A Semantic Mention Graph Augmented Model for Document-Level Event Argument Extraction

Jian Zhang[1], Changlin Yang[1], Haiping Zhu[1,2]*, Qika Lin[3], Fangzhi Xu[1], Jun Liu[1,2]
[1]School of Computer Science and Technology, Xi'an Jiaotong University
[2]National Engineering Lab for Big Data Analytics, Xi'an, China
[3]National University of Singapore
{zhangjian062422,yangchanglin,Leo981106}@stu.xjtu.edu.cn,
{zhuhaiping,liukeen}@xjtu.edu.cn, linqika@nus.edu.sg

Abstract

Document-level Event Argument Extraction (DEAE) aims to identify arguments and their specific roles from an unstructured document. The advanced approaches on DEAE utilize prompt-based methods to guide pre-trained language models (PLMs) in extracting arguments from input documents. They mainly concentrate on establishing relations between triggers and entity mentions within documents, leaving two unresolved problems: a) independent modeling of entity mentions; b) document-prompt isolation. To this end, we propose a semantic mention Graph Augmented Model (GAM) to address these two problems in this paper. Firstly, GAM constructs a semantic mention graph that captures relations within and between documents and prompts, encompassing co-existence, co-reference and co-type relations. Furthermore, we introduce an ensembled graph transformer module to address mentions and their three semantic relations effectively. Later, the graph-augmented encoder-decoder module incorporates the relation-specific graph into the input embedding of PLMs and optimizes the encoder section with topology information, enhancing the relations comprehensively. Extensive experiments on the RAMS and WikiEvents datasets demonstrate the effectiveness of our approach, surpassing baseline methods and achieving a new state-of-the-art performance.

Keywords: document-level event argument extraction, semantic mention graph, ensembled graph transformer, graph-augmented PLMs

1. Introduction

Document-level Event Extraction (DEE) stands as an essential technology in the construction of event graphs (Xu et al., 2021) in the field of natural language processing (NLP) (Hirschberg and Manning, 2015; Hedderich et al., 2021; Bojun and Yuan, 2023). Within the realm of DEE, Document-level Event Argument Extraction (DEAE) plays a crucial role in transforming unstructured text into a structured event representation, thereby enabling support for various downstream tasks like recommendation systems (Roy and Dutta, 2022), dialogue systems (Ni et al., 2023) and some reasoning applications (Wang et al., 2023a). DEAE strives to extract all arguments from the entity mentions in a document and assign them specific roles with a given trigger word representing the event type. As depicted in Fig. 1, the trigger word is *set off* and the task is to extract arguments of the predefined argument roles of the event type *Conflict*, e.g., *attacker* and *explosiveDevice*. In recent researches, significant strides have been made in DEAE thanks to the success of pre-trained language models (PLMs) and the prompt-tuning paradigm. An unfilled prompt p is initialized by argument placeholders based on the event ontology(Li et al., 2021). For example, the prompt for *Conflict* type in Fig. 1 is "*Attacker* $\langle arg1 \rangle$ *exploded explosiveDevice* $\langle arg2 \rangle$ *using instrument* $\langle arg3 \rangle$ *to attack target* $\langle arg4 \rangle$ *at place* $\langle arg5 \rangle$". We define argument placeholders in the prompt as mask mentions, e.g., "*attacker* $\langle arg1 \rangle$". The advanced approaches on DEAE utilize prompt-based methods to guide PLMs in extracting arguments from input documents. These studies on DEAE (Lin et al., 2022a; Ma et al., 2022; Zeng et al., 2022) consider using different prompts to instruct PLMs, but there remains two unsolved problems: a) independent modeling of entity mentions; b) document-prompt isolation.

On one hand, the relevance among entity mentions within the document is crucial but frequently overlooked. These entity mentions share a clear and significant connection that demands careful consideration in DEAE. This relevance is universal and invaluable, enabling DEAE to grasp the completeness of events and the correlation structure within documents. Taking co-reference relations as an example, arguments appear in various forms across different sentences within the document, creating co-reference instances. As shown in Fig. 1, the entity mention *Aaron Driver* appears multiple times in various forms of expressions (in green), such as *a Canadian man* and *Harun Abdurahman*, conveying an identical semantic meaning. The same phenomenon also exists in the co-existence relation among entity mentions, wherein

* Corresponding author.

LREC-COLING 2024, pages 1577–1587

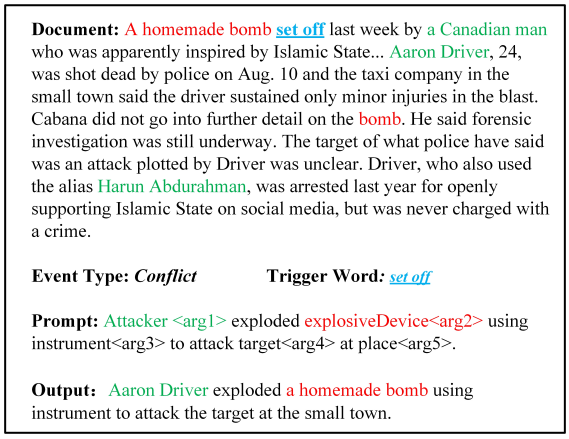

Figure 1: An illustration of DEAE including the relevance among entity mentions with the same color labeled in the document and the co-type relation between the prompt and the document with the same color labeled.

co-existence relations denote the presence of entity mentions or masked mentions within the same sentence. Surprisingly, previous studies (Du and Cardie, 2020; Liu et al., 2021; Wei et al., 2021) often overlook this aspect, obscuring this vital correlation.

On the other hand, the document-prompt isolation is both valuable and underappreciated. Generally, arguments should only be extracted from entity mentions of the same type in the appropriate context. The co-type relation between documents and prompts provides essential guidance for accurately determining the positions of arguments. In other words, the co-type relation refers to the same type attributes between masked mentions and entity mentions. As demonstrated in Fig. 1, the argument role *attacker* in the prompt and the entity mention *Aaron Driver* are of the same type, namely, *PERSON*, indicating a co-type phenomenon. Previous studies (Lin et al., 2022a; Ma et al., 2022; Zeng et al., 2022) ignore the document-prompt isolation problem, neglecting the co-type relation between documents and prompts when directly feeding them into PLMs.

To this end, we propose a semantic mention Graph Augmented Model (GAM) to alleviate the above two problems in this paper. Within the semantic mention graph, the semantics highlights the internal meaning of mentions and models this through the relations between mentions. To address the independent modeling of entity mentions, GAM considers the co-existence and co-reference relations among entity mentions. For the document-prompt isolation problem, the co-type relation between mask mentions and entity mentions are incorporated. Specifically, we first construct a semantic mention graph module to model these three semantic relations. It includes nodes representing entity mentions and mask mentions, connected by the aforementioned relations. For instance, nodes like *Aaron Driver* and *Harun Abdurahman* are connected by an edge labeled *co-reference*. Then, the three types of relations are depicted in three adjacent matrices, which are aggregated into a fused attention bias. The node sequence and fused attention bias are fed into the ensembled graph transformer for encoding. Lastly, we integrate node embeddings into initial embeddings as input and employ the fused topology information as attention bias to boost the PLMs. The main contributions of our work are as follows:

- This research introduces a universal framework GAM[1], in which we construct a semantic mention graph incorporating three types of relations within and between the documents and the prompts initially. It is the first work in simultaneously addressing the independent modeling of entity mentions and document-prompt isolation as far as we know.
- We propose an ensembled graph transformer module and a graph-augmented encoder-decoder module to handle the three types of relations. The former is utilized to handle the mentions and their three semantic relations, while the latter integrates the relation-specific graph into the input embedding and optimizes the encoder section with topology information to enhance the performance of PLMs.
- Extensive experiments report that GAM achieves the new state-of-the-art performance on two benchmarks and further analysis validates the effectiveness of the different relations in semantic mention graph construction module, ensembled graph transformer module and graph-augmented encoder-decoder module in our model.

2. Related Works

In this section, we introduce the current researches on DEAE, mainly consisting the sequence model and graph model for event extraction.

2.1. DEAE Based on Sequence Model

From the early stages, semantic role labeling (SRL) has been utilized for extracting event arguments in various studies (Yang et al., 2018; Zheng et al., 2019; Xu et al., 2021; Wang et al., 2023b). Some studies initially identify entities within the document and subsequently assign these entities specific argument roles. Lin et al. (2020) began by identifying

[1]The code for the framework and the experimental data are stored in the repository: https://github.com/exoskeletonzj/gam.

candidate entity mentions, followed by their assignment of specific roles through multi-label classification.

Later, certain studies have approached DEAE as a question-answering (QA) task. Methods (Du and Cardie, 2020; Liu et al., 2021) based on QA involve querying arguments by answering questions predefined through templates one by one, treating DEAE as a machine reading comprehension task. Wei et al. (2021) toke into account the implicit interactions among roles by imposing constraints on each other within the template. However, this method tends to lead to error accumulation.

Alongside the emergence of sequence-to-sequence models, specifically generative PLMs like BART (Lewis et al., 2020) and T5 (Raffel et al., 2020), generating all arguments in the sequence of target event has become possible. Some studies (Li et al., 2021; Du et al., 2021; Lu et al., 2021) employ sequence-to-sequence models to extract arguments efficiently. Furthermore, accompanied by sequence-to-sequence models, prompt-tuning methods have also emerged. Recent works on DEAE (Lin et al., 2022a; Ma et al., 2022; Zeng et al., 2022) explore the utilization of various prompts to guide PLMs in extracting arguments.

Up to now, these studies have proposed some solutions to DEAE tasks at different levels, but they rarely consider the entity mentions' relevance directly. Under the latest paradigm of prompt-tuning with generative PLMs, they have not considered the explicit interaction between prompts and documents.

2.2. DEAE Based on Graph Model

Graph model is a crucial kind of methods in information extraction, particularly in recent years, where it evaluates documents by constructing various graphs on DEAE tasks. Zheng et al. (2019) first introduced an entity directed acyclic graph to efficiently address DEE. Xu et al. (2021) implemented cross-entity and cross-sentence information exchange by constructing heterogeneous graphs. Xu et al. (2022b) constructed abstract meaning representation (Banarescu et al., 2013) semantic graphs to manage long distance dependencies between trigger and arguments across sentences.

However, these methods based on graph model simply transforms the document into graph structures and then utilize a classification model to assign specific roles to entity mentions. This paradigm make no use of PLMs and is inefficient in extracting all arguments for a given event simultaneously.

Limiting the consideration to just the sequence model or solely the graph model is incomplete. Our research motivation lies in the organic fusion of these two approaches, enabling our method to harness the strengths of both the latest sequence model and graph model.

3. Methodology

This section begins by introducing the task formulation. We formulate DEAE task as a prompt-based span extraction problem. Given an input instance $\left(X, t, e, R^{(e)}\right)$, where $X = \{x_1, x_2, ..., x_n\}$ denotes the document, $t \subseteq X$ denotes the trigger word, e denotes the event type and $R^{(e)}$ denotes the set of event-specific role types, we aim to extract a set of spans A as the output. Each $a^{(r)} \in A$ is a segmentation of X and represents an argument corresponding to $r \in R^{(e)}$.

GAM leverages the relations among entity mentions and mask mentions to enhance PLMs for event argument extraction. Our model, depicted in Figure 2, comprises three key components: a) semantic mention graph construction from the context, consisting of co-existence, co-reference and co-type relations; b) ensembled graph transformer module for handling the dependencies and interactions in the graph; c) graph-augmented encoder-decoder module with PLMs for argument generation. Subsequent sections will outline our task formulation and elaborate on each component in detail.

3.1. Semantic Mention Graph Construction

One crucial problem in extracting arguments from the document is mitigating the relevance among entity mentions, as well as the relevance between entity mentions and mask mentions, by capturing co-existence, co-reference and co-type information. Therefore, we introduce a graph construction module that adopts the semantic mention graph to provide a robust semantic structure. This approach facilitates interactions among entity mentions and mask mentions, offering logical meanings of the document from a linguistically-driven perspective to enhance language understanding.

Primarily, as demonstrated in the section 1, GAM generates an unfilled prompt p with argument placeholders. GAM initially concatenates the document X with corresponding prompt p respectively to form the input sequences. In DEAE tasks, all extracted arguments should originate from entity mentions in the document. In the prompt-tuning paradigm, the extracted arguments are ultimately filled with placeholders, represented by mask mentions in the prompt. Consequently, we treat all entity mentions mask mentions as nodes in the semantic mention graph. In this module, GAM constructs the semantic mention graph from three perspec-

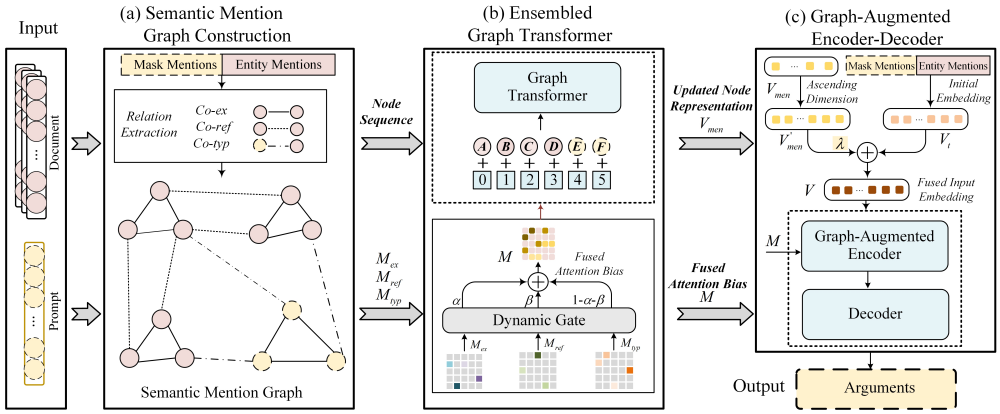

Figure 2: The architecture of GAM. The left part is an input example of the document and a corresponding prompt. The graph construction module (*a*) constructs a semantic mention graph including co-existence, co-reference and co-type relations from entity mentions and mask mentions. The ensembled graph transformer module (*b*) handles the text features combined with three semantic relations. Finally, the graph-augmented encoder-decoder module (*c*) is utilized to conduct the feature fusion and predict the arguments.

tives by extracting three types of relations, including co-existence relation within entity mentions and mask mentions, co-reference relation between entity mentions and the co-type relation between mask mentions and entity mentions.

3.1.1. Co-existence Relation

In the co-existence relation, GAM focuses on mentions within the same sentence. Intuitively, entity mentions in the same sentence represent all specific information and they are more likely to become arguments for the same event. Mask mentions also represent the same event. The aggregation of the co-existence relation within the mask mentions enables the subsequent sub-modules to better understand which argument roles are present in the current event, thus better reflecting the complete event ontology information in the graph. Therefore, we construct the co-existence relation to enhance the same sentence connection.

If nodes m_i and m_j are in the same sentence, we establish a direct connection between mentions m_i and m_j. These connections confined within a single sentence in the document or prompt. This relation is reflected in the adjacent matrix $\mathbf{M}_{ex} \in \mathbb{R}^{K \times K}$ of the co-existence relation, where $\mathbf{M}_{ex}[m_i, m_j] = 1$, where K is the total number of the nodes, i.e., the sum of the entity mentions and mask mentions.

Consider Fig. 1 for example, in the same sentence, entity mentions *a homemade bomb* and *Aaron Driver* have an edge connecting them. Similarly, mask mentions *attacker* $\langle arg1 \rangle$ and *explosiveDevice* $\langle arg2 \rangle$ also share a direct connection within the same sentence.

3.1.2. Co-reference Relation

The co-reference relation aims to make better use of co-reference information between entity mentions. As introduced in the section 1, it is evident that the co-reference commonly exists in the entire document, showcasing a significant characteristic of co-reference. Hence, we focus on constructing a co-reference relation that captures co-reference relations among entity mentions throughout the entire document.

While, the number of the nodes K is the same as the count of mentions with the co-existence relation. Following the extraction of co-reference relation using the tool *fastcoref* (Otmazgin et al., 2022), we establish direct connection between co-reference entity mentions m_k and m_l. Notably, these connections can occur within sentences or across sentences in the document. Such linkage is represented in the adjacent matrix $\mathbf{M}_{ref} \in \mathbb{R}^{K \times K}$ of the co-reference relation as $\mathbf{M}_{ref}[m_k, m_l] = 1$.

Note that in Fig. 1, the co-reference entity mentions *Aaron Driver*, *a Canadian man*, *Harun Abdurahman* and *the driver*. There is a direct connection between each of them respectively.

3.1.3. Co-type Relation

The co-type relation comprises entity mentions and mask mentions, detailing the relation between the two. Unlike previous methods, we consider the explicit connection between entity mentions and mask mentions in our approach.

A fundamental and logical assumption is that each mask mention should be filled with the same type of entity mentions. In other words, each mask mention should be associated with the same type of entity mentions. Consequently, we compose

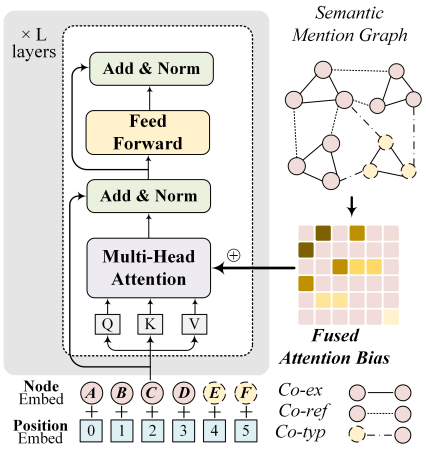

Figure 3: The illustration of graph transformer. The inputs are the node sequence as well as the node position and the outputs are omitted. The Co-ex, Co-ref and Co-typ semantic mention relations are fused as a attention bias.

the third relation to establish co-type connections between mask mentions and entity mentions.

For consistency, the number of the nodes, denoted as K, aligns with the count in the previous two relations. Directed connections can be established between mask mention m_s and entity mention m_t of the same type. These connections link entity mentions in the document to mask mentions in the prompt. These relations are represented in the adjacent matrix $\mathbf{M}_{typ} \in \mathbb{R}^{K\times K}$ of the co-type relation, where $\mathbf{M}_{typ}[m_s, m_t] = 1$.

As depicted in Fig. 1, the mask mentions *attacker* $\langle arg1 \rangle$ and entity mention *Aaron Driver* share the same type, GAM establishes a connection between the two.

3.2. Ensembled Graph Transformer

Several studies (Zhang et al., 2020; Dwivedi and Bresson, 2020) have highlighted drawbacks in graph neural network, including the problem of over-smoothing (Li et al., 2018). Consequently, we have incorporated the individual approach of graph transformer (Ying et al., 2021; Cai and Lam, 2020). Following the extraction of the three types of relations, we utilize ensembled graph transformer structures (Xu et al., 2022a) to handle them collectively.

The merged graph transformer is visually represented in Fig. 3 for a concise overview. First of all, We define text markers as $\langle \mathbf{tgr} \rangle$ / $\langle /\mathbf{tgr} \rangle$ and insert them into the document X before and after the trigger word, respectively. It is essential to obtain the original feature embedding for each node. Given the concatenated sequence of the i^{th} document:

$$\tilde{x_i} = [x_1, x_2, ..., \langle \mathbf{tgr} \rangle, x_{tgr}, \langle /\mathbf{tgr} \rangle, ..., x_n], \quad (1)$$

where x_j represents the j^{th} token in the document, and tgr denotes the index of the trigger word. The document $\tilde{x_i}$ is then encapsulated, together with a prompt template p, using a function denoted as $\lambda(\cdot, \cdot)$:

$$X_p = \lambda(p, \tilde{x_i}) = [CLS]p[SEP]\tilde{x_i}[SEP], \quad (2)$$

where $[CLS]$ and $[SEP]$ serve as separators in BART, X_p denotes the concatenated input sequence of prompt p and the document $\tilde{x_i}$.

We utilize the BART model as the encoder for obtaining the token-level representation $\mathbf{V}_t \in \mathbb{R}^{N\times d}$ of X_p, where N is the token numbers of the input sequence and d is the dimension of the hidden state. Subsequently, we extract the order of entity mentions and mask mentions from $\mathbf{V}_t$. To obtain the embedding of node m_k with the length L, we average the token embedding constituting the node:

$$\mathbf{v}_k = \frac{1}{L}\sum_{i=1}^{L} \mathbf{v}_i^{(k)}. \quad (3)$$

We integrate positional embedding and node embedding to maintain the consistency of node order within the document:

$$\mathbf{V_i} = \mathbf{V_{token}} + Position(\mathbf{V_{token}}), \quad (4)$$

where $\mathbf{V_{token}} = [\mathbf{v}_1; \mathbf{v}_2; ...; \mathbf{v}_K]$ and $\mathbf{V_{token}} \in \mathbb{R}^{K\times d}$. The function $Position(\cdot)$ generates a d-dimensional embedding for each node within the input sequence.

This module revolves around multi-head attention mechanism. Firstly, to incorporate graph information into the transformer architecture, we first obtain the fused topology information M. Considering the attention bias $\mathbf{M}_{ex}$, $\mathbf{M}_{ref}$ *and* $\mathbf{M}_{typ} \in \mathbb{R}^{K\times K}$, these three biases, although having the same dimension, may not contribute equally to the final prediction. To aggregate them effectively, GAM assigns proper hyper-parameters to balance their influence. The representation of M is as follows:

$$\mathbf{M} = \alpha\mathbf{M}_{ex} + \beta\mathbf{M}_{ref} + (1-\alpha-\beta)\mathbf{M}_{typ}. \quad (5)$$

Hence, GAM employs the obtained matrix M as attention bias to adjust the self attention formula:

$$Att(Q, K, V)' = \text{softmax}(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}} + \mathbf{M}) \cdot V. \quad (6)$$

where matrices $Q, K, V \in \mathbb{R}^{K\times d_k}$ is the projection of $\mathbf{V_i}$ by projection matrices $\mathrm{W}^{\mathrm{Q}}, \mathrm{W}^{\mathrm{K}}, \mathrm{W}^{\mathrm{V}} \in \mathbb{R}^{d\times d_k}$.

To learn diverse feature representations and improve the adaptability of the graph, we implement multi-head attention mechanism with a specified number of heads, denoted as MH:

$$MH(Q, K, V) = [Head_1; ...; Head_H] \cdot \mathrm{W}^{\mathrm{O}}, \quad (7)$$

DataSet	Split	Doc	Event	Argument
RAMS	Train	3,194	7,394	17,026
	Dev	399	924	2,188
	Test	400	871	2,023
WikiEvents	Train	206	3,241	4,542
	Dev	20	345	428
	Test	20	365	556

Table 1: Data statistics of RAMS and WikiEvents.

where $\mathrm{W}^{\mathrm{O}} \in \mathbb{R}^{(H*d_k)\times d_k}$ is the linear projection matrix, $Head_i = Att_i(Q,K,V)'$.

To better capture the diversity and complexity of the attention module, we fuse the last two hidden layers as the updated node features:

$$\mathbf{V}_{men} = 0.5 \cdot \mathbf{V}^{(L-1)} + 0.5 \cdot \mathbf{V}^{(L)}, \tag{8}$$

where $\mathbf{V}_{men} \in \mathbb{R}^{K\times d}$, and $\mathbf{V}^{(L-1)}$, $\mathbf{V}^{(L)} \in \mathbb{R}^{K\times d}$ denote the hidden states of the last two layers.

3.3. Graph-Augmented Encoder-Decoder Model

As the previous methods on DEAE (Lin et al., 2022a; Ma et al., 2022; Zeng et al., 2022) adopt, we choose and expand pre-trained language model BART as our encoder-decoder model.

We have obtained the token-level representation $\mathbf{V}_t$ and the updated mention node representation $\mathbf{V}_{men}$. To maintain dimension consistency, we broadcast the feature of each node to all the tokens it encompasses. The transformed features are denoted as $\mathbf{V}_t$, $\mathbf{V}'_{men} \in \mathbb{R}^{N\times d}$.

To enhance the ability to perceive semantic mentions, we integrate node embedding into the initial embedding. GAM then configures a proper weight to balance these two features. The resulting fused input embedding $\mathbf{V}$ is as follows:

$$\mathbf{V} = LN(\mathbf{V}_t + \lambda \cdot \mathbf{V}'_{men}), \tag{9}$$

where $LN(\cdot)$ denotes the layer normalization operation. Then $\mathbf{V}$ as the input embedding is fed into BART.

To further enhance the effectiveness, GAM incorporates a graph-augmented encoder section of BART. GAM employs the fused topology information $\mathbf{M}$ as attention bias, similar to the graph transformer module. The representation of the self attention formula is adjusted as Eq. 6.

For each instance, the graph-augmented BART module can be employed to generate a completed template, replacing the placeholder tokens with the extracted arguments. The model parameter θ is trained by minimizing the argument extraction loss, which is the conditional probability computed over all instances:

$$\mathcal{L} = -\sum \log_{p_\theta}(y|X,t,p). \tag{10}$$

4. Experiments

4.1. Datasets and baselines

We conduct comprehensive experiments on two widely recognized DEAE benchmark datasets: RAMS (Ebner et al., 2020) and WikiEvents (Li et al., 2021), which have been extensively utilized in previous studies (Lin et al., 2022a; Ma et al., 2022; Zeng et al., 2022). As shown in table 1, the RAMS dataset comprises 3,993 paragraphs, annotated with 139 event types and 65 argument roles. The WikiEvents dataset consists of 246 documents, annotated with 50 event types and 59 argument roles.

We deem an argument span as correctly identified when its offsets align with any of the reference arguments of the current event (i.e., **Argument Identification**), and as correctly classified when its role matches (i.e., **Argument Classification**). Furthermore, we evaluate the argument extraction performance using Head Match F1 and Coref Match F1 metrics on the WikiEvents dataset, where Head Match indicates alignment with the head of the span, and Coref Match indicates an exact match of the span with all co-reference spans. In the case of the latter, full credit is assigned when the extracted argument is coreferential with the gold-standard argument.

We compare GAM with several state-of-the-art models in two categories: (1) **FEAE** (Wei et al., 2021), **EEQA** (Du and Cardie, 2020), **BART-Gen** (Li et al., 2021), **PAIE** (Ma et al., 2022) on RAMS dataset; (2) **BERT-CRF** (Shi and Lin, 2019), **ONEIE** (Lin et al., 2020), **BART-Gen** (Li et al., 2021), $\mathbf{EA^2E}$ (Zeng et al., 2022) on WikiEvents dataset. Among them, **BERT-CRF** is a semantic role labeling method, **ONEIE** is a graph-based method, **FEAE** and **EEQA** utilize QA patterns, whereas **BART-Gen**, **PAIE**, and $\mathbf{EA^2E}$ employ different prompts directly.

4.2. Implementation Details

GAM extends upon the BART-style encoder-decoder transformer structure. Each model, including baselines and GAM, is trained for 4 epochs with a batch size of 4, utilizing NVIDIA-V100 with 32GB DRAM. The model is optimized using the Adam optimizer with a learning rate of 3e-5, $\alpha = 0.3$, $\beta = 0.4$ and $\lambda = 0.015$. These hyper-parameters are meticulously selected through grid search, based on model's performance on the development set.[2]

[2]The learning rate is chosen in {3e-5, 5e-5}, α and β is chosen from {0.1, 0.2, 0.3, 0.4, 0.6, 0.7, 0.8}, and λ is chosen from {0.01, 0.015, 0.02, 0.03, 0.04, 0.05}.

Model	Argument Identification						Argument Classification					
	Head Match			Coref Match			Head Match			Coref Match		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BERT-CRF	72.66	53.82	61.84	74.58	55.24	63.47	61.87	45.83	52.65	63.79	47.25	54.29
ONEIE	68.16	56.66	61.88	70.09	58.26	63.63	63.46	52.75	57.61	65.17	54.17	59.17
BART-Gen	70.43	71.94	71.18	71.83	73.36	72.58	65.39	66.79	66.08	66.78	68.21	67.49
$\mathbf{EA^2E}$	76.51	72.82	74.62	77.69	73.95	75.77	70.35	66.96	68.61	71.47	68.03	69.7
GAM	**79.05**	**72.97**	**75.89**	**80.36**	**74.08**	**77.09**	**73.47**	**67.07**	**70.12**	**74.59**	**68.96**	**71.66**
GAM w/o co-ex	78.34	71.66	74.85	80.28	73.09	76.52	72.86	66.80	69.70	73.69	67.29	70.34
GAM w/o co-ref	75.63	70.24	72.84	76.07	70.72	73.30	69.85	64.05	66.82	70.53	64.74	67.51
GAM w/o co-typ	76.44	70.95	73.59	78.62	72.34	75.35	71.96	67.16	69.48	72.81	66.46	69.49
GAM w/o G.T.	78.46	70.52	74.28	79.45	71.4	75.21	71.34	64.12	67.54	72.33	65.01	68.48
GAM w/o N.E.	77.08	72.29	74.61	78.03	73.18	75.53	70.64	66.25	68.38	71.59	67.14	69.29
GAM w/o bias	76.85	70.16	73.35	77.82	71.05	74.28	70.23	64.12	67.04	71.21	65.01	67.97

Table 2: Overall performance on WikiEvents dataset. In the results, the best-performing model is highlighted, and the second best is underlined. **G.T.**: graph transformer module. **N.E.**: node embedding module. **bias**: attention bias for graph transformer and BART encoder module.

Model	Argument Identification	Argument Classification
FEAE	53.5	47.4
EEQA	48.7	46.7
BART-Gen	51.2	47.1
EEQA-BART	51.7	48.7
PAIE	55.6	53.0
GAM	**56.83**	**54.20**
GAM w/o co-ex	54.52	52.19
GAM w/o co-ref	52.86	50.65
GAM w/o co-typ	53.16	51.82
GAM w/o G.T.	54.24	53.02
GAM w/o N.E.	53.64	51.17
GAM w/o bias	53.94	52.45

Table 3: Overall performance on RAMS dataset.

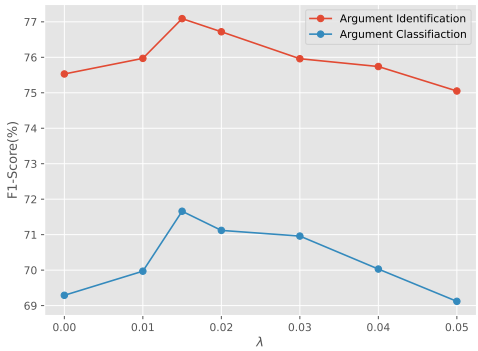

Figure 4: The illustration of ablation study on WikiEvents dataset. The model performances under different λ.

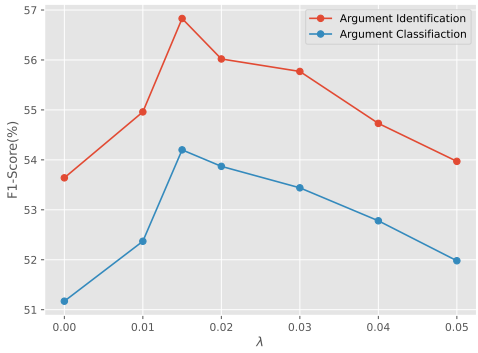

Figure 5: The different performance on RAMS dataset under different λ.

4.3. Comparison Results

Tables 2 and 3 demonstrate the superior performance of our proposed GAM compared to strong baseline methods across various datasets and evaluation metrics. Specifically, on the WikiEvents dataset, our model achieves a notable 1.32% improvement in absolute argument identification F1 and a 1.96% improvement in argument classification F1. Similarly, on the RAMS dataset, GAM exhibits the improvements with a 1.23% increase in argument identification and 1.20% in argument classification F1 scores. These results underscore the outstanding performance of our proposed method.

Furthermore, our graph-augmented encoder-decoder model outperforms graph-based methods and directly prompt-tuning encoder-decoder methods, including ONEIE and BART-Gen. From the experimental results shown in Tables 2 and 3, we can conclude that: (1) Compared to graph-based methods, GAM can utilize rich information from the graph to enhance the initial embedding and the encoder part of encoder-decoder model. (2) Compared to directly prompt-tuning encoder-decoder methods, These results emphasize the effectiveness of our semantic mention graphs in leveraging the BART architecture, enhancing semantic interactions within documents, and bridging the gap between documents and prompts.

4.4. Ablation Studies

In this section, we assess the effectiveness of our primary components by systematically removing each module one at a time. The components are as following: (1) three types of relations. Here, we analyze the gain brought by these relations by con-

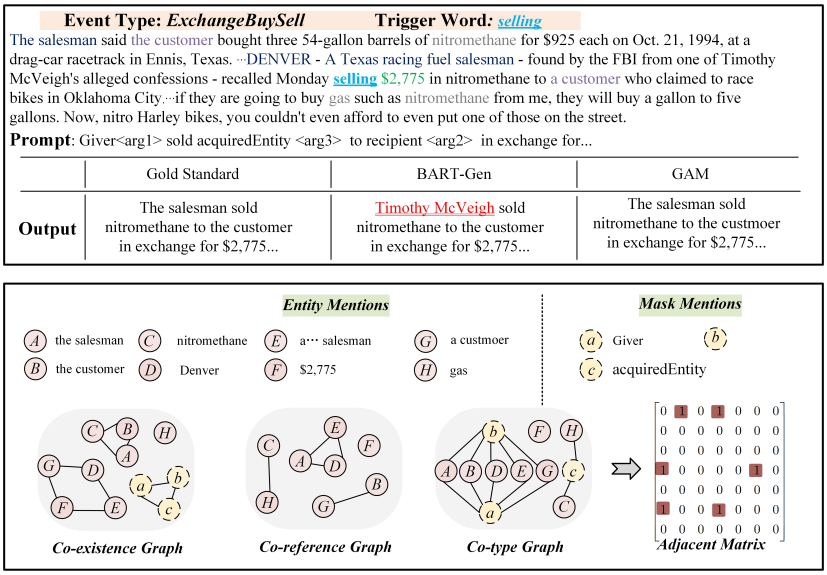

Figure 6: The illustration of an DEAE case. This case mainly showcases the construction of semantic mention graph and compares the extraction results between BART-Gen and GAM.

sidering the removal of one of the three; (2) graph transformer. We exclude the graph transformer, thereby disregarding the updating of node representation in the semantic mention graph; (3) node embedding. We eliminate the node embedding component from the input of the BART encoder-decoder module, retaining only the initial embedding; (4) attention bias. We withhold the attention bias from both the graph transformer and the BART encoder module.

The results of ablation studies are summarized in Table 2 and Table 3. We can observe that all of the three types of relations, graph transformer, node embedding and attention bias modules can help boost the performance of DEAE.

Regarding the module of the semantic mention graph construction, we remove one of the three relations at a time to observe the decrease it causes. According to the ablation results, the co-reference relation contributes the most among the three types of relations, followed by co-type relation and co-existence relation. It is evident that the co-reference relation significantly reduces ambiguity, enhances the accuracy of DEAE, and provides a more comprehensive, consistent, and precise semantic representation. The decrease brought by the co-type relation follows closely behind because, ideally, the correctly extracted arguments should all come from corresponding co-type entity mentions in the original document.

Moreover, the absence of the graph transformer module leads to a obvious drop in performance, with F1 score decreasing by more than 2 points on both RAMS and WikiEvents datasets. This clearly emphasizes the crucial role of the graph transformer in updating nodes. Similar patterns are observed in other modules, underscoring the effectiveness of each component in enhancing argument extraction. We are pleasantly surprised to discover that withholding attention bias from both the graph transformer and the BART encoder module resulted in the largest decrease, excluding the semantic mention graph construction module. This is because, in the transformer architecture, the attention mechanism tends to allocate more attention to the emphasized parts.

In particular, Dropping out all of the above modules—essentially eliminating all components related to the graph—results in the variant model regressing to BART-Gen. BART-Gen is a standard model that relies solely on prompts and PLMs. Upon reviewing the results in Table 2 and Table 3, GAM outperforms BART-Gen by 4.17% on the WikiEvents dataset and 5.5% on the RAMS dataset. This comparison strongly emphasizes the significant performance enhancement achieved by the graph-enhanced model over BART.

4.5. Supplementary Analysis

Throughout the experiments, hyper-parameters are employed in many places. Due to space constraints, we focus on analyzing one specific parameter—namely, the node embedding weight λ used to consolidate the initial embedding fed into BART. As shown in Eq. 9, $\mathbf{V}_t$ refers to the initial embedding of the document, while $\mathbf{V}'_{men}$ refers to the updated node embedding, reflecting the GAM model's modeling of nodes in the semantic mention graph, addressing the independent modeling of entity mentions and document-prompt isolation. λ is the balanced weight of $\mathbf{V}'_{men}$, enhancing the input of the graph-augmented encoder-decoder module. The results corresponding to different values of λ

are presented in Fig. 4 and Fig. 5.

The results demonstrate that the optimal performance is achieved when λ is set to 0.015. A decrease in the hyper-parameter λ implies less consideration of node features and underutilization of semantic information. Conversely, as λ increases, additional semantic information is incorporated into the initial embedding. However, this might be detrimental to subsequent decoder stages because the encoder-decoder architecture heavily depends on the transmission of initial embedding within this context.

5. Case Study

Fig. 6 presents a representative example from the WikiEvents dataset, illustrating the process of graph-augmented DEAE. Initially, the graph construction module comprises nodes representing all entity mentions and mask mentions, along with edges depicting three semantic mention relations. In this instance, nodes are represented as circles in green and gray. GAM generates the semantic mention graph based on these relations. The connections efficiently capture the co-existence, co-reference and co-type information within and between the document and the prompt, highlighting GAM's interpretability capability.

Finally, GAM accurately extracts arguments corresponding to their respective roles using an unfilled prompt p. As depicted in Fig. 6, the output of BART-Gen differs from that of GAM. When compared to the gold standard, BART-Gen incorrectly identifies the argument role *Giver* due to its failure in considering the three types of relations within and between the document and the prompt. Conversely, GAM accurately aligns with the gold standard.

While effective, GAM can inadvertently propagate errors during graph construction. Furthermore, a scenario might arise where an argument role lacks a corresponding argument in the document. In such cases, the co-type relation may still assign edges of the same type of entity mention to these mask mention nodes.

6. Conclusion

We propose an end-to-end framework named semantic mention Graph Augmented Model to address the independent modeling of entity mentions and the document-prompt isolation problems. Firstly, GAM constructs a semantic mention graph by creating three types of relations: co-existence, co-reference and co-type relations within and between mask mentions and entity mentions. Secondly, The ensembled graph transformer module is utilized to handle the mentions and their three semantic relations. Lastly, the graph-augmented encoder-decoder module integrates the relation-specific graph into the input embedding and optimize the encoder section with topology information to enhance the performance of PLMs. Extensive experiments report that GAM achieves the new state-of-the-art performance on two benchmarks.

In the future, we plan to delve into DEAE within the framework of Large Language Models (LLM) (Xu et al., 2023). Due to the ambiguity (Liu et al., 2023) and polysemy (Laba et al., 2023) inherent in entity mentions within documents, LLM faces limitations in DEAE. We aim to leverage the semantic mention graph to provide guidance to LLM in DEAE. Furthermore, we will strive to integrate prior knowledge and employ logical reasoning (Lin et al., 2022b, 2023) to enhance event extraction with greater precision and interpretability.

Acknowledgement

This work was supported by National Key Research and Development Program of China (2022YFC3303600), National Natural Science Foundation of China (62137002, 62293553, 62176207, 62192781, 62277042 and 62250009), "LENOVO-XJTU" Intelligent Industry Joint Laboratory Project, Natural Science Basic Research Program of Shaanxi (2023-JC-YB-593), the Youth Innovation Team of Shaanxi Universities, XJTU Teaching Reform Research Project "Acquisition Learning Based on Knowledge Forest", Shaanxi Undergraduate and Higher Education Teaching Reform Research Program(Program No.23BY195).

Bibliographical 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.

Huang Bojun and Fei Yuan. 2023. Utility-probability duality of neural networks. *arXiv preprint arXiv:2305.14859*.

Deng Cai and Wai Lam. 2020. Graph transformer for graph-to-sequence learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7464–7471.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 671–683.

Xinya Du, Alexander M Rush, and Claire Cardie. 2021. Template filling with generative transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 909–914.

Vijay Prakash Dwivedi and Xavier Bresson. 2020. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.

Julia Hirschberg and Christopher D Manning. 2015. Advances in natural language processing. *Science*, 349(6245):261–266.

Yurii Laba, Volodymyr Mudryi, Dmytro Chaplynskyi, Mariana Romanyshyn, and Oles Dobosevych. 2023. Contextual embeddings for ukrainian: A large language model approach to word sense disambiguation. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 11–19.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI conference on artificial intelligence*.

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,{NAACL-HLT} 2021*, volume 2021.

Jiaju Lin, Qin Chen, Jie Zhou, Jian Jin, and Liang He. 2022a. CUP: curriculum learning based prompt tuning for implicit event argument extraction. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 4245–4251. ijcai.org.

Qika Lin, Jun Liu, Rui Mao, Fangzhi Xu, and Erik Cambria. 2023. TECHS: temporal logical graph networks for explainable extrapolation reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1281–1293.

Qika Lin, Jun Liu, Fangzhi Xu, Yudai Pan, Yifan Zhu, Lingling Zhang, and Tianzhe Zhao. 2022b. Incorporating context graph with logical reasoning for inductive relation prediction. In *The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 893–903. ACM.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7999–8009.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2023. We're afraid language models aren't modeling ambiguity. *arXiv preprint arXiv:2304.14399*.

Jian Liu, Yufeng Chen, and Jinan Xu. 2021. Machine reading comprehension as data augmentation: A case study on implicit event argument extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2725.

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.

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. Association for Computational Linguistics.

Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, and Erik Cambria. 2023. Recent advances in deep learning based dialogue systems: A systematic survey. *Artificial intelligence review*, 56(4):3055–3155.

Shon Otmazgin, Arie Cattan, and Yoav Goldberg. 2022. F-coref: Fast, accurate and easy to use coreference resolution. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 48–56.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Deepjyoti Roy and Mala Dutta. 2022. A systematic review and research perspective on recommender systems. *Journal of Big Data*, 9(1):59.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Jianing Wang, Nuo Chen, Qiushi Sun, Wenkang Huang, Chengyu Wang, and Ming Gao. 2023a. Hugnlp: A unified and comprehensive library for natural language processing. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 5111–5116, New York, NY, USA. Association for Computing Machinery.

Jianing Wang, Qiushi Sun, Xiang Li, and Ming Gao. 2023b. Boosting language models reasoning with chain-of-knowledge prompting.

Kaiwen Wei, Xian Sun, Zequn Zhang, Jingyuan Zhang, Guo Zhi, and Li Jin. 2021. Trigger is not sufficient: Exploiting frame-aware knowledge for implicit event argument 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 4672–4682.

Fangzhi Xu, Jun Liu, Qika Lin, Yudai Pan, and Lingling Zhang. 2022a. Logiformer: a two-branch graph transformer network for interpretable logical reasoning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1055–1065.

Fangzhi Xu, Zhiyong Wu, Qiushi Sun, Siyu Ren, Fei Yuan, Shuai Yuan, Qika Lin, Yu Qiao, and Jun Liu. 2023. Symbol-llm: Towards foundational symbol-centric interface for large language models. *arXiv preprint arXiv:2311.09278*.

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.

Runxin Xu, Peiyi Wang, Tianyu Liu, Shuang Zeng, Baobao Chang, and Zhifang Sui. 2022b. 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.

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, System Demonstrations*, pages 50–55.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform bad for graph representation? *arXiv preprint arXiv:2106.05234*.

Qi Zeng, Qiusi Zhan, and Heng Ji. 2022. Ea2e: Improving consistency with event awareness for document-level argument extraction. In *2022 Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2649–2655. Association for Computational Linguistics (ACL).

Jiawei Zhang, Haopeng Zhang, Congying Xia, and Li Sun. 2020. Graph-bert: Only attention is needed for learning graph representations. *arXiv preprint arXiv:2001.05140*.

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)*, pages 337–346.