Document-Level Event Extraction via Information Interactivion Based on Event Relation and Argument Correlation

Bangze Pan[1], **Yang Li**[2], **Suge Wang**[1,4]*, **Xiaoli Li**[3], **Deyu Li**[1,4], **Jian Liao**[1], **JianXing Zheng**[1]

[1]School of Computer and Information Technology, Shanxi University, Taiyuan, China
[2]School of Finance, Shanxi University of Finance and Economics, Taiyuan, China
[3]Institute for Infocomm Research, A*STAR, Singapore
[4]Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan, China
{bangze_pan, liyangprimrose}@163.com
{wsg, lidy, jxzheng, liaoj}@sxu.edu.cn
xlli@i2r.a-star.edu.sg

Abstract

Document-level Event Extraction (DEE) is a vital task in NLP as it seeks to automatically recognize and extract event information from a document. However, current approaches often overlook intricate relationships among events and subtle correlations among arguments within a document, which can significantly impact the effectiveness of event type recognition and the extraction of cross-sentence arguments in DEE task. This paper proposes a novel **C**orrelation **A**ssociation **I**nteractive **Net**work (CAINet), comprising two key components: *event relationship graph* and *argument correlation graph*. In particular, the *event relationship graph* models the relationship among various events through structural associations among event nodes and sentence nodes, to improve the accuracy of event recognition. On the other hand, the *arguments correlation graph* models the correlations among arguments by quantifying the strength of association among arguments, to effectively aggregate cross-sentence arguments, contributing to the overall success of DEE. Furthermore, we use the large language model to execute DEE task experiments. Experimental results show the proposed CAINet outperforms existing state-of-the-art models and large language models in terms of F1-score across two benchmark datasets.

Keywords: Document-level event extraction, Events relationship, Arguments correlation

1. Introduction

Document-level Event Extraction (**DEE**) aims at the recognition of event types and their associated arguments for event roles within a document. DEE exhibits several key characteristics:(1) **Multi-Events and Multi-Mentions**: It's common for a document to mention multiple events, and in some cases, a single event type can appear multiple times. As illustrated in Figure 1, a document contains instances of the *Equity Pledge* event and *Equity Freeze* event, with the *Equity Pledge* event appearing twice, each with a distinct cluster of arguments. (2) **Scattered Arguments**: In the majority of cases, the arguments for a given event are dispersed across multiple sentences within the document. Figure 1 highlights the arguments for the three event mentions are distributed across two to four different sentences.

Current methods still exhibit limitations in addressing two critical aspects of DEE: (1) **Integrating Event Relationships**: Recent work by (Zheng et al., 2019;Yang et al., 2021;Liang et al., 2022) have made strides in integrating sentence-to-entity and entity-to-entity relationships using variant Transformer models. However, these approaches often overlook DEE's multi-event and multi-mention characteristics. For instance, in Figure 1, we can observe that Event 1's role *Pledger*, Event 2's role *Pledger*, and Event 3's role *EquityHolder* is the same entity *Chengdu Maitian Investment Co. Ltd.*, highlighting the interconnectedness of three event mentions within a document. (2) **Handling Scattered Arguments:** Notably, (Xu et al., 2021) constructs a sentence-entity heterogeneous graph that fully connects sentence nodes and argument nodes based on entity positions, while (Wang et al., 2023) introduces proxy nodes in heterogeneous graphs to aggregate information from sentence and entity nodes unidirectionally. Nevertheless, these approaches still face limitations in addressing the characteristic of scattered arguments. As depicted in Figure 1, the arguments for Event 3 span across four different sentences, and similar challenges arise for Event 1 and Event 2 which have an expanded contextual span, necessitating the establishment of appropriate semantic relations among these arguments.

To address these limitations, we propose a novel document-level event extraction approach, named as the **C**orrelation **A**ssociation **I**nteractive **Net**work (**CAINet**). It explores the interplay between multiple events and the correlation among their respec-

* Corresponding authors

LREC-COLING 2024, pages 5156–5166
20-25 May, 2024.

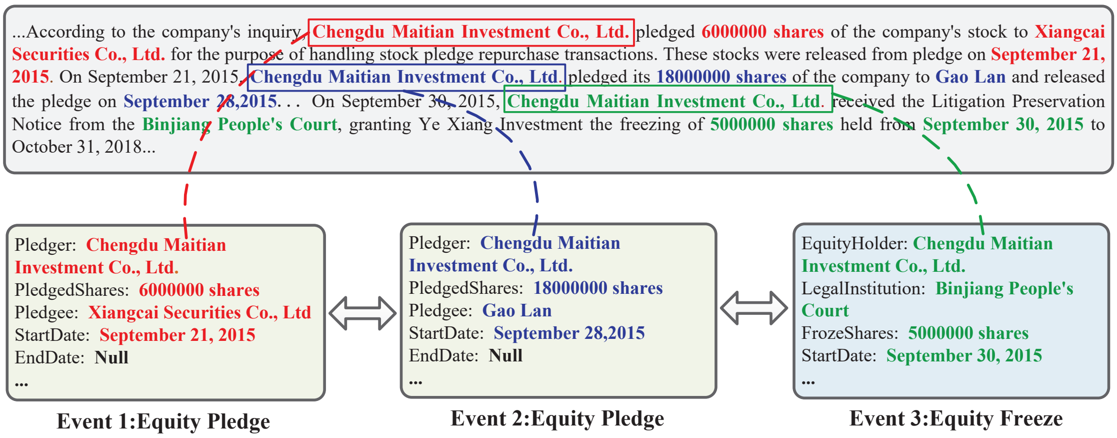

Figure 1: An example of a document that contains two event types. There are three events mentiones in the document, and the arguments of the same event are marked with the same color. We can observe that there have a same argument in the three event mentions, it have some relationship in them, and no sentence can contain all the arguments of the event.

tive arguments within the document simultaneously. Specifically, in CAINet, a heterogeneous graph interaction network is constructed, which comprises event nodes, sentence nodes, and entity nodes, and modeling their interactions through two key sub-graphs: event relationship graph and argument correlation graph. Finally, event nodes are used to capture global insights and determine event types. It also aggregates information from entity nodes to extract each argument associated with an event record. This comprehensive model significantly enhances document-level event extraction by considering both event relationships and argument correlations.

In summary, our main contributions are the followings:

- We propose a novel approach that contains a heterogeneous graph interaction network for document-level event extraction. Different from existing methods, we constructs an event sub-graph that focuses on event relationships and dynamically updates the representation of event nodes.

- We design a strategy to identify the correlation among arguments, and construct an argument correlation graph. It enhances the connection among cross-sentence arguments, thereby improving argument extraction performance.

- Our experimental results indicate that the CAINet outperforms existing state-of-the-art models on two challenging public datasets. Notably, we conduct experiments using both prompt-based and fine-tuning methods with the large language model, and the results show that CAINet is more effective than the large language model for DEE task.

2. Related Work

Historically, the predominant focus in related research has been on sentence-level event extraction (SEE). (Chen et al., 2015) introduced a pipeline model, breaking event extraction into two distinct subtasks: recognizing trigger words and extracting arguments. (Nguyen et al., 2016) devised a joint extraction model capable of simultaneously identifying trigger words and arguments. Notably, other approaches, such as the ones put forth by (Liu et al., 2018; Yan et al., 2019), harnessed dependency tree information for mining trigger words, thus enabling the determination of event types and their associated arguments. More recently, (Li et al., 2020;Lyu et al., 2021) have explored multi-turn Question-Answer (QA) techniques to enhance SEE's effectiveness. Additionally, several researchers (Paolini et al., 2021;Lu et al., 2021; Li et al., 2021) have employed sequence labeling for distinguishing event types and arguments. The advent of prompt learning has seen adopting prompt learning methodologies, crafting various prompt templates to harness the knowledge of pre-trained models in SEE tasks (Lin et al., 2021; Hsu et al., 2021; Ma et al., 2022).

However, it's essential to note that these SEE methods have exhibited limitations when applied to cross-sentence event extraction, while Document-Level Event Extraction (**DEE**) represents a more prevalent requirement in real-world scenarios. Recently, DEE has garnered great attention from both academic and industrial communities. DEE presents unique challenges, necessitating the modeling of long-term dependencies across multiple sentences and the extraction of arguments scattered throughout a document, spanning multiple events. To tackle these challenges, researchers have often framed DEE as a filling-table task. For instance, (Yang et al., 2018) introduced an initial model for critical event detection. While (Zheng et al., 2019) devised a path extension method for directed acyclic graphs to decode DEE, which has since become a foundation for many subsequent studies. (Lu et al., 2022) focused on filling intermediate information for DEE by capturing event-related clues. These approaches effectively integrate sentence and entity information, yet they tend to overlook the intricate relationships among events within a document. To address this, (Xu et al., 2021) introduced a heterogeneous graph to model sentences and entities in documents, although there remained limitations in capturing event dependencies. (Zhu et al., 2022) devised an entity clustering method that correlated entity groups with event types, though it didn't achieve the optimal experimental results. Notably, (Wang et al., 2023) developed a method employing proxy nodes to aggregate information from sentences and entities, optimizing them using the Hausdorff minimum distance approach. However, these methods often overlooked the correlation among arguments, which is a crucial aspect of DEE. Furthermore, more recent research by (Xu et al., 2022; Hsu et al., 2023; Yang et al., 2023) has explored the use of the abstract meaning representation(Banarescu et al., 2013) for document-level event argument extraction.

However, existing methods primarily follow a pipeline-like approach to model sentences information, they have somewhat overlooked the intricate dependencies among events and the essential correlations among arguments for DEE. Our proposed model adopts a heterogeneous graph interaction network approach to solves this deficiency.

3. The Proposed Method

Following previous studies (Wang et al., 2023), we formulate a heterogeneous graph that contains event nodes, sentence nodes, and entity nodes for DEE task. Figure 2 schematically shows our approach, which consists of three components: (1)**Document Representation and Entity Extraction:** Representing documents as dense vectors and extracting all entities from the document. (2)**Sub-Graph Construction and Heterogeneous Graph Fusion:** Defining two types of sub-graphs to effectively capture the interdependence among events and the correlation among arguments. (3)**Event Record Generation:** Employing the event nodes to determine the type of event, filling the entities to their corresponding event roles subsequently, and generating final event records.

3.1. Document Representation and Entity Extraction

Given a document D, CAINet extracts all entities from the document, which is treated as a sequence labeling problem. It means that a sequence labeling model is designed to assign a B(Begin), I(Inside), O(Other) label to each token in the document.

We employ BERT (Devlin et al., 2019) to encode each document, which is represented as $D_s = \{Sen_i\}_{i=1}^{N_s}$, where Sen_i denotes the vector representation of the *i-th* sentence, N^s is the number of sentences. Then D_s serves as the input of the linear classifier used to derive the document's tag sequence *S*. This enables us to generate the entity collection for the document $\{e_i\}_{i=1}^{N_e}$, where e_i is the vector representation of the *i-th* entity, and N_e is the total number of entities. During the training, our objective is to minimize the following losses:

$$\mathcal{L}_{ner} = -\sum_{i=1}^{N_s} \hat{y}_i log(\hat{D}_i) \quad (1)$$

Where $\hat{D}_i$ is the *i-th* label sequence of document, $\hat{y}_i$ is the *i-th* gold standard tag sequence.

3.2. Sub-Graph Construction and Heterogeneous Graph Fuse

Event Relationship Graph (ERG): In DEE, a document often contains multiple events. Recognizing the interplay and dependence among these events can significantly enhance event extraction. Thus, we construct the Event Relationship Graph (ERG) $G_{event} = (V_e, E_e)$, where V_e represents the nodes of the ERG and E_e represents the edges of the ERG.

For node set V_e, our graph comprises primarily event nodes and sentence nodes. We enhance contextual information exchange by feeding the sentence vector $\{Sen_i\}_{i=1}^{N_s}$ into a Transformer model. Consequently, we acquire the sentence vector representation $\{S_i\}_{i=1}^{N_s}$, which serves as the foundation for initializing the sentence node representation. We define *n* event nodes, where *n* is a fixed hyperparameter. These event nodes are initialized with random values to encode their node embeddings $\{h_i\}_{i=1}^{n}$, where h_i is the vector representation of the *i-th* event node, with each event node capable of representing an actual event mention or an empty event.

For edge E_e, we establish unidirectional connections between all sentence nodes *s* and each event node *h* as $\{S_i \longrightarrow h_j, i \in (0, N_s), j \in (0, n)\}$, with each event node serving as an aggregator of document information. Subsequently, we establish bidirectional connections $\{h_i \rightleftharpoons h_j, i \in (0, n), j \in (0, n)\}$ among all event

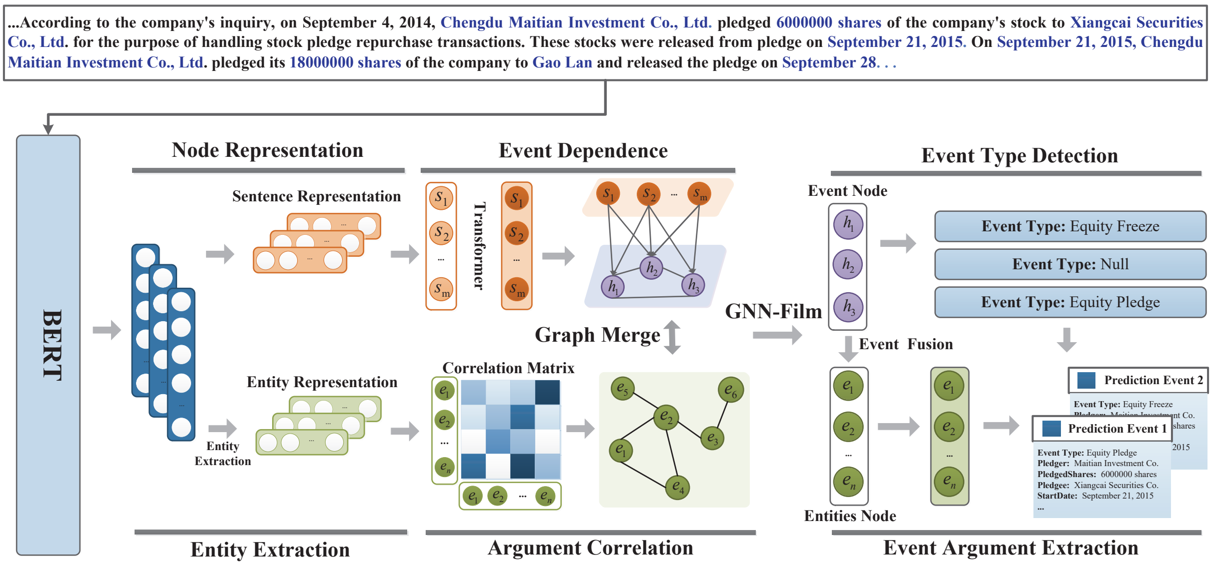

Figure 2: Overall architecture of our proposed CAINet model.

nodes h, facilitating comprehensive information exchange among them. This ERG proves to be an effective means of modeling the intricate relationships between events within the document.

Arguments Correlation Graph (ACG): Arguments in documents are frequently dispersed across multiple sentences, making it essential to model the interconnections between them. In this study, we construct the Argument Correlation Graph (ACG) $G_{argument} = (V_a, E_a)$, where V_a denotes the nodes of arguments, and E_a represents the connections between these arguments.

For node set V_a, our graph predominantly comprises entity nodes. We adopt $\{e_i\}_{i=1}^{N_e}$ to establish the initial representation for each entity node, where e_i denotes the representation of the *i-th* entity node.

For edge E_a, we extensive data analysis conducted for DEE task has revealed a discernible correlation among arguments within the same event. Following this insight, we assess the potential correlations between all entity pairs in the document, and establish bidirectional connections among relevant entities. More specifically, we devise a method to discern these argument correlations.

To handle entities dispersed across different sentences, we concatenate all entity vector representations, and input them into a Transformer model to enhance the holistic contextual awareness, as illustrated below:

$$\hat{e} = Transformer(concat(\{e_i\}_{i=1}^{N_e})) \tag{2}$$

Once we have acquired entity representations, we apply a dot-scaled similarity function to estimate their semantic correlation matrix as follows:

$$\tilde{e_i} = \hat{e_i} \times W_i + b_i \tag{3}$$

$$\tilde{e_j} = \hat{e_j} \times W_j + b_j \tag{4}$$

$$A_{i,j} = sigmoid(\tilde{e_i}^\top \tilde{e_j} / \sqrt{d_h}) \tag{5}$$

Where W_i,W_j,b_i,b_j is trainable parameter, $A_{i,j}$ represents the correlation strength of the entity pair of the *i-th* entity and the *j-th* entity, $\sqrt{d_h}$ is nodes hidden representation.

We set a predetermined threshold γ to determine the existence of correlations between each pair of entities. If the strength of correlation between entity pairs exceeds this threshold γ, we establish bidirectional connections between them in the form of $\{e_i \rightleftharpoons e_j, (i, j) \in \{i, j; A_{i,j} \geq \gamma\}\}$, thereby enabling information exchange among all correlated entities within the ACG. This approach effectively captures and models the correlations among entities.

Graph Fuse: We posit that when two entities stem from the same sentence, a correlation invariably exists, which aligns with the inherent logic of natural language. Consequently, we design the inner-sen argument edges, which bidirectional connections entities originating from the same sentence, represented as $\{e_i \rightleftharpoons e_j, (e_i, e_j) \in S_k, k \in (0, N_s)\}$ to bolster the relationships among intra-sentence entities.

Once the sub-graphs have been constructed, the next step involves fusing these two sub-graphs into a unified heterogeneous graph. To achieve this, we establish unidirectional connections from all entity nodes *e* to each event node *h*, symbolized as $\{e_i \longrightarrow h_j, i \in (0, N_e), j \in (0, n)\}$, thereby capturing the dependencies among events and the correlations among arguments within the framework of a heterogeneous graph.

When it comes to the design of heterogeneous graph convolution techniques, each node enhances its embedding by aggregating information from its neighboring nodes. In this study, we adopt a Graph Neural Network with Feature-wise Linear Modulation (Brockschmidt and Marc, 2020) for updating nodes. This approach introduces a hypernetwork, enabling each node to autonomously engage a polymerization function with distinct parameters.

We adopt the formula below to update each node, and the nodes updated through GNN-Film are represented as follows:

$$v = GNN - Film(\nu, \varepsilon) \tag{6}$$

Where ν is the updated representation of the node.

3.3. Event Records Generation

Event Type Prediction: Utilizing event nodes, the event type is predicted which is present in a document. Our approach involves connecting all event nodes and inputting them into a Transformer model to facilitate comprehensive information exchange among the event nodes, ultimately enabling us to classify the event type for each node:

$$p_i^{event} = softmax(MLP(v_i^h)) \tag{7}$$

Here, p_i^{event} represents the probability distribution of the event type for the *i-th* event node. An empty event type is included in the event type table, signifying that the event node doesn't correspond to any actual event type. The event type for the event node *i* is determined as follows:

$$c_i = argmax(p_i^{event}) \tag{8}$$

Event Argument Prediction: Since the same entity can have multiple mentions within a document, variations may exist in the vector representation of the same entity across different events. To address this, we utilize event nodes to apply a diversity-enhancing aggregation process to each entity. We use a Multi-Head Attention mechanism to integrate event node information into the entity representations:

$$\widehat{e}_{i,k} = MultiAttention(v_i^h, e_k, e_k) \tag{9}$$

Where v_i^h is representation of the *i-th* event node, e_k is representation of the *k-th* entity node. $\widehat{e}_{i,k}$ is polymerize representation of entity e_k regarding event node h_i. The role probability distribution of entity e_k regarding event h_i is given as follows:

$$p_{i,k}^{role} = softmax(MLP(v_i^h; \widehat{e}_{i,k})) \tag{10}$$

Where [;] is the concatenation operation. The role type set encompasses all the roles associated with the event type c_i , including an empty role to indicate that the entity does not assume any role within the event type c_i. The role type corresponding to the entity e_k is determined as follows:

$$a_{i,k} = argmax(p_{i,k}^{role}) \tag{11}$$

Finally, each event mention consists of an event type c_i and a set of arguments $\{a_{i,k}\}$. If multiple entities are classified into the same role in an event, we retain the entity with the highest probability as the argument for that particular role.

3.4. The Model Optimization based on Hausdorff Minimum

Inspired by (Wang et al., 2023), our optimization approach is set-based and employs the Hausdorff minimum distance (Schutze et al., 2012). Specifically, let the set of all predicted event sets in a document be U_z, the set of all ground-truth events be U_y, the event subset $u_i = (c_i, a_i)$ represents the *i-th* predicted event tuple, where c_i is event type, and a_i is the set of arguments. Furthermore, $u_j = (c_j, a_j)$ represents the j-th ground-truth event set in the document. The distance $d(u_i, u_j)$ between the event set u_i and the event set u_j is defined as follows:

$$d(u_i, u_j) = \sigma(c_i, c_j) + \frac{1}{|a_j|}\sum_{k=1}^{|a_j|}\sigma(a_{i,k}, a_{j,k}) \tag{12}$$

Where σ represents cross-entropy loss.

Our objective is to minimize the value of $D(U_z, U_y)$,which is obtained through the following optimization formula:

$$D(U_z, U_y) = \sum_{u_i \in U_z} \min_{u_j \in U_y} d(u_i, u_j) \tag{13}$$

Where $D(U_z, U_y)$ represents the distance between the predicted event set U_z and ground-truth event set U_y.

The final loss of the model is given by the following:

$$\mathcal{L}_{all} = D(U_z, U_y) + \mathcal{L}_{ner} \tag{14}$$

4. Experiments

4.1. Dataset

We use two benchmark datasets to evaluate our model's performance, namely, ChFinAnn(Zheng et al., 2019) and DuEE-Fin(Han et al., 2022) (1)ChFinAnn is the largest and most authoritative dataset for document-level event extraction tasks. Comprising 32,040 financial documents, it encompasses five distinct types of events. Notably, it features a distribution where 71% the documents pertain to single-event scenarios, while the remaining 29% involve multiple events. Additionally, a substantia

98% of the arguments are dispersed across multiple sentences. (2)DuEE-Fin the latest dataset for document-level event extraction. It includes 11,900 financial documents and encompasses 13 different event types. Similar to ChFinAnn, it exhibits a mixture of document types, with 67% being single-event documents and 33% representing scenarios with multiple events. Due to the absence of ground-truth annotations in the test documents, we extract a subset of 1,000 documents from the training data to serve as the test set for our experiments.

4.2. Experimental Setup and Evaluation Metrics

Experimental Setup: We employ the MacBert as a word vector embedding, with a hidden size of 512. We utilize 2-layer heterogeneous graph convolution method to update the node representations. The threshold for argument correlation is set to 0.5. We employ Adam optimizer(Kingma and Bac, 2015) and a learning rate 1e-4 for 100 epochs on an NVIDIA RTX A6000 GPU.

Evaluation Metrics: We follow the standard evaluation criteria (Zheng et al., 2019). Specifically, for each predicted event record, our approach selects the golden event record by matching it with the same event type and the closest argument. Subsequently, we gauge the alignment between their respective argument records to compute the final F1-score.

4.3. Baselines

We have chosen eight highly representative baseline models for comprehensive comparison: (1)**DCFEE**(Yang et al., 2018) extracts arguments from a central sentence within the document. It encompasses two variants **DCFEE-O** only extracts a single event mention, and **DCFEE-M** extracts multiple event mentions concurrently. (2)**Doc2EDAG**(Zheng et al., 2019) utilizes a path extension strategy to extract multiple events and their corresponding arguments.(3)**Greedy-Dec** is a variant of Doc2EDAG, employing a greedy strategy to populate the event table. (4)**DEPPN**(Yang et al., 2021) employs multiple Transformer frameworks to aggregate relationships among event types, roles, and arguments. (5) **GIT**(Xu et al., 2021) is the first to adopt a graph model, which connects sentence nodes and entity nodes in the token order of the document. (6)**PTPCG**(Zhu et al., 2022) aggregates arguments in the form of cliques and extracts events based on these argument cliques. (7)**ReDEE**(Liang et al., 2022) introduces a multi-scale relationship enhancement converter built on the Transformer framework. (8)**ProCNet**(Wang et al., 2023) incorporates proxy nodes to aggregate both sentence and argument information into these nodes using a graph model. The model is further optimized using the Hausdorff minimum distance method.

4.4. Overall Results

Table 1 shows the results of all models on ChFinAnn and DuEE-Fin datasets. The results clearly demonstrate that CAINet attains the highest F1-score on both datasets, affirming its universality and effectiveness. In comparison to the latest model, ProCNet, we still achieve improvements of 0.9% and 0.4%, respectively. Additionally, we segment the test dataset into two subsets: the single event dataset (S) and the multi-events dataset (M). It's evident that all models perform better on single-event scenarios compared to multi-event ones, underscoring the challenge posed by multi-event DEE tasks. CAINet is specifically designed to capture dependencies among multiple events and correlations between arguments, which is crucial in addressing multi-event DEE scenarios. This ability is the key reason why CAINet manages to enhance the F1-score for multi-event extraction in both datasets. Moreover, on ChFinAnn, where ample training data is available, CAINet significantly improves the F1-score for multi-event extraction.

To delve deeper into the models' performance across various event types, we use a more robust ChFinAnn dataset and conduct a comprehensive analysis of results across all event types. The outcomes of each model for the five event types are presented in Table 2. Notably, CAINet outperforms others in EF, EU, and EP event types. For ER, CAINet closely rivals the performance of ProCNet. However, CAINet's performance in the EO event type is not as remarkable. In EO events, the documents tend to be longer than the dataset's average document length. Compared to other models, CAINet places less emphasis on modeling sentence relationships. Consequently, GIT and ReDEE outperform CAINet when dealing with documents significantly longer than the dataset's average length. Nevertheless, it's important to note that this emphasis on sentence relationships can cause these models to be less effective in capturing event relationships and argument correlations, which are crucial for the overall performance of event extraction tasks.

4.5. Ablation Experiment

To verify the efficacy of the module, we design three ablation experiments. The -ACG experiment removes the Arguments Correlation Graph. The -EG experiment removes the Event Relationship Graph and relies on a type classifier to determine event types based solely on sentence node information. Finally, the -Inner experiment eliminates edges that connect entities within the same sentence. The

Model	ChFinAnn					DuEE-Fin				
	P	R	F1	F1(S.)	F1(M.)	P	R	F1	F1(S.)	F1(M.)
DCFEE-O	68.0	63.3	65.6	69.9	50.3	59.8	55.5	57.6	62.7	53.3
DCFEE-M	63.0	64.6	63.8	65.5	50.5	50.2	55.5	52.7	57.1	49.5
Doc2EDAG	82.7	75.2	78.8	83.9	67.3	67.1	60.1	63.4	69.1	58.7
Greedy-Dec	82.5	53.7	65.1	80.2	36.9	66.0	50.6	57.3	67.8	47.4
DEPPN	83.7	76.4	79.9	85.9	68.4	69.0	33.5	45.1	54.2	21.8
GIT*	83.6	76.9	80.1	87.5	72.1	69.8	65.9	67.8	73.7	63.8
PTPCG*	83.2	74.9	78.8	87.4	71.3	71.0	61.7	66.0	72.6	64.1
ReDEE*	83.9	79.9	81.9	88.7	74.1	77.0	72.0	74.4	78.9	70.6
ProCNet*	**84.4**	80.9	82.7	89.5	75.3	78.6	72.6	75.5	79.9	72.0
CAINet(Our)	84.3	**82.9**	**83.6**	**89.8**	**76.5**	**79.1**	**72.9**	**75.8**	**80.3**	**72.2**

Table 1: Results of all Models on ChFinAnn and DuEE-Fin, *represent we reproduce the results using their open-source codes

Model	EF	ER	EU	EO	EP
DCFEE-O	51.1	83.1	45.3	46.6	63.9
DCFEE-M	45.6	80.8	44.2	44.9	62.9
Doc2EDAG	70.2	87.3	71.8	75.0	77.3
Greedy-Dec	58.9	78.9	51.2	51.3	62.1
DEPPN	73.5	87.4	74.4	75.8	78.4
GIT	73.4	90.8	74.2	76.3	77.6
PTPCG	71.1	92.0	68.0	71.4	75.8
ReDEE	74.1	90.7	75.3	**78.1**	80.1
ProCNet	75.6	**93.8**	75.8	71.9	81.4
CAINet(Our)	**76.0**	93.6	**77.2**	72.9	**81.7**

Table 2: Results of each Model on the Five Event Types in ChFinAnn

	ChFinAnn			DuEE-Fin		
	P	R	F1	P	R	F1
All	**84.3**	**82.9**	**83.6**	**79.1**	**72.9**	**75.8**
-ACG	84.0	82.2	83.1	78.6	72.1	75.2
-EG	82.1	80.9	81.5	77.6	70.5	73.8
-Inner	85.2	81.2	83.3	78.8	72.4	75.4

Table 3: The Impact of Remove each Module on the Results of CAINet

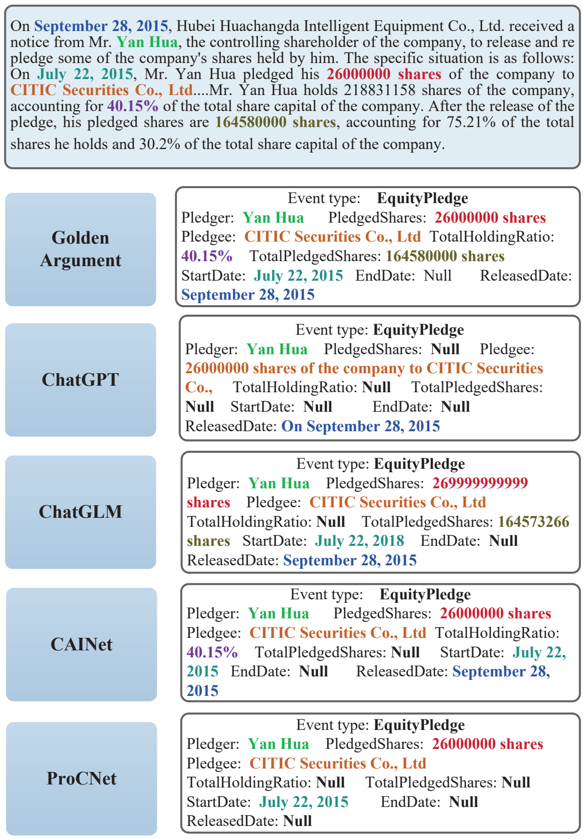

Figure 3: Performance comparison of across four models with a concrete example. Both CAINet and ProCNet outperform the large language model. Particularly, CAINet demonstrates the most impressive performance, excelling notably in cross-sentence argument recognition.

results are presented in Table 3. It is evident that the F1-score for the -ACG and -Inner configurations decreases by 0.5% and 0.3%, as well as 0.4% and 0.2%, on the ChFinAnn and DuEE-Fin datasets, respectively. Notably, the reduction in performance is more pronounced for the -ACG configuration compared to -Inner, underscoring the significance of addressing the challenge posed by arguments scattered across multiple sentences in DEE. Furthermore, the results reveal that the F1-score for the -EG configuration decreased by 2.1% on ChFinAnn and 1.8% on DuEE-Fin. This suggests that employing event nodes proves to be more effective than directly relying on sentence information to determine event types. The rationale behind this is that each event node encapsulates all the document information and only necessitates consideration of a single event type, substantially reducing the risk of error propagation.

4.6. Event Extraction and Large Language Models

Since the advent of generative large language models, such as ChatGPT, the landscape of NLP research has significantly evolved, and posing new challenges. The existing large language models are mainly divided into general-purpose LLM and open-source LLM. In this paper, we primarily utilize ChatGPT and ChatGLM to investigate the capabilities of these large language models in DEE task. We have designed different methods to complete DEE tasks based on their differences in characteristics. (1) **ChatGPT**: It is a commonly general-purpose LLM, we adopt a prompt method, devise effective prompts for ChatGPT to complete the DEE task. (2) **ChatGLM**: It is a commonly open-source LLM. For the ChatGLM, we aim to further inspire potential of large language models by fine-tuning the ChatGLM model using ChFinAnn's training dataset.

	P	R	F1
ChatGPT	37.3	34.2	35.7
ChatGLM	74.2	71.9	73.0
CAINet	**84.3**	**82.9**	**83.6**

Table 4: The Results of Two Large Language Models on DEE

The results presented in Table 4 clearly demonstrate that CAINet outperforms both ChatGPT and ChatGLM. Notably, the results from ChatGPT are substantially lower than those of ChatGLM, showing a notable 38.3% performance gap. This suggests that prompts alone may not be sufficient to fully harness the knowledge and capabilities of large language models. In comparison to ChatGLM, CAINet achieves a remarkable 10.6% improvement in F1-score. Even though ChatGLM has been trained on an extensive corpus and has acquired a wide range of linguistic knowledge, its ability to handle complex tasks appears to be somewhat limited.

4.7. Case Study

We conduct a case study to further validate the effectiveness of CAINet, as presented in Figure 3, where we analyze the characteristics of ChatGPT, ChatGLM, ProCNet, and CAINet. The results reveal that ChatGPT struggles to effectively recognize the majority of arguments, with many of its outputs being either 'NULL' or lengthy, continuous fragments. In contrast, ChatGLM demonstrates sensitivity to textual arguments, successfully extracting arguments for most textual categories, but struggles with the recognition of numerical arguments. This limitation is attributed to its auto-regressive generation nature, which makes it less adept at handling numerical arguments. This, in turn, suggests that extractive methods tend to be more effective than generative ones in the context of DEE tasks. Both ProCNet and CAINet exhibit relatively stable results. CAINet, in comparison to ProCNet, excels in extracting cross-sentence arguments, a feature attributed to the argument correlation graph in CAINet, which facilitates interaction with global arguments through correlation. Nevertheless, both models encounter challenges in modeling long-distance dependencies, underscoring the intricate nature of DEE tasks.

4.8. Impact of the Number of Events Nodes

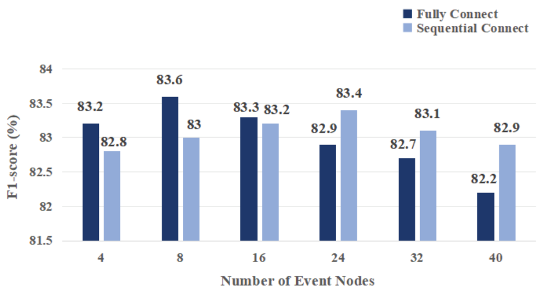

Figure 4: The impact of different numbers of event nodes on experimental results

This section delves into the impact of the number of event nodes on overall performance. We conduct experiments with varying numbers of event nodes and the results in terms of F1-score are depicted in Figure 4. Note **Full Connect** denotes bidirectional connections between all event nodes, while **Sequential Connect** signifies that each event node is only bidirectionally connected to one other event node. The insights derived from Figure 4 reveal the following observations:

(1)The **Full Connect** configuration yields optimal results with 8 event nodes, while the **Sequential Connect** setup performs best with 24 event nodes. Beyond these optimal numbers, the F1-scores start to decline, highlighting that an excessive number of event nodes may lead to misidentification of event types, whereas too few event nodes might result in some events being overlooked within the document.

(2)The **Full Connect**, in comparison to **Sequential Connect**, achieves optimal results with a smaller number of event nodes. This underscores the presence of inherent relationships among all events within a document. Additionally, **Full Connect** can expedite the training process, further enhancing its practicality.

5. Conclusion

In this paper, we propose a novel model that adeptly integrates event dependencies and argument correlations. Our proposed method utilizes graph models to comprehensively represent events, sentences, and entities, establishing the Event Relationship Graph and Argument Correlation Graph. These structures are further optimized using the Hausdorff minimum distance method. The experimental results show the superior performance of our proposed method, surpassing both the state-of-the-art model and the capabilities of ChatGPT and ChatGLM.

6. Limitations

In our work, it's important to acknowledge that our current model relies on a random initialization method to define event node representations. Yet, an intriguing challenge remains in achieving independence among different event representations. Additionally, an area warranting further exploration is the explicit modeling of sentence interrelations. Our future research endeavors will be directed toward advancing our understanding of these relationships among sentences, as this represents a significant avenue for improvement.

Acknowledgements

The authors would like to thank all anonymous reviewers for their valuable comments and suggestions which have significantly improved the quality and presentation of this paper. The works described in this paper are supported by the National Key Research and Development Program of China (2022QY0300-01), National Natural Science Foundation of China (62376143, 62106130, 62076158, 62072294, 62272286), Natural Science Foundation of Shanxi Province, China (20210302124084),Scientifc and Technological Innovation Programs of Higher Education Institutions in Shanxi, China (2021L284), and CCF-Zhipu AI Large Model Foundation of China (CCF-Zhipu202310).

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Brockschmidt and Marc. 2020. Gnn-film: Graph neural networks with feature-wise linear modulation. In *International Conference on Machine Learning*, pages 1144–1152. PMLR.

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 (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Cuiyun Han, Jinchuan Zhang, Xinyu Li, Guojin Xu, Weihua Peng, and Zengfeng Zeng. 2022. Duee-fin: A large-scale dataset for document-level event extraction. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 172–183. Springer.

I Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, et al. 2021. Degree: A data-efficient generation-based event extraction model. *arXiv preprint arXiv:2108.12724*.

I-Hung Hsu, Zhiyu Xie, Kuan-Hao Huang, Prem Natarajan, and Nanyun Peng. 2023. AMPERE: AMR-aware prefix for generation-based event argument extraction model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10976–10993, Toronto, Canada. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Bac. 2015. Adam: A method for stochastic optimization. In *In 3rd International Conference on Learning Representations*, San Diego, CA, USA. ICLR.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event extraction as multi-turn question answering. In *Findings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 829–838, Online. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Yuan Liang, Zhuoxuan Jiang, Di Yin, and Bo Ren. 2022. RAAT: Relation-augmented attention transformer for relation modeling in document-level event extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4985–4997, Seattle, United States. Association for Computational Linguistics.

Jiaju Lin, Jin Jian, and Qin Chen. 2021. Eliciting knowledge from language models for event extraction. *arXiv preprint arXiv:2109.05190*.

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*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Shudong Lu, Gang Zhao, Si Li, and Jun Guo. 2022. Explainable document-level event extraction via back-tracing to sentence-level event clues. *Knowledge-Based Systems*, 248:108715.

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 *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332, Online. Association for Computational Linguistics.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774, Dublin, Ireland. Association for Computational Linguistics.

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: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *In 9th International Conference on Learning Representations*, Online.

Oliver Schutze, Xavier Esquivel, Adriana Lara, and Carlos A Coello Coello. 2012. Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522.

Xinyu Wang, Lin Gui, and Yulan He. 2023. Document-level multi-event extraction with event proxy nodes and hausdorff distance minimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10118–10133, Toronto, Canada. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 3533–3546, Online. Association for Computational Linguistics.

Runxin Xu, Peiyi Wang, Tianyu Liu, Shuang Zeng, Baobao Chang, and Zhifang Sui. 2022. A two-stream AMR-enhanced model for document-level event argument extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5025–5036, Seattle, United States. 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 *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5766–5770, Hong Kong, China. Association for Computational Linguistics.

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*, pages 50–55, Melbourne, Australia. 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 (Volume 1: Long Papers)*, pages 6298–6308.

Yuqing Yang, Qipeng Guo, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2023. An AMR-based link prediction approach for document-level event argument extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 12876–12889, Toronto, Canada. 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*, pages 337–346, Hong Kong, China. Association for Computational Linguistics.

Tong Zhu, Xiaoye Qu, Wenliang Chen, Zhefeng Wang, Baoxing Huai, Nicholas Yuan, and Min Zhang. 2022. Efficient document-level event extraction via pseudotrigger-aware pruned complete graph. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, page 4552–4558. International Joint Conferences on Artificial Intelligence Organization.