Document-level Event Extraction via Heterogeneous Graph-based Interaction Model with a Tracker

Runxin Xu[1], **Tianyu Liu**[1], **Lei Li**[3] and **Baobao Chang**[1,2*]
[1]Key Laboratory of Computational Linguistics, Peking University, MOE, China
[2]Peng Cheng Laboratory, Shenzhen, China
[3]ByteDance AI Lab
runxinxu@gmail.com, lileilab@bytedance.com
{tianyu0421,chbb}@pku.edu.cn

Abstract

Document-level event extraction aims to recognize event information from a whole piece of article. Existing methods are not effective due to two challenges of this task: a) the target event arguments are scattered across sentences; b) the correlation among events in a document is non-trivial to model. In this paper, we propose Heterogeneous **G**raph-based **I**nteraction Model with a **T**racker (GIT) to solve the aforementioned two challenges. For the first challenge, GIT constructs a heterogeneous graph interaction network to capture global interactions among different sentences and entity mentions. For the second, GIT introduces a *Tracker* module to track the extracted events and hence capture the interdependency among the events. Experiments on a large-scale dataset (Zheng et al., 2019) show GIT outperforms the existing best methods by 2.8 F1. Further analysis reveals GIT is effective in extracting multiple correlated events and event arguments that scatter across the document. Our code is available at https://github.com/RunxinXu/GIT.

1 Introduction

Event Extraction (EE) is one of the key and challenging tasks in Information Extraction (IE), which aims to detect events and extract their arguments from the text. Most previous methods (Chen et al., 2015; Nguyen et al., 2016; Liu et al., 2018; Yang et al., 2019; Du and Cardie, 2020b) focus on sentence-level EE, extracting events from a single sentence. The sentence-level model, however, fails to extract events whose arguments spread in multiple sentences, which is much more common in real-world scenarios. Hence, extracting events at the document-level is critical. It has attracted much attention recently (Yang et al., 2018; Zheng et al., 2019; Du and Cardie, 2020a; Du et al., 2020).

[1] On **Nov 6, 2014**, the company received a letter of share reduction from **Mingting Wu**, the shareholder of the company. [2] **Mingting Wu** decreased his holding of **7.2 million** shares of the company on the Shenzhen Stock Exchange on **Nov 6, 2014**. [3] The **7.2 million** shares of the company **Mingting Wu** reduced this time were transferred to **Xiaoting Wu**. [4] **Xiaoting Wu** is the daughter of **Mingting Wu**, and they were identified as persons acting in concert according to relevant regulations.

EventType	EquityHolder	TradedShares	StartDate	···
EU	Mingting Wu	7.2 million	Nov 6, 2014	···
EO	Xiaoting Wu	7.2 million	Nov 6, 2014	···

Figure 1: An example document from a Chinese dataset proposed by Zheng et al. (2019) in the financial domain, and we translate it into English for illustration. Entity mentions are colored. Due to space limitation, we only show four associated sentences and three argument roles of each event type. The complete original document can be found in Appendix C. EU: Equity Underweight, EO: Equity Overweight.

Though promising, document-level EE still faces two critical challenges. **Firstly**, the arguments of an event record may scatter across sentences, which requires a comprehensive understanding of the cross-sentence context. Figure 1 illustrates an example that one *Equity Underweight* (EU) and one *Equity Overweight* (EO) event records are extracted from a financial document. It is less challenging to extract the EU event because all the related arguments appear in the same sentence (Sentence 2). However, for the arguments of EO record, *Nov 6, 2014* appears in Sentence 1 and 2 while *Xiaoting Wu* in Sentence 3 and 4. It would be quite challenging to identify such events without considering global interactions among sentences and entity mentions. **Secondly**, a document may express several correlated events simultaneously, and recognizing the interdependency among them is

[*] Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 3533–3546
August 1–6, 2021. ©2021 Association for Computational Linguistics

fundamental to successful extraction. As shown in Figure 1, the two events are interdependent because they correspond to exactly the same transaction and therefore share the same *StartDate*. Effective modeling on such interdependency among the correlated events remains a key challenge in this task.

Yang et al. (2018) extracts events from a central sentence and query the neighboring sentences for missing arguments, which ignores the cross-sentence correspondence between augments. Though Zheng et al. (2019) takes a first step to fuse the sentences and entities information via Transformer, they neglect the interdependency among events. Focusing on single event extraction, Du and Cardie (2020a) and Du et al. (2020) concatenate multiple sentences and only consider a single event, which lacks the ability to model multiple events scattered in a long document.

To tackle the aforementioned two challenges, in this paper, we propose a Heterogeneous **G**raph-based **I**nteraction Model with a **T**racker (GIT) for document-level EE. To deal with scattered arguments across sentences, we focus on the *Global Interactions* among sentences and entity mentions. Specifically, we construct a heterogeneous graph interaction network with mention nodes and sentence nodes, and model the interactions among them by four types of edges (i.e., sentence-sentence edge, sentence-mention edge, intra-mention-mention edge, and inter-mention-mention edge) in the graph neural network. In this way, GIT jointly models the entities and sentences in the document from a global perspective.

To facilitate the multi-event extraction, we target on the *Global Interdependency* among correlated events. Concretely we propose a *Tracker* module to continually tracks the extracted event records with a global memory. In this way, the model is encouraged to incorporate the interdependency with other correlated event records while predicting.

We summarize our contributions as follows:

- We construct a heterogeneous graph interaction network for document-level EE. With different heterogeneous edges, the model could capture the global context for the scattered event arguments across different sentences.

- We introduce a novel *Tracker* module to track the extracted event records. The *Tracker* eases the difficulty of extracting correlated events, as interdependency among events would be taken into consideration.

- Experiments show GIT outperforms the previous state-of-the-art model by 2.8 F1 on the large-scale public dataset (Zheng et al., 2019) with $32,040$ documents, especially on cross-sentence events and multiple events scenarios (with 3.7 and 4.9 absolute increase on F1).

2 Preliminaries

We first clarify some important notions. a) **entity mention**: a text span within document that refers to an entity object; b) **event argument**: an entity playing a specific event role. Event roles are predefined for each event type; c) **event record**: an entry of a specific event type containing arguments for different roles in the event. For simplicity, we use **record** for short in the following sections.

Following Zheng et al. (2019), given a document composed of sentences $\mathcal{D} = \{s_i\}_{i=1}^{|\mathcal{D}|}$ and a sentence containing a sequence of words $s_i = \{w_j\}_{j=1}^{|s_i|}$, the task aims to handle three sub-tasks : 1) **entity extraction**: extracting entities $\mathcal{E} = \{e_i\}_{i=1}^{|\mathcal{E}|}$ from the document to serve as argument candidates. An entity may have multiple mentions across the document. 2) **event types detection**: detecting specific event types that are expressed by the document. 3) **event records extraction**: finding appropriate arguments for the expressed events from entities, which is the most challenging and also the focus of our paper. The task does not require to identify event triggers (Zeng et al., 2018; Liu et al., 2019b), which reduces manual effort of annotation and the application scenarios becomes more extensive.

3 Methodology

As shows in Figure 2, GIT first extracts candidate entities through sentence-level neural extractor (Sec 3.1). Then we construct a heterogeneous graph to model the interactions among sentences and entity mentions (Sec 3.2), and detect event types expressed by the document (Sec 3.3). Finally we introduce a *Tracker* module to continuously track all the records with global memory, in which we utilize the global interdependency among records for multi-event extraction (Sec 3.4).

3.1 Entity Extraction

Given a sentence $s = \{w_j\}_{j=1}^{|s|} \in \mathcal{D}$, we encode s into a sequence of vectors $\{g_j\}_{j=1}^{|s_i|}$ using Trans-

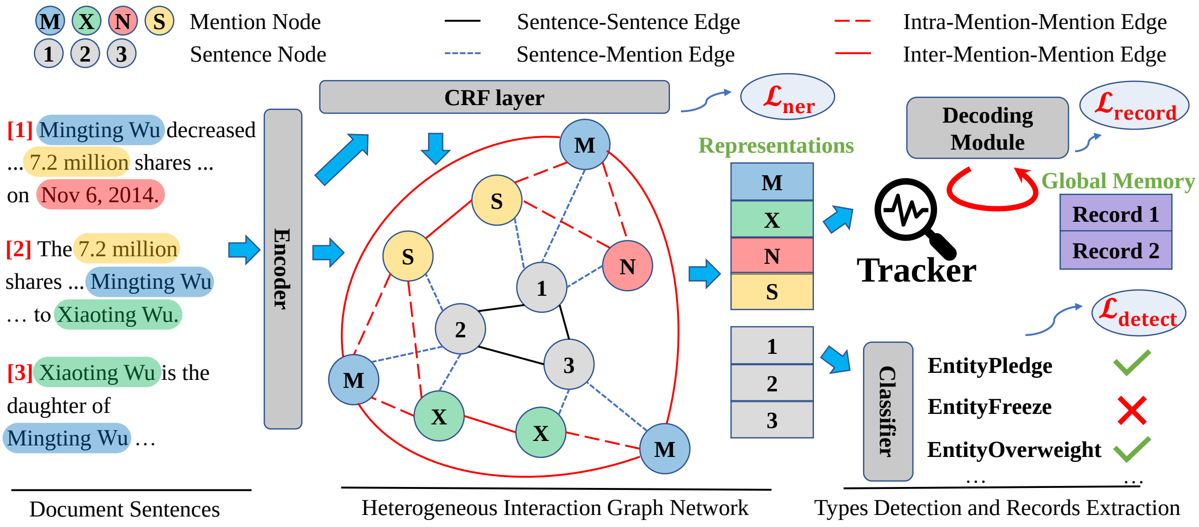

M X N S Mention Node
1 2 3 Sentence Node
Sentence-Sentence Edge
Sentence-Mention Edge
Intra-Mention-Mention Edge
Inter-Mention-Mention Edge
[1] Mingting Wu decreased ... 7.2 million shares ... on Nov 6, 2014.
[2] The 7.2 million shares ... Mingting Wu ... to Xiaoting Wu.
[3] Xiaoting Wu is the daughter of Mingting Wu ...
Encoder
CRF layer
$\mathcal{L}_{ner}$
Representations
M
X
N
S
1
2
3
Decoding Module
$\mathcal{L}_{record}$
Global Memory
Record 1
Record 2
Tracker
$\mathcal{L}_{detect}$
Classifier
EntityPledge
EntityFreeze
EntityOverweight
...
Document Sentences
Heterogeneous Interaction Graph Network
Types Detection and Records Extraction

Figure 2: Overview of our GIT. Firstly, sentences of the document are fed into the encoder to obtain contextualized representation, followed by a CRF layer to extract entities. Then GIT constructs a heterogeneous graph interaction network with mention nodes and sentence nodes, which captures the global interactions among them based on GCNs. After obtaining document-aware representations of entities and sentences, GIT detects event types and extracts records through the decoding module with a *Tracker*. The *Tracker* tracks extracted records with global memory, based on which the decoding module incorporates global interdependency among correlated event records. Different entities are marked by different colors. M: Mingting Wu. X: Xiaoting Wu. N: Nov 6, 2014. S: 7.2 million.

former (Vaswani et al., 2017):

$$\{g_1, \ldots, g_{|s|}\} = \texttt{Transformer}(\{w_1, \ldots, w_{|s|}\})$$

The word representation of w_j is a sum of the corresponding token and position embeddings.

We extract entities at the sentence level and formulate it as a sequence tagging task with BIO (Begin, Inside, Other) schema. We leverage a conditional random field (CRF) layer to identify entities. For training, we minimize the following loss:

$$\mathcal{L}_{\text{ner}} = -\sum_{s \in \mathcal{D}} \log P(y_s|s) \tag{1}$$

where y_s is the golden label sequence of s. For inference, we use Viterbi algorithm to decode the label sequence with the maximum probability.

3.2 Heterogeneous Graph Interaction Network

An event may span multiple sentences in the document, which means its corresponding entity mentions may also scatter across different sentences. Identifying and modeling these entity mentions in the cross-sentence context is fundamental in document EE. Thus we build a heterogeneous graph $\mathcal{G}$ which contains entity mention nodes and sentence nodes in the document $\mathcal{D}$. In the graph $\mathcal{G}$, interactions among multiple entity mentions and sentences can be explicitly modeled. For each entity mention node e, we initialize node embedding $h_e^{(0)} = \texttt{Mean}(\{g_j\}_{j \in e})$ by averaging the representation of the contained words. For each sentence node s, we initialize node embedding $h_s^{(0)} = \texttt{Max}(\{g_j\}_{j \in s}) + \texttt{SentPos}(s)$ by max-pooling all the representation of words within the sentence plus sentence position embedding.

To capture the interactions among sentences and mentions, we introduce four types of edges.

Sentence-Sentence Edge (S-S) Sentence nodes are fully connected to each other with S-S edges. In this way, we can easily capture the global properties in the document with sentence-level interactions, e.g., the long range dependency between any two separate sentences in the document would be modeled efficiently with S-S edges.

Sentence-Mention Edge (S-M) We model the local context of an entity mention in a specific sentence with S-M edge, specifically the edge connecting the mention node and the sentence node it belongs to.

Intra-Mention-Mention Edge (M-M$_{\text{intra}}$) We connect distinct entity mentions in the same sentences with M-M$_{\text{intra}}$ edges. The co-occurrence of mentions in a sentence indicates those mentions are likely to be involved in the same event. We explicitly model this indication by M-M$_{\text{intra}}$ edges.

Inter-Mention-Mention Edge (M-M$_{\text{inter}}$) The entity mentions that corresponds to the same entity are fully connected with each other by M-M$_{\text{inter}}$ edges. As in document EE, an entity usually corresponds to multiple mentions across sentences, we thus use M-M$_{\text{inter}}$ edge to track all the appearances of a specific entity, which facilitates the long distance event extraction from a global perspective.

In Section. 4.5, experiments show that all of these four kinds of edges play an important role in event detection, and the performance would decrease without any of them.

After heterogeneous graph construction [*], we apply multi-layer Graph Convolution Network (Kipf and Welling, 2017) to model the global interactions inspired by Zeng et al. (2020). Given node u at the l-th layer, the graph convolutional operation is defined as follows:

$$h_u^{(l+1)} = \texttt{ReLU}\left(\sum_{k\in\mathcal{K}} \sum_{v\in\mathcal{N}_k(u)\bigcup\{u\}} \frac{1}{c_{u,k}} W_k^{(l)} h_v^{(l)}\right)$$

where $\mathcal{K}$ represents different types of edges, $W_k^{(l)} \in \mathbb{R}^{d_m\times d_m}$ is trainable parameters. $\mathcal{N}_k(u)$ denotes the neighbors for node u connected in k-th type edge and $c_{u,k}$ is a normalization constant. We then derive the final hidden state h_u for node u,

$$h_u = W_a[h_u^{(0)}; h_u^{(1)}; \ldots ; h_u^{(L)}]$$

where $h_u^{(0)}$ is the initial node embedding of node u, and L is the number of GCN layers.

Finally, we obtain the sentence embedding matrix $S = [h_1^\top \; h_2^\top \; \ldots \; h_{|\mathcal{D}|}^\top] \in \mathbb{R}^{d_m\times|\mathcal{D}|}$ and entity embedding matrix $E \in \mathbb{R}^{d_m\times|\mathcal{E}|}$. The i-th entity may have many mentions, where we simply use string matching to detect entity coreference following Zheng et al. (2019) , and the entity embedding E_i is computed by the average of its mention node embedding, $E_i = \texttt{Mean}(\{h_j\}_{j\in\text{Mention}(i)})$. In this way, the sentences and entities are interactively represented in a context-aware way.

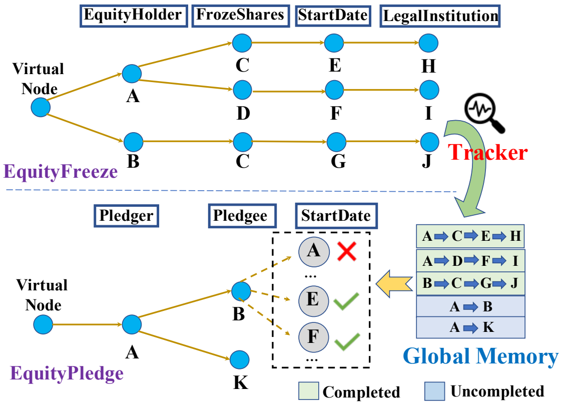


Figure 3: The decoding module of GIT. Three *Equity Freeze* records have been extracted completely, and GIT is predicting the *StartDate* role for the *Equity Pledge* records (in the dashed frame ⁞), based on the global memory where *Tracker* tracks the records on-the-fly. Both entity E and F are predicted as the legal *StartDate* role while A is not. Pre-defined argument roles are shown in the blue box, and GIT extracts records in this order. Capital letters (A-K) refer to different entities. A path from root to leaf node represents one unique event record.

3.3 Event Types Detection

Since a document can express events of different types, we formulate the task as a multi-label classification and leverage sentences feature matrix S to detect event types:

$$A = \texttt{MultiHead}(Q, S, S) \in \mathbb{R}^{d_m\times T}$$
$$R = \texttt{Sigmoid}(A^\top W_t) \in \mathbb{R}^T$$

where $Q \in \mathbb{R}^{d_m\times T}$ and $W_t \in \mathbb{R}^{d_m}$ are trainable parameters, and T denotes the number of possible event types. `MultiHead` refers to the standard multi-head attention mechanism with Query/Key/Value. Therefore, we derive the event types detection loss with golden label $\widehat{R} \in \mathbb{R}^T$:

$$\begin{aligned}\mathcal{L}_{\text{detect}} = -\sum_{t=1}^{T} \mathbb{I}\left(\widehat{R}_t = 1\right) \log P\left(R_t|\mathcal{D}\right) \\ + \mathbb{I}\left(\widehat{R}_t = 0\right) \log\left(1 - P\left(R_t|\mathcal{D}\right)\right)\end{aligned} \tag{2}$$

3.4 Event Records Extraction

Since a document is likely to express multiple event records and the number of records cannot be known in advance, we decode records by expanding a tree orderly as previous methods did (Zheng et al., 2019). However, they treat each record independently. Instead, *to incorporate the interdependency among event records*, we propose a *Tracker* module, which improves the model performance.

To be self-contained, we introduce the ordered tree expanding in this paragraph. In each step,

[*] Traditional methods in sentence-level EE also utilize graph to extract events (Liu et al., 2018; Yan et al., 2019), based on the dependency tree. However, our interaction graph is heterogeneous and have no demands for dependency tree.

we extract event records of a specific event type. The arguments extraction order is predefined so that the extraction is modeled as a constrained tree expanding task[†]. Taking *Equity Freeze* records as an example, as shown in Figure 3, we firstly extract *EquityHolder*, followed by *FrozeShares* and others. Starting from a virtual root node, the tree expands by predicting arguments in a sequential order. As there may exist multiple eligible entities for the event argument role, the current node will expand several branches during extraction, with different entities assigned to the current role. This branching operation is formulated as *multi-label* classification task. In this way, each path from the root node to the leaf node is identified as a unique event record.

Interdependency exists extensively among different event records. For example, as shown in Figure 1, an *Equity Underweight* event record is closely related to an *Equity Overweight* event record, and they may share some key arguments or provide useful reasoning information. To take advantage of such interdependency, we propose a novel *Tracker* module inspired by memory network (Weston et al., 2015). Intuitively, the *Tracker* continually tracks the extracted records on-the-fly and store the information into a global memory. When predicting arguments for current record, the model will query the global memory and therefore make use of useful interdependency information of other records.

In detail, for the i-th record path consisting of a sequence of entities, the *Tracker* encodes the corresponding entity representation sequence $U_i = [E_{i1}, E_{i2}, ...]$ into an vector G_i with an LSTM (last hidden state) and add event type embedding. Then the compressed record information is stored in the global memory G, which is shared across different event types as shown in Figure 3. For extraction, given a record path $U_i \in \mathbb{R}^{d_m \times (J-1)}$ with the first $J-1$ arguments roles, we predict the J-th role by injecting role-specific information into entity representations, $\overline{E} = E + \texttt{Role}_J$, where $\texttt{Role}_J$ is the role embedding for the J-th role. Then we concatenate $\overline{E}$, sentences feature S, current entities path U_i, and the global memory G, followed by a transformer to obtain new entity feature matrix $\widetilde{E} \in \mathbb{R}^{d_m \times |\mathcal{E}|}$, which contains global role-specific information for all entity candidates.[‡]

$$[\widetilde{E}, \widetilde{S}, \widetilde{U}_i, \widetilde{G}] = \texttt{Transformer}([\overline{E}; S; U_i; G])$$

We treat the path expansion as a multi-label classification problem with a binary classifier over $\widetilde{E}_i$, i.e., predicts whether the i-th entity is the next argument role for the current record and expand the path accordingly as shown in Figure 3.

During training, we minimize the following loss:

$$\mathcal{L}_{\text{record}} = -\sum_{n \in N_{\mathcal{D}}} \sum_{t=1}^{|\mathcal{E}|} \log P(y_t^n | n) \qquad (3)$$

where $N_{\mathcal{D}}$ denotes the nodes set in the event records tree, and y_t^n is the golden label. If the t-th entity is validate for the next argument in node n, then $y_t^n = 1$, otherwise $y_t^n = 0$.

3.5 Training

We sum the losses coming from three sub-tasks with different weight respectively in Eq. (1), (2) and (3) as follows:

$$\mathcal{L}_{\text{all}} = \lambda_1 \mathcal{L}_{\text{ner}} + \lambda_2 \mathcal{L}_{\text{detect}} + \lambda_3 \mathcal{L}_{\text{record}}$$

More training details are shown in Appendix A.

4 Experiments

4.1 Dataset

We evaluate our model on a public dataset proposed by Zheng et al. (2019)[§], which is constructed from Chinese financial documents. It consists of up to $32,040$ documents which is the largest document-level EE dataset by far. It focuses on five event types: *Equity Freeze* (EF), *Equity Repurchase* (ER), *Equity Underweight* (EU), *Equity Overweight* (EO) and *Equity Pledge* (EP), with 35 different kinds of argument roles in total. We follow the standard split of the dataset, $25,632/3,204/3,204$ documents for training/dev/test set. The dataset is quite challenging, as a document has 20 sentences and consists of 912 tokens on average. Besides, there are roughly 6 sentences involved for an event record, and 29% documents express multiple events.

[†] We simply adopt the order used by Zheng et al. (2019).

[‡] To distinguish different parts in the concatenated vector, we also add segment embedding, which is omitted in Eq. 3.4.

[§] https://github.com/dolphin-zs/Doc2EDAG/blob/master/Data.zip

Model	EF	ER	EU	EO	EP	Overall
DCFEE-S	46.7	80.0	47.5	46.7	56.1	60.3
DCFEE-M	42.7	73.3	45.8	44.6	53.8	56.6
Greedy-Dec	57.7	79.4	51.2	50.0	54.2	61.0
Doc2EDAG	71.0	88.4	69.8	73.5	74.8	77.5
GIT (ours)	**73.4**	**90.8**	**74.3**	**76.3**	**77.7**	**80.3**

Table 1: F1 scores on test set. GIT achieves the best performance. We also list the results reported in Zheng et al. (2019) in Appendix B, and GIT consistently outperforms other baselines. EF/ER/EU/EO/EP refer to specific event types, and Overall denotes micro F1.

4.2 Experiments Setting

In our implementation of GIT, we use 8 and 4 layers Transformer (Vaswani et al., 2017) in encoding and decoding module respectively. The dimensions in hidden layers and feed-forward layers are the same as previous work (Zheng et al., 2019), i.e., 768 and $1,024$. We also use $L = 3$ layers of GCN, and set dropout rate to 0.1, batch size to 64. GIT is trained using Adam (Kingma and Ba, 2015) as optimizer with $1e-4$ learning rate for 100 epochs. We set $\lambda_1 = 0.05$, $\lambda_2 = \lambda_3 = 1$ for the loss function.

4.3 Baselines and Metrics

Yang et al. (2018) proposes DCFEE that extracts arguments from the identified central sentence and queries surrounding sentences for missing arguments. The model has two variants, **DCFEE-S** and **DCFEE-M**. DCFEE-S produces one record at a time, while DCFEE-M produces multiple possible argument combinations by the closest distance from the central sentence. Besides, **Doc2EDAG** (Zheng et al., 2019) uses transformer encoder to obtain sentence and entity embeddings, followed by another transformer to fuse cross-sentence context. Then multiple events are extracted simultaneously. **Greedy-Dec** is a variant of Doc2EDAG, which produces only one record greedily.

Three sub-tasks of the document-level EE are all evaluated by F1 score. Due to limited space, we leave the results of *entity extraction* and *event types detection* in Appendix B, which shows GIT only slightly outperform Doc2EDAG, because we mainly focus on event record extraction and the methods are similar to Doc2EDAG for these two sub-tasks. In the following, we mainly report and analyze the results of *event record extraction*.

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 (ours)	**81.9**	**85.7**	**80.0**	**75.7**

Table 2: F1 scores on four sets with growing average number of involved sentences for records (increases from I to IV). The highest improvement of GIT comes from event records involving the most sentences (Set IV) by 3.7 F1 score compared with Doc2EDAG.

4.4 Main Results

Overall performance. The results of the overall performance on the document-level EE dataset is illustrated in Table 1. As Table 1 shows, our GIT consistently outperforms other baselines, thanks to better modelling of global interactions and interdependency. Specifically, GIT improves 2.8 micro F1 compared with the previous state-of-the-art, Doc2EDAG, especially 4.5 improvement in *Equity Underweight* (EU) event type.

Cross-sentence records scenario. There are more than 99.5% records of the test set are cross-sentence event records, and the extraction becomes gradually more difficult as the number of their involved sentences grows. To verifies the effectiveness of GIT to capture cross-sentence information, we first calculate the average number of sentences that the records involve for each document, and sort them in ascending order. Then we divide them into four sets I/II/III/IV with equal size. Documents in Set. IV is considered to be the most challenging as it requires the most number of sentences to successfully extract records. As Table 2 shows, GIT consistently outperforms Doc2EDAG, especially on the most challenging Set. IV that involves the most sentences, by 3.7 F1 score. It suggests that GIT can well capture global context and mitigate the arguments-scattering challenge, with the help of the heterogeneous graph interaction network.

Multiple records scenario. GIT introduces the *tracker* to make use of global interdependency among event records, which is important in multiple records scenario. To illustrate its effectiveness, we divide the test set into single-record set (S.) containing documents with one record, and multi-record set (M.) containing those with multiple records. As shown in Table. 3, F1 score on M.

Model	EF		ER		EU		EO		EP		Overall	
	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
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
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
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
GIT (ours)	**81.9**	**65.9**	**93.0**	**71.7**	**82.0**	**64.1**	**80.9**	**70.6**	**85.0**	**73.5**	**87.6**	**72.3**

Table 3: F1 scores on single-record (S.) and multi-record (M.) sets.

Model	F1	I	II	III	IV
GIT	**80.3**	**81.9**	**85.7**	**80.0**	**75.7**
- S-S	-1.4	-0.9	-0.1	-1.9	-2.3
- S-M	-1.0	-1.6	-1.7	-0.7	-0.7
- M-M$_{\text{intra}}$	-1.3	-0.5	-1.4	-2.4	-1.5
- M-M$_{\text{inter}}$	-1.1	-0.5	-1.6	-1.4	-1.7
- Graph	-2.0	-1.8	-1.5	-2.0	-2.5

Table 4: The decrease of F1 scores on ablation study for GIT's heterogeneous graph interaction network. Removing the heterogeneous graph leads to significant drop on F1, especially for records involving the most sentences (i.e., -2.5 F1 on Set IV).

Model	P	R	F1	S.	M.
GIT	**82.3**	**78.4**	**80.3**	**87.6**	**72.3**
GIT-OT	-0.6	-0.4	-0.5	-0.8	-0.7
GIT-OP	-1.0	-1.6	-1.2	-1.0	-1.5
GIT-NT	-2.8	+0.1	-1.3	-1.3	-1.5

Table 5: Performance of GIT on ablation study for the $Tracker$ module. The removal of the *Tracker* (GIT-NT) brings about higher F1 decrease on M. than that on S.. S.: Single-record set, M.: Multi-record set.

is much lower than that on S., indicating it is challenging to extract multiple records. However, GIT still surpasses other strong baselines by $4.9 \sim 35.3$ on multi-record set (M.). This is because GIT is aware of other records through the $Tracker$ module, and leverage the interdependency information to improve the performance[¶].

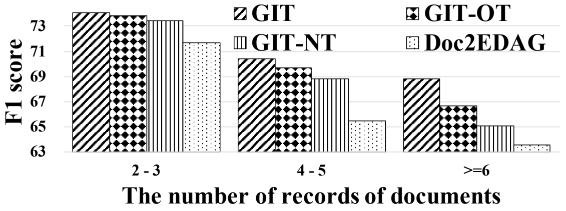


Figure 4: F1 scores on documents with different number of event records. The F1 gap between w/ (GIT) and w/o *Tracker* (GIT-NT) becomes wider as the number of event records of documents increases.

4.5 Analysis

We conduct further experiments to analyze the key modules in GIT more deeply.

On the effect of heterogeneous graph interaction network. The heterogeneous graph we constructed contains four types of edges. To explore their functions, we remove one type of edges at a time, and remove the whole graph network finally. Results are shown in Table 4, including micro F1 and F1 on the four sets, which are divided by the number of involved sentences for records as we did before. The micro F1 would decreases $1.0 \sim 1.4$ without a certainty type of edge. Besides, removing the whole graph causes an significant drop by 2.0 F1, especially for Set IV by 2.5, which requires the most number of sentences to extract the event record. It demonstrates that the graph interaction network helps improve the performance, especially on records involving many sentences, and all kinds of edges play an important role for extraction.

On the effect of *Tracker* module. GIT can leverage interdependency among records based on the information of other event records tracked by *Tracker*. To explore its effect, firstly, we remove the global interdependency information between records of different event types, by clearing the global memory whenever we extract events for an-

[¶] Nguyen et al. (2016) maintain three binary matrices to memorize entities and events states. Although they aim at sentence-level EE that contains fewer entities and event records, it would be also interesting to compare with them and we leave it as future work.

… **[5]** The shareholder of the company, **Quanlie Chen**, pledged **52.4 million** to **GDZQ Co., Ltd.** in 2018, and supplemented the pledge recently because of the decline of the share price. … **[7]** Since the borrowings have been paid off, **Quanlie Chen** completed the pledge cancellation procedures of **35.5 million** that were pledged to **GTJA Co., Ltd.** on Nov 7, 2018. **[8]** As of today, **Quanlie Chen** holds a total of **325.4 million** of the company, and there are still **218.6 million** in pledge status. …

Doc2EDAG

No.	Pledger	PledgedShares	Pledgee	TotalHoldingShares	TotalPledgedShares	…
1	Quanlie Chen	35.5 million	GTJA Co., Ltd.	325.4 million	218.6 million	…
2	Quanlie Chen	52.4 million	GDZQ Co., Ltd.	NULL	NULL	…

GIT

No.	Pledger	PledgedShares	Pledgee	TotalHoldingShares	TotalPledgedShares	…
1	Quanlie Chen	35.5 million	GTJA Co., Ltd.	325.4 million	218.6 million	…
2	Quanlie Chen	52.4 million	GDZQ Co., Ltd.	325.4 million	218.6 million	…

Figure 5: The case study of our proposed GIT and Doc2EDAG, with their key prediction difference colored in red. Related entities are colored in blue. GIT successfully extract *TotalHoldingShares* and *TotalPledgedShares* for Record 2, while Doc2EDAG fails. The complete content are provided in Appendix C.

other new event type (GIT-**O**wn **T**ype). Next, we remove all the tracking information except the own path for a record, to explore whether the tracking of other records makes effect indeed (GIT-**O**wn **P**ath). Finally, we remove the whole *Tracker* module (GIT-**N**o **T**racker). As Table 5 shows, the F1 in GIT-OT/GIT-OP decreases by 0.5/1.2, suggesting the interdependency among records of both the same and different event types do play an essential role. Besides, their F1 decrease in M. by 0.7/1.5 are more than those in S. by 0.8/1.0, verifying the effectiveness of the *Tracker* in multi-event scenarios. Moreover, the performances are similar between GIT-OP and GIT-NT, which also provides evidence that other records do help. We also reveal F1 on documents with different number of records in Figure 4. The gap between models with or without *Tracker* raises as the number of records increases, which validates the effectiveness of our *Tracker*.

4.6 Case Study

Figure 5 demonstrates a case of the predictions of Doc2EDAG and GIT for *Equity Pledge* (EP) event types. The *TotalHoldingShares* and *TotalPledgedShares* information lies in Sentence 8, while the *PledgedShares* and *Pledgee* information for Record 2 lies in Sentence 5. Though Doc2EDAG fails to extract these arguments in Record 2 (colored in red), GIT succeeds because it can capture interactions between long-distance sentences, and utilize the information of Record 1 (*325.4 million* and *218.6 million*) thanks to the *Tracker* model.

5 Related Work

Sentence-level Event Extraction. Previous approaches mainly focus on sentence-level event extraction. Chen et al. (2015) propose a neural pipeline model that identifies triggers first and then extracts argument roles. Nguyen et al. (2016) use a joint model to extract triggers and argument roles simultaneously. Some studies also utilize dependency tree information (Liu et al., 2018; Yan et al., 2019). To utilize more knowledge, some studies leverage document context (Chen et al., 2018; Zhao et al., 2018), pre-trained language model (Yang et al., 2019), and explicit external knowledge (Liu et al., 2019a; Tong et al., 2020) such as WordNet (Miller, 1995). Du and Cardie (2020b) also try to extract events in a Question-Answer way. These studies usually conduct experiments on sentence-level event extraction dataset, ACE05 (Walker et al., 2006). However, it is hard for the sentence-level models to extract multiple qualified events spanning across sentences, which is more common in real-world scenarios.

Document-level Event Extraction. Document-level EE has attracted more and more attention recently. Yang and Mitchell (2016) use well-defined features to handle the event-argument relations across sentences, which is, unfortunately, quite nontrivial. Yang et al. (2018) extract events from a central sentence and find other arguments from neighboring sentences separately. Although Zheng et al. (2019) use Transformer to fuse sentences and entities, interdependency among events is neglected. Du and Cardie (2020a) try to encode the sentences in a multi-granularity way and Du et al. (2020) leverage a seq2seq model. They conduct experiments on MUC-4 (Sundheim, 1992) dataset with $1,700$ documents and 5 kinds of entity-based arguments, and it is formulated as a table-filling task, coping with single event record of single event type. However, our work is different from these studies in that a) we utilize heterogeneous graph to model the global interactions among sentences and mentions to capture cross-sentence context, b) and we leverage the global interdependency through *Tracker* to extract multiple event records of multiple event types.

6 Conclusion

Although promising in practical application, document-level EE still faces some challenges such as arguments-scattering phenomenon and multiple correlated events expressed by a single document. To tackle the challenges, we introduce Heterogeneous **G**raph-based **I**nteraction Model with a **T**racker (GIT). GIT uses a heterogeneous graph interaction network to model global interactions among sentences and entity mentions. GIT also uses a *Tracker* to track the extracted records to consider global interdependency during extraction. Experiments on large-scale public dataset (Zheng et al., 2019) show GIT outperforms previous state-of-the-art by 2.8 F1. Further analysis verifies the effectiveness of GIT especially in cross-sentence events extraction and multi-event scenarios.

Acknowledgments

The authors would like to thank Changzhi Sun, Mingxuan Wang, and the anonymous reviewers for their thoughtful and constructive comments. This paper is supported in part by the National Key R&D Program of China under Grand No.2018AAA0102003, the National Science Foundation of China under Grant No.61936012 and 61876004.

References

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS)*.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. 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 (ACL-IJCNLP)*.

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xinya Du and Claire Cardie. 2020a. Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Xinya Du and Claire Cardie. 2020b. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xinya Du, Alexander M. Rush, and Claire Cardie. 2020. Document-level event-based extraction using generative template-filling transformers. *arXiv preprint arXiv:2008.09249*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations (ICLR)*.

Jian Liu, Yubo Chen, and Kang Liu. 2019a. Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*.

Shulin Liu, Yang Li, Feng Zhang, Tao Yang, and Xinpeng Zhou. 2019b. Event detection without triggers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Beth M. Sundheim. 1992. Overview of the fourth Message Understanding Evaluation and Conference. In *Fourth Message Uunderstanding Conference (MUC-4)*.

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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. In *Philadelphia: Linguistic Data Consortium*.

Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, Ziyue Huang, Qipeng Guo, Hao Zhang, Haibin Lin, Junbo Zhao, Jinyang Li, Alexander J Smola, and Zheng Zhang. 2019. Deep graph library: Towards efficient and scalable deep learning on graphs. *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. In *3rd International Conference on Learning Representations (ICLR)*.

Haoran Yan, Xiaolong Jin, Xiangbin Meng, Jiafeng Guo, and Xueqi Cheng. 2019. Event detection with multi-order graph convolution and aggregated attention. 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)*.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

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 *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. Double graph based reasoning for document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Ying Zeng, Yansong Feng, Rong Ma, Zheng Wang, Rui Yan, Chongde Shi, and Dongyan Zhao. 2018. Scale up event extraction learning via automatic training data generation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*.

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Document embedding enhanced event detection with hierarchical and supervised attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. 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)*.

A Training Details

To mitigate the error propagation due to the gap between training and inference phrase (i.e., the extracted entities are ground truth during training but predicted results during inference), we adopt scheduled sampling strategy (Bengio et al., 2015) as Zheng et al. (2019) did. We gradually switch the entity extraction results from golden label to what the model predicts on its own. Specifically, from epoch 10 to epoch 20, we linearly increase the proportion of predicted entity results from 0% to 100%. We implement GIT under PyTorch (Paszke et al., 2017) and DGL (Wang et al., 2019) based on codes provided by Zheng et al. (2019).

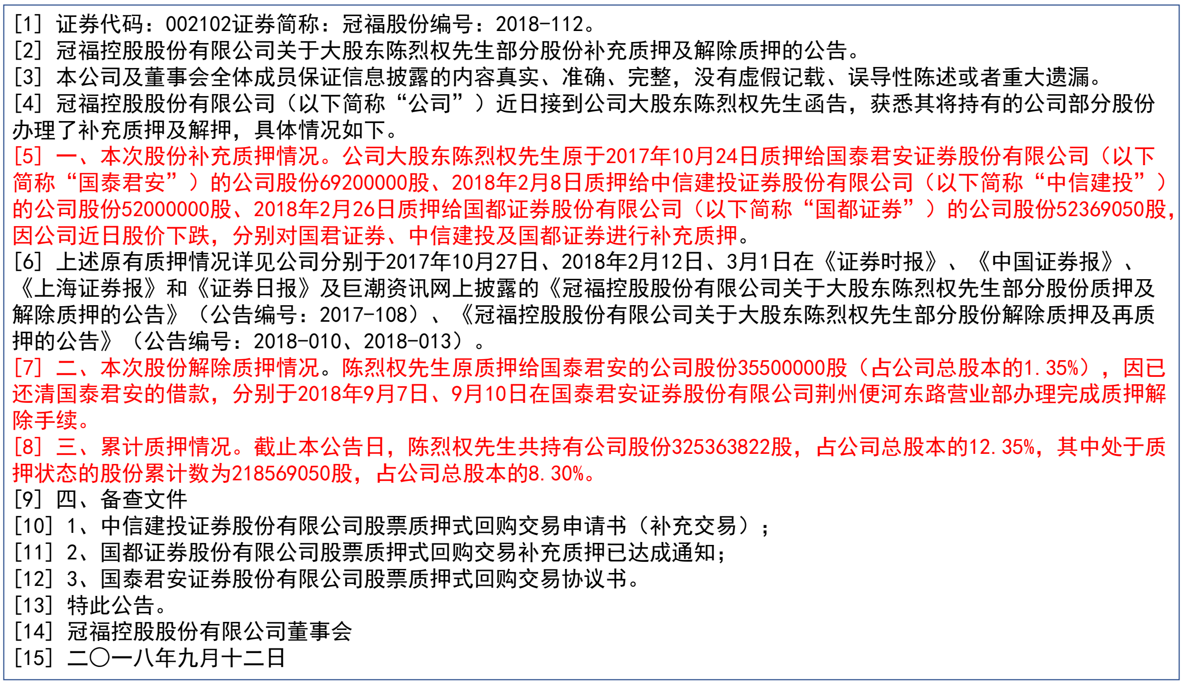
All the experiments (including the baselines) are run with the same 8 Tesla-V100 GPUs and the same version of python dependencies to ensure the fairness.

Hyperparameters trials are listed in Table 6. The value of hyperparameters we finally adopted are in bold. Note that we do not tune all the hyperparameters, and make little effort to select the best hyperparameters for our GIT.

We choose the final checkpoints for test according to the Micro F1 performance on the dev set. Table 9 illustrates the best epoch in which the model achieves the highest Micro F1 on the dev set and their according F1 score.

B Additional Evaluation Results

We have showed the evaluation results of event records extraction in the paper for document-level



[1] 证券代码：002102证券简称：冠福股份编号：2018-112。
[2] 冠福控股股份有限公司关于大股东陈烈权先生部分股份补充质押及解除质押的公告。
[3] 本公司及董事会全体成员保证信息披露的内容真实、准确、完整，没有虚假记载、误导性陈述或者重大遗漏。
[4] 冠福控股股份有限公司（以下简称“公司”）近日接到公司大股东陈烈权先生函告，获悉其将持有的公司部分股份办理了补充质押及解押，具体情况如下。
[5] 一、本次股份补充质押情况。公司大股东陈烈权先生原于2017年10月24日质押给国泰君安证券股份有限公司（以下简称“国泰君安”）的公司股份69200000股、2018年2月8日质押给中信建投证券股份有限公司（以下简称“中信建投”）的公司股份52000000股、2018年2月26日质押给国都证券股份有限公司（以下简称“国都证券”）的公司股份52369050股，因公司近日股价下跌，分别对国君证券、中信建投及国都证券进行补充质押。
[6] 上述原有质押情况详见公司分别于2017年10月27日、2018年2月12日、3月1日在《证券时报》、《中国证券报》、《上海证券报》和《证券日报》及巨潮资讯网上披露的《冠福控股股份有限公司关于大股东陈烈权先生部分股份质押及解除质押的公告》（公告编号：2017-108）、《冠福控股股份有限公司关于大股东陈烈权先生部分股份解除质押及再质押的公告》（公告编号：2018-010、2018-013）。
[7] 二、本次股份解除质押情况。陈烈权先生原质押给国泰君安的公司股份35500000股（占公司总股本的1.35%），因已还清国泰君安的借款，分别于2018年9月7日、9月10日在国泰君安证券股份有限公司荆州便河东路营业部办理完成质押解除手续。
[8] 三、累计质押情况。截止本公告日，陈烈权先生共持有公司股份325363822股，占公司总股本的12.35%，其中处于质押状态的股份累计数为218569050股，占公司总股本的8.30%。
[9] 四、备查文件
[10] 1、中信建投证券股份有限公司股票质押式回购交易申请书（补充交易）；
[11] 2、国都证券股份有限公司股票质押式回购交易补充质押已达成通知；
[12] 3、国泰君安证券股份有限公司股票质押式回购交易协议书。
[13] 特此公告。
[14] 冠福控股股份有限公司董事会
[15] 二〇一八年九月十二日

Figure 6: The original complete document corresponding to the case study in Figure 5. Sentences in red color are presented in Figure 5.

Hyperparameters	**Value**
Batch Size	32, **64**
Learning Rate	**0.0001**
Dropout	**0.1**
Layers of GCN	1, 2, **3**, 4, 5
Number of Epochs	**100**
λ_1	**0.05**
λ_2	**1.00**
λ_3	**1.00**
Gradient Accumulation Steps	**8**
Layers of Transformer in Entity Extractor	**8**
Layers of Transformer in Decoder Module	**4**
Hyperparameter Search Trials	10

Table 6: Hyperparameters for our proposed GIT.

Model	**P**	**R**	**F1**
DCFEE-S	86.5	88.6	87.6
DCFEE-M	86.6	89.0	87.8
Greedy-Dec	87.5	89.8	88.6
Doc2EDAG	88.0	90.0	89.0
GIT (ours)	85.8	92.6	89.1

Table 7: Results of **entity extraction** sub-task on the test set. The performance of different models are similar, for the reason that they all utilize the same structure and methods to extract entities.

event extraction. In this section, we also illustate the results of entity extraction in Table. 7 and event types detection in Table. 8. Moreover, the comprehensive results of event record extraction is shown in Table. 10, including results reported in Zheng et al. (2019) with precison, recall and F1 score.

C Complete Document for the Examples

We show an example document in Figure 1 in the paper. To better illustrate, we translate it from Chinese into English and make some simplication. Here we present the original complete document example in Figure 7. For the specific meanings of argument roles, we recommend readers to refer to (Zheng et al., 2019).

We also demonstrate an case study in Figure 5 in the paper. Now we also show its original Chinese version in Figure 6.

Model	**EF**	**ER**	**EU**	**EO**	**EP**	**Overall**
DCFEE-S	81.5	94.0	82.3	85.7	93.8	91.4
DCFEE-M	79.8	92.4	78.9	84.2	92.9	90.0
Greedy-Dec	99.3	99.9	96.8	95.4	99.6	99.0
Doc2EDAG	99.0	99.8	96.8	94.1	99.5	98.9
GIT (ours)	98.8	99.8	97.9	96.6	99.6	99.2

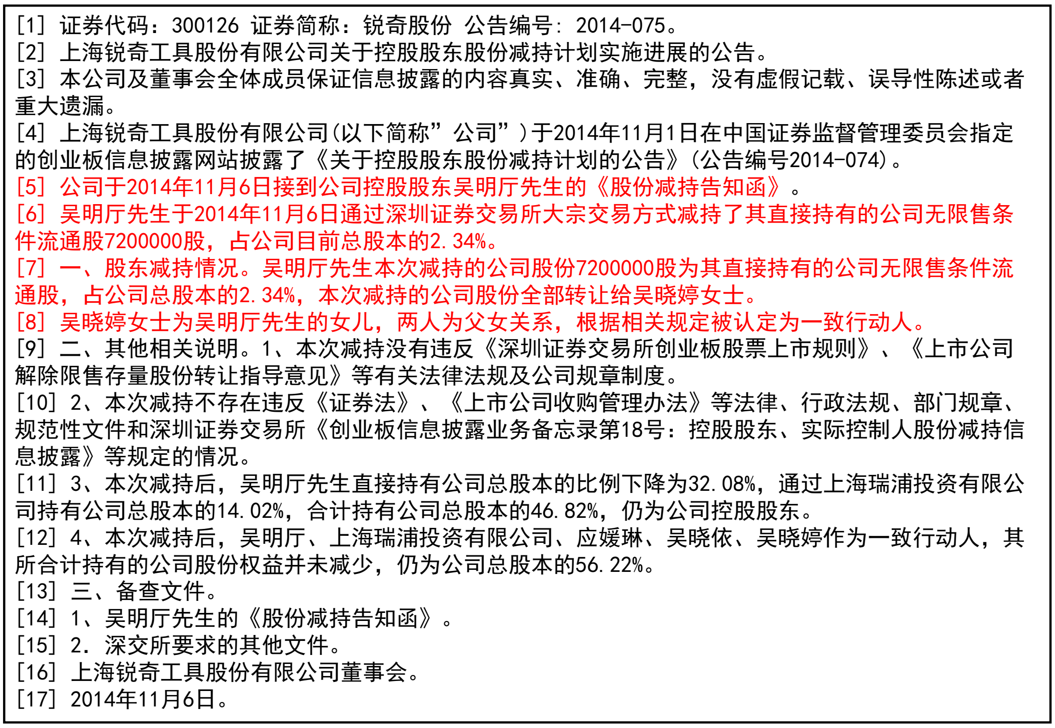
Table 8: F1 scores results of **event types detection** sub-task on the test set. All the models obtains more than 90.0 micro F1 score. GIT slightly outperform Doc2EDAG.

Model	**Best Epoch**	**EF**	**ER**	**EU**	**EO**	**EP**	**Overall**
DCFEE-S	86	51.3	73.0	44.1	51.4	58.6	58.7
DCFEE-M	87	52.5	69.1	43.9	47.2	55.9	55.8
Greedy-Dec	90	57.5	76.0	55.1	49.3	57.0	59.1
Doc2EDAG	89	75.2	85.2	71.6	80.0	77.9	78.7
GIT (ours)	89	**78.3**	**87.6**	**74.7**	**80.9**	**79.8**	**80.7**

Table 9: The best epoch in which the models achieve the highest **micro F1 score on the dev set** and the corresponding performance.

Model	EF			ER			EU			EO			EP			Overall		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
DCFEE-S◇	66.0	41.6	51.1	84.5	81.8	83.1	62.7	35.4	45.3	51.4	42.6	46.6	64.3	63.6	63.9	-	-	-
DCFEE-M◇	51.8	40.7	45.6	83.7	78.0	80.8	49.5	39.9	44.2	42.5	47.5	44.9	59.8	66.4	62.9	-	-	-
Greedy-Dec◇	**79.5**	46.8	58.9	83.3	74.9	78.9	68.7	40.8	51.2	69.7	40.6	51.3	**85.7**	48.7	62.1	-	-	-
Doc2EDAG◇	77.1	64.5	70.2	91.3	83.6	87.3	80.2	65.0	71.8	**82.1**	69.0	75.0	80.0	74.8	77.3	-	-	-
DCFEE-S♠	61.1	37.8	46.7	84.5	76.0	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 (ours)♠	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**

Table 10: Comprehensive results of event record extraction. Results with ◇ are results reported in Zheng et al. (2019). Results with are ♠ results we implement on our own. Our GIT consistently outperform other baselines.



[1] 证券代码：300126 证券简称：锐奇股份 公告编号：2014-075。

[2] 上海锐奇工具股份有限公司关于控股股东股份减持计划实施进展的公告。

[3] 本公司及董事会全体成员保证信息披露的内容真实、准确、完整，没有虚假记载、误导性陈述或者重大遗漏。

[4] 上海锐奇工具股份有限公司(以下简称”公司”)于2014年11月1日在中国证券监督管理委员会指定的创业板信息披露网站披露了《关于控股股东股份减持计划的公告》(公告编号2014-074)。

[5] 公司于2014年11月6日接到公司控股股东吴明厅先生的《股份减持告知函》。

[6] 吴明厅先生于2014年11月6日通过深圳证券交易所大宗交易方式减持了其直接持有的公司无限售条件流通股7200000股，占公司目前总股本的2.34%。

[7] 一、股东减持情况。吴明厅先生本次减持的公司股份7200000股为其直接持有的公司无限售条件流通股，占公司总股本的2.34%，本次减持的公司股份全部转让给吴晓婷女士。

[8] 吴晓婷女士为吴明厅先生的女儿，两人为父女关系，根据相关规定被认定为一致行动人。

[9] 二、其他相关说明。1、本次减持没有违反《深圳证券交易所创业板股票上市规则》、《上市公司解除限售存量股份转让指导意见》等有关法律法规及公司规章制度。

[10] 2、本次减持不存在违反《证券法》、《上市公司收购管理办法》等法律、行政法规、部门规章、规范性文件和深圳证券交易所《创业板信息披露业务备忘录第18号：控股股东、实际控制人股份减持信息披露》等规定的情况。

[11] 3、本次减持后，吴明厅先生直接持有公司总股本的比例下降为32.08%，通过上海瑞浦投资有限公司持有公司总股本的14.02%，合计持有公司总股本的46.82%，仍为公司控股股东。

[12] 4、本次减持后，吴明厅、上海瑞浦投资有限公司、应媛琳、吴晓依、吴晓婷作为一致行动人，其所合计持有的公司股份权益并未减少，仍为公司总股本的56.22%。

[13] 三、备查文件。

[14] 1、吴明厅先生的《股份减持告知函》。

[15] 2. 深交所要求的其他文件。

[16] 上海锐奇工具股份有限公司董事会。

[17] 2014年11月6日。

EquityUnderweight

EquityHolder	吴明厅
TradedShares	720000股
StartDate	2014年11月6日
EndDate	2014年11月6日
LaterHolding Shares	NULL
AveragePrice	NULL

EquityOverweight

EquityHolder	吴晓婷
TradedShares	720000股
StartDate	2014年11月6日
EndDate	2014年11月6日
LaterHolding Shares	720000股
AveragePrice	NULL

Figure 7: The original complete document corresponding to the running example in Figure 1. Sentences in red color are presented in Figure 1.