Exploring Sentence Community for Document-Level Event Extraction

Yusheng Huang[1,2] and **Weijia Jia**[2,1,*]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]BNU-UIC Institute of Artificial Intelligence and Future Networks, Beijing Normal University
Guangdong Key Lab of AI and Multi-Modal Data Processing
BNU-HKBU United International College
{huangyusheng,jiawj}@sjtu.edu.cn

Abstract

Document-level event extraction is critical to various natural language processing tasks for providing structured information. Existing approaches by sequential modeling neglect the complex logic structures for long texts. In this paper, we leverage the entity interactions and sentence interactions within long documents, and transform each document into an undirected unweighted graph by exploiting the relationship between sentences. We introduce the *Sentence Community* to represent each event as a subgraph. Furthermore, our framework SCDEE maintains the ability to extract multiple events by sentence community detection using graph attention networks and alleviate the role overlapping issue by predicting arguments in terms of roles. Experiments demonstrate that our framework achieves competitive results over state-of-the-art methods on the large-scale document-level event extraction dataset.

1 Introduction

Document-level Event Extraction (DEE) aims to identify events in a long text with pre-specified types and corresponding event-specific argument roles. Figure 1 illustrates an DEE example for *Covid-19 Tracking* type with 5 arguments spreading across multiple sentences.

Generating document-level events is beneficial for a variety of natural language processing downstream tasks, such as knowledge base construction (Li et al., 2018), article summarization (Lee et al., 2003), and question answering (Srihari and Li, 2000), since it can produce valuable structured information. However, the complex logic structures in long documents have made it a more challenging task than Sentence-level Event Extraction (SEE) that extracts the event from the sentence.

Recently, a wide variety of deep neural network models (Nguyen et al., 2016; Yang et al., 2018; Sha et al., 2018; Yang et al., 2019; Ahmad et al., 2020; Ma et al., 2020a) have been proposed for event extraction, which could capture the semantic dependencies (mainly sequential dependencies) through recurrent neural networks or Transformer-based networks. However, existing models are mainly designed for sentence-level event extraction, omitting the complex interactions among entities or sentences in a long document. Therefore, document-level event extraction remains under-explored in spite of its importance. Intuitively, for long texts,

(1) **Entity Interaction**: Entities existing in the same sentence have a higher probability of being arguments of the same event. For example, in Figure 1, entities "*Israel*" and "*120*" in **[S3]** tend to portray the same event.

(2) **Sentence Interaction**: Sentences containing the same entity tend to narrate the same event. For example, in Figure 1, **[S1]-[S4]** containing the same entity "*Israel*" incline to depict the same event.

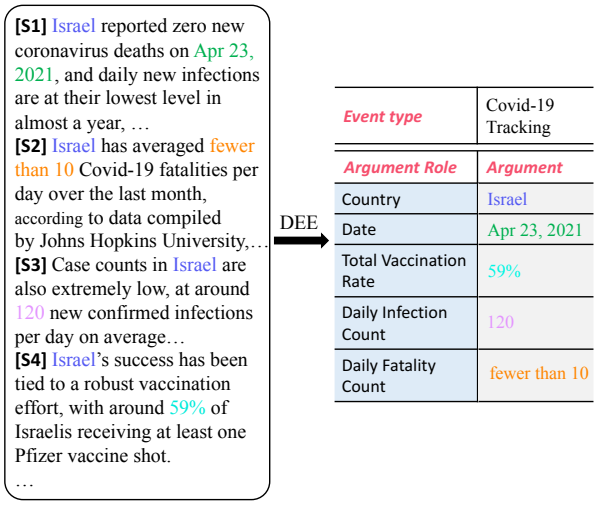

Figure 1: A DEE example for *Covid-19 Tracking* type with 5 related argument roles: *Country*, *Date*, *Total Vaccination Rate*, *Daily Infection Count*, and *Daily Fatality Count*.

*Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 340–351
November 7–11, 2021.

Considering the above properties, in this paper, we propose to build document graphs based on these interactions and bring the document-level event extraction from sequential modeling to graphical document representation, which could be exploited to handle multiple problems in DEE.

Specifically, we firstly propose a novel method that transforms each document into an undirected unweighted graph. Each sentence presents one node considering the entity interaction, and we assign each node with a comprehensively encoded attribute vector based on BERT (Devlin et al., 2019). Besides, the edges are constructed by entity co-occurrences between sentences in view of the sentence interaction. Compared with sequential modeling, graph structure maintains the capability to drain the information from long-distance sentences to their related sentences through much fewer transitions.

Second, we propose the so-called *Sentence Community* to represent each event as a subgraph of the constructed document graph. Specifically, we designate the sentence community by sentences that contain the arguments required for each event. In this way, the selected sentences also contain information about the corresponding event type. Therefore, each sentence community contains all the information needed for the event. Each sentence community corresponds to the related sentence nodes and edges in the document graph.

Third, we are able to mitigate the following issues based on our graphical representation: (1) *Multi-event issue.* Extracting multiple events for DEE is challenging because of argument scattering and overlapping. [1] In the real world, long texts are prone to contain multiple events. To extract multiple events, we employ Graph Attention Networks (GAT) (Velickovic et al., 2018) with the multi-head graph attention to detect overlapping sentence communities (Shchur and Günnemann, 2019), then we classify event types and extract corresponding arguments with an entity-level attention mechanism for each sentence community. (2) *Role overlapping issue.* An interesting problem in DEE is *role overlapping issue*, which refers to the phenomenon that an argument can play multiple roles, and few attentions have been paid to the problem. For example, in sentence "*On Mar 3 2021, FedEx pledges $2 billion toward sustainable energy initiatives*", the "*Mar 3 2021*" plays both the role "*StartDate*" and the role "*EndDate*" at the same time. We mitigate this issue by predicting arguments in terms of roles.

In summary, our contributions include:

- We propose a novel graph construction method for long documents with the comprehensively encoded attribute vector for each sentence node.
- We propose a novel framework SCDEE that explores **S**entence **C**ommunity for **D**ocument-level **E**vent **E**xtraction, which alleviates the multi-event issue and the role overlapping issue.
- We perform a thorough evaluation of our framework and show the effectiveness on a large-scale document-level event extraction dataset.

2 Methodology

In this section, we present our proposed framework. We first introduce the document graph construction method. Then we present the GNN-based sentence community detection approach. Finally, we explain the event type and argument classification module. An overview is shown in Figure 2.

2.1 Document Graph Construction

We denote one document D as a sequence of sentences $D = [s_1, ..., s_i, ..., s_N]$. For each document, we construct an undirected unweighted graph $G = (V, E)$, where the number of nodes $V = \{v_1, v_2, ..., v_N\}$ equals the number of sentences and $E = \{(u, v) \in V \times V : A_{uv} = 1\}$ is the set of edges where $A \in \{0, 1\}^{N \times N}$ is a binary adjacency matrix.

Adjacency Matrix. The adjacency matrix is constructed based on the entity co-occurrences between sentences. For each sentence, entities are recognized by the well-performed BI-LSTM-CRF (Huang et al., 2015) model. Then we set $A_{ij} = A_{ji} = 1$ for any sentences s_i and s_j containing the same entity. Besides, we add self-loops for A, i.e. $A_{ii} = 1$ for $1 \leq i \leq N$.

Node Attribute Vector. To comprehensively encode the sentence information for each node, the attribute vector is constructed based on two segments: (1) the entity-level feature vector α that presents the information of event argument candidates, and (2) the sentence-level feature vector β that reflects the information of the event type.

[1] Overlapping means events might share arguments.

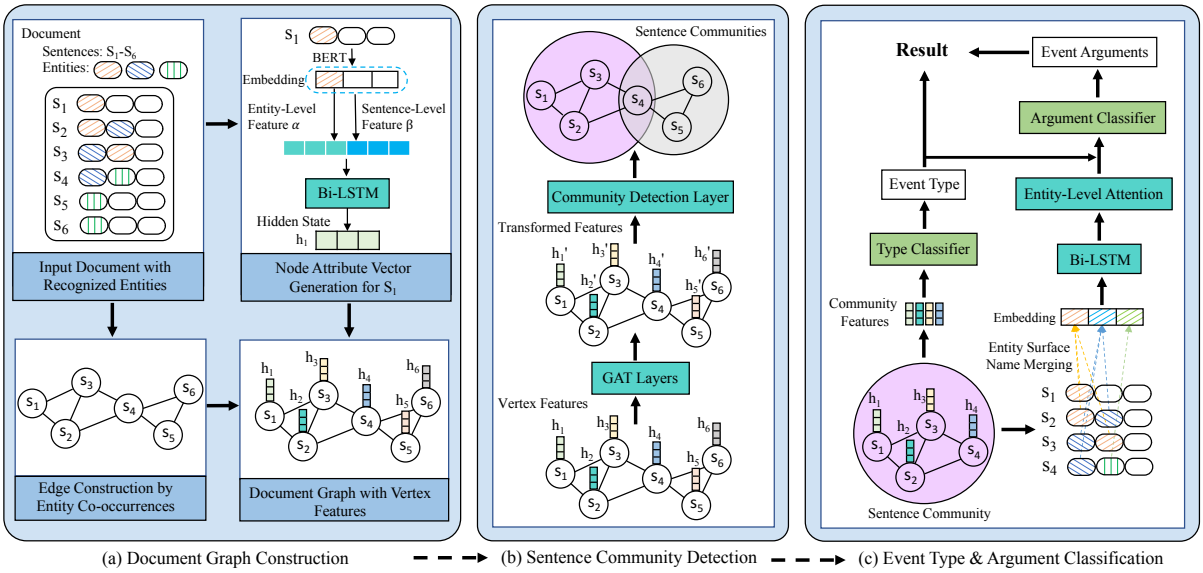

Figure 2: An overview of our SCDEE architecture. The input document contains 6 sentences with 2 events. Arguments of the first event (in orange and blue) are scattered in S1-S4, which form the first sentence community (in purple). Arguments of the second event (in green) are scattered in S4-S6, which form the second sentence community (in grey). The two sentence communities overlap on S4.

Specifically, for each sentence s_i containing N_i words, we employ BERT representation model on s_i and obtain the embedding vector of the last layer $B_i \in \mathbb{R}^{N_i \times d_B}$, where d_B denotes the hidden layer dimensionality of BERT. For each recognized entity in s_i covering the j_{th} to k_{th} tokens, we obtain the entity embedding $e_i \in \mathbb{R}^{d_B}$ by conducting a max-pooling operation on corresponding index range of B_i, i.e.,

$$e_i = \text{maxpool}(B_{i,j}, B_{i,j+1}, ..., B_{i,k}) \quad (1)$$

Then we conduct another max-pooling operation on all the existing l entities in s_i to obtain the fix-sized entity-level feature vector $\alpha \in \mathbb{R}^{d_B}$,

$$\alpha = \text{maxpool}(e_1, e_2, ..., e_l) \quad (2)$$

The sentence-level feature vector β is obtained by max-pooling on B_i.

Finally, we employ a Bi-LSTM layer on the concatenation of α and β to get the node attribute vector $h_i \in \mathbb{R}^D$,

$$h_i = \text{Bi}-\text{LSTM}(\alpha \| \beta) \quad (3)$$

where D is the dimensionality of Bi-LSTM hidden states, and $\|$ denotes the concatenation operation.

2.2 Sentence Community Detection

Given the constructed document graph $G = (V, E)$ with N vertices and node attribute vectors $\mathbf{h} = [h_1, h_2, ..., h_N]$, we first generate the target sentence community for each event within the document. Then we propose to utilize GAT networks to detect overlapping sentence communities as nodes might be shared by several sentence communities.

Target Sentence Community. For a document containing C events and N sentences, we construct a binary affiliation matrix $F \in \{0, 1\}^{N \times C}$ with each column representing one sentence community, and we set $F_{i,j} = 1$ if the ith sentence contains any argument of the jth event. Each sentence may be assigned to multiple sentence communities or no sentence community, depending on whether these sentence communities overlap with each other.

Community Detection via GAT. We employ GAT to model the information flow between nodes and predict overlapping sentence communities. There are several advantages of utilizing GNN-based models for overlapping sentence community detection. First, GNN could capture long-range dependencies between sentences through edges. Second, GNN tends to produce similar community affiliation vectors for the densely connected subgraphs.

In our implementation, we exploit GAT for sentence community detection. The local node attribute vectors can be further aggregated into more informative vectors by attention mechanism over its neighbor features. Besides, GAT does not depend on upfront access to the global graph structure as the attention mechanism is applied in a shared man-

ner to all edges in a graph. Therefore, it is directly applicable to *inductive* learning, which means it could predict communities inductively on graphs that are completely unseen during training.

In general, the input to GAT layers is an undirected unweight graph $G = (V, E)$ with the adjacency matrix A and node attribute vectors $\mathbf{h} = [h_1, ..., h_i, ..., h_N]$, $h_i \in \mathbb{R}^D$. We use D' to denote the cardinality of GAT outputs. We briefly describe the GAT layer used in our implementation. The attention score α_{ij} that indicates the importance of the neighbor node j to the attended node i is

$$\alpha_{ij} = \frac{\exp\left(\sigma\left(\vec{\mathbf{a}}^T\left[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j\right]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\sigma\left(\vec{\mathbf{a}}^T\left[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k\right]\right)\right)} \quad (4)$$

where σ is LeakyReLU activation, $\vec{\mathbf{a}} \in \mathbb{R}^{2D'}$ is a fully connected layer, $\cdot^T$ represents transposition, $\mathbf{W} \in \mathbb{R}^{D'\times D}$ denotes a weight matrix, and $\mathcal{N}_i$ denotes the neighbors of node i.

We employ the multi-head attention mechanism with K heads to capture more information from different representation subspaces:

$$\vec{h}'_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}^k\mathbf{W}^k\vec{h}_j\right) \quad (5)$$

Then we obtain the predicted feature matrix $X \in \mathbb{R}^{N\times(K\cdot D')}$ by stacking the K-head GAT outputs $h'_i \in \mathbb{R}^{K\cdot D'}, i = 1, 2, ..., N$.

We employ a Multi-Layer Perceptron (MLP) on X with hidden dimensions of $2C$, and we further reshape it as $\mathbb{R}^{N\times 2\times C}$ with 2 being the cardinality of the target affiliation matrix $F \in \{0, 1\}^{N\times C}$. Then we employ softamx on X, i.e.,

$$P = \mathrm{softmax}(\mathrm{reshape}(\mathrm{MLP}(X))) \quad (6)$$

where values along the second dimension of $P \in \mathbb{R}^{N\times 2\times C}$ represent the probabilities of nodes affiliating sentence communities.

We assign node $v_i \in V$ to sentence community $c \in C$ if the corresponding probability is more than half, and we can further obtain our predicted affiliation matrix $F' \in \{0, 1\}^{N\times C}$.

Besides, we calculate the high-dimension cross-entropy loss $\mathcal{L}_{CD}$ based on P and the target affiliation matrix F:

$$\mathcal{L}_{CD} = -\frac{1}{N\times C}\sum_{1\le i\le N, 1\le j\le C}\log P^{i,:,j}_{F^{i,j}} \quad (7)$$

2.3 Event Type & Argument Classification

2.3.1 Event Type Classification

We predict the event type for the sentence community j based on the predicted affiliation matrix $F' \in \{0, 1\}^{N\times C}$. First, the embedding for the event E_{event} is obtained by conducting a max-pooling operation on the selected node attribute vectors,

$$E_{event} = \mathrm{maxpool}(F'^T_{:,j} \odot \mathbf{h}) \quad (8)$$

where $\odot$ denotes element-wise product.

Then, for all pre-defined V target event types, the event type is predicted by applying a fully connected layer on the event embedding E_{event} with softmax function to estimate the probability distribution, i.e.,

$$\hat{p} = \mathrm{softmax}(\mathbf{W}E_{event} + \mathbf{b}) \quad (9)$$

where $\mathbf{W} \in \mathbb{R}^{V\times D}$ and $\mathbf{b} \in \mathbb{R}^V$ are weights.

The loss function for event type classification $\mathcal{L}_{ET}$ is the cross-entropy loss,

$$\mathcal{L}_{ET} = -\log \hat{p}_{y_{ET}} \quad (10)$$

where y_{ET} is the label of the event type.

2.3.2 Event Argument Classification

Given the sentences in each sentence community and the predicted event type, we extract the corresponding arguments. First, we take out the entities within these sentences and their embeddings as depicted in Equation 1. For entities preserving the same surface name, we merge their embeddings by max-pooling operation. Then, we obtain m entity embeddings with distinct surface names, which are denoted as $E \in \mathbb{R}^{m\times d_B}$. We employ a Bi-LSTM layer to make the embeddings more informative and obtain $E' \in \mathbb{R}^{m\times L}$ with L being the hidden size of Bi-LSTM.

Entity-Level Attention Layer. To capture the associations between entities, we further design an entity-level attention mechanism to aggregate information. The attention score $\alpha_i \in \mathbb{R}^m$ (similarity or relatedness) is calculated as follows

$$r_i = \tanh(\mathbf{W}E'_{i,:} + \mathbf{b}) \quad (11)$$

$$\alpha_i = \mathrm{softmax}(r_i) = \frac{\exp(r_i)}{\sum_{t=1}^m \exp(r_t)} \quad (12)$$

where $\mathbf{W} \in \mathbb{R}^L$, $\mathbf{b} \in \mathbb{R}$ are weights.

Then the final entity embedding $F_i \in \mathbb{R}^{2L}$ is computed by:

$$F_i = [\sum_{j=1,j!=i}^{m} \alpha_j * E'_j, E'_i] \quad (13)$$

Role Overlapping Issue. We predict arguments for each argument role to mitigate this issue. First, we feed the final entity embedding F_i to a sigmoid function to simulate the relative scores for argument classification instead of the ordinary softmax classifier:

$$\hat{p}_{EA} = \text{sigmoid}(\mathbf{W}F_i + \mathbf{b}) \quad (14)$$

where $\mathbf{W} \in \mathbb{R}^{C\times 2L}$, $\mathbf{b} \in \mathbb{R}^C$ are weights, and C denotes the number of roles corresponding to the predicted event type.

Then for each role, we select the entity with the highest score that exceeds the threshold p_0 as the argument. In this way, an entity can be the argument for multiple roles.

We assume the ground truth label for each role is $\mathbf{y} \in \mathbb{R}^C$, where $y^i \in \{0, 1\}$ denotes whether the entity is the argument, and we utilize the binary cross-entropy loss $\mathcal{L}_{EA}$ for argument classification as follows

$$\mathcal{L}_{EA} = -\sum_{i=1}^{C} y^i \log \hat{p}^i_{EA} + \left(1 - y^i\right) \log(1 - \hat{p}^i_{EA}) \quad (15)$$

2.4 Objective Function

We utilize the weighted summation of $\mathcal{L}_{CD}$, $\mathcal{L}_{ET}$, $\mathcal{L}_{EA}$ as our final loss, i.e.,

$$\mathcal{L}_{all} = \lambda_1 \mathcal{L}_{CD} + \lambda_2 \mathcal{L}_{ET} + \lambda_3 \mathcal{L}_{EA} \quad (16)$$

where λ_1, λ_2 and λ_3 are hyper-parameters.

3 Experiment

3.1 Experimental Setup

Dataset. We conduct the experiments on the large-scale Chinese Financial event extraction dataset constructed by (Zheng et al., 2019). This dataset contains 32040 documents in total. The major feature of the dataset is that around 29% documents contain multiple events, which makes extracting multiple events an inevitable issue. There are five pre-defined event types: Equity Freeze (EF), Equity Repurchase (ER), Equity Underweight (EU), Equity Overweight (EO), and Equity Pledge (EP) with 8, 6, 6, 6, and 9 pre-defined roles, respectively. The training set accounts for 80%, and both the development and test set account for 10%. The detailed statistics are shown in Table 1.

Event	Train	Dev	Test	Total
EF	806	186	204	1196
ER	1862	297	282	3677
EU	5268	677	346	5847
EO	5101	570	1138	6017
EP	12857	1491	1254	15602
All	25632	3204	3204	32040

Table 1: Dataset statistics for the training set, development set and test set.

We can see from Table 1 that the number of EP type is much larger than other types. Therefore, in each epoch, we randomly sample 40% of the EP type, which is a similar size of EU and EO types. Besides, we randomly resample the documents so that the number of single-event, double-event, and triple-event documents are the same.

Implementation Details. In our experiments, we set the hidden dimensions of all the LSTM layers used in our framework to be 250, and set the dropout rate to be 0.2 in order to avoid overfitting. We employ a one-layer GAT model with $K = 3$ attention heads computing $D' = 200$ features per head (for a total of 400 features). In the event argument classification part, the probability threshold p_0 is set to be 0.5 to mitigate the role overlapping issue. During training, we set $\lambda_1 = 3$, $\lambda_2 = \lambda_3 = 1$ in the objective function. We employ the Adam (Kingma and Ba, 2015) to optimize the model parameters with the initial learning rate being 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. We implement our model in PyTorch 1.7.1 with one NVIDIA Titan Xp GPU. For all experiments, we set the maximal number of training epochs to be 50.

Evaluation Metrics. The goal of DEE is to correctly predict the event type and extract the related arguments. Following (Zheng et al., 2019), for each document, we select the most similar predicted event record when the predicted event type is correct, and then we calculate the event-role-specific true positive, false positive, and false negative statistics until no target event records left. Then we aggregate all the statistics for each event type and present the precision and F1 scores in the percentage format.

Model	EF		ER		EU		EO		EP		Average	
	P.	F1	P.	F1	P.	F1	P.	F1	P.	F1	P.	F1
DCFEE-O	66.0	51.1	84.5	83.1	62.7	45.3	51.4	46.6	64.3	63.9	65.8	58.0
DCFEE-M	51.8	45.6	83.7	80.8	49.5	44.2	42.5	44.9	59.8	62.9	57.5	55.7
GreedyDec	79.5	58.9	83.3	78.9	68.7	51.2	69.7	51.3	**85.7**	62.1	77.4	60.5
Doc2EDAG	77.1	70.2	91.3	87.3	80.2	71.8	82.1	**75.0**	80.0	77.3	82.1	76.3
SCDEE	**88.4**	**80.4**	**93.7**	**90.5**	**84.3**	**75.1**	**85.5**	70.1	84.4	**78.1**	**87.2**	**78.9**

Table 2: Precision and F1 scores for each event type and the averaged performance on the test set. Bold denotes the best result. Due to space limitation, we report the recall scores of each event type in Appendix A.

3.2 Experimental Results and Analysis

Baseline Models. In order to comprehensively evaluate our framework, we compare it with these following state-of-the-art baselines:

• **DCFEE** (Yang et al., 2018) employs the argument-completion strategy to generate the document-level event record by utilizing the arguments from sentences-level event extraction results. In order to handle multi-event extraction, **DCFEE-O** and **DCFEE-M** (Zheng et al., 2019) are proposed by producing one event record and multiple possible argument combinations from one key-event sentence respectively.

• **Doc2EDAG** (Zheng et al., 2019) generates an entity-based directed acyclic graph to extract multiple events from documents. Besides, the **GreedyDec** fills one event table entry greedily by using recognized entity roles, which shares the same architecture with Doc2EDAG.

Main Results. Table 2 presents the performance comparison of different models. Overall, our framework SCDEE outperforms all other methods on the test set and improves 5.1% and 2.6% on the averaged precision and F1-scores over the state-of-the-art Doc2EDAG model. Specifically, compared with DCFEE-O and DCFEE-M, our framework achieves better results both in precision and F1-scores on all the five event types. When compared with GreedyDec that holds relatively high precision, our framework still improves 9.8% on the averaged precision.

Performance Analysis. Concretely, we transform the long document into a graph and provide shortcuts for closely related sentences in a sentence community. Compared with DCFEE-O and DCFEE-M that predicted missing arguments from surrounding sentences, we believe the improvements of DCFEE should give credit to the graph structure and the GAT layer, which alleviate the long-range dependency issue. When comparing with GreedyDec that extracts events greedily using the recognized entity roles, we consider the reason may lie in the stronger association between entities within the same sentence, which means that these entities are more likely to portray the same event. The overall performance of the strongest baseline Doc2EDAG is slightly inferior to our model. Though Doc2EDAG generates multiple events by path-expanding subtasks, they ignore the role overlap problem in DEE. We further alleviate this problem by predicting arguments in terms of roles in our framework.

Model	EF	ER	EU	EO	EP	Avg
SCDEE	80.4	90.5	75.1	70.1	78.1	78.9
-GAT	-3.7	-2.5	-0.4	-2.5	-1.2	-2.0
-ELA	-3.5	-1.2	-1.7	-3.0	-1.0	-2.1
-ROI	-7.4	-0.1	-4.3	-6.1	-3.4	-4.2

Table 3: Overall F1-scores decreasing of ablation experiments. Avg denotes the averaged scores.

Type	SCDEE		-GAT		-ELA		-ROI	
	P.	F1	P.	F1	P.	F1	P.	F1
S	**92.4**	**88.7**	91.3	87.6	92.2	87.9	92.3	85.6
M	**78.9**	**65.8**	78.0	63.2	78.8	62.7	74.8	60.6

Table 4: Overall precision and F1-scores for documents containing single event (S) and multiple events (M).

3.3 Ablation Study

As shown in Table 3, we conduct ablation experiments by evaluating three key designs to demonstrate the effectiveness of components in our framework.

• **–GAT**. We investigate the effectiveness of the GAT layers in our framework. To be fair, we replace the GAT layer with a fully connected layer. Experimental results show the effectiveness of the GAT networks on our framework.

• **–ELA**. We remove the entity-level attention layer that aims to capture the association between

Heads	EF	ER	EU	EO	EP	Avg
1-head	79.2	**92.3**	76.1	68.9	77.8	78.8
2-head	77.6	91.5	**78.6**	68.0	76.8	78.5
3-head	**80.4**	90.5	75.1	**70.1**	78.1	**78.9**
4-head	78.8	89.6	78.1	68.9	**78.2**	78.7
5-head	77.3	90.5	75.2	67.5	77.9	77.7
6-head	78.7	90.0	74.4	66.7	76.2	77.2

Table 5: F1-scores for the single-layer GAT network with different number of heads.

entities. We show that the attention layer is helpful to incorporate the information from other entities and improve the overall performance.

• **–ROI**. We replace the sigmoid function and binary cross-entropy loss in the event argument classification with the general softmax classifier and cross-entropy loss respectively in order to explore how the role overlapping issue affects the experimental results. We find that F1 scores of the EF and EO types drop significantly, which might mean that they suffer the most from this issue.

3.4 Single & Multiple DEE Analysis

We conduct experiments to study the performance of our framework on single-event and multi-event documents, and the influence of the aforementioned three key components. As shown in Table 4, we find that (1) for single-event documents, our framework achieves superior performance in terms of both precision and F1-scores. In addition, **–GAT** leads to the most decrease in precision, and **–ROI** causes the most F1-score decrease, which means that the role overlapping issue might be the critical obstacle. (2) For multiple-event documents, our framework achieves fairish performance. Besides, **–ROI** results in noteworthy performance degradation both in precision and F1-scores. It demonstrates that the role overlapping issue hinders the performance of multiple event extraction.

3.5 Effect of GAT Architecture

We conduct experiments to see how the model's performance is affected by the GAT network architecture. First, we perform a set of experiments on a single-layer GAT network with a different number of heads. Experimental results in Table 5 show that there is no notable difference between 1-head and 4-head GAT. However, more time is needed for convergence as parameters are increasing. But more heads lead to performance degradation.

Layer	2-1 Head	2-2 Head	2-3 Head
1-1 Head	**76.5**	74.3	65.3
1-2 Head	74.3	73.6	62.4
1-3 Head	72.7	72.1	58.6

Table 6: Overall f1-scores for the two-layer GAT networks with different number of heads. i-j *Head* denotes the i_{th} layer with j heads.

The deeper, the better? We further investigate the framework performance using two-layer GAT networks with different numbers of heads. We employ the exponential linear unit (ELU) (Clevert et al., 2016) as the activation function between layers. As described in Table 6, the overall F1 scores significantly drop whether we increase the number of heads in the first or the second layer. The possible reason for the overall performance dropping may lie in the over-smoothing issue (Zhou et al., 2018) that the node attribute vectors tend to converge to similar values.

3.6 Time complexity

In news articles, entities are usually extracted in advance by highly efficient tools in real-world industry applications. For the document graph construction $G = (V, E)$, let N_s be the number of sentences, N_e be the number of all extracted entities, and N_u denotes the number of entities with distinct surface names. Then generating node attribute vectors requires $O(N_e)$ complexity. For the sentence community detection, the GAT layer requires $O(N_s \cdot D \cdot D' + |E| \cdot D')$ with D and D' representing the input and output dimensionality, and the complexity of node assignment is $O(N_s)$. For the argument classification, the complexity of the entity attention layer is $O(N_u)$. Notably, N_s, N_e, and N_u are far less than the length of documents, which makes our model work efficiently.

3.7 Case Study

We visualize the graph structure of the document and analyse its property as shown in Figure 3.

First, as shown in Figure 3(a), two thirds of the sentences contain no entity. Our framework could filter the noise sentences and focus on informative sentences, which is an advantage compared with the baseline DCFEE.

Second, in Figure 3(b), from the perspective of sentence community, the document graph is composed of two overlapping sentence commu-

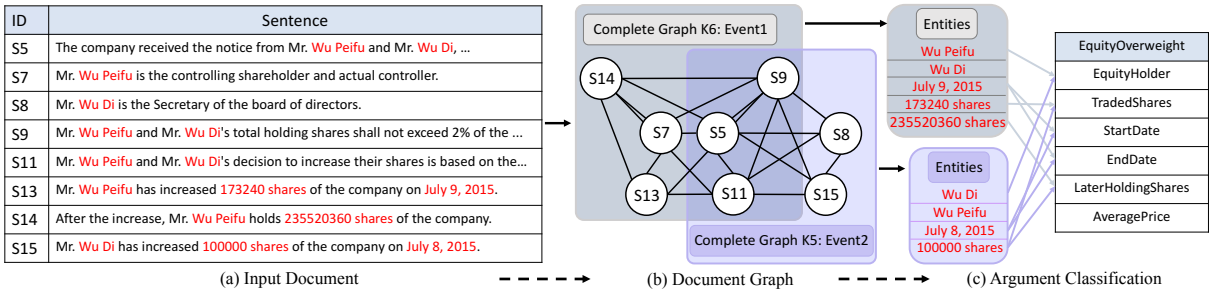

Figure 3: (a) an example of the document containing 24 sentences with 2 *EquityOerweight* events. We exclusively present the 8 sentences with recognized entities (in red). (b) an example of the document graph with two sentence communities. The first sentence community corresponds to the complete graph *K6*. The second community corresponds to the complete graph *K5*. (c) an example of argument classification. An entity might be classified into multiple roles if these roles overlap.

nities. Notably, the first sentence community corresponds to the complete graph $K6$ since all the sentence nodes share the entity *Wu Peifu*. The second sentence community corresponds to the complete graph $K5$ with all the sentence nodes sharing the entity *Wu Di*. Sentences related to each event are densely connected under the definition of sentence community.

Third, as depicted in Figure 3(c), our framework could reduce the irrelevant argument candidates for each event as compared with our baseline Doc2EDAG. Entities within each sentence community are more closely related.

The above results verify that graphical representation is advantageous for document-level event extraction.

4 Related Work

Event Extraction (EE), a challenging sub-task of information extraction, has been recently studied under two paradigms: the sentence-level EE and document-level EE.

Sentence-level Event Extraction mainly follows the requirements of ACE event extraction task (Doddington et al., 2004) that aims to detect the event trigger and arguments from a sentence. This task can be further decomposed into two sub-tasks: *Event Detection* that aims to identify the event triggers (Feng et al., 2016; Liu et al., 2017; Zhao et al., 2018; Yan et al., 2019; Cui et al., 2020; Lai et al., 2020a,b) and *Event Argument Role Labeling* that aims to predict whether words or phrases participate in the event argument roles (Wang et al., 2019; Yun et al., 2019; Pouran Ben Veyseh et al., 2020; Ma et al., 2020b; Ahmad et al., 2020; Zhang et al., 2020). Furthermore, various researches have been dedicated to extracting event triggers and arguments simultaneously (Sha et al., 2018; Yang et al., 2019; Tang et al., 2020; Du and Cardie, 2020b).

Document-level Event Extraction aims to identify event types and corresponding event argument roles. Compared with sentence-level event extraction, the main difference is that it is no longer necessary to identify the event trigger words explicitly.

From the perspective of modeling, Yang et al. (2018) employ a sequence tagging model to extract document-level events by utilizing sentence-level results. Zheng et al. (2019) propose an end-to-end model that transforms the DEE task into several sequential path-expanding sub-tasks with each final path being a predicted event record. Du and Cardie (2020a) show that longer text might hurt the model performance, and a multi-granularity reader is proposed to incorporate sentence-level and paragraph-level information. Huang and Peng (2020) propose to leverage Deep Value Networks (DVN) that captures cross-event dependencies to jointly resolving both the entity and event coreferences for DEE. Du et al. (2020) introduce an end-to-end generative transformer-based model to extract arguments across sentence boundaries.

5 Conclusion

In this paper, we propose a novel document-level event extraction framework that explores the sentence community for the event extraction task, which alleviates the multi-event issue and the role overlapping issue. In our framework, we introduce the document graph construction method that transforms a document into an undirected unweighted graph, which establishes associations between related sentences, and we employ the graph attention networks to capture the associations between sentences and further assign sentences to communities.

The experimental results validate the effectiveness of our proposed framework.

Acknowledgments

This work is supported by Guangdong Key Lab of AI and Multi-modal Data Processing, Chinese National Research Fund (NSFC) Project No. 61872239; BNU-UIC Institute of Artificial Intelligence and Future Networks funded by Beijing Normal University (Zhuhai) and AI-DS Research Hub, BNU-HKBU United International College (UIC), Zhuhai, Guangdong, China.

References

Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. 2020. GATE: graph attention transformer encoder for cross-lingual relation and event extraction. *CoRR*, abs/2010.03009.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and accurate deep network learning by exponential linear units (elus). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Shiyao Cui, Bowen Yu, Tingwen Liu, Zhenyu Zhang, Xuebin Wang, and Jinqiao Shi. 2020. Edge-enhanced graph convolution networks for event detection with syntactic relation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2329–2339, Online. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

George R. Doddington, Alexis Mitchell, Mark A. Przybocki, Lance A. Ramshaw, Stephanie M. Strassel, and Ralph M. Weischedel. 2004. The automatic content extraction (ace) program tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-2004)*.

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 2020, Online, July 5-10, 2020*, pages 8010–8020. Association for Computational Linguistics.

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 2020, Online, November 16-20, 2020*, pages 671–683. 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.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. A language-independent neural network for event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics.

Kung-Hsiang Huang and Nanyun Peng. 2020. Efficient end-to-end learning of cross-event dependencies for document-level event extraction. *CoRR*, abs/2010.12787.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020a. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5405–5411, Online. Association for Computational Linguistics.

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020b. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5405–5411. Association for Computational Linguistics.

Chang-Shing Lee, Yea-Juan Chen, and Zhi-Wei Jian. 2003. Ontology-based fuzzy event extraction agent for chinese e-news summarization. *Expert Syst. Appl.*, 25(3):431–447.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing narrative event evolutionary graph for script event prediction. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4201–4207. ijcai.org.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Vancouver, Canada. Association for Computational Linguistics.

Jie Ma, Shuai Wang, Rishita Anubhai, Miguel Ballesteros, and Yaser Al-Onaizan. 2020a. Resource-enhanced neural model for event argument extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 3554–3559. Association for Computational Linguistics.

Jie Ma, Shuai Wang, Rishita Anubhai, Miguel Ballesteros, and Yaser Al-Onaizan. 2020b. Resource-enhanced neural model for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3554–3559, Online. Association for Computational Linguistics.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 300–309. The Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Graph transformer networks with syntactic and semantic structures for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3651–3661, Online. Association for Computational Linguistics.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5916–5923. AAAI Press.

Oleksandr Shchur and Stephan Günnemann. 2019. Overlapping community detection with graph neural networks. *CoRR*, abs/1909.12201.

Rohini K. Srihari and Wei Li. 2000. A question answering system supported by information extraction. In *6th Applied Natural Language Processing Conference, ANLP 2000, Seattle, Washington, USA, April 29 - May 4, 2000*, pages 166–172. ACL.

Zheng Tang, Gus Hahn-Powell, and Mihai Surdeanu. 2020. Exploring interpretability in event extraction: Multitask learning of a neural event classifier and an explanation decoder. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, ACL 2020, Online, July 5-10, 2020*, pages 169–175. Association for Computational Linguistics.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Xiaozhi Wang, Ziqi Wang, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, Maosong Sun, Jie Zhou, and Xiang Ren. 2019. HMEAE: Hierarchical modular event argument 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)*, pages 5777–5783, Hong Kong, China. 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 (EMNLP-IJCNLP)*, 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 ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, pages 50–55. Association for Computational Linguistics.

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 Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5284–5294. Association for Computational Linguistics.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard H. Hovy. 2020. A two-step approach for implicit event argument detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7479–7485. Association for Computational Linguistics.

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 (Volume 2: Short Papers)*, pages 414–419, Melbourne, Australia. 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.

Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2018. Graph neural networks: A review of methods and applications. *CoRR*, abs/1812.08434.

A Additional Evaluation Results

We present the evaluation results of each event type with precision, recall, and F1 score in Table 7.

Model	EF			ER			EU			EO			EP		
	P.	**R.**	**F1**	**P.**	**R.**	**F1**	**P.**	**R.**	**F1**	**P.**	**R.**	**F1**	**P.**	**R.**	**F1**
DCFEE-O	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
GreedyDec	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
SCDEE	**88.4**	**73.7**	**80.4**	**93.7**	**87.5**	**90.5**	**84.3**	**67.8**	**75.1**	**85.5**	59.4	70.1	84.4	72.7	**78.1**

Table 7: Comprehensive results for each event type on the test set. Bold denotes the best result.