One Small and One Large for Document-level Event Argument Extraction

Jiaren Peng[1*], **Hongda Sun**[2*], **Wenzhong Yang**[1 †], **Fuyuan Wei**[1], **Liang He**[3], **Liejun Wang**[1],

[1]School of Computer Science and Technology (School of Cyberspace Security), Xinjiang University, China
[2]Gaoling School of Artificial Intelligence, Renmin University, China
[3]Department of Electronic Engineering, Tsinghua University, China
1354527247@qq.com, sunhongda98@ruc.edu.cn, yangwenzhong@xju.edu.cn, wfy@stu.xju.edu.cn,
heliang@tsinghua.edu.cn, wljxju@xju.edu.cn

Abstract

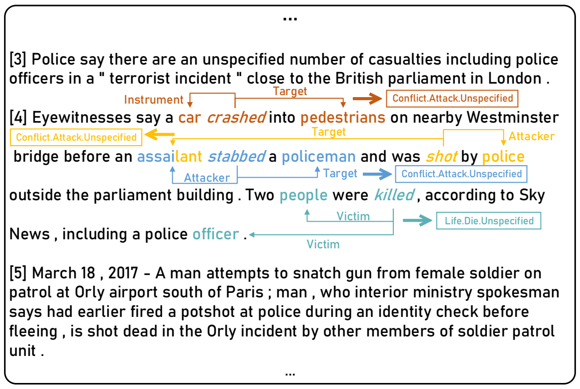
Document-level Event Argument Extraction (EAE) faces two challenges due to increased input length: 1) difficulty in distinguishing semantic boundaries between events, and 2) interference from redundant information. To address these issues, we propose two methods. The first method introduces the Co and Structure Event Argument Extraction model (CsEAE) based on Small Language Models (SLMs). CsEAE includes a co-occurrences-aware module, which integrates information about all events present in the current input through context labeling and co-occurrences event prompts extraction. Additionally, CsEAE includes a structure-aware module that reduces interference from redundant information by establishing structural relationships between the sentence containing the trigger and other sentences in the document. The second method introduces new prompts to transform the extraction task into a generative task suitable for Large Language Models (LLMs), addressing gaps in EAE performance using LLMs under Supervised Fine-Tuning (SFT) conditions. We also fine-tuned multiple datasets to develop an LLM that performs better across most datasets. Finally, we applied insights from CsEAE to LLMs, achieving further performance improvements. This suggests that reliable insights validated on SLMs are also applicable to LLMs. We tested our models on the Rams, WikiEvents, and MLEE datasets. The CsEAE model achieved improvements of 2.1%, 2.3%, and 3.2% in the Arg-C F1 metric compared to the baseline, PAIE (Ma et al. 2022). For LLMs, we demonstrated that their performance on document-level datasets is comparable to that of SLMs [1].

Introduction

Event Argument Extraction (EAE) aims to extract structured event information composed of arguments corresponding to event roles from text (Peng et al. 2024). As shown in Figure 1, given a trigger and event type, along with a predefined list of roles for the event type, the model needs to extract the corresponding token spans as arguments for each role. This structured information can enhance the performance of downstream tasks such as question answering (Costa, Gottschalk, and Demidova 2020), dialogue systems (Zhang, Chen, and Bui 2020), and recommendation systems (Li et al. 2020).

[*]These authors contributed equally.
[†]corresponding author

[1]All code is available at https://github.com/simon-p-j-r/CsEAE



...

[3] Police say there are an unspecified number of casualties including police officers in a " terrorist incident " close to the British parliament in London .

[4] Eyewitnesses say a car *crashed* into pedestrians on nearby Westminster bridge before an assailant *stabbed* a policeman and was *shot* by police outside the parliament building . Two people were *killed* , according to Sky News , including a police officer .

[5] March 18 , 2017 – A man attempts to snatch gun from female soldier on patrol at Orly airport south of Paris ; man , who interior ministry spokesman says had earlier fired a potshot at police during an identity check before fleeing , is shot dead in the Orly incident by other members of soldier patrol unit .

...

Instrument
Target
Conflict.Attack.Unspecified
Conflict.Attack.Unspecified
Target
Attacker
Attacker
Target
Conflict.Attack.Unspecified
Victim
Life.Die.Unspecified
Victim

Figure 1: An EAE instance from the WikiEvents dataset.

As the length of document-level input texts increases, document-level EAE faces two critical challenges: (1) difficulty in distinguishing semantic boundaries between events (He, Hu, and Tang 2023). As shown in Figure 1, the four trigger words *crashed*, *stabbed*, *shot*, and *killed*, each trigger four events. The argument distribution of these events is extremely dense, and different events can share the same token span as arguments corresponding to different roles. These dense and overlapping events make the semantic boundaries between them blurry. (2) The volume of information received by the model increases significantly; however, this information includes not only useful data for the extraction task but also a large amount of redundant information that interferes with the task (Xu et al. 2022). For example, in the sentence [5], the presence of person nouns such as *man*, *female* and *soldier* can mislead the extraction of the *Victim* role for the *Life.Die.Unspecified* event triggered by *killed*. However, previous work has not simultaneously addressed both of these issues (Ma et al. 2022; Xu et al. 2022; He, Hu, and Tang 2023).

To address these issues, we proposed two methods, with the first being the co and structure EAE model (CsEAE) based on Small Language Models (SLMs). CsEAE enhances the boundaries of the model's focus from both event and sen-

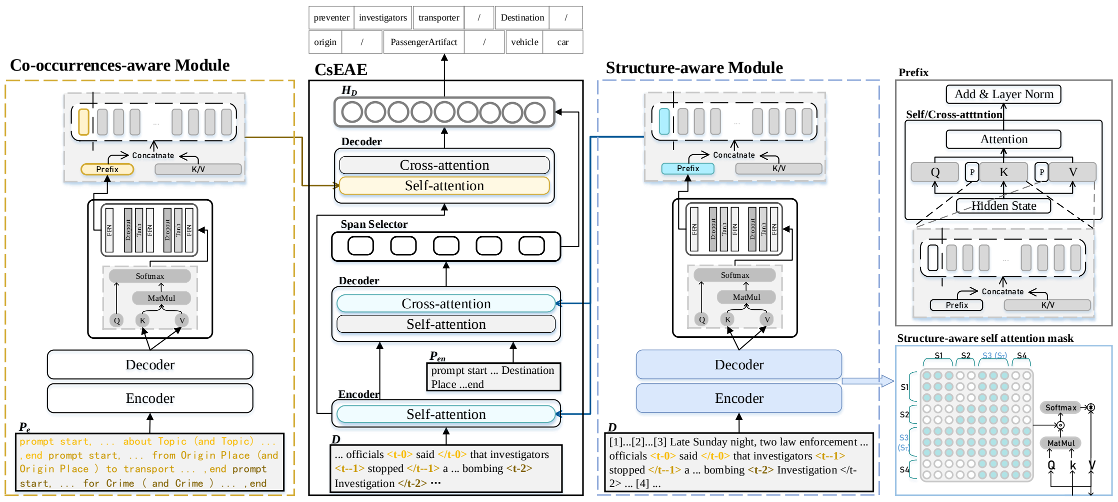

Co-occurrences-aware Module
CsEAE
Structure-aware Module
Prefix
preventer
investigators
transporter
Destination
origin
PassengerArtifact
vehicle
car
Concatnate
Prefix
K/V
Softmax
MatMul
Decoder
Encoder
Cross-attention
Self-attention
Span Selector
prompt start ... Destination Place ...end
... officials <t-0> said </t-0> that investigators <t--1> stopped </t--1> a ... bombing <t-2> Investigation </t-2> ···
[1]...[2]...[3] Late Sunday night, two law enforcement ... officials <t-0> said </t-0> that investigators <t--1> stopped </t--1> a ... bombing <t-2> Investigation </t-2> ... [4] ...
Add & Layer Norm
Self/Cross-atttntion
Attention
Hidden State
Structure-aware self attention mask

Figure 2: Overview of CsEAE. The yellow attention represents the concatenation of co-occurrences-aware module, while the blue attention represents the concatenation of structure-aware module.

tence perspectives. From the event perspective, to help the model capture semantic boundaries between events, we introduced a co-occurrence-aware module. This module identifies all co-occuring events in the input by marking triggers and encoding related prompts. From the sentence perspective, while event mentions are document-level, event information is often within a single sentence. For instance, in the WikiEvents dataset, over 94% of arguments are in the same sentence as the trigger; in the Rams dataset, over 82%; and in the MLEE dataset, over 99%. This highlights the importance of the information in the trigger sentence for the extraction task. To emphasize this, we structured the knowledge around the trigger sentence and its relationship with other sentences in the document. This approach helps the model selectively gather relevant information from other sentences, reducing distractions from redundant information.

Additionally, we proposed a second method based on Large Language Models (LLMs). We designed prompts tailored to LLMs for each dataset and performed Supervised Fine-Tuning (SFT) on the LLMs. This approach addresses a gap in the EAE field, which previously lacked fine-tuned LLMs (Ma et al. 2023; Chen et al. 2024; Zhou et al. 2023). Inspired by the use of large-scale high-quality data for continuous pretraining (Yang et al. 2024), we attempted multi-dataset fine-tuning to make the LLMs more familiar with event extraction tasks. On this basis, we also conducted enhanced training on the LLMs using additional datasets.

Finally, inspired by CsEAE, where co-occurrence- and structure-aware interactions enhance the model's ability to capture event boundaries and reduce interference from redundant information, we applied these insights to LLMs. This led to further performance improvements and introduced a novel perspective: the reliable insights validated on SLMs are also applicable to LLMs.Our contributions are summarized below:

• We propose the CsEAE model, which incorporates a co-occurrences-aware module to capture semantic boundaries between events. Additionally, it uses a structure-aware module to build structured perception information, allowing the model to minimize interference from redundant information.

• We designed different prompts for various datasets and further used SFT to enhance the performance of LLMs. Additionally, we proposed multiple datasets SFT and supplementary dataset enhancement training, which led to even better performance.

• We applied insights from SLMs to LLMs, resulting in further performance improvements. This shows that reliable insights validated on SLMs are also effective for LLMs.

Related Works

Document-level Event Argument Extraction

With the capability to extract events across multiple sentences, document-level EAE has garnered significant research interest. Some studies incorporate abstract meaning representation into the extraction task (Xu et al. 2022; Yang et al. 2023; Hsu et al. 2023b). BART-Gen (Li, Ji, and Han 2021) utilizes a prompt-based generative approach to generate event arguments end-to-end, and subsequently, PAIE (Ma et al. 2022) introduces more effective manually crafted prompts, using slot prompts to extract arguments by filling slots. TabEAE (He, Hu, and Tang 2023) defines EAE as a table-filling problem, enabling the extraction of all events present in the input simultaneously. However, the aforementioned models did not simultaneously address capturing the semantic boundaries between events and As shown in the Figure 2, CsEAE explicitly addresses both of the issues by incorporating co-occurrences- and structure-aware modules.

Large Language Models for Event Argument Extraction

The success of LLMs (Touvron et al. 2023) has been widely recognized, and in recent years, there has been increasing research on the development of LLMs in the field of event extraction. Such as (Ma et al. 2023; Zhou et al. 2023; Ma et al.

2024; Chen et al. 2024) have explored the performance of LLMs in event extraction tasks. However, these studies typically rely on In-context Learning (ICL). While this approach significantly conserves computational resources, it often results in less satisfactory outcomes compared to SLMs. In this paper, we move beyond the limitations of ICL and employ SFT, enabling LLMs to learn how to perform event extraction more effectively. We also found that multiple dataset SFT can improve the extraction capabilities of LLMs. Building on this, we introduced supplementary dataset enhancement training. Finally, we incorporated the insights derived from CsEAE into LLMs, achieving further improvements.

CsEAE Model

In this section, we will provide a detailed introduction to each component of CsEAE.

Basic Architecture

In the Figure 2, given the input $\mathcal{D}$ and the prompt p_{e_n} corresponding to the event type to be extracted, we fed $\mathcal{D}$ into an encoder with a structure-aware prefix, resulting in $H_{\mathcal{D}}^{enc}$. Then, $H_{\mathcal{D}}^{enc}$ is passed through a decoder with a co-occurrences-aware prefix to obtain the contextual representation of $\mathcal{D}$, referred to as the event-oriented context representation $H_{\mathcal{D}}$. This process can be formulated as:

$$H_{\mathcal{D}}^{enc} = Encoder_{Sap}(\mathcal{D}), \tag{1}$$

$$H_{\mathcal{D}} = Decoder_{Cap}(H_{\mathcal{D}}^{enc}, H_{\mathcal{D}}^{enc}). \tag{2}$$

Where Sap represents structure-aware prefix, Cap represents co-occurrences-aware prefix.

To create the span selector θ, we need to interactively encode each token representation of $\mathcal{D}$ with p_{e_n} at a deep level. Specifically, we will input $H_{\mathcal{D}}^{enc}$ and p_{e_n} together into the Decoder after concatenating with the structure-aware prefix, obtaining its context-oriented prompt representation H_{pt}. We formalize it as:

$$H_{pt} = Decoder_{Sap}(H_{\mathcal{D}}^{enc}, p_{e_n}). \tag{3}$$

Co-occurrences-aware Module

Co-occurrences-aware module introduces event co-occurrences-aware interaction through three aspects: context labeling, prompt extraction and co-occurrences prefix.

Context Labeling Given the input of the model $\mathcal{D} = \{t_1, t_2, \ldots, t_n\}$, where t_i represents the i-th token in the input. Given $E = \{e_0, e_1, \ldots, e_l\}$, where e_i represents one event appearing in $\mathcal{D}$, and l represents the number of events appearing in $\mathcal{D}$. Given all the triggers $T = \{e_0^t, e_1^t, \ldots, e_l^t\}$, where e_i^t represents the trigger corresponding to event e_i, and e_i^t corresponds one-to-one with e_i. We will annotate all token spans corresponding to triggers in $\mathcal{D}$ according to the order in which the triggers e_i^t appear in $\mathcal{D}$. Specifically, for the trigger e_n^t corresponding to the event e_n being extracted, we will annotate its appearance in $\mathcal{D}$ using special characters <t- -1>and </t- -1>.

For triggers e_j^t corresponding to other events existing in $\mathcal{D}$, we will annotate them according to the order of appearance in $\mathcal{D}$ using <t-k>and </t-k>, where k is calculated starting from 0 and incremented by 1.

Prompt Extraction Given $P_e = \{p_{e_1}, p_{e_2}, \ldots, p_{e_l}\}$, where p_{e_i} represents the prompt corresponding to event e_i. Notice that p_{e_i}, e_i^t and e_i are uniquely paired. In this paper, we utilize prompts proposed in PAIE (Ma et al. 2022) for the Rams and WikiEvents datasets and those in TabEAE (He, Hu, and Tang 2023) for the MLEE dataset. To fully utilize the semantic information provided by the prompts, we first concatenate all prompts P_e corresponding to events mentioned in $\mathcal{D}$. Then, we encode them into the SLMs to obtain dense vector representations W_C for all co-occurring event prompts. Finally, the information of W_C is integrated into the prefixes.

Co-occurrences Prefix After constructing the co-occurrences-aware matrix W_C for the current event mention $\mathcal{D}$, we condense W_C into prefixes (Li and Liang 2021; Hsu et al. 2023b), which then participate in the model's generation. As shown in the Figure 2. Firstly, we introduce a learnable vector of length len, which serves as the Q vector for multi-head attention, where len is a tunable hyperparameter controlling the final length of the prefixes to be fed into the SLMs, we set it as 40. Then, W_C is used as the K and V vectors in multi-head attention computation, which is computed with the Q vector. After multi-head attention computation, we obtain a set of compressed dense vector $\mathcal{P}$, which then undergoes a series of linear layers. Finally, $\mathcal{P}$ is evenly split into c segments $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_c\}$, each with a length of len, where c is the number of transformer layers in the SLMs. This results in prefixes that can be concatenated into the SLMs for computation.

Structure-aware Module

structure-aware module introduces structure-aware interaction through two aspects: structural relationship and structure prefix.

Structural Relationship For different document inputs, as shown in Figure 2 (blue part on the right), we designed a structure-aware self-attention mask M_s, which treats sentences as units and trains the model to be structure-aware across the entire document. Specifically, given the document-level input $\mathcal{D} = \{S_1, S_2, \ldots, S_m\}$, where S_i represents the i-th sentence in $\mathcal{D}$, and given the trigger e_n^t of the current event to be extracted, located in sentence S_n, M_s restricts the receptive field of all sentences except S_n, allowing these sentences to focus only on themselves and S_n. In contrast, S_n can attend to all sentences.

We can obtain the structure-aware dense vector representation W_S for the event mention $\mathcal{D}$ as follows:

$$W_S = Decoder(Encoder(\mathcal{D}, M_s)). \tag{4}$$

Structure Prefix Finally, following the same approach as described in Section Co-occurrences-aware Prefix, the information from W_S is integrated into the prefixes and participates in the model's generation.

Span Selection

After obtaining H_{pt}, we extract the slot representation ψ_k corresponding to the pre-defined roles from H_{pt}, where k represents the k-th slot. Then, we convert ψ_k into a span selector specific to that slot θ_k (Ma et al. 2022; Du and Cardie 2020a). Next, apply the span selector θ_k directly to the event-oriented context representation H_D to determine the argument's token span $[p_k^{(start)}; p_k^{(end)}]$.

$$\begin{aligned}
\psi_k^{(start)} &= \psi_k \circ w^{(start)} \in R^h, \\
\psi_k^{(end)} &= \psi_k \circ w^{(end)} \in R^h, \\
\text{logit}_k^{(start)} &= \psi_k^{(start)} H_{\mathcal{D}} \in R^L, \\
\text{logit}_k^{(end)} &= \psi_k^{(end)} H_{\mathcal{D}} \in R^L, \\
p_k^{(start)} &= \text{Softmax}(\text{logit}_k^{(start)}) \in R^L, \\
p_k^{(end)} &= \text{Softmax}(\text{logit}_k^{(end)}) \in R^L.
\end{aligned} \tag{5}$$

Where $\theta = [w^{(start)}; w^{(end)}] \in R^{h\times 2}$ is a learnable parameter matrix shared by all span selectors, $\circ$ represents element-wise multiplication. $\theta_k = [\psi_k^{(start)}; \psi_k^{(end)}]$ is the span selector specific to the slot corresponding to the role, L demotes the context lengrth.

We define the loss function $\mathcal{L}$ as follows:

$$\begin{aligned}
\mathcal{L}_k(\mathcal{D}) &= -(\log p_k^{(start)}(s_k) + \log p_k^{(end)}(e_k)), \\
\mathcal{L} &= \sum_{\mathcal{D}\in B} \sum_k \mathcal{L}_k(\mathcal{D}).
\end{aligned} \tag{6}$$

Where B ranges over all context in dataset and k ranges over all slots in prompt p_{e_n} for $\mathcal{D}$, and (s_k, e_k) represents the token span of the most likely argument corresponding to the role in H_D.

During the inference phase, we predefine spans $\mathcal{C}$ that cover all possible spans within a predefined length and include a special span (0, 0) to represent the absence of any corresponding argument. Then, we utilize the span selector θ_k to compute scores for all spans using the following method:

$$\text{score}_k(i,j) = \text{logit}_k^{(start)}(i) + \text{logit}_k^{(end)}(j). \tag{7}$$

Where i and j represent the start and end indices of each span in the set of spans.

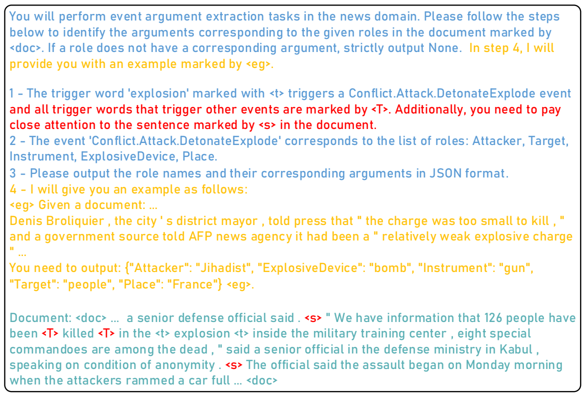
Based on the scores, we determine the predicted final span by selecting the span with the highest score.

$$(\widehat{s_k}, \widehat{e_k}) = \arg \max_{(i,j)\in\mathcal{C}} \text{score}_k(i,j). \tag{8}$$

For the issue of multiple arguments of the same role, we utilize the Hungarian algorithm (Kuhn 1955; Ma et al. 2022). For the problem of allocating multiple slots corresponding to a single role, we employ Bipartite Matching (Carion et al. 2020; Yang et al. 2021; Ma et al. 2022).

Generalization in LLMs

In this section, we will provide a detailed explanation of how to use LLMs for EAE and further improvements.



You will perform event argument extraction tasks in the news domain. Please follow the steps below to identify the arguments corresponding to the given roles in the document marked by <doc>. If a role does not have a corresponding argument, strictly output None. In step 4, I will provide you with an example marked by <eg>.

1 - The trigger word 'explosion' marked with <t> triggers a Conflict.Attack.DetonateExplode event and all trigger words that trigger other events are marked by <T>. Additionally, you need to pay close attention to the sentence marked by <s> in the document.
2 - The event 'Conflict.Attack.DetonateExplode' corresponds to the list of roles: Attacker, Target, Instrument, ExplosiveDevice, Place.
3 - Please output the role names and their corresponding arguments in JSON format.
4 - I will give you an example as follows:
<eg> Given a document: ...
Denis Broliquier , the city ' s district mayor , told press that " the charge was too small to kill , " and a government source told AFP news agency it had been a " relatively weak explosive charge " ...
You need to output: {"Attacker": "Jihadist", "ExplosiveDevice": "bomb", "Instrument": "gun", "Target": "people", "Place": "France"} <eg>.

Document: <doc> ... a senior defense official said . <s> " We have information that 126 people have been <T> killed <T> in the <t> explosion <t> inside the military training center , eight special commandoes are among the dead , " said a senior official in the defense ministry in Kabul , speaking on condition of anonymity . <s> The official said the assault began on Monday morning when the attackers rammed a car full ... <doc>

Figure 3: Prompt for LLMs on WikiEvents. The blue parts represent $\mathcal{I}$, the yellow parts represent $\mathcal{E}$, the green parts represent $\mathcal{Q}$ and the red parts represent co-occurrences- and structure-aware interactions.

Prompt Design

Given the input $\mathcal{D}$, we designed a corresponding prompt $\mathcal{P}_{\mathcal{L}}(\mathcal{D})$ for LLMs. As shown in the Figure 3, the prompt $\mathcal{P}_{\mathcal{L}}(\mathcal{D})$ is divided into three parts:

$$\mathcal{P}_{\mathcal{L}}(\mathcal{D}) = [\mathcal{I}; \mathcal{E}; \mathcal{Q}]. \tag{9}$$

The first part is the instruction $\mathcal{I}$, which describes the task and provides basic information such as the trigger, roles, and output format. The second part is the example $\mathcal{E}$, which provides a single example (one-shot) to the LLMs. We identified corresponding examples for each event type from the training set and the example should include as many arguments as possible from the input. The third part is the question $\mathcal{Q}$. We use <doc>for input to separate the $\mathcal{Q}$ from other components in the prompt.

Supervised Fine-Tuning

SFT is the critical stage that endows the model with high-quality extraction capabilities. Through training data, the model can effectively leverage the latent knowledge accumulated during pre-training to understand and respond to extraction instructions (Yang et al. 2024).

A high-quality pre-training corpus can significantly enhance the performance of LLMs, even to the extent of breaking through scaling laws (Gunasekar et al. 2023). Inspired by this, and considering the complexity of the EAE domain (Ma et al. 2023), we sequentially merged multiple datasets and fine-tuned the LLMs using the combined dataset. To further exploit the improvements from multiple dataset SFT and enhance the model's sensitivity to extraction tasks, we incorporated additional datasets into the multiple dataset SFT, conducting enhanced training on the LLMs.

CsLLMs

In CsEAE, we optimized the model using event co-occurrences- and structure-aware interactions of the document. This brings up an important question: does insights that has been validated to be effective for extraction in SLMs also work effectively in LLMs?

Model	PLM	Rams		WikiEvents		MLEE	
		Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
EEQA*	RoBERTa	51.9	47.5	60.4	57.2	70.3	68.7
TSAR*	RoBERTa	57.0	52.1	<u>71.1</u>	65.8	72.6	71.5
BART-Gen*	BART	51.2	47.1	66.8	62.4	71.0	69.8
TabEAE-m2s	RoBERTa	56.2	51.4	69.7	64.9	-	-
TabEAE-m2m	RoBERTa	55.9	50.9	70.3	64.6	74.0	72.9
PAIE	BART	55.3	51.0	68.9	64.2	71.3	70.1
CsEAE	BART	<u>57.5</u>	<u>53.1</u>	70.9	<u>66.5</u>	<u>74.3</u>	<u>73.3</u>

Table 1: Overall performance of CsEAE and baselines. * means the value from the TabEAE's paper. All experiments utilized a large-scale PLM. The highest scores are underlined.

We believe this is a crucial question, as it can bridge future developments on LLMs with the extensive work previously done on SLMs. Therefore, we also incorporated event co-occurrences- and structure-aware interactions into the prompt. In the Figure 3, the changes are highlighted in red. Specifically, we introduced co-occurrences-aware interaction in the $\mathcal{Q}$ by marking the triggers and introduced structure-aware interaction by marking the sentence containing the trigger. Additionally, we guided the model in the $\mathcal{I}$ to pay attention to these marked pieces of information. We refer to the fine-tuned LLMs, which integrate the information mentioned above, as CsLLMs.

Experiments

Experimental Setup

Datasets We used the three most commonly employed datasets for document-level event argument extraction (EAE): Rams (Ebner et al. 2020), WikiEvents (Li, Ji, and Han 2021), and MLEE (Pyysalo et al. 2012). We preprocessed the data following previous methods (Trieu et al. 2020; Ma et al. 2022; He, Hu, and Tang 2023). To further enhance model training, we also incorporated sentence-level EAE datasets, specifically ACE (Doddington et al. 2004) and GENEVA (Parekh et al. 2023), applying preprocessing techniques from prior research (Hsu et al. 2022, 2023b; Parekh et al. 2023).

Additionally, to more comprehensively validate the effectiveness of CsEAE, we applied the data processing methods used in TextEE (Huang et al. 2024) to WikiEvents and RAMS. These methods included standardization of data assumptions, normalization of data processing steps, and standardization of dataset splits (5 times).

Implementation Details We used PyTorch and a single NVIDIA A40 Tensor Core GPU with 45GB to train all models and reproduce experiments of other models. We used BART (Lewis et al. 2020) as the backbone for CsEAE. During model training the learning rate was set to 2e-5. We used the methods provided by LLama-Factory [2] for model's SFT, employing LoRA-based (Hu et al. 2021) fine-tuning with a rank r of 8 and a dropout rate of 0.1. The batch size was set to 4, and training was conducted for 3 epochs on each dataset.

[2]https://github.com/hiyouga/LLaMA-Factory

Evaluation Metrics Fllowed by previous works (Ma et al. 2022; He, Hu, and Tang 2023), We used the Arg-I F1 and Arg-C F1 metrics to evaluate the model's performance on the argument identification and argument classification.

It should be noted that our use of Arg-I and Arg-C corresponds to Arg-I+ and Arg-C+ as defined in TextEE, meaning that predicted arguments are attached to the correct trigger words. In all experiments in this paper, Arg-I and Arg-C is equivalent to Arg-I+ and Arg-C+.

Baselines For SLMs, we categorized the baseline models into two groups:

(1) Classification-based models: EEQA (Du and Cardie 2020b), TSAR (Xu et al. 2022), TagPrime-C and TagPrime-CR (Hsu et al. 2023a);

(2) Generation-based models: Bart-Gen (Li, Ji, and Han 2021), X-Gear (Huang et al. 2022), AMPERE (Hsu et al. 2023b), PAIE (Ma et al. 2022), TabEAE (He, Hu, and Tang 2023).

For LLMs, we categorized the baseline models into two groups too:

(1) Open-AI: Chat-GPT [3], GPT-4o, GPT-4o-mini;

(2) Open-source: Llama3-8B [4] (Touvron et al. 2023), Llama3-8B-Instruct [5].

Main Results

CsEAE We evaluate the proposed model CsEAE and baseline methods under all benchmarks. In the Table 1, our model outperformed all baselines on the Rams and MLEE datasets.

Compared to the baseline model PAIE (Ma et al. 2022), CsEAE achieves improvements on the Rams dataset, with increases of 2.2% and 2.1%, respectively. On the WikiEvents dataset, CsEAE shows improvements of 2.0% in Arg-I and 2.3% in Arg-C metrics. Similarly, on the MLEE dataset, CsEAE achieves improvements of 3.0% in Arg-I and 3.2% in Arg-C metrics. The consistent improvements of 2% or more across all datasets demonstrate the effectiveness of the structure- and co-occurrences-aware modules in document-level EAE tasks.

[3]The versions of model we use are: gpt-3.5-turbo-0125

[4]https://huggingface.co/meta-llama/Meta-Llama-3-8B

[5]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

Model	PLM	Rams		WikiEvents	
		Arg-I	Arg-C	Arg-I	Arg-C
TagPrime-C*	RoBERTa	54.4	48.3	68.6	64.0
TagPrime-CR*	RoBERTa	54.1	49.7	68.4	65.5
EEQA*	BART	48.9	44.7	48.4	46.1
BART-Gen*	BART	50.4	45.4	68.1	63.9
X-Gear*	BART	52.1	46.2	55.4	52.4
Ampere*	BART	52.0	46.8	56.2	53.3
PAIE	BART	56.4	51.9	68.5	64.5
CsEAE	BART	<u>56.8</u>	<u>52.3</u>	<u>69.3</u>	<u>65.7</u>

Table 2: All experiments in the table above used the data processing methods described in TextEE, and the results are averaged over five data splits. * means the value from the TextEE's paper. All experiments utilized a large-scale PLM.

Model	WikiEvents		Rams		MLEE	
	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
In-Context Learning (ICL)						
GPT-3.5	18.12	16.04	34.30	27.64	21.16	15.46
GPT4o-mini	20.42	17.99	35.47	30.04	25,85	22.34
GPT4o	25.58	23.37	41.58	35.70	28.04	24.92
Llama3	10.34	9.50	23.05	18.79	0.07	0.07
Llama3-Instruct	0.00	0.00	0.00	0.00	0.00	0.00
Supervised Fine-tuning						
Llama3	65.82	60.68	37.00	33.26	72.63	71.09
Llama3-Instruct	65.88	60.54	55.06	49.82	70.85	69.76
CsLLMs	66.33	62.80	55.35	50.25	74.80	73.87
Multiple Datasets Supervised Fine-tuning						
Doc	66.73	62.99	55.76	50.74	73.35	71.96
CsLLMs (Doc)	69.92	65.66	56.14	50.99	75.34	74.10
Multiple Datasets Supervised Fine-tuning using additional datasets						
News	69.12	63.70	56.12	51.32	-	-
News+MLEE	68.92	65.12	54.96	50.82	70.83	69.58
News+GENEVA	57.85	54.54	44.12	41.04	-	-
ALL	68.27	63.96	56.83	51.62	72.04	70.87
CsLLMs (ALL)	<u>70.89</u>	<u>66.53</u>	<u>57.19</u>	<u>51.84</u>	<u>75.93</u>	<u>74.89</u>

Table 3: Overall performance of LLMs. Doc represents training using the WikiEvents, Rams, and MLEE; News represents training using the ACE, Rams, and WikiEvents, ALL signifies that all five datasets were used for training. CsLLMs refers to the fine-tuning process that incorporates prompts enhanced with co-occurrences- and structure-aware interactions. During the multiple dataset SFT, we used Llama3-Instruct as the LLMs.

We also utilized the data preprocessing method provided by TextEE, dividing the dataset into five subsets while allowing for multi-word triggers, accounting for overlapping argument spans, and retaining all instances without filtering. The final results, shown in the Table 2, represent the average performance across these five splits. Even under such stringent conditions, CsEAE consistently outperforms all baselines, demonstrating its superior effectiveness.

LLMs As shown in the Table 3, in the ICL setting, the Open-AI series models demonstrated superior performance compared to the Open-resource models. Notably, instruct-type models have shown relatively poor performance during ICL. However, after fine-tuning, they outperformed base models on some datasets.

After SFT, the extraction capabilities of the LLMs improved significantly. Further improvements were observed when the model was fine-tuned using multiple datasets, demonstrating that the LLMs robust memory capacity can handle diverse datasets simultaneously and learn common extraction-enhancing abilities from them. Additionally, after incorporating two extra sentence-level datasets for enhanced training, the model achieved better performance.

Moreover, incorporating co-occurrences- and structure-aware interactions into the prompts led to additional performance gains compared to models fine-tuned on single datasets without such enhancements. This indicates that beneficial extraction-related insights identified in SLMs is also applicable and effective in LLMs.

We attribute the lower performance of CsLLMs (ALL) on Rams compared to CsEAE to the incomplete integration of structure-aware elements in the prompt. While structure-aware interaction has been proven to be the most effective

GENEVA		
Model	Arg-I	Arg-C
In-Context Learning (ICL)		
GPT-3.5	33.07	27.97
GPT4o-mini	35.17	31.06
GPT4o	42.98	39.55
Llama3	4.70	3.61
Llama3-Instruct	0.35	0.29
Supervised Fine-tuning		
Llama3	28.98	27.88
Llama3-Instruct	66.07	62.42
News+GENEVA	64.22	61.06
ALL	63.91	61.03
CsLLMs (ALL)	67.99	64.71

Table 4: Overall performance of LLMs on GENEVA.

module for improving Rams performance in CsEAE (analysis on ablation studies), but we are unable to fully constrain the model's focus through the prompt alone.

To analyze the generalization challenges of LLMs in broader domains and their applicability in real-world scenarios, we conducted extensive experiments on the GENEVA dataset, which includes 115 event types and 220 distinct roles across general-domain, sentence-level data. The experimental results are presented in the table 4. Surprisingly, unlike in domain-specific document-level datasets, multiple datasets SFT does not enhance model performance on GENEVA. However, incorporating co-occurrences- and structure-aware interactions into the prompt improves the model's performance on document-level datasets, allowing for better extraction on GENEVA. This indicates that the model learns to capture co-occurrences- and structure-aware information from the three document-level datasets, such that, even though sentence-level datasets cannot directly embed structure-aware information in prompt construction, the model can leverage what it learned from document-level data to assist in extraction. Additionally, it becomes evident that LLMs do not perform well on general-domain datasets like GENEVA. Its best performance, an Arg-C score of 64.71, falls short compared to best results of SLMs (Huang et al. 2024). We attribute this to the fact that many event types in GENEVA are quite similar, and fine-tuning an 8B-parameter model using prompt + LoRA struggles to discern numerous labels and their subtle interactions during extraction (Ma et al. 2023).

Analysis

Ablation Studies

The Table 5 show that even a single type of interaction can enhance the model's performance across all datasets, with each interaction type providing varying levels of improvement. The structure-aware module significantly improves performance on the Rams dataset, increasing the Arg-C metric by 1%. Conversely, the co-occurrence-aware module significantly boosts performance on the WikiEvents and MLEE datasets, increasing the Arg-C metric by 1.7% and 2.8%, respectively. We analyzed that the significant improvement in Rams by structure-aware module is due to its stable sentence structure, where each document consists of five sentences, allowing the model to learn more consistent structural information. The notable improvement of the co-occurrences-aware module on the WikiEvents and MLEE datasets is attributed to the higher number of events in instances, where the auxiliary information provided by the co-occurrences-aware module leads to a greater performance boost in complex event scenarios. CsEAE not only retains the benefits of individual interaction features but also integrates multiple types of interaction without causing interference.

Capturing the Event Semantic Boundary

Following TabEAE, we analyzed CsEAE's ability to capture event semantic boundaries on the WikiEvents and MLEE datasets from two perspectives: inter-event semantics and intra-event semantics.

Inter-event semantics We divided the both datasets based on the overlap, where overlap indicates instances where different events use the same token span as arguments, and N_O denotes instances without event overlap. As observed from the Table 6, CsEAE achieved overall improvements across all metrics on both datasets and performed particularly well in handling instances with overlap.

Inner-event semantics We divided the roles in the both datasets based on their distance from the trigger. Specifically, we defined the argument distance as the value obtained by subtracting the index of the argument's head word from the index of its corresponding trigger's head word. Since the model predicts all arguments corresponding to a role at once, we defined the distance between a role and the trigger, $\mathcal{D}$, as the maximum argument distance among all arguments for that role. As shown in the Figure 4, where negative values indicate the argument is to the left of the trigger and positive values indicate the argument is to the right. The results show that CsEAE achieved the best performance across multiple ranges on both datasets and demonstrated a trend where the improvement increased with greater distances.

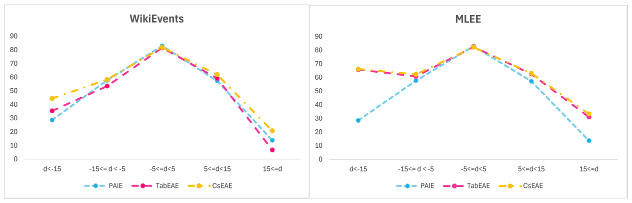


Figure 4: The performance of different EAE models in extracting arguments at different distances from the triggers. We only measure the Arg-C F1 metric.

Structure-aware Interaction for Document

To analyze the effectiveness of the model in performing extraction centered around the sentence containing the trigger word, we conducted an analysis on RAMS, which has

Model	Rams		WikiEvents		MLEE	
	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
w/o str&occur	55.3	51.0	68.9	64.2	71.3	70.1
add str	55.8	52.0	70.5	64.8	72.0	70.9
add occur	55.9	51.6	70.5	65.9	73.9	72.9
CsEAE	<u>57.5</u>	<u>53.1</u>	<u>70.9</u>	<u>66.5</u>	<u>74.3</u>	<u>73.3</u>

Table 5: Ablation study on all benchmarks, str: structure-aware interaction, occur: co-occurrences-aware interaction.
aai25

Model	WikiEvents				MLEE			
	N_O (296)		Overlap (69)		N_O (734)		Overlap (1460)	
	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
TabEAE	70.7	65.4	66.1	63.0	78.0	77.0	68.9	67.6
PAIE	68.8	63.9	68.9	65.0	76.8	75.7	64.8	63.4
CsEAE	<u>71.0</u>	<u>66.0</u>	<u>70.6</u>	<u>68.4</u>	<u>78.7</u>	<u>77.8</u>	<u>69.0</u>	<u>67.9</u>

Table 6: The performance in extracting the arguments of overlapping events. The numbers in parentheses represent the quantity of the corresponding data type within the dataset.

the highest number of cross-sentence arguments. We defined the distance D between a role and the trigger as the maximum argument distance among all arguments for that role. When the trigger and the maximum argument are in the same sentence, D=0; when they are not, D≠0. In the Table 7, CsEAE achieved a 3.23% improvement in the Arg-C metric compared to PAIE when D=0. This improvement significantly contributed to CsEAE's overall lead over PAIE in all datasets. The substantial improvement at D=0 also demonstrates that the model's approach of centering the document structure around the trigger's sentence effectively helps focus attention on the core content of the sentence, reducing the distraction from redundant information.

Model	Rams (Arg-C F1)		
	D=0	D≠0	Overall
PAIE	58.7	35.3	51.0
TabEAE	61.2	31.8	51.4
CsEAE	<u>61.9</u>	<u>35.5</u>	<u>53.1</u>

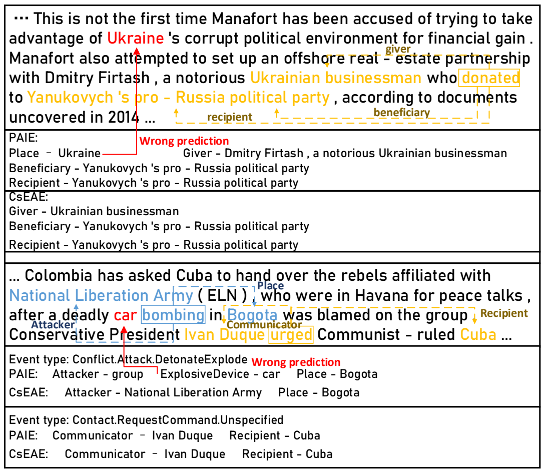
Table 7: Model performance on cross-sentence arguments.

Case Study

In the first case in the Figure 5, PAIE incorrectly predicts *Ukraine* from the previous sentence as the argument for role *Place*, while CsEAE avoids this interference. In the second example, PAIE incorrectly identifies *car* as the argument for *ExplosiveDevice*, whereas CsEAE, by incorporating more event information, avoids this mistake.

Conclusion

We proposed CsEAE, which aids the model in capturing semantic boundaries between events by incorporating interaction of all co-occurring events within the input. Additionally, it establishes the structural relationships within the input to reduce the distraction caused by redundant information. Moreover, we addressed the gap in previous research


··· This is not the first time Manafort has been accused of trying to take advantage of Ukraine 's corrupt political environment for financial gain . Manafort also attempted to set up an offshore real - estate partnership with Dmitry Firtash , a notorious Ukrainian businessman who donated to Yanukovych 's pro - Russia political party , according to documents uncovered in 2014 ...

PAIE:
Place - Ukraine
Giver - Dmitry Firtash , a notorious Ukrainian businessman
Beneficiary - Yanukovych 's pro - Russia political party
Recipient - Yanukovych 's pro - Russia political party

CsEAE:
Giver - Ukrainian businessman
Beneficiary - Yanukovych 's pro - Russia political party
Recipient - Yanukovych 's pro - Russia political party

... Colombia has asked Cuba to hand over the rebels affiliated with National Liberation Army (ELN) , who were in Havana for peace talks , after a deadly car bombing in Bogota was blamed on the group . Conservative President Ivan Duque urged Communist - ruled Cuba ...

Event type: Conflict.Attack.DetonateExplode
PAIE: Attacker - group ExplosiveDevice - car Place - Bogota
CsEAE: Attacker - National Liberation Army Place - Bogota

Event type: Contact.RequestCommand.Unspecified
PAIE: Communicator - Ivan Duque Recipient - Cuba
CsEAE: Communicator - Ivan Duque Recipient - Cuba

Figure 5: Two test cases from Rams and WikiEvents.

regarding fine-tuning LLMs in the EAE domain, achieving further improvements through multiple dataset fine-tuning. Lastly, we intend to apply the reliable insights developed in CsEAE to LLMs, offering a new perspective: reliable insights validated on SLMs are also applicable to LLMs.

Acknowledgement

The article is supported by the "Tianshan Talent" Research Project of Xinjiang (No.2022TSYCLJ0037). The National Natural Science Foundation of China (No.62262065). The Science and Technology Program of Xinjiang (No.2022B01008).

Thanks to Professor Rui Yan (Renmin University) for the support and assistance, and to Haokun Geng and Guanghan Li (Xinjiang University) for their support.

Thanks to all the reviewers for their valuable feedback and to all the contributors who continuously advance the development of open-source projects in the EE community.

References

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-End Object Detection with Transformers. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 213–229. Cham: Springer International Publishing. ISBN 978-3-030-58452-8.

Chen, R.; Qin, C.; Jiang, W.; and Choi, D. 2024. Is a Large Language Model a Good Annotator for Event Extraction? In Wooldridge, M. J.; Dy, J. G.; and Natarajan, S., eds., *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 17772–17780. AAAI Press.

Costa, T. S.; Gottschalk, S.; and Demidova, E. 2020. Event-QA: A Dataset for Event-Centric Question Answering over Knowledge Graphs. In d'Aquin, M.; Dietze, S.; Hauff, C.; Curry, E.; and Cudré-Mauroux, P., eds., *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, 3157–3164. ACM.

Doddington, G. R.; Mitchell, A.; Przybocki, M. A.; Ramshaw, L. A.; Strassel, S. M.; and Weischedel, R. M. 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, May 26-28, 2004, Lisbon, Portugal*. European Language Resources Association.

Du, X.; and Cardie, C. 2020a. Event Extraction by Answering (Almost) Natural Questions. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 671–683. Association for Computational Linguistics.

Du, X.; and Cardie, C. 2020b. Event Extraction by Answering (Almost) Natural Questions. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 671–683. Association for Computational Linguistics.

Ebner, S.; Xia, P.; Culkin, R.; Rawlins, K.; and Durme, B. V. 2020. Multi-Sentence Argument Linking. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 8057–8077. Association for Computational Linguistics.

Gunasekar, S.; Zhang, Y.; Aneja, J.; Mendes, C. C. T.; Del Giorno, A.; Gopi, S.; Javaheripi, M.; Kauffmann, P.; de Rosa, G.; Saarikivi, O.; et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

He, Y.; Hu, J.; and Tang, B. 2023. Revisiting Event Argument Extraction: Can EAE Models Learn Better When Being Aware of Event Co-occurrences? In Rogers, A.; Boyd-Graber, J. L.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, 12542–12556. Association for Computational Linguistics.

Hsu, I.; Huang, K.; Boschee, E.; Miller, S.; Natarajan, P.; Chang, K.; and Peng, N. 2022. DEGREE: A Data-Efficient Generation-Based Event Extraction Model. In Carpuat, M.; de Marneffe, M.; and Ruíz, I. V. M., eds., *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, 1890–1908. Association for Computational Linguistics.

Hsu, I.; Huang, K.; Zhang, S.; Cheng, W.; Natarajan, P.; Chang, K.; and Peng, N. 2023a. TAGPRIME: A Unified Framework for Relational Structure Extraction. In Rogers, A.; Boyd-Graber, J. L.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, 12917–12932. Association for Computational Linguistics.

Hsu, I.; Xie, Z.; Huang, K.; Natarajan, P.; and Peng, N. 2023b. AMPERE: AMR-Aware Prefix for Generation-Based Event Argument Extraction Model. In Rogers, A.; Boyd-Graber, J. L.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, 10976–10993. Association for Computational Linguistics.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Huang, K.; Hsu, I.; Natarajan, P.; Chang, K.; and Peng, N. 2022. Multilingual Generative Language Models for Zero-Shot Cross-Lingual Event Argument Extraction. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 4633–4646. Association for Computational Linguistics.

Huang, K.; Hsu, I.; Parekh, T.; Xie, Z.; Zhang, Z.; Natarajan, P.; Chang, K.; Peng, N.; and Ji, H. 2024. TextEE: Benchmark, Reevaluation, Reflections, and Future Challenges in Event Extraction. In Ku, L.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, 12804–12825. Association for Computational Linguistics.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2): 83–97.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the*

Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, 7871–7880. Association for Computational Linguistics.

Li, M.; Zareian, A.; Lin, Y.; Pan, X.; Whitehead, S.; Chen, B.; Wu, B.; Ji, H.; Chang, S.; Voss, C. R.; Napierski, D.; and Freedman, M. 2020. GAIA: A Fine-grained Multimedia Knowledge Extraction System. In Celikyilmaz, A.; and Wen, T., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, 77–86. Association for Computational Linguistics.

Li, S.; Ji, H.; and Han, J. 2021. Document-Level Event Argument Extraction by Conditional Generation. In Toutanova, K.; Rumshisky, A.; Zettlemoyer, L.; Hakkani-Tür, D.; Beltagy, I.; Bethard, S.; Cotterell, R.; Chakraborty, T.; and Zhou, Y., eds., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, 894–908. Association for Computational Linguistics.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, 4582–4597. Association for Computational Linguistics.

Ma, M. D.; Wang, X.; Kung, P.; Brantingham, P. J.; Peng, N.; and Wang, W. 2024. STAR: Boosting Low-Resource Information Extraction by Structure-to-Text Data Generation with Large Language Models. In Wooldridge, M. J.; Dy, J. G.; and Natarajan, S., eds., *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 18751–18759. AAAI Press.

Ma, Y.; Cao, Y.; Hong, Y.; and Sun, A. 2023. Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples! In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, 10572–10601. Association for Computational Linguistics.

Ma, Y.; Wang, Z.; Cao, Y.; Li, M.; Chen, M.; Wang, K.; and Shao, J. 2022. Prompt for Extraction? PAIE: Prompting Argument Interaction for Event Argument Extraction. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 6759–6774. Association for Computational Linguistics.

Parekh, T.; Hsu, I.-H.; Huang, K.-H.; Chang, K.-W.; and Peng, N. 2023. GENEVA: Benchmarking Generalizability for Event Argument Extraction with Hundreds of Event Types and Argument Roles. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3664–3686.

Peng, J.; Yang, W.; Wei, F.; and He, L. 2024. Prompt for extraction: Multiple templates choice model for event extraction. *Knowledge-Based Systems*, 289: 111544.

Pyysalo, S.; Ohta, T.; Miwa, M.; Cho, H.; Tsujii, J.; and Ananiadou, S. 2012. Event extraction across multiple levels of biological organization. *Bioinform.*, 28(18): 575–581.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Trieu, H.; Tran, T. T.; Nguyen, A. D.; Nguyen, A.; Miwa, M.; and Ananiadou, S. 2020. DeepEventMine: end-to-end neural nested event extraction from biomedical texts. *Bioinform.*, 36(19): 4910–4917.

Xu, R.; Wang, P.; Liu, T.; Zeng, S.; Chang, B.; and Sui, Z. 2022. A Two-Stream AMR-enhanced Model for Document-level Event Argument Extraction. In Carpuat, M.; de Marneffe, M.; and Ruíz, I. V. M., eds., *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, 5025–5036. Association for Computational Linguistics.

Yang, H.; Sui, D.; Chen, Y.; Liu, K.; Zhao, J.; and Wang, T. 2021. Document-level Event Extraction via Parallel Prediction Networks. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, 6298–6308. Association for Computational Linguistics.

Yang, S.; Zhao, H.; Zhu, S.; Zhou, G.; Xu, H.; Jia, Y.; and Zan, H. 2024. Zhongjing: Enhancing the Chinese Medical Capabilities of Large Language Model through Expert Feedback and Real-World Multi-Turn Dialogue. In Wooldridge, M. J.; Dy, J. G.; and Natarajan, S., eds., *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 19368–19376. AAAI Press.

Yang, Y.; Guo, Q.; Hu, X.; Zhang, Y.; Qiu, X.; and Zhang, Z. 2023. An AMR-based Link Prediction Approach for Document-level Event Argument Extraction. In Rogers, A.; Boyd-Graber, J. L.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, 12876–12889. Association for Computational Linguistics.

Zhang, T.; Chen, M.; and Bui, A. A. T. 2020. Diagnostic Prediction with Sequence-of-sets Representation Learning for Clinical Events. In Michalowski, M.; and Moskovitch, R., eds., *Artificial Intelligence in Medicine - 18th International Conference on Artificial Intelligence in Medicine, AIME 2020, Minneapolis, MN, USA, August 25-28, 2020, Proceedings*, volume 12299 of *Lecture Notes in Computer Science*, 348–358. Springer.

Zhou, H.; Qian, J.; Feng, Z.; Lu, H.; Zhu, Z.; and Mao, K. 2023. Heuristics-driven link-of-analogy prompting: Enhancing large language models for document-level event argument extraction. *arXiv preprint arXiv:2311.06555*.

Reproducibility Checklist

This paper:

1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes)
2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes)
3. Provides well marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes)

Does this paper make theoretical contributions? (yes)

If yes, please complete the list below.

1. All assumptions and restrictions are stated clearly and formally. (yes)
2. All novel claims are stated formally (e.g., in theorem statements). (yes)
3. Proofs of all novel claims are included. (yes)
4. Proof sketches or intuitions are given for complex and/or novel results. (yes)
5. Appropriate citations to theoretical tools used are given. (yes)
6. All theoretical claims are demonstrated empirically to hold. (yes)
7. All experimental code used to eliminate or disprove claims is included. (yes)

Does this paper rely on one or more datasets? (yes)

If yes, please complete the list below.

1. A motivation is given for why the experiments are conducted on the selected datasets (yes)
2. All novel datasets introduced in this paper are included in a data appendix. (yes)
3. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (NA)
4. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations. (yes)
5. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available. (yes)
6. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing. (NA)

Does this paper include computational experiments? (yes)

If yes, please complete the list below.

1. Any code required for pre-processing data is included in the appendix. (yes)
2. All source code required for conducting and analyzing the experiments is included in a code appendix. (yes)
3. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (yes)
4. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes)
5. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. (NA)
6. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks. (yes)
7. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. (yes)
8. This paper states the number of algorithm runs used to compute each reported result. (yes)
9. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information. (yes)
10. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank). (yes)
11. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments. (yes)
12. This paper states the number and range of values tried per (hyper-)parameter during development of the paper, along with the criterion used for selecting the final parameter setting. (yes)