoldingRatio
[SHARE1]	[TIME1]	PledgedShares2ShartDate
[SHARE1]	[TIME2]	PledgedShares2ReleaseDate
[ORG2]	[SHARE2]	Pledgee2TotalHoldingShares
[ORG2]	[RATIO1]	Pledgee2TotalHoldingRatio
[ORG2]	[TIME1]	Pledgee2StartDate
[ORG2]	[TIME2]	Pledgee2EndDate
[SHARE2]	[RATIO1]	TotalHoldingShares2TotalHoldingRatio
[SHARE2]	[SHARE2]	TotalHoldingShares2TotalPledgedShares
[SHARE2]	[TIME1]	TotalHoldingShares2StartDate
[SHARE2]	[TIME2]	TotalHoldingShares2ReleaseDate
[RATIO1]	[SHARE2]	TotalHoldingRatio2TotalPledgedShares
[RATIO1]	[TIME1]	TotalHoldingRatio2StartDate
[RATIO1]	[TIME2]	TotalHoldingRatio2ReleaseDate
[TIME1]	[TIME2]	StartDate2ReleaseDate
[ORG1]	[ORG3]	Pledger2Pledgee
[SHARE2]	[ORG3]	PledgedShares2Pledgee
[ORG3]	[SHARE2]	Pledgee2TotalPledgedShares
[ORG3]	[RATIO1]	Pledgee2TotalHoldingRatio
[ORG3]	[TIME2]	Pledgee2StartDate

Table 9: Complete relation triplets.

A.5 More Training Settings

For all native transformers and RAATs, the dimensions of hidden layers and feed-forward layers are set to 768 and 1,024 respectively. During training, we set the learning rate $lr = 5e^{-5}$, batch size $b = 64$. The four loss weights are set to $\lambda_1 = \lambda_3 = 0.05, \lambda 2 = 1.0, \lambda_4 = 0.95$. We use 8 V100 GPUs and set gradient accumulation steps to 8. The train epoch are set to 100, and the best epoch are selected by the best validation score on development set for the evaluation of test set. And we use Adam to optimize the whole learning task.

Relation Statistics for ChiFinAnn			
Relation type	**#Train.**	**#Dev.**	**#Test.**
Pledger2PledgedShares	20002	2567	2299
Pledger2Pledgee	20002	2576	2299
PledgedShares2Pledgee	20002	2576	2299
StartDate2EndDate	19615	2239	1877
Pledger2TotalHoldingShares	18552	2412	2173
PledgedShares2TotalHoldingShares	18552	2412	2173
Pledgee2TotalHoldingShares	18552	2412	2173
TotalHoldingShares2TotalHoldingRatio	17403	2416	2162
Pledger2TotalHoldingRatio	16465	2180	1923
PledgedShares2TotalHoldingRatio	16465	2180	1923
Pledgee2TotalHoldingRatio	16465	2180	1923
Pledger2StartDate	15839	2247	2047
PledgedShares2StartDate	15839	2247	2047
Pledgee2StartDate	15839	2247	2047
TotalHoldingShares2StartDate	15237	2296	2106
EquityHolder2StartDate	14725	1423	1058
Pledger2TotalPledgedShares	14549	2024	1842
PledgedShares2TotalPledgedShares	14549	2024	1842
Pledgee2TotalPledgedShares	14549	2024	1842
TotalHoldingShares2TotalPledgedShares	14369	1999	1813
EquityHolder2EndDate	14357	1280	881
EquityHolder2TradedShares	14245	1269	886
TradedShares2EndDate	14003	1226	843
TradedShares2StartDate	13749	1202	836
TotalHoldingRatio2TotalPledgedShares	13221	1823	1653
TotalHoldingRatio2StartDate	13093	2023	1837
Pledger2ReleasedDate	11215	873	707
PledgedShares2ReleasedDate	11215	873	707
Pledgee2ReleasedDate	11215	873	707
TotalHoldingRatio2ReleasedDate	10949	855	698
TotalPledgedShares2StartDate	10451	1712	1596
TotalHoldingRatio2ReleasedDate	9913	775	630
TotalPledgedShares2ReleasedDate	9472	715	609
StartDate2ReleasedDate	7106	586	484
EquityHolder2LaterHoldingShares	6317	507	346
TradedShares2LaterHoldingShares	6317	507	346
Pledger2EndDate	6189	1062	1806
PledgedShares2EndDate	6189	1062	1806
Pledgee2EndDate	6189	1062	1806
TotalHoldingShares2EndDate	6125	1046	1048
EndDateLaterHoldingShares	6094	469	304
StartDate2LaterHoldingShares	5885	446	309

Relation type	**#Train.**	**#Dev.**	**#Test.**
TotalHoldingRatio2EndDate	5185	898	866
EquityHolder2AveragePrice	3732	323	164
TradedShares2AveragePrice	3732	323	164
TotoalPledgedShares2EndDate	3691	772	823
EndDate2AveragePrice	3623	310	156
StartDate2AveragePrice	3551	306	149
EndDate2ReleaseDate	2717	265	215
CompanyName2LowestTradingPrice	2455	815	1219
CompanyName2RepurchasedShares	2447	812	1219
CompanyName2HighestTradingPrice	2446	811	1216
HighestTradingPrice2LowestTradingPrice	2431	801	1205
LowestTradingPrice2RepurchasedShares	2422	800	1207
HighestTradingPrice2RepurchasedShares	2413	796	1201
LaterHoldingShares2AveragePrice	1703	106	61
CompanyName2RepurchaseAmount	1512	617	1803
LowestTradingPrice2RepurchaseAmount	1492	605	1068
CompanyName2ClosingDate	1488	586	998
HighestTradingPrice2RepurchaseAmount	1482	603	1066
RepurchasedShares2ClosingDate	1482	585	989
RepurchasedShares2RepurchaseAmount	1479	602	1068
LowestTradingPrice2ClosingDate	1463	574	984
HighestTradingPrice2ClosingDate	1454	570	980
EquityHolder2FrozeShares	1361	324	330
EquityHolder2LegalInstitution	1361	324	330
FrozeShares2LegalInstitution	1361	324	330
EquityHolder2TotalHoldingShares	1197	307	293
FrozeShares2TotalHoldingShares	1197	307	293
LegalInstitution2TotalHoldingShares	1197	307	293
EquityHolder2TotalHoldingRatio	1069	263	269
FrozeShares2TotalHoldingRatio	1069	263	269
LegalInstitution2TotalHoldingRatio	1069	263	269
FrozeShares2StartDate	976	221	222
LegalInstitution2StartDate	976	221	222
ClosingDateRepurchaseAmount	811	436	882
FrozeShares2EndDate	354	54	38
LegalInstitution2EndDate	354	54	38
EquityHolder2UnfrozeDate	235	8	18
FrozeShares2UnfrozeDate	235	8	18
LegalInstitution2UnfrozeDate	235	8	18
TotalHoldingShares2UnfrozeDate	194	8	18
TotalHoldingRatio2UnfrozeDate	163	7	17
StartDate2UnfrozeDate	87	4	7
EndDate2UnfrozeDate	30	2	1
Total	621010	83502	80829

Table 10: Relation statistics of ChiFinAnn dataset.

ID	Selected Sentences of a Document
S4	Chongqing Wanli New Energy Co., Ltd. (hereinafter referred to as the “Company” or “the Company”) received on September 21, 2018 that [ORG1] …
S5	On [TIME1], Nanfang Tongzheng pledged its [SHARE1] unrestricted tradable shares of the company to [ORG2] for …
S6	Nanfang Tongzheng has released all the above-mentioned [SHARE1] shares pledged to [ORG2], and … on [TIME2] …
S8	According to … [ORG3], Nanfang Tongcheng pledged its [SHARE2] unrestricted tradable shares of the company to [ORG3], …([TIME2]) until …
S9	As … [SHARE2] shares of the company, accounting for [RATIO1] of the company’s total share capital, and the cumulative number of ledged shares is [SHARE2] …

Event Records for the Equity Pledge (EP) Event Type (Ground Truth)

Record ID	Pledger	Pledged Shares	Pledgee	Total Holding Shares	Total Holding Ratio	Total Pledged Shares	Start Date	Release Date
1	[ORG1]	[SHARE1]	[ORG2]	[SHARE2]	[RATIO1]	[SHARE2]	[TIME1]	[TIME2]
2	[ORG1]	[SHARE2]	[ORG3]	[SHARE2]	[RATIO1]	[SHARE2]	[TIME2]	-

Event Records for the Equity Pledge (EP) Event Type (Our Model)

Record ID	Pledger	Pledged Shares	Pledgee	Total Holding Shares	Total Holding Ratio	Total Pledged Shares	Start Date	Release Date
1	[ORG1]	[SHARE1]	[ORG2]	[SHARE2]	[RATIO1]	[SHARE2]	[TIME1]	[TIME2]
2	[ORG1]	[SHARE2]	[ORG3]	[SHARE2]	[RATIO1]	[SHARE2]	-	-

Event Records for the Equity Pledge (EP) Event Type (GIT)

Record ID	Pledger	Pledged Shares	Pledgee	Total Holding Shares	Total Holding Ratio	Total Pledged Shares	Start Date	Release Date
1	[ORG1]	[SHARE1]	[ORG2]	[SHARE2]	[RATIO1]	-	[TIME1]	[TIME2]
-	-	-	-	-	-	-	-	-

Figure 4: Case study.