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

Anonymous ACL submission

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 dialogue systems (Zhang et al., 2020) and recommendation systems (Han et al., 2025).

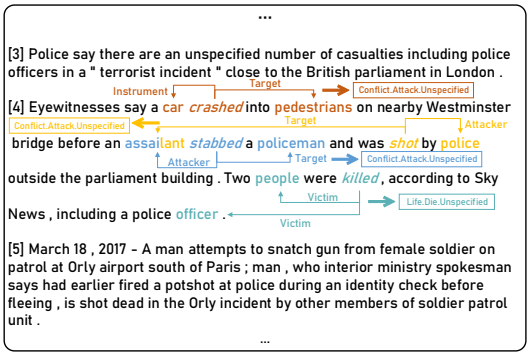

Figure 1: An EAE instance from the WikiEvents dataset.

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 et al., 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 et al., 2023; Liu et al., 2024).

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 sentence 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-occurring 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.

2 CsEAE Model

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

2.1 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:

$$\begin{aligned} H_{\mathcal{D}}^{enc} &= Encoder_{Sap}(\mathcal{D}), \\ H_{\mathcal{D}} &= Decoder_{Cap}(H_{\mathcal{D}}^{enc}, H_{\mathcal{D}}^{enc}). \end{aligned} \tag{1}$$

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:

$$H_{pt} = Decoder_{Sap}(H_{\mathcal{D}}^{enc}, p_{e_n}). \tag{2}$$

2.2 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.

2.2.1 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.

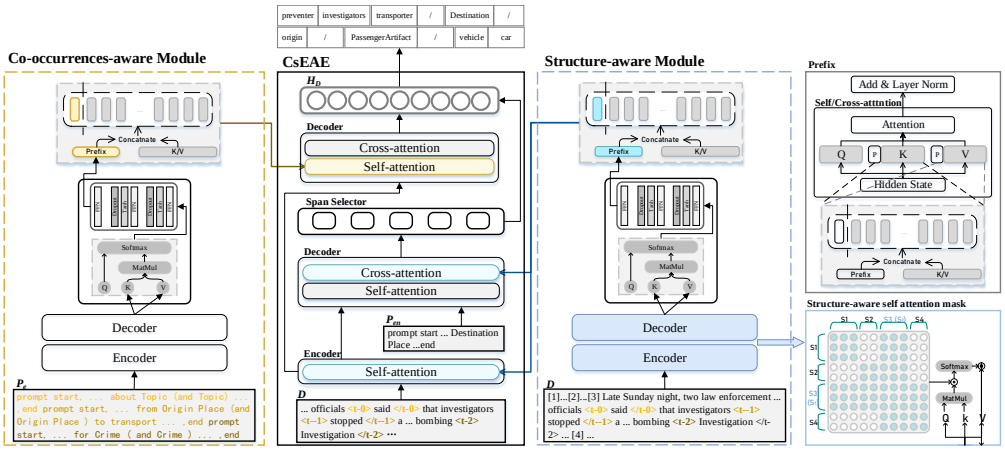

Figure 2: Overview of CsEAE. The yellow attention represents the concatenation of co-occurrences-aware module, while the blue attention represents the concatenation of structure-aware module.

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.

2.2.2 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 et al., 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.

2.2.3 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.

2.3 Structure-aware Module

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

2.3.1 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:

$$W_S = Decoder(Encoder(\mathcal{D}, M_s)). \quad (3)$$

2.3.2 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.

2.4 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 $[p_k^{(start)}; p_k^{(end)}]$.

$$\begin{aligned}
\psi_k^{(start)} &= \psi_k \circ w^{(start)} \in R^h, \\
\psi_k^{(end)} &= \psi_k \circ w^{(end)} \in R^h, \\
\text{logit}_k^{(start)} &= \psi_k^{(start)} H_{\mathcal{D}} \in R^L, \\
\text{logit}_k^{(end)} &= \psi_k^{(end)} H_{\mathcal{D}} \in R^L, \\
p_k^{(start)} &= \text{Softmax}(\text{logit}_k^{(start)}) \in R^L, \\
p_k^{(end)} &= \text{Softmax}(\text{logit}_k^{(end)}) \in R^L.
\end{aligned} \quad (4)$$

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:

$$\begin{aligned}
\mathcal{L}_k(\mathcal{D}) &= -(\log p_k^{(start)}(s_k) + \log p_k^{(end)}(e_k)), \\
\mathcal{L} &= \sum_{\mathcal{D} \in B} \sum_k \mathcal{L}_k(\mathcal{D}).
\end{aligned} \quad (5)$$

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:

$$\text{score}_k(i, j) = \text{logit}_k^{(start)}(i) + \text{logit}_k^{(end)}(j). \quad (6)$$

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

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).

3 Generalization in LLMs

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

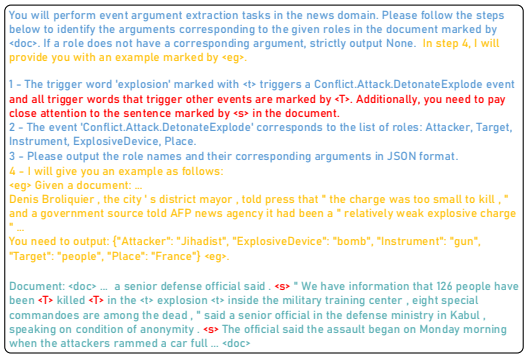

Figure 3: Prompt for LLMs on WikiEvents. The blue parts represent $\mathcal{I}$, the yellow parts represent $\mathcal{E}$, the green parts represent $\mathcal{Q}$ and the red parts represent co-occurrences- and structure-aware interactions.

3.1 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:

$$\mathcal{P}_{\mathcal{L}}(\mathcal{D}) = [\mathcal{I}; \mathcal{E}; \mathcal{Q}]. \quad (7)$$

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.

3.2 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.

3.3 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?

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.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

We used the three most commonly employed datasets for document-level event argument extraction (EAE): Rams (Ebner et al., 2020), WikiEvents (Li et al., 2021), and MLEE (Pyysalo et al., 2012). We preprocessed the data following previous methods (Trieu et al., 2020; Ma et al., 2022; He et al., 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). We leave the dataset details in Appendix B.

4.1.2 Implementation Details

Please refer to Appendix C for details.

4.1.3 Evaluation Metrics

Fllowed by previous works (Ma et al., 2022; He et al., 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 all experiments in this paper, Arg-I and Arg-C is equivalent to Arg-I+ and Arg-C+ as defined in TextEE. More details in Appendix D

4.1.4 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); and (2) Generation-based models: Bart-Gen (Li et al., 2021), PAIE (Ma et al., 2022), TabEAE (He et al., 2023), DEEIA (Liu et al., 2024). For LLMs, we categorized the baseline models into two groups too: (1) Open-AI: Chat-GPT [1], GPT-4o, GPT-4o-mini; and (2) Open-source: Llama3-8B (Touvron et al., 2023), Llama3-8B-Instruct [2]. More details are provided in the Appendix E.

4.2 Main Results

4.2.1 CsEAE

We evaluate the proposed model CsEAE and baseline methods under all benchmarks. In the Table 1,

[1]The versions of model we use are: gpt-3.5-turbo-0125
[2]https://huggingface.co/meta-llama

Model	Rams		WikiEvents		MLEE	
	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
EEQA*	51.9	47.5	60.4	57.2	70.3	68.7
TSAR*	57.0	52.1	71.1	65.8	72.6	71.5
BART-Gen*	51.2	47.1	66.8	62.4	71.0	69.8
DEEIA	55.9	51.3	69.7	64.5	73.5	72.5
TabEAE-m2s	56.2	51.4	69.7	64.9	-	-
TabEAE-m2m	55.9	50.9	70.3	64.6	74.0	72.9
PAIE	55.3	51.0	68.9	64.2	71.3	70.1
CsEAE	57.5	53.1	70.9	66.5	74.3	73.3

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.

Model	Rams		WikiEvents	
	Arg-I	Arg-C	Arg-I	Arg-C
TagPrime-C*	54.4	48.3	68.6	64.0
TagPrime-CR*	54.1	49.7	68.4	65.5
EEQA*	48.9	44.7	48.4	46.1
BART-Gen*	50.4	45.4	68.1	63.9
PAIE	56.4	51.9	68.5	64.5
CsEAE	56.8	52.3	69.3	65.7

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.

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.

We also utilized the data preprocessing method provided by TextEE. 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. In the Appendix G, we provide a detailed performance breakdown of PAIE and CSEAE across five splits.

4.2.2 CsLLMs

We present the performance of various models under the ICL setting in the Appendix F.

As shown in the Table 3, 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 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. In the Appendix I, we present more detailed experiments, including generalization experiments for LLMs and analysis experiments on the LLMs.

5 Analysis

5.1 Ablation Studies

The Table 4 show that even a single type of interaction can enhance the model's performance across all datasets, with each interaction type providing

Model	WikiEvents		Rams		MLEE	
	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
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)	70.89	66.53	57.19	51.84	75.93	74.89

Table 3: Overall performance of LLMs. **Doc** represents training using the WikiEvents, Rams, and MLEE (since they are all document-level datasets); **News** represents training using the ACE, Rams, and WikiEvents (since they are all datasets from the news domain). **ALL** signifies that all five datasets were used for training. During the multiple dataset SFT, we used Llama3-Instruct as the LLMs.

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	57.5	53.1	70.9	66.5	74.3	73.3

Table 4: Ablation study on all benchmarks, str: structure-aware interaction, occur: co-occurrences-aware interaction.

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	71.0	66.0	70.6	68.4	78.7	77.8	69.0	67.9

Table 5: The performance in extracting the arguments of overlapping events. The numbers in parentheses represent the quantity of the corresponding data type within the dataset.

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.

5.2 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 and intra-event semantics. We also analyze the performance of the models when handling multiple event inputs in the Appendix H.

5.2.1 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 5, CsEAE achieved overall improvements across all metrics on both datasets and performed particularly well in handling instances with overlap.

5.2.2 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.

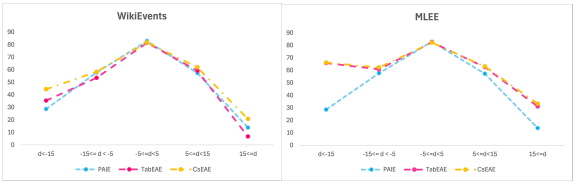

Figure 4: Performance of different models in extracting arguments at different distances from the triggers.

5.3 Structure-aware Interaction

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 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 6, 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 6: Performance on cross-sentence arguments.

5.4 Case Study

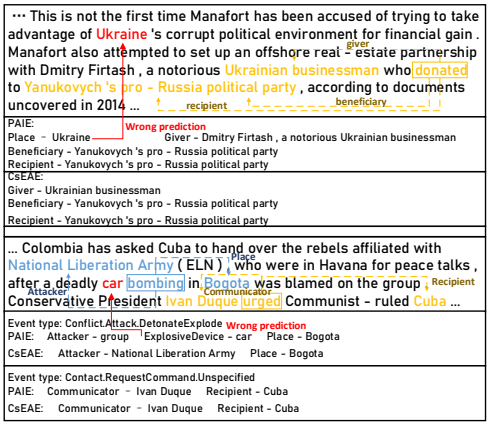

Figure 5: Two test cases from Rams and WikiEvents.

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.

6 Conclusion

We proposed CsEAE, enhancing event semantic boundary detection via co-occurring events interaction and structural relationships to reduce redundancy. Additionally, we fine-tuned LLMs on multiple datasets, bridging EAE gaps. Lastly, we offer a new perspective: reliable insights validated on SLMs are also applicable to LLMs.

7 Ethics

We use a generative approach for argument extraction, which may occasionally produce offensive content during the extraction process (though the probability is very low and did not occur in our experiments). Therefore, we recommend that users thoroughly review the generated content before practical application.

8 Limitation

For the generation of LLMs, we have only designed basic prompts, and there are many methods that can enhance the information contained in the prompts to improve model performance, such as example selection, among others.

Furthermore, the powerful memory of LLMs may allow us to move beyond focusing solely on EAE as a task and explore the potential for LLMs to handle most tasks in the information extraction domain.

References

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020*, pages 213–229, Cham. Springer International Publishing.

Ruirui Chen, Chengwei Qin, Weifeng Jiang, and Dongkyu Choi. 2024. Is a large language model a good annotator for event extraction? In *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*, pages 17772–17780. AAAI Press.

George R. Doddington, Alexis Mitchell, Mark A. Przybocki, Lance A. Ramshaw, Stephanie M. Strassel, and Ralph M. Weischedel. 2004. The automatic content extraction (ACE) program - tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal*. European Language Resources Association.

Xinya Du and Claire Cardie. 2020a. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 671–683. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020b. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 671–683. Association for Computational Linguistics.

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, ACL 2020, Online, July 5-10, 2020*, pages 8057–8077. Association for Computational Linguistics.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Xiaofeng Han, Xiangwu Meng, and Yujie Zhang. 2025. Exploiting multiple influence pattern of event organizer for event recommendation. *Information Processing & Management*, 62(2):103966.

Yuxin He, Jingyue Hu, and Buzhou Tang. 2023. Revisiting event argument extraction: Can EAE models learn better when being aware of event co-occurrences? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12542–12556. 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, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1890–1908. Association for Computational Linguistics.

I-Hung Hsu, Kuan-Hao Huang, Shuning Zhang, Wenxin Cheng, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2023a. TAGPRIME: A unified framework for relational structure extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12917–12932. Association for Computational Linguistics.

I-Hung Hsu, Zhiyu Xie, Kuan-Hao Huang, Prem Natarajan, and Nanyun Peng. 2023b. AMPERE: amr-aware prefix for generation-based event argument extraction model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10976–10993. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang,

and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Kuan-Hao Huang, I-Hung Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, and Heng Ji. 2024. Textee: Benchmark, reevaluation, reflections, and future challenges in event extraction. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 12804–12825. Association for Computational Linguistics.

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

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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. 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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 894–908. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *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*, pages 4582–4597. Association for Computational Linguistics.

Wanlong Liu, Li Zhou, Dingyi Zeng, Yichen Xiao, Shaohuan Cheng, Chen Zhang, Grandee Lee, Malu Zhang, and Wenyu Chen. 2024. Beyond single-event extraction: Towards efficient document-level multi-event argument extraction. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 9470–9487. Association for Computational Linguistics.

Mingyu Derek Ma, Xiaoxuan Wang, Po-Nien Kung, P. Jeffrey Brantingham, Nanyun Peng, and Wei Wang. 2024. STAR: boosting low-resource information extraction by structure-to-text data generation with large language models. In *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*, pages 18751–18759. AAAI Press.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 10572–10601. Association for Computational Linguistics.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6759–6774. Association for Computational Linguistics.

Tanmay Parekh, I-Hung Hsu, Kuan-Hao Huang, Kai-Wei Chang, and Nanyun Peng. 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)*, pages 3664–3686.

Jiaren Peng, Wenzhong Yang, Fuyuan Wei, and Liang He. 2024. Prompt for extraction: Multiple templates choice model for event extraction. *Knowledge-Based Systems*, 289:111544.

Sampo Pyysalo, Tomoko Ohta, Makoto Miwa, Han-Cheol Cho, Junichi Tsujii, and Sophia Ananiadou. 2012. Event extraction across multiple levels of biological organization. *Bioinform.*, 28(18):575–581.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hai-Long Trieu, Thy Thy Tran, Anh-Khoa Duong Nguyen, Anh Nguyen, Makoto Miwa, and Sophia Ananiadou. 2020. Deepeventmine: end-to-end neural nested event extraction from biomedical texts. *Bioinform.*, 36(19):4910–4917.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5783–5788. 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, NAACL 2022, Seattle, WA,*

United States, July 10-15, 2022, pages 5025–5036. Association for Computational Linguistics.

Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. 2021. Document-level event extraction via parallel prediction networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6298–6308. Association for Computational Linguistics.

Songhua Yang, Hanjie Zhao, Senbin Zhu, Guangyu Zhou, Hongfei Xu, Yuxiang Jia, and Hongying Zan. 2024. Zhongjing: Enhancing the chinese medical capabilities of large language model through expert feedback and real-world multi-turn dialogue. In *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*, pages 19368–19376. AAAI Press.

Yuqing Yang, Qipeng Guo, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2023. An amr-based link prediction approach for document-level event argument extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12876–12889. Association for Computational Linguistics.

Tianran Zhang, Muhao Chen, and Alex A. T. Bui. 2020. Diagnostic prediction with sequence-of-sets representation learning for clinical events. In *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*, pages 348–358. Springer.

Hanzhang Zhou, Junlang Qian, Zijian Feng, Hui Lu, Zixiao Zhu, and Kezhi Mao. 2023. Heuristics-driven link-of-analogy prompting: Enhancing large language models for document-level event argument extraction. *arXiv preprint arXiv:2311.06555*.

A Related Works

A.1 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 et al., 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 et al., 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.

A.2 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.

B Datasets

We used three of the most commonly used datasets for document-level Event Argument Extraction (EAE), namely Rams (Ebner et al., 2020), WikiEvents (Li et al., 2021), and MLEE (Pyysalo et al., 2012). All three datasets are in English, and we followed previous methods to preprocess the data (Ma et al., 2022; Li et al., 2021; He et al., 2023; Pyysalo et al., 2012; Xu et al., 2022; Yang et al., 2023).

B.1 Rams

This dataset contains 9124 annotated events from English online news articles, defining 39 event types and 65 roles. Each document data consists of five sentences and is commonly used in research in

the field of document-level EE/EAE. Since the original dataset is stored on an event-by-event basis, to accommodate the co-occurrence-aware, we merged all events appearing in the same document. Unlike previous preprocessing methods, to facilitate the structure-aware, we retained the 'sents' field, which records the sentence-level segmentation of the current document.

B.2 WikiEvents

It consists of events recorded in English Wikipedia along with news articles mentioning these events. It provides 246 document-level data, containing 50 event types and 59 predefined roles. It is commonly used in research on document-level event extraction. The number of sentences composing each document data varies.

B.3 MLEE

This dataset consists of abstracts from biomedical publications, defining 23 event types. Since the original dataset does not have a separate validation set, we followed previous work and used the training set as the validation set to prevent data leakage (He et al., 2023). The final results were evaluated on the test set. All performance metrics of the models on the MLEE dataset in this paper are based on their performance on the test set (He et al., 2023).

Detailed statistics of the above datasets are listed in Table 7.

Dataset	Rams	WikiEvents	MLEE
Event types	139	50	23
Args per event	2.33	1.40	1.29
Events per text	1.25	1.78	3.32
Events			
Train	7329	3241	4442
Dev	924	345	-
Test	871	365	2200

Table 7: The table above shows the basic information for the all datasets, where Args stands for Arguments.

We classified the data from the three document-level datasets based on the number of events occurring, as shown in the Figure 6. There are significantly more instances with multiple events in MLEE compared to WikiEvents and Rams.

Additionally, We used two sentence-level datasets for enhanced training: ACE(Doddington et al., 2004) and GENEVA (Parekh et al., 2023). ACE was chosen due to its extensive use in sentence-level event extraction, EAE, and event detection tasks, as well as its relevance to the news domain, which aligns with Rams and WikiEvents. GENEVA was selected because of its broad range of covered domains.

B.4 ACE

It is a widely used dataset in the field of information extraction, consisting of newswire, broadcast news and telephone conversations. The dataset includes three languages: Arabic, Chinese and English. We used the English part of the dataset. We preprocessed the data using the same data processing method as previous works (Wadden et al., 2019; Ma et al., 2022; Hsu et al., 2023b).

B.5 GENAVA

This dataset is a widely used dataset in the EAE field, consisting of English data from genaral fields. We preprocessed the data using the same data processing method as in previous work (Parekh et al., 2023; Ma et al., 2022; Hsu et al., 2023b).

C 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 [3] 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.

D Evaluation Metrics

Following the same evaluation metrics as in prior works (Li et al., 2021; Hsu et al., 2022; Ma et al., 2022; Yang et al., 2023; He et al., 2023; Xu et al., 2022) for all datasets, we used the Arg-I F1 score and Arg-C F1 score to evaluate the model's performance on Argument Identification and Argument Classification tasks, respectively.

We considered TP as true positives, FN as false negatives, and FP as false positives. Recall (R) can be calculated using TP / (TP + FP), and precision (P) can be calculated using TP / (TP + FN). The F1 score combines both recall and precision, defined as F1 = 2 * P * R / (P + R).

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

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

Table 8: The performance of various models under the ICL settings.

Model	WikiEvents									
	split1		split2		split3		split4		split5	
	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
PAIE	56.78	51.95	55.07	50.48	57.32	52.62	56.92	52.13	56.14	52.48
CsEAE	56.33	52.01	55.85	51.68	58.80	53.18	56.50	52.02	56.52	52.64
Model	Rams									
	split1		split2		split3		split4		split5	
	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
PAIE	71.47	67.37	70.31	64.94	64.46	61.14	65.87	62.76	70.61	66.72
CsEAE	71.38	67.98	69.09	64.31	67.10	63.28	67.99	65.41	70.94	67.45

Table 9: The performance of CsEAE and PAIE across all five splits on TextEE benchmark.

• Arg-I: an argument is correctly identified from event mention.

• Arg-C: an argument is correctly classified if its offset and the role's label both match the ground truth.

Since the Arg-C score reflects whether the model extracts the correct arguments and associates them with the appropriate roles to generate the correct structured events, the EAE task places more emphasis on the Arg-C F1 score.

E Baselines

We compared CsEAE with a series of strong baselines, which are categorized into classification-based models and generation-based models.

The classification-based models include:

• EEQA (Du and Cardie, 2020b): the model redefines the EE task as a question-answering task, extracting event parameters in an end-to-end manner.

• TSAR (Xu et al., 2022): the model utilizes the Two-Stream Abstract meaning Representation enhanced span-based event argument extraction model.

• TagPrime (Hsu et al., 2023a): the model is a sequence labeling model that enhances its suitability for extracting relational information under specific conditions by appending prompt words containing information about given conditions to the input text.

Generation-based models include:

• PAIE (Ma et al., 2022): the model utilizes a span selector for decoding and extracting arguments.

• TabEAE (He et al., 2023): the model extends the PAIE into a non-autoregressive generation framework. TabEAE(m2s) means the use of a Multi-Single Training-inference Scheme, TabEAE(m2m) means the use of a Multi-Multi Training-inference Scheme. In all analysis experiments, the TabEAE metric used on the WikiEvents and Rams datasets was the m2s model, while the metric used on the MLEE dataset was the m2m model.

• DEEIA (Liu et al., 2024) [4]: the model adopts a multi-event prompt mechanism that can simultaneously extract arguments from all events within a document.

[4]We used six seeds provided by the authors to train the model. The reported results for RAMS and MLEE are the average of the six models. However, due to the extremely poor performance of the model on the WikiEvents dataset when seed=22 (Arg-I=61.07, Arg-C=55.57), we excluded the model corresponding to this seed when calculating DEEIA's performance on WikiEvents. Instead, we used the average of the models corresponding to the remaining five seeds.

Model	MLEE							
	Arg-I				Arg-C			
	Event=1	Event=2	Event=3	Event>=4	Event=1	Event=2	Event=3	Event>=4
	175	312	342	1371	175	312	342	1371
PAIE	81.6	79.4	73.6	68.2	81.4	78.2	72.4	67.0
TabEAE	81.3	81.4	77.3	71.2	81.3	80.1	76.5	70.0
CsEAE	80.8	81.0	78.4	71.5	80.8	80.3	77.3	70.4

Table 10: The table above compares the performance of EAE models based on Small Language Models (SLMs) on event mentions with different numbers of events in the MLEE datasets. "Event=1" indicates that there is only one triggered event in the event mention, and the number below represents the quantity of such event mentions in the corresponding dataset's test set. "Event>=4" indicates that the event mention has four or more triggered events.

Model	MLEE							
	Arg-I				Arg-C			
	Event=1	Event=2	Event=3	Event>=4	Event=1	Event=2	Event=3	Event>=4
	175	312	342	1371	175	312	342	1371
Base	79.3	76.1	74.6	70.1	78.7	75.2	72.7	69.0
CsLLMs	81.4	78.6	78.3	73.0	81.4	77.1	77.2	72.0

Table 11: The table above compares the performance of EAE models based on LLMs on event mentions with different numbers of events in the MLEE datasets. The Base model refers to Llama3-Instruct fine-tuned using all five datasets, but it does not incorporate the co-occurrence-aware and structure-aware prompts.

F In-Context Learning with LLMs

As shown in the Table 8, 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.

G CsEAE and PAIE on TextEE Benchmark

We tested our model using the TextEE Benchmark, dividing the dataset into five subsets while allowing for multi-word triggers, accounting for overlapping argument spans, and retaining all instances without filtering. As shown in the Table 9, the performance of CsEAE and PAIE is compared across all five splits of the dataset. CsEAE consistently outperforms PAIE in the Arg-C metric across most splits. This demonstrates that the improvements achieved by CsEAE are not coincidental but rather the result of the genuine enhancements in extraction performance brought by the incorporation of structure-aware and co-occurrences-aware mechanisms.

H Analysis of CsEAE

In this section, we will conduct an additional experiment on CsEAE to demonstrate that it can achieve better performance than other models when faced with complex event types.

H.1 Multiple Event Extraction

We categorized the data from the three datasets based on the number of events occurring, as shown in the Figure 6. It is evident that there are significantly more instances with multiple events in MLEE compared to WikiEvents and Rams. To analyze the performance of CsEAE when facing an increase in the number of events in documents, we conducted experiments on the MLEE dataset.

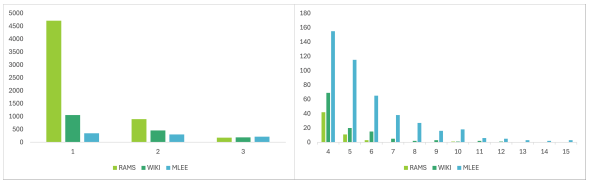

Figure 6: The figure above illustrates the distribution of the number of events per instance across the three datasets. The horizontal axis represents the number of events, while the vertical axis represents the number of instances.

The Table 10 show that compared to PAIE, CsEAE achieves improvements in handling instances with multiple events. Specifically, in cases where the number of events is greater than or equal to four (Event >= 4), CsEAE achieves improve-

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
Base	70.3	66.0	61.6	58.0	75.8	74.9	65.6	64.2
CsLLMs	73.5	68.2	68.1	66.4	80.6	79.6	70.4	69.2

Table 12: The performance of LLMs in extracting the arguments of overlapping events.

ments of 3.3% in Arg-I and 3.4% in Arg-C metrics on the MLEE dataset. Furthermore, it also exhibits slightly superior performance compared to TabEAE in handling instances with multiple events. This indicates the superiority of CsEAE in handling instances with multiple events.

I Analysis of CsLLMs

In this section, we will conduct a comprehensive analysis of CsLLMs.

I.1 Multiple event Extraction

Similar to CsEAE, as shown in the Table 11, we evaluated the performance of CsLLMs in handling multi-event instances on the MLEE dataset. As shown in the table, the model achieved significant improvements in all cases after adding co-occurrences-aware information.

I.2 Capturing the Event Semantic Boundary

Similar to CsEAE, we evaluated the model's ability to capture the event semantic boundaries on the WikiEvents and MLEE datasets from two aspects: Inter-event semantics and Inner-event semantics.

Inter-event semantics. As shown in the Table 12, in the Overlap scenarios on the MLEE dataset, CsLLMs outperformed the base model by 5.1% and 6.6% in Arg-I and Arg-C metrics, respectively. This indicates a significant improvement in the model's ability to capture Inter-event semantics.

Inner-event semantics. As shown in the Figure 7, the comprehensive improvement of the model across multiple different d indicates an enhanced ability to capture inner-event semantics.

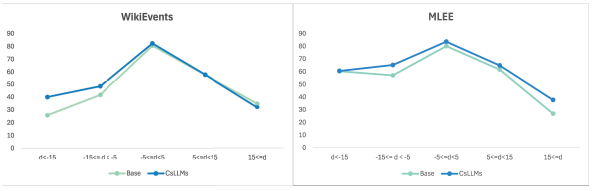

Figure 7: The performance of different LLMs in extracting arguments at different distances from the triggers.

I.3 Structure-aware Interaction for Document

As shown in the Table 13, the improvement of CsLLMs compared to the base model demonstrates the effectiveness of introducing structure-aware.

Model	Rams (Arg-C)		
	D=0	D≠0	All
Base	61.8	26.7	51.6
CsLLMs	61.7	27.7	51.8

Table 13: "All" refers to all the data in the test set.

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 14: Overall performance of LLMs on GENEVA.

I.4 Generalization of LLMs

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 14. 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).