Generation-Augmented and Embedding Fusion in Document-Level Event Argument Extraction

Xingjian Lin[1], Shengfei Lyu[2*], Xin Wang[1], Qiuju Chen[1], Huanhuan Chen[1*]
[1]University of Science and Technology of China
[2]Nanyang Technological University
linxingjian@mail.ustc.edu.cn, shengfei.lyu@ntu.edu.sg, wz520@mail.ustc.edu.cn, {qqchern, hchen}@ustc.edu.cn

Abstract

Document-level event argument extraction is a crucial task that aims to extract arguments from the entire document, beyond sentence-level analysis. Prior classification-based models still fail to explicitly capture significant relationships and heavily relies on large-scale datasets. In this study, we propose a novel approach called Generation-Augmented and Embedding Fusion. This approach first uses predefined templates and generative language models to produce an embedding capturing role relationship information, then integrates it into the foundational embedding derived from a classification model through a noval embedding fusion mechanism. We conduct the extensive experiments on the RAMS and WikiEvents datasets to demonstrate that our approach is more effective than the baselines, and that it is also data-efficient in low-resource scenarios.

1 Introduction

Document-Level Event Argument Extraction (DEAE) is a crucial task in information extraction, focusing on extracting arguments and their roles associated with specific event types within a document (Xu et al., 2021). Figure 1 illustrates an example of DEAE: the event *conflict.attack* is triggered by the word *attacked*. The arguments identified are *machete, man, soldiers, mall*, with their respective roles as *instrument, attacker, target, place*.

Typical DEAE models can usually be categorized into generation-based models (Hsu et al., 2022; Zeng et al., 2022) and classification-based ones (Zhang et al., 2020; Xu et al., 2022). Generation-based models typically employ an argument-specified template in conjunction with contextual input to generate arguments for each extraction task (Nguyen et al., 2023). In the given example, the template can be defined as *<arg1> attacked <arg2> using <arg3> at <arg4> place*, wherein a semantic relationship *attacked* is established between *arg1* and *arg2* roles, which has been shown to be beneficial for argument extraction (Liu et al., 2023b). In contrast, classification-based models identify candidate spans and classify their roles independently (Li et al., 2023), which still fail to explicitly capture significant relationships between roles. These relationships primarily stems from their task-specific architectural designs inherent in these models. Moreover, such models are constrained by their heavy reliance on exhaustive entity annotations for joint training (Zhang et al., 2023). These limitations may restrict classification-based models' extraction performance and necessitate urgent improvements in this field.

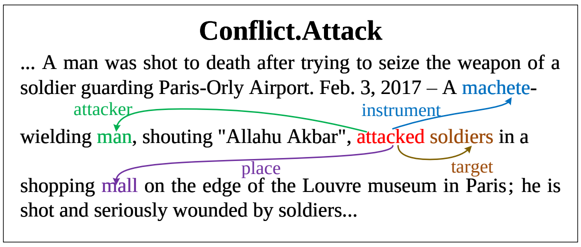


Figure 1: An example of DEAE. The event is triggered by *executed*, underlined words represent Arguments, and the arcs represent their Roles.

In this paper, we propose a novel approach Generation-Augmented and Embedding Fusion (**GAEF**), which contains Generation-Augmented Module (GAM) and Embedding Fusion Module (EFM). GAEF aims to address the aforementioned limitations by integrating role relationship information into the classification task using generative techniques and embedding fusion mechanisms. Specifically, GAM first utilizes a predefined event template as initial semantic structure and applies generative language models to fill in this template to produce multiple candidate arguments in its placeholders. Instead of using these arguments

[*]Corresponding authors.

Proceedings of the 31st International Conference on Computational Linguistics, pages 4078–4084
January 19–24, 2025.

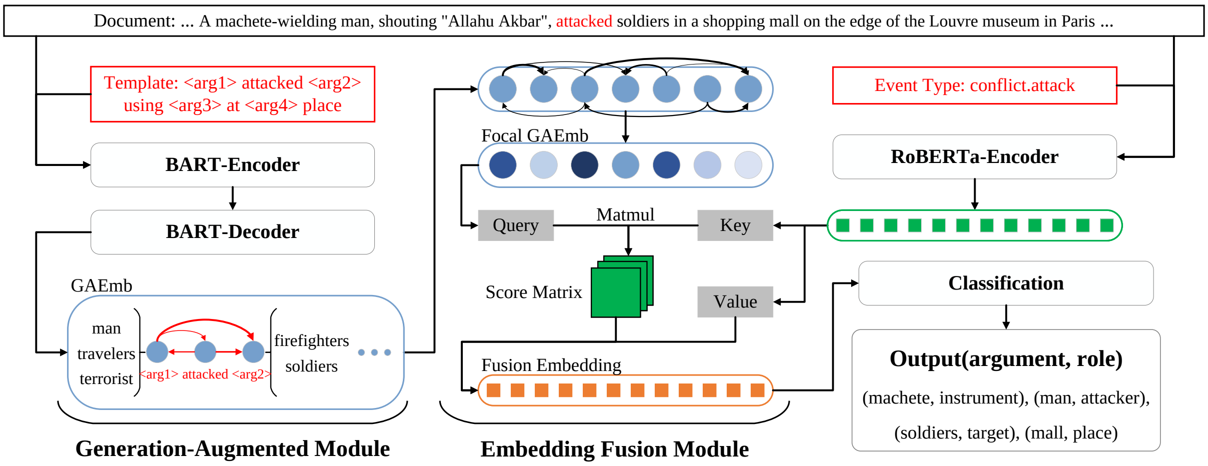


Figure 2: The overall architecture of GAEF, with an example of extracting the arguments and their roles.

directly, GAM employs a unified embedding technique to encapsulate their semantic information into a high-dimensional embedding, which preserves comprehensive contextual details derived from the generative model. EFM employs the generation-augmented embedding as a query, utilizing a novel embedding fusion mechanism to fuse it with the foundational embedding derived from a classification-based model. The fusion embedding not only encapsulates generative information, but also preserves the task-specific features inherent in the foundational model. This mechanism facilitates the incorporation of nuanced and comprehensive role relationship information into the classification process, thereby enhancing extraction performance.

GAEF not only provides rich role relationship information for classification tasks, but also reduces reliance on large-scale datasets, achieving superior performance with limited training data by leveraging external generation-augmented embeddings (see Section 4.4). Our contributions are summarized as follows: (1) We propose GAEF comprising GAM and EFM, amalgamates the embedding, containing role relations information generated by GAM, into a classification-based model using EFM. (2) We conduct extensive experiments on RAMS and WikiEvents datasets, demonstrating the effectiveness of GAEF.

2 Related Work

Document-level event argument extraction is an important and challenging task in Information Extraction, which aims to discover event arguments with specific roles from a document (Zheng et al., 2019; Tong et al., 2020; Wei et al., 2021; Wang et al., 2023). This field has evolved significantly since early effort in exploring the MUC-4 template-filling task (Chinchor, 1991).

Some researchers have proposed new methods based on classification-based models. A two-step method (Zhang et al., 2020) decomposed the problem into two steps: argument head-word detection and head-to-span expansion. TSAR (Xu et al., 2022) encoded the document using a two-stream encoding module to utilize local and global information. TT-BECG (Wan et al., 2023) proposed an edge-enhanced joint framework using Graph Neural Networks. Additionally, a chain-of-reasoning paradigm with the discrete First-Order Logic rules was introduced (Liu et al., 2023a).

With the advancement of generative models, another group of researchers have leveraged these techniques to achieve improved performance across various benchmarks. BART-Gen (Li et al., 2021) and EA2E (Zeng et al., 2022) developed conditional generative models using predefined templates. IPGPF (Huang et al., 2023) proposed an Iteratively Parallel Generation method with a Pre-Filling strategy. More recent approaches explored in-context learning with Large Language Models to reduce their reliance on extensive labeled data (Zhou et al., 2024).

3 Method

In this chapter, we first define the task and then introduce two core modules of GAEF: GAM and EFM. Figure 2 shows the overall architecture of our GAEF.

3.1 Task Definition

We define a document $\mathcal{D}$ consists of N words, denoted as $\mathcal{D} = \{w_1, w_2 \ldots w_N\}$, and the event type set $\mathcal{E}$. For each event $e \in \mathcal{E}$, the corresponding trigger word is denoted as tri_e and the argument role set is denoted as $\mathcal{R}_e$. Then, given a document $\mathcal{D}$ and the trigger $tri_e \in \mathcal{D}$ triggering the event type $e \in \mathcal{E}$, our task aims to detect all $(role, span)$ pairs, where $role \in \mathcal{R}_e$ is argument role, and $span \in \mathcal{D}$ is continuous words corresponding to $role$.

3.2 Generation-Augmented Module

In GAM, we first use a predefined event template as the initial semantic structure. Following (Li et al., 2021), we select a specific template associated with each event type, and the template explicitly delineates the relationships between role elements. This one-to-one mapping mechanism allows us to select the appropriate template for each event based on its type, with distinct templates corresponding to different event types. For a document context with the template, we utilize BART (Lewis et al., 2020) as our generative model, and the input is defined as follows:

$$g = \text{<d>}\ document\ \text{</d>}\ \text{<t>}\ template\ \text{</t>}, \quad (1)$$

where <d> and <t> are special tokens. We use BART-encoder to encode g:

$$H_g^{enc} = \text{BART-encoder}(g), \quad (2)$$

where H_g^{enc} stores the semantic information of both the document context and the template. Note that we aim to obtain the embeddings of candidate arguments filled in the placeholders, rather than the arguments themselves. The embeddings capture all alternatives considered during the generation process, encompassing all potential candidate arguments for the generative output. Unlike the final output token, which represents only the highest probability word and thus may propagate errors if incorrect, embeddings encapsulate a comprehensive representation of all information generated during the process to mitigate the impact of error propagation. Since the current embedding is generated based on its previous ones in an auto-regressive way, we employ BART-decoder to decode H_g^{enc} and obtain individual sub-embedding corresponding to each token:

$$E_{n+1}^{dec} = \text{BART-decoder}(H_g^{enc}, E_1^{dec}, \ldots, E_n^{dec}), \quad (3)$$

where $E_1^{dec}, \ldots, E_n^{dec}$ are all sub-embedding before E_{n+1}^{dec}. Finally, we concatenate each sub-embedding to obtain full embedding E_g^{dec}. We denote it as Generation-Augmented Embedding (GAEmb), which contains contextual details of both roles and their relationships.

3.3 Embedding Fusion Module

In EFM, we employ GAEmb as a query and fuse it with the foundational embedding derived from a classification-based model through an embedding fusion mechanism, resulting in a fusion embedding. Prior to this integration, we conduct a preprocessing step on GAEmb to enhance focus on roles and strengthen the relationships between them. We pass it through different linear transformations, obtaining the Query, Key, and Value matrices:

$$Q = E_g^{dec} W_q, K = E_g^{dec} W_k, V = E_g^{dec} W_v, \quad (4)$$

where $W_q, W_k, W_v \in \mathbb{R}^{d \times d}$ are all trainable weight matrices. Then we get attention score matrix S through the dot product of the Query and Key:

$$S = QK^T / \sqrt{d}. \quad (5)$$

Normalizing and multiplying the attention score matrix with the Value, we obtain $\hat{E}_g^{dec}$:

$$\hat{E}_g^{dec} = \text{Softmax}(S) \cdot V. \quad (6)$$

Next, we utilize a novel embedding fusion mechanism so as to fuse $\hat{E}_g^{dec}$ into a foundational embedding derived from classification-based model. For a document context and the event type et, the input is as follows:

$$c = \text{<d>}\ document\ \text{</d>}\ \text{<e>}\ et\ \text{</e>}, \quad (7)$$

where <d> and <e> are special tokens. Then we use RoBERTa (Liu et al., 2019) as our classification-based model to encode c and obtain the input embedding:

$$H_c^{enc} = \text{RoBERTa}(c). \quad (8)$$

We use $\hat{E}_g^{dec}$ as the new Query. By using it, we can enhance the classification task from the perspective of roles and their relationships. Specifically, through the cross-attention mechanism between $\hat{E}_g^{dec}$ and H_g^{enc}, we obtain the fusion embedding:

$$F = \text{Softmax}\left(\hat{E}_g^{dec} W_q' \left(H_g^{enc} W_k'\right)^T / \sqrt{d}\right) \cdot \left(H_g^{enc} W_v'\right), \quad (9)$$

where $W'_q, W'_k, W'_v \in \mathbb{R}^{d \times d}$ are trainable weight matrices. The fusion embedding contains both role relationship information derived from generative model and the task-specific features inherent in the classification-based model, which can provide robust performance for DEAE.

3.4 Classification Module

Finally, we process the fusion embedding to extract arguments in classifiication module. Referring to (Liu et al., 2023b), we employ the boundary loss to identify the start and end positions of arguments. The boundary loss is formulated as:

$$\mathcal{L}_b = -\sum_{i=1}^{|\mathcal{D}|}[y_i^s \log P_i^s + (1 - y_i^s)\log(1 - P_i^s) + y_i^c \log P_i^c + (1 - y_i^c)\log(1 - P_i^c)], \tag{10}$$

where y_i^s and y_i^e represent the gold labels, and P_i^s (P_i^c) denotes the predicted probability of word w_i being the first (last) word of a gold argument span, respectively. For role prediction, we utilize a feed-forward network that takes multiple features as input. These features include span representation, trigger representation, their absolute difference, element-wise multiplication, event type embedding, and span length embedding. Specifically, the role prediction $P(r_{i:j})$ for a candidate span $s_{i:j}$ is computed as:

$$P(r_{i:j}) = \text{FFN}(\mathbf{I}_{i:j}), \tag{11}$$

where $\mathbf{I}_{i:j}$ represents the concatenation of all aforementioned features. To enhance focus on informative positive samples, we incorporate focal loss:

$$\mathcal{L}_c = -\sum_{i=1}^{|\mathcal{D}|}\sum_{j=1}^{|\mathcal{D}|}\alpha[1 - P(r_{i:j} = y_{i:j})]^{\gamma} \cdot \log P(r_{i:j} = y_{i:j}). \tag{12}$$

The final training objective combines both losses, $\mathcal{L}_c$ and $\mathcal{L}_b$, weighted by a hyperparameter λ:

$$\mathcal{L} = \lambda\mathcal{L}_b + \mathcal{L}_c. \tag{13}$$

4 Experiment

4.1 Experimental Setup

Datasets. We use RAMS (Ebner et al., 2020) and WikiEvents (Li et al., 2021), two commonly used datasets in DEAE, for our experiments. Both datasets provide complex, cross-sentence event extraction scenarios. RAMS is constructed from news articles and discussions, including 9,124 events, 139 event types, and 65 role types. WikiEvents is constructed from Wikipedia articles, including 3,951 events, 50 event types and 59 role types. The detailed data statistics are shown in Table 1.

Dataset	Split	Doc.	Event	Argument
RAMS	Train	3,194	7,329	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	566

Table 1: The detailed data statistics of RAMS and WikiEvents.

Models	**RAMS**		**WikiEvents**		
	Arg-I	Arg-C	Arg-I	Arg-C	Head-C
BERT-CRF∗	-	40.3	-	32.3	43.3
EEQA∗	46.4	44.0	54.3	53.2	56.9
BART-Gen∗	50.9	44.9	47.5	41.7	44.2
PAIE∗	54.7	49.5	68.9	63.4	66.5
EDGE∗	55.2	49.7	68.2	62.8	65.9
GAEF	**56.8**	**50.0**	**70.1**	**64.1**	**67.3**

Table 2: Overall performances on RAMS and WikiEvents. ∗ means the results from original paper. All models here are all base-scale.

Metrics. We use three evaluation metrics to measure performance. Argument Identification **(Arg-I)**: If an argument's offset matches any golden arguments, the argument is correctly identified. Argument Classification **(Arg-C)**: If the argument is correctly identified and its role is also correct, the argument is correctly classified. Argument Head Classification **(Head-C)**: For WikiEvents dataset, Head-C is only concerned about the head position of an argument matching. We use F1 score to evaluate performance.

Baseline. We compare GAEF with several DEAE models: **BERT-CRF** (Shi and Lin, 2019), **EEQA** (Du and Cardie, 2020), **BART-Gen** (Li et al., 2021), **PAIE** (Ma et al., 2022) and **EDGE** (Li et al., 2023).

4.2 Overall Performance

Table 2 compares GAEF with the baselines. From the results, we can conclude that: On RAMS dataset, GAEF yields an improvement of 1.6∼10.4 Arg-I F1 and 0.3∼9.7 Arg-C F1. On WikiEvents dataset, it outperforms 1.2∼15.8 Arg-I

Method	Arg-I	Arg-C	Head-C
GAEF	**70.1**	**64.1**	**67.3**
w/o GAM	67.7	59.2	63.5
w/o Focal GAEmb	68.0	62.1	65.4

Table 3: Ablation Study on WikiEvents.

F1, 0.7∼31.8 Arg-C F1, and 0.8∼24.0 Head-C F1 than all the baselines. Notably, we achieve a significant improvement in the Arg-I metric, which indicates that GAEF can extract arguments better. This enhancement highlights the effectiveness of our approach in allowing it to more accurately identify and extract relevant arguments from documents.

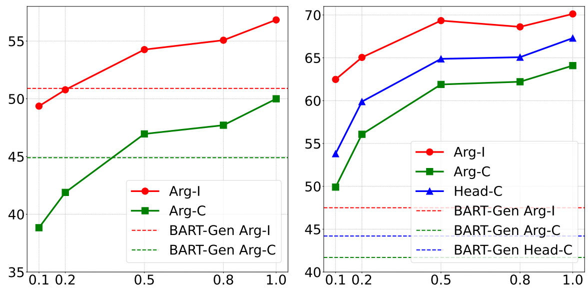
4.3 Ablation Study

To make our experiments more complete, we also conduct the ablation study on WikiEvents to investigate the capabilities of each component in GAEF, as shown in Table 3. "w/o GAM" means removing GAM to quantify its effectiveness. "w/o Focal GAEmb" means removing the process of enhancing GAEmb to focus on roles and relationships in EFM, showing the importance of this process.

We can find that each component can help DEAE, to be specific: (1) Without GAM, the performance drops dramatically by 2.4 Arg-I F1 and 4.9 Arg-C F1. This indicates that GAE can enhance the overall robustness in the classification-based model. (2) Without emphasizing GAEmb in EFM, the performance drops dramatically by about 2.1 Arg-I F1 and 2.0 Arg-C F1. This indicates that it can emphasize roles and relationships within GAEmb, thereby positively affecting subsequent classification tasks.

4.4 Low-Resource Learning

To better evaluate the efficacy of the proposed GAEF under low-resource training settings, we asymptotically increase the training data to analyze the performance on both datasets, and the results are shown in Figure 3. As can be seen, GAEF achieves superior performance relative to the BART-Gen baseline while utilizing only 10% of training set on WikiEvents. This remarkable performance demonstrates the model's efficiency in learning from limited data. It significantly reduces dependency on large-scale training data and exhibits promising potential in low-resource learning experimental settings.



(a) RAMS (b) WikiEvents

Figure 3: Experiment results in low-resource learning. The x-axis represents the percentage of the training data, and the y-axis represents the F1 score.

5 Conclusion

In this paper, we propose a novel approach GAEF, including two core modules: GAM and EFM, to improve extraction performance for the DEAE task. GAM takes a document and a template, which contains role relationship information, into a generative model to generate an embedding while EFM integrates the embedding into a classification-based model. Extensive experiments confirm that GAEF performs more effectively than the baselines, demonstrating its potential in low-resource scenarios.

Limitations

While GAEF demonstrates strong performance, there are several limitations to consider:

- Computational complexity: The generation and fusion processes may increase computational requirements compared to simpler classification-based model.

- Template dependency: The effectiveness of the GAM relies on well-designed templates. Developing comprehensive templates for diverse event types could be challenging and time-consuming.

Acknowledgment

This research is supported by National Nature Science Foundation of China (No.62137002, 62176245). We thank the anonymous reviewers for their constructive comments that help improve the manuscript.

References

Nancy Chinchor. 1991. MUC-3 evaluation metrics. In *Third Message Understanding Conference (MUC-3): Proceedings of a Conference Held in San Diego, California, May 21-23, 1991*.

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, Online. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. DEGREE: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1890–1908, Seattle, United States. Association for Computational Linguistics.

Guanhua Huang, Runxin Xu, Ying Zeng, Jiaze Chen, Zhouwang Yang, and Weinan E. 2023. An iteratively parallel generation method with the pre-filling strategy for document-level event extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10834–10852, Singapore. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Hao Li, Yanan Cao, Yubing Ren, Fang Fang, Lanxue Zhang, Yingjie Li, and Shi Wang. 2023. Intra-event and inter-event dependency-aware graph network for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6362–6372, Singapore. Association for Computational Linguistics.

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

Jian Liu, Chen Liang, Jinan Xu, Haoyan Liu, and Zhe Zhao. 2023a. Document-level event argument extraction with a chain reasoning paradigm. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9570–9583, Toronto, Canada. Association for Computational Linguistics.

Wanlong Liu, Shaohuan Cheng, Dingyi Zeng, and Qu Hong. 2023b. Enhancing document-level event argument extraction with contextual clues and role relevance. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12908–12922, Toronto, Canada. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

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

Chien Nguyen, Hieu Man, and Thien Nguyen. 2023. Contextualized soft prompts for extraction of event arguments. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4352–4361, Toronto, Canada. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. *CoRR*, abs/1904.05255.

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020. Improving event detection via open-domain trigger knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5887–5897, Online. Association for Computational Linguistics.

Qizhi Wan, Changxuan Wan, Keli Xiao, Dexi Liu, Chenliang Li, Bolong Zheng, Xiping Liu, and Rong Hu. 2023. Joint document-level event extraction via token-token bidirectional event completed graph. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10481–10492, Toronto, Canada. Association for Computational Linguistics.

Xin Wang, Shengfei Lyu, Xiangyu Wang, Xingyu Wu, and Huanhuan Chen. 2023. Temporal knowledge graph embedding via sparse transfer matrix. *Information Sciences*, 623:56–69.

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, Online. Association for Computational Linguistics.

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

Runxin Xu, Peiyi Wang, Tianyu Liu, Shuang Zeng, Baobao Chang, and Zhifang Sui. 2022. A two-stream AMR-enhanced model for document-level event argument extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5025–5036, Seattle, United States. Association for Computational Linguistics.

Qi Zeng, Qiusi Zhan, and Heng Ji. 2022. EA^2E: Improving consistency with event awareness for document-level argument extraction. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2649–2655, Seattle, United States. Association for Computational Linguistics.

Meiru Zhang, Yixuan Su, Zaiqiao Meng, Zihao Fu, and Nigel Collier. 2023. COFFEE: A contrastive oracle-free framework for event extraction. In *Proceedings of the First Workshop on Matching From Unstructured and Structured Data (MATCHING 2023)*, pages 33–44, Toronto, ON, Canada. Association for Computational Linguistics.

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020. A two-step approach for implicit event argument detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7479–7485, Online. 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)*, pages 337–346, Hong Kong, China. Association for Computational Linguistics.

Hanzhang Zhou, Junlang Qian, Zijian Feng, Lu Hui, Zixiao Zhu, and Kezhi Mao. 2024. LLMs learn task heuristics from demonstrations: A heuristic-driven prompting strategy for document-level event argument extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11972–11990, Bangkok, Thailand. Association for Computational Linguistics.