Exploiting Coreference and Schema Structure for Document-level Event Extraction

Anonymous ACL submission

Abstract

Document-level event extraction (DEE) extracts structured information of events from a document. Previous studies focus on improving the model architecture. We argue that exploiting data characteristics is also important. We propose to utilize coreference information to obtain better document-level entity representations, and propose the concept of core roles to adjust the schema structure to alleviate error propagation. Experiments demonstrate that our data exploitation methods significantly improve the performance of existing models on both the role-level and record-level metrics. Our code is available at this link.

1 Introduction

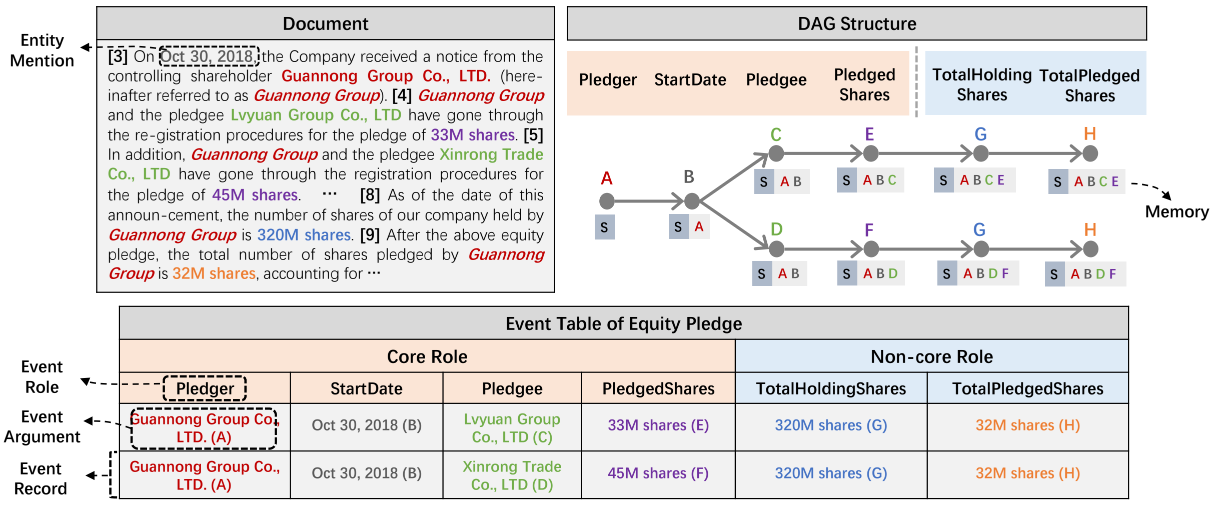
Document-level event extraction (DEE) aims to detect events and extract event arguments from a document. DEE has attracted much attention currently (Yang et al., 2018a; Du and Cardie, 2020; Du et al., 2020). There are two main challenges for DEE: multiple events and argument scattering. Figure 1 demonstrates this where there are two *Equity Pledge* event records, and the event arguments of each record scatter across sentences.

To address these challenges, recently an end-to-end model Doc2EDAG (Zheng et al., 2019) has been proposed to perform the DEE task over financial documents. Doc2EDAG transforms the event table into a directed acyclic graph (DAG) and iteratively extracts each event role. For example, in Figure 1, the first event record is extracted from A to B to H in the DAG structure. When extracting a role, such as *PledgedShares*, the model applies a memory tensor to gather information from preceding event arguments A, B, and C.

DEE is a relatively new NLP task. Current researches on DEE (Zheng et al., 2019; Xu et al., 2021; Yang et al., 2021) overlook some critical data characteristics, including various coreferential mentions in the document, as well as the dependence and independence among roles in an event schema. We propose the DEE model with **C**oreference **A**ggregation and Schema Structure A**D**justment Extraction (CAD) to exploit these data characteristics as follows.

First, we leverage more **coreference information** to enhance entity representations. Though entity mentions scatter across sentences, some of them may refer to the same entities. Utilizing such coreference is especially critical to document-level extraction. For example, in Figure 1, a company is referred to as its full name *Guannong Group Co., LTD.* and its abbreviated name *Guannong Group*. The former only appears once in sentence [3], while the latter appears in sentences [3,4,5,8,9], which is conducive to extracting the two event records. However, as the popular DEE dataset ChFinAnn (Zheng et al., 2019) is constructed through distant supervision, only the full-name mention of an entity is annotated as the event argument. Previous methods only utilize this kind of mention and rely on the neural encoder to implicitly capture mention interactions. We introduce a coreference aggregation module, which first detects coreferential mentions using vary simple patterns and then encodes them to get better document-level entity representations.

Second, we leverage the "**core role**" concept to alleviate the error propagation problem. Previous methods extract arguments of an event in sequence, and each depends on the information of previously extracted arguments. Arguments at the back of the path have long pre-paths, thus errors from preceding arguments may propagate and accumulate along the path and influence the succeeding extractions. For example, when extracting the last argument of the upper record in Figure 1, they use the information of A, B, C, E, G. However, an Equity Pledge event can be uniquely determined by the first four roles. So, we only need the information of A, B, C, E to extract H in this case. We introduce the concept of core roles which is the



[3] On Oct 30, 2018, the Company received a notice from the controlling shareholder Guannong Group Co., LTD. (hereinafter referred to as *Guannong Group*). [4] *Guannong Group* and the pledgee Lvyuan Group Co., LTD have gone through the re-gistration procedures for the pledge of 33M shares. [5] In addition, *Guannong Group* and the pledgee Xinrong Trade Co., LTD have gone through the registration procedures for the pledge of 45M shares. ··· [8] As of the date of this announ-cement, the number of shares of our company held by *Guannong Group* is 320M shares. [9] After the above equity pledge, the total number of shares pledged by *Guannong Group* is 32M shares, accounting for ···

Event Table of Equity Pledge

Core Role				Non-core Role	
Pledger	StartDate	Pledgee	PledgedShares	TotalHoldingShares	TotalPledgedShares
Guannong Group Co., LTD. (A)	Oct 30, 2018 (B)	Lvyuan Group Co., LTD (C)	33M shares (E)	320M shares (G)	32M shares (H)
Guannong Group Co., LTD. (A)	Oct 30, 2018 (B)	Xinrong Trade Co., LTD (D)	45M shares (F)	320M shares (G)	32M shares (H)

Figure 1: An example (translated) from the ChFinAnn dataset Zheng et al. (2019). **Document** part shows sentences [3] to [9]. The **Event Table of Equity Pledge** part shows two event records described in this document. Event arguments are marked with different colors. The Document part highlights mentions of these arguments with the same colors and the abbreviations are shown in italic. The **DAG structure** part shows the DAG transformed from the event table in which arguments are denoted using capital letters for brevity. Below each node, we illustrate its memory state during extraction. "S" denotes the initial memory.

signature of an event record that uniquely determines a record, and non-core roles, on the contrary, only provide accessory information. In Figure 1, we show the core roles of the *Equity Pledge* in the orange background. By utilizing the core role, we propose to adjust the schema structure and apply a shallower memory to discard the information of non-core roles to alleviate the error propagation.

Experiments on a large dataset show that our proposed methods can significantly improve model performance, especially on record-level metrics. Such improvement may inspire people to think about why sophisticated models can not capture such simple data patterns, and guide future model design.

2 Related Work

Most of the previous EE approaches focus on the sentence-level event extraction (SEE) (Liu et al., 2018; Yan et al., 2019; Liu et al., 2020; Ahmad et al., 2021). In recent years, some researchers have shifted their attention to DEE, which is common and important in real-world scenarios. Yang et al. (2018b) propose a two-stage framework that can extract document-level events through heuristic-based argument completion. It pays attention to the argument scattering challenge but ignores the multiple events challenge. Zheng et al. (2019) obtain a large-scale dataset ChFinAnn through distant supervision on Chinese financial documents, which paves the way for later researches. They also propose an end-to-end framework named Doc2EDAG which can extract multiple events in a document. Xu et al. (2021) extend the Doc2EDAG with a GNN to model mention interactions and a tracker module to capture the interdependency among extracted events. Besides the EDAG framework, Yang et al. (2021) introduce a multi-granularity decoder to extract structured events in a parallel manner. These researches focus on the model architecture. To the best of our knowledge, there is no work to improve DEE from the perspective of data exploitation.

3 Method

Our CAD method contains two main steps. **Entity encoding** step extracts mentions from sentences and aggregates coreferential mentions to get entity embeddings (Sec 3.1). Then, **EDAG generation** step takes these entities as candidates and iteratively extracts event roles as a DAG (Sec 3.2). This framework follows Doc2EDAG (Zheng et al., 2019). But we introduce the whole framework to be self-contained. Our methods focus on coreference utilization and schema structure adjustment.

Before introducing the details of the model, we first clarify some key notions (refer to Figure 1): a) **event role**: an event role is a predefined field of the event table; b) **event argument**: an event argument is an entity that plays a specific event role; c) **event record**: an event record contains event arguments that describe an event. Given a document $\mathcal{D} = \{s_i\}_{i=1}^{n}$, where sentence $s_i = \{w_j\}_{j=1}^{|s_i|}$ is a sequence of tokens, DEE extracts all event records from the document with a pre-defined event schema.

3.1 Entity Extraction and Encoding

Sentence-level encoding. We extract entity mentions on each sentence with a sequence labeling model. Given a sentence $s_i = \{w_j\}_{j=1}^{m} \in \mathcal{D}$, we encode it into a sequence of vectors $\{v_j\}_{j=1}^{m}$ using a transformer (Vaswani et al., 2017) (encoder-1). We leverage a conditional random field (CRF) layer to identify entity mentions. Then, for each entity mention, we output an embedding by max-pooling over its covered token embeddings. For each sentence s_i, we apply the max-pooling over all token embeddings to get the sentence embedding c_i.

Document-level encoding. We further jointly encode all entity mention embeddings and sentence embeddings in the document to capture document-level interactions with another transformer (encoder-2), and get $\{c_1^d, ..c_n^d, m_1^d, ..m_k^d\}$, where c_i^ds are document level sentence embeddings and m_i^ds are the document level mention embeddings, and k is the number of mentions.

Coreference detection and aggregation. Previous methods only regard mentions with the same surface name as arguments. We propose to incorporate more coreferential mentions. As a prime attempt, we apply hand-crafted patterns. Specifically, we find two simple but effective patterns: "[Full name], hereafter referred to as [short name]" and "[Full name] (referred to as [short name])". Applying more advanced coreference detection methods may be interesting future work. Then, we use the max-pooling over coreferential mentions (same expression or abbreviation) of an entity to get a single embedding e_i^d for each entity.

Finally, the entity encoding step outputs document-level sentence embeddings $\mathcal{C} = \{c_i^d\}_{i=1}^{n}$ and entity embeddings $\mathcal{E} = \{e_i^d\}_{i=1}^{|\mathcal{E}|}$.

3.2 EDAG Generation

We first identify what types of events are described in a given document, and then extract event records for each event type.

Event type detection. First, we take the max-pooling over sentence embeddings $\mathcal{C}$ to get the document embedding. Then we stack a linear classifier to detect event types (multi-label classification).

Event Record Extraction. The event records are extracted by several path expanding sub-tasks.

For each detected event type, we extract the event roles iteratively following a predefined role order, as shown in Figure 1. For example, when extracting the Pledgee role of the Equity Pledge event, there is only one pre-path A-B. We concatenate embeddings of sentences and preceding entities, $[c_1^d, \cdots, c_n^d, e_A^d, e_B^d]$, as memory tensor $\mathcal{M}$ and feed $[\mathcal{M}, \mathcal{E}]$ into a transformer (encoder-3) to get $\{e_1^o, ...e_{|\mathcal{E}|}^o\}$. Then, a linear classifier classifies each e_i^o into positive or negative class. In the example, entities C and D are positive. So we get two paths A-B-C and A-B-D, and we need to predict the next role for each path.

Schema structure adjustment. During the path expanding procedure, the memory tensor $\mathcal{M}$ is used to gather the path history information and indicate the state of the procedure. Previous methods store all entities in the pre-path into $\mathcal{M}$, which exists the error propagation risk.

We propose to adjust the schema structure to address this issue. We first manually distinguish the core roles for each event type with expert knowledge and adjust the role order so that the core roles are in front of the non-core roles. When extracting non-core roles, we only include embeddings of core roles in the memory, as they have already uniquely determined an event and non-core roles should not affect each other. This is demonstrated in the memory of node H in Figure 1.

4 Experiments

4.1 Experimental Setup

Dataset. We evaluate our model on the public dataset ChFinAnn proposed by Zheng et al. (2019), which is constructed from real-world Chinese financial documents. It contains 32040 documents and focus 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 event roles in total.

We strictly follow the experiment settings of Doc2EDAG (Zheng et al., 2019).

Metrics. we evaluate models on the role-level and record-level micro-averaged F1 scores. The record-level F1 score indicates that an event record can be counted as true positive only if all predicted roles are consistent with the gold event record. Real-world applications focus on how many correct records can be extracted, which is reflected by record-level metrics.

Model	Role-level			Record-level		
	P.	R.	F1	P.	R.	F1
Doc2EDAG	**84.6**	70.7	77.1	36.3	33.1	34.6
DE-PPN	-	-	77.9	-	-	-
GIT	82.3	78.4	80.3	42.0	41.7	41.8
D2E*	80.5	72.8	76.4	34.5	33.5	34.0
D2E*+SA	80.5	73.9	77.0	37.9	36.5	37.2
CAD-D2E*	82.4	76.7	79.4	38.9	37.8	38.3
GIT$_4$	84.4	73.8	78.7	36.5	34.5	35.5
CAD-GIT$_4$	84.2	**78.9**	**81.4**	**43.8**	**42.6**	**43.2**

Table 1: Role-level and record-level precision(P.), recall (R.), and F1 scores evaluated on the test set.

Model	Data-CO			Data-NC		
	P.	R.	F1	P.	R.	F1
D2E*+SA	76.7	68.6	72.4	84.1	79.3	81.7
CAD-D2E*	**80.1**	**74.0**	**77.0**	**84.6**	**79.3**	**81.9**
	4.4%↑	7.9%↑	6.4%↑	0.6%↑	0	0.2%↑

Table 2: Role-level precision(P.), recall (R.), F1 scores evaluated on Data-CO and Data-NC sets.

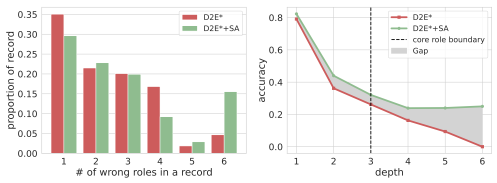


Figure 2: Detailed error analysis of the schema adjustment on *Equity Underweight* events.

Baselines. We compare with **Doc2EDAG** (Zheng et al., 2019), **DE-PPN** (Yang et al., 2021) and **GIT** (Xu et al., 2021). We apply CAD on Doc2EDAG and GIT. For Doc2EDAG, we test these variants: 1) **D2E***. Doc2EDAG with the same role order as the CAD for comparison; 2) **D2E*+SA**, D2E* only with the schema adjustment; 3) **CAD-D2E***; D2E* with schema adjustment and coreference aggregation. For GIT, as 8-layer GIT can't fit into a 12G memory GPU, we alter the encoder layer to 4 (**GIT**$_4$), the same as Doc2EDAG. And we also apply CAD on it (**CAD-GIT**$_4$).

4.2 Results and Analysis

Results. The performance results are shown in Table 1. We report GIT and DE-PPN results from the original paper. CAD improves 3.0, 2.7 role-level micro F1 compared with the D2E* and GIT$_4$, respectively. The improvements are more significant under record-level metrics, which are 4.3, 7.7 micro F1 respectively, which indicates CAD is more capable of extracting a complete event record. Note that the 4-layer CAD-GIT$_4$ can even outperform the 8-layer GIT. Experiments on these models demonstrate the effectiveness of CAD. We further show the results on each event type in the appendix.

Analysis. We take CAD-D2E* as an example and further analyze where the improvements come from. From the ablation test, we can observe that 1) the coreference module is of prime importance to the role level, bringing about 2.4 scores increase on F1; 2) the schema adjustment makes contributions to the record level, improving 3.2 scores on F1.

a) **Coreference aggregation**. Our patterns extract 17,656 new coreferential mentions in the dataset. According to whether there exists coreference in a document, we split the origin test set into two parts: data with and without coreference (Data-CO vs. Data-NC). We show the evaluation results on these two sets in Table 2. Adding the coreference aggregation results in more performance increase on the Data-CO compared with the Data-NC, which indicates that this module can help to obtain better entity representations.

b) **Schema structure adjustment**. We take event type *Equity Underweight* as example. As shown in Figure 2. In the left sub-figure, we show the proportions of wrong records vs. the number of wrong arguments in a record. The distribution of D2E*+SA relatively skews to the right. This indicates the wrongly extracted roles of D2E*+SA are more likely to appear in the same record, leading to higher record-level metrics. In the right sub-figure, we further analyze the role extraction accuracy along the depth of the DAG when there are several extraction errors on the preceding roles. Roles before the boundary (1,2,3) are core roles. As the extraction deepens, the accuracy of D2E* decreases sharper than D2E*+SA. Because D2E*+SA extracts non-core roles only depending on core roles, and thus alleviates the error propagation problem.

5 Conclusion and Future Work

We argue that besides the model architecture, the data characteristics are also indispensable to improving DEE. Note that: 1) The core role concept can be applied to a wide range of event types in other domains. 2) Advanced coreference detection can be interesting future work. 3) Our significant improvement may lead people to rethink why complex models can't capture these data characteristics and further inspire the future model design.

References

Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. 2021. Gate: Graph attention transformer encoder for cross-lingual relation and event extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Xinya Du and Claire Cardie. 2020. 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 2020, Online, July 5-10, 2020*, pages 8010–8020. Association for Computational Linguistics.

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

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly Multiple Events Extraction via Attention-based Graph Information Aggregation. In *EMNLP*, pages 1247–1256. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3533–3546. Association for Computational Linguistics.

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 5766–5770. Association for Computational Linguistics.

Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018a. DCFEE: A document-level chinese financial event extraction system based on automatically labeled training data. In *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, pages 50–55. Association for Computational Linguistics.

Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018b. DCFEE: A document-level chinese financial event extraction system based on automatically labeled training data. In *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, pages 50–55. Association for Computational Linguistics.

Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. 2021. Document-level event extraction via parallel prediction networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6298–6308. Association for Computational Linguistics.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 337–346. Association for Computational Linguistics.

Model	EF			ER			EU			EO			EP		
	P.	**R.**	**F1**	**P.**	**R.**	**F1**	**P.**	**R.**	**F1**	**P.**	**R.**	**F1**	**P.**	**R.**	**F1**
Doc2EDAG	72.8	62.1	67.0	**93.9**	82.6	87.9	**81.2**	59.6	68.8	79.9	65.8	72.2	**82.7**	68.4	74.9
D2E*	77.8	59.9	67.7	91.0	85.1	88.0	79.0	58.5	67.2	**81.6**	65.2	**72.5**	76.2	71.4	73.7
D2E*+SA	73.2	61.4	66.8	90.9	**87.6**	**89.2**	77.8	56.0	65.1	75.1	69.7	72.3	77.6	71.8	74.5
CAD-D2E*	**80.1**	**63.4**	**70.8**	91.1	85.3	88.1	76.2	**65.1**	**70.2**	73.7	**71.1**	72.4	80.7	**76.3**	**78.5**
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
GIT_4	78.5	66.2	71.9	**94.0**	85.5	89.5	80.0	65.4	72.0	78.3	70.5	74.2	**82.0**	71.0	76.1
CAD-GIT_4	**80.9**	63.9	71.4	92.5	**90.0**	**91.2**	78.6	66.3	71.9	**80.9**	71.5	75.9	81.8	**78.0**	**79.9**

Table 3: Overall role-level precision (P.), recall (R.) and F1 scores evaluated on the test set.

Model	EF			ER			EU			EO			EP		
	P.	**R.**	**F1**	**P.**	**R.**	**F1**	**P.**	**R.**	**F1**	**P.**	**R.**	**F1**	**P.**	**R.**	**F1**
Doc2EDAG	29.1	**25.5**	**27.1**	43.3	42.1	42.7	38.2	35.3	36.7	**35.7**	33.1	34.4	32.9	28.9	30.8
D2E*	28.3	23.3	25.6	47.2	47.2	47.2	33.5	31.7	32.6	33.8	30.8	32.2	28.7	28.4	28.6
D2E*+SA	25.2	22.1	23.6	**53.8**	**54.0**	**53.9**	**42.9**	35.6	38.9	34.0	33.5	33.8	30.9	30.0	30.4
CAD-D2E*	**29.6**	23.6	26.3	48.6	48.2	48.4	41.5	**40.0**	**40.8**	34.8	**35.7**	**35.3**	**35.2**	**34.3**	**34.7**
GIT	**32.5**	**28.8**	**30.5**	59.1	59.3	59.2	**45.8**	**42.4**	**44.0**	**38.1**	**37.5**	**37.8**	34.3	34.9	34.6
GIT_4	29.6	26.4	27.9	47.7	46.8	47.2	39.5	40.6	40.1	33.5	33.1	33.3	31.1	28.4	29.7
CAD-GIT_4	31.6	26.1	28.6	**62.0**	**62.1**	**62.0**	41.1	41.1	41.1	36.2	34.7	35.4	**37.4**	**36.5**	**36.9**

Table 4: Overall record-level precision (P.), recall (R.) and F1 scores evaluated on the test set.

Appendix

Event	#Train	#Dev	#Test	#Total
EF	608	125	164	897
ER	0	0	0	0
EU	1,637	130	121	2,888
EO	2,218	266	160	2,644
EP	9,005	1,247	1,065	11,317
All	13,468	1,768	1,510	17,746

Table 5: Number of detected coreference in each event type.

We train Doc2EDAG using the official code, and the role-level micro-F1 score is 77.1, which is close to results reported in previous studies, 76.3 and 77.5 (Zheng et al., 2019; Xu et al., 2021). The 8-layer GIT model can't fit into a 12G memory GPU, so we obtain the best prediction result from the authors and conduct analysis on that result. We acknowledge Xu et al. (2021) for sharing their results. For DE-PPN, we can only get the role-level F1 from the original paper. Due to the lack of time, we can't conduct experiments on this model and the results we report are based on a single run. We will do more experiments later.

We report detailed results on each event type in Table 3 and Table 4. Although 4-layer CAD-GIT_4 can only outperform 8-layer GIT on ER and EP events, CAD-GIT_4 still outperforms GIT on the overall record-level metrics, as EP events account for almost half of the dataset.

Table 5 shows the number of coreference detected in each event type. The distribution is related to the improvement of our coreference method. For example, EP has the most coreference mentions so that CAD-D2E* performs much better than D2E*+SA, and ER has no coreference so that CAD-D2E* performs similar or slightly worse than D2E*+SA.