Joint-DEE: A Joint Framework for End-to-End Document-Level Event Extraction

Anonymous ACL submission

Abstract

Document-level event extraction (DEE) is typically addressed in a cascaded pipeline—trigger extraction, coreference resolution, and argument extraction are performed sequentially. This paradigm suffers from error propagation and isolated task modeling, making it difficult to capture long-range contextual dependencies and cross-task interactions (e.g., coreference clues aiding argument role assignment). To address these limitations, we propose a joint document-level extraction framework that unifies these subtasks into a single model. Specifically, trigger mentions and entity mentions are first extracted via parallel sequence labeling, and then jointly organized through a unified relation graph that explicitly models trigger coreference, entity coreference, and trigger–argument associations within one global structure. To further enhance cross-sentence entity aggregation and role consistency, we introduce an auxiliary entity tracking task that maintains entity states across the document. Experimental results on multiple benchmarks demonstrate consistent improvements over strong cascade and joint baselines under both topic-centric and non-topic-centric settings.

1 Introduction

Document-level event extraction aims to identify and structurally represent multiple interrelated events from an entire document. This task typically consists of several closely related subtasks: (1) trigger extraction, which focuses on identifying event-triggering expressions in a document and classifying them into predefined event types; (2) argument extraction, which, conditioned on a given trigger, aims to identify and classify entities, times, locations, and other participants involved in the event according to their semantic roles; and (3) coreference resolution, which plays a particularly important role in document-level event extraction by determining whether trigger mentions or entity mentions at different positions refer to the same real-world event or object, thereby enabling cross-sentence and cross-paragraph event aggregation and consistent modeling.

The core challenges of the task lie in modeling cross-sentence contextual information, capturing dependencies among events, and handling the complex many-to-many matching between events and their arguments. Unlike sentence-level event extraction, event cues in the document-level setting are often scattered across multiple sentences, requiring a holistic understanding of the entire document to accurately interpret event semantics.

Existing studies have demonstrated promising progress, yet they still suffer from several limitations.

Firstly, many document-level approaches overlook the intrinsic dependencies among subtasks, instead adopting cascaded pipelines tailored to individual tasks, which often leads to error propagation and suboptimal global modeling.

Secondly, existing methods make limited use of the document-level coreference information. Although there are traditional tool-based approaches based on syntactic dependency structures, they still struggle to handle long-range event aggregation across sentences or paragraphs, resulting in insufficient event integration and argument matching.

Thirdly, the dense presence of triggers and entities in document-level texts, as well as the complex relations among them, poses challenges for both training and inference. Existing generative methods based on large language models (LLMs) produce excessively long outputs, highlighting the need for discriminative models that combine extraction with document-level semantic interaction modeling.

To address the first limitation, we propose a document-level event extraction framework that jointly performs trigger extraction and entity extraction, while leveraging a relation matching mod-

ule to capture the rich relationships between the extracted triggers and entities and further identify the entities as arguments.

The relation matching module, designed to address the second limitation, models trigger coreference, entity coreference, and argument extraction in a graph structure, enabling the capture of structured document-level relations.

To mitigate the difficulty of learning complex document-level relation matching, we further introduce an **entity tracking** task during training defined as follows: given a seed entity, the model identifies all its coreference mentions and determines its semantic roles under different triggers. This auxiliary task improves relation matching while maintaining effective document-level information aggregation. In addition, the framework adapts a generative language model into a bidirectional encoder by replacing its causal attention to preserve its semantic understanding at the document-level. In general, the proposed framework enables unified end-to-end training for document-level event extraction.

Due to the issue of incomplete event annotations in existing datasets, we adopt two experimental settings. In the **non-topic-centric** setting, no prior information is provided, and the model directly performs the complete extraction process. In the **topic-centric** setting, a target event type set is provided in advance, and the model extracts triggers and arguments restricted to these specified types. We treat the topic-centric and non-topic-centric settings as two parallel evaluation protocols, and compare methods only within the same protocol. This setting partially alleviates the problems of missing or redundant event annotations in existing datasets, while also better aligning with real-world application scenarios that focus only on specific event types.

In summary, our main contributions are as follows:

(1) We propose Joint-DEE, a unified framework for end-to-end document-level event extraction, which jointly models trigger extraction, entity extraction, coreference resolution, and argument assignment.

(2) We design a relation matching module that operates over candidate trigger and entity spans, enabling coordinated modeling of trigger coreference, entity coreference, and trigger–argument relations.

(3) We introduce an auxiliary entity tracking task to enhance document-level entity aggregation and argument consistency. Experiments under both topic-centric and non-topic-centric settings demonstrate the effectiveness of our framework.

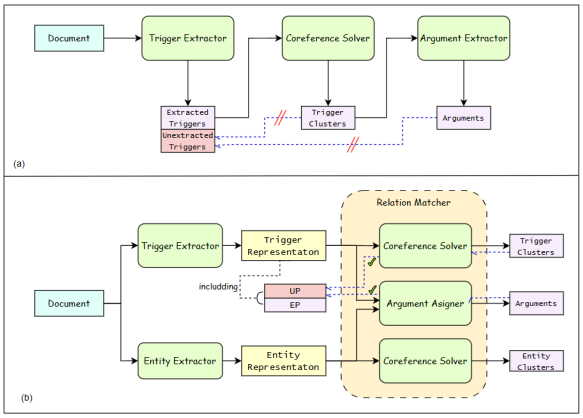

Figure 1: Comparison between a cascaded framework and the proposed joint framework. (a) In the cascaded paradigm, trigger extraction, coreference resolution, and argument extraction are executed sequentially, such that unextracted triggers are pruned early and cannot contribute to downstream inference, leading to severe error propagation. (b) In the proposed joint framework, trigger extraction and entity extraction are conducted in parallel, and their outputs are integrated through a unified relation matching module. This design allows upstream uncertain predictions (UP) to remain informative for downstream modules rather than being prematurely discarded, while also enabling information aggregation from both entity-level and coreference-level signals.

2 Related Work

2.1 Sentence-Level Event Extraction

Event Extraction (EE) aims to extract critical information (triggers and relevant entities) from unstructured text and identify them as structured semantic roles. Early studies on EE focused on sentence-level data. Neural approaches relied on CNN- (Nguyen and Grishman, 2015) and RNN-based (Nguyen et al., 2016) architectures for trigger and argument modeling. With the success of BERT, pretrained language models (PLMs) have been widely used in event extraction tasks (Yang et al., 2019).

Despite these advances, early sentence-level methods are inherently limited: they ignore cross-sentence and document-level context, struggle with multi-event interactions, and fail to exploit global discourse coherence, which motivates the shift toward document-level event extraction.

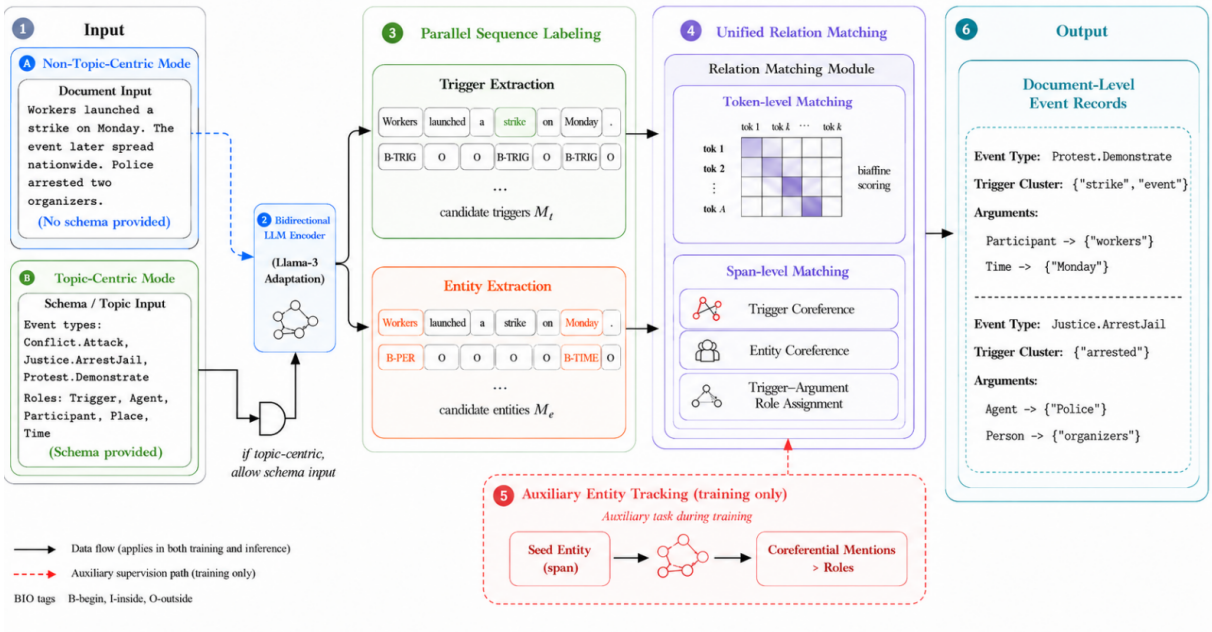

Figure 2: Architecture of our LLM-based document level event extraction framework

2.2 Document-Level Event Extraction

With the growing demand for processing longer texts, numerous document-level event extraction(**DEE**) datasets have been proposed, such as WikiEvents (Li et al., 2021), RAMS (Ebner et al., 2020), MAVEN (Wang et al., 2020), MAVEN-Arg (Wang et al., 2024), DiscourseEE (Sharif et al., 2024).

Classification methods (Huang et al., 2022; Ma et al., 2020; Hsu et al., 2023a) solve DEE task by labeling the long sequence with Tags or classifying tokens to semantic roles. These methods benefit from their flexibility in adapting to both Event Detection (**ED**) task and Event Argument Extraction (**EAE**) task. To address the limitation of sequence-labeling methods in modeling nested and overlapping spans, semantic parsing methods improve document-level reasoning by explicitly modeling structural dependencies among spans. Abstract Meaning Representation (AMR), for instance, has been incorporated to introduce prior information of document structure into event extraction (Pouran Ben Veyseh et al., 2020; Zhang and Ji, 2021; Xu et al., 2022; Hsu et al., 2023b).

Generative approaches address event extraction from a sequence generation perspective. Machine reading comprehension and question answering strategies (Huang et al., 2023; Lu et al., 2021) facilitate the understanding of event structures by identifying and extracting key elements from text through a question-driven approach. PAIE (Ma et al., 2022), TableEAE (He et al., 2023) and HMPEAE (Zhang et al., 2024) identify optimal text pieces as roles by span selectors. However, such methods typically rely on separate queries or lengthy sequential decoding for each role, resulting in high decoding costs and limited ability to model global interactions across events and arguments.

2.3 Joint Information Extraction

Traditional information extraction (IE) pipelines often decompose the task into sequential subtasks, such as named entity recognition (NER) followed by relation extraction (RE). In contrast, DyGIE++ (Lin et al., 2020) and OneIE (Wadden et al., 2019) formulate IE as joint span-graph construction, where entities, relations, and events are modeled within a unified structure. Generative models have also been explored for structured information extraction. For example, BART-based models have been applied to span prediction (Yan et al., 2021), while DeepStruct (Wang et al., 2022) and GenIE (Josifoski et al., 2022) generate triples for structured prediction. MIRROR (Zhu et al., 2023) and KnowCoder-X (Zuo et al., 2025) further construct multi-task methods on large-scale datasets, with the former (generative approaches) typically relying on classification networks, while the latter (discriminative approaches) leverage large language models. However, these approaches provide limited support for cross-sentence event interaction

and document-level event aggregation.

3 Preliminary

The document-level event extraction task can be formalized as follows: given a document $D = \{w_1, w_2, \ldots, w_n\}$ and a predefined event schema set $\mathcal{Y}$ (where each event type $y \in \mathcal{Y}$ is associated with a set of semantic roles $\mathcal{R}_y$), document-level event extraction aims to identify and structurally represent all events of predefined types from the entire document.

Input. The input comprise the document D and the event schema $\mathcal{Y}$ (including role definitions). In the **topic-centric** setting, an additional target event type subset $\mathcal{T} \subseteq \mathcal{Y}$ is provided, restricting the model to focus only on these specified types.

Intermediate Representations. The model first extracts a set of **trigger mentions** $\mathcal{M}_t = \{(s_i, y_i)\}$, where s_i denotes a text span and y_i its predicted event type. These mentions are then grouped into **trigger clusters** $\mathcal{C}_t = \{C_t^k\}$ via trigger coreference resolution, with each cluster representing an independent event instance. Concurrently, the model identifies a set of **entity mentions** $\mathcal{M}_e = \{(s_j, \text{type}_j)\}$, which are aggregated into **entity clusters** $\mathcal{C}_e = \{C_e^m\}$ through entity coreference resolution, each cluster corresponding to a real-world object.

Output. The final output is a set of document-level **event records** $\mathcal{E} = \{e_1, e_2, \ldots\}$. Each record e_k consists of an event type $y \in \mathcal{Y}$, a trigger cluster $C_t \in \mathcal{C}_t$ containing all trigger mentions of this event, and a collection of argument-role pairs mapping each predefined role $r \in \mathcal{R}_y$ to its corresponding filler entity cluster $C_e \in \mathcal{C}_e$, with $\emptyset$ indicating an unfilled role.

Despite these advances, early sentence-level methods are inherently limited: they ignore cross-sentence and document-level context, struggle with multi-event interactions, and fail to exploit global discourse coherence, which motivates the shift toward document-level event extraction.

4 Framework

4.1 Overview

As shown in Figure 2, our Joint-DEE framework operates as an end-to-end pipeline built upon a shared bidirectional LLM encoder. The encoder (Sec 4.2) first produces document-level contextual representations $\mathbf{H} \in \mathbb{R}^{n \times d}$ for the input document D. Two parallel sequence labeling heads then consume $\mathbf{H}$ to extract candidate trigger mentions $\mathcal{M}_t$ and entity mentions $\mathcal{M}_e$. These candidate spans, together with the encoded representations $\mathbf{H}$, are fed into a unified relation matching module (Sec 4.3) that jointly performs trigger coreference clustering, entity coreference clustering, and trigger–entity role assignment within a single graph structure. During training, an auxiliary entity tracking task (Sec 4.4) is introduced. The resulting clusters and role links are finally decoded into structured event records $\mathcal{E}$.

Topic-Centric Adaptation. Under the topic-centric setting, the model restricts both its input schema description and output label space to a user-specified event type subset $\mathcal{T} \subseteq \mathcal{Y}$. Concretely, a schema prefix enumerating only the target types and their associated roles is prepended to the document, focusing the encoder on the relevant semantic space from the outset. Simultaneously, a category mask is applied to the trigger classification layer so that predictions are drawn exclusively from $\mathcal{T} \cup \{\text{None}\}$, and the role classifier activates only the dimensions corresponding to the currently predicted event type. This input–output co-restriction suppresses non-target event predictions, alleviates noise caused by incomplete annotations, and aligns the model with real-world scenarios where analysts prioritize specific event topics.

4.2 Bidirectional LLM-Encoder

Given an input document $D = \{w_1, w_2, \ldots, w_n\}$, the adapted bidirectional LLM encoder produces contextualized token representations:

$$\mathbf{H} = \text{Enc}_\theta(D), \quad \mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_n] \in \mathbb{R}^{n \times d}. \tag{1}$$

Trigger extraction and entity extraction are formulated as two parallel sequence labeling tasks:

$$\mathbf{o}_i^t = \text{MLP}_t(\mathbf{h}_i), \quad \mathbf{o}_i^e = \text{MLP}_e(\mathbf{h}_i). \tag{2}$$

Let $\hat{y}_i^t$ and $\hat{y}_i^e$ denote the gold trigger and entity labels of token w_i. The sequence labeling losses are:

$$\begin{aligned} \mathcal{L}_{trigger} &= \frac{1}{n} \sum_{i=1}^{n} \ell_{\text{ce}}(\mathbf{o}_i^t, \hat{y}_i^t), \\ \mathcal{L}_{entity} &= \frac{1}{n} \sum_{i=1}^{n} \ell_{\text{ce}}(\mathbf{o}_i^e, \hat{y}_i^e). \end{aligned} \tag{3}$$

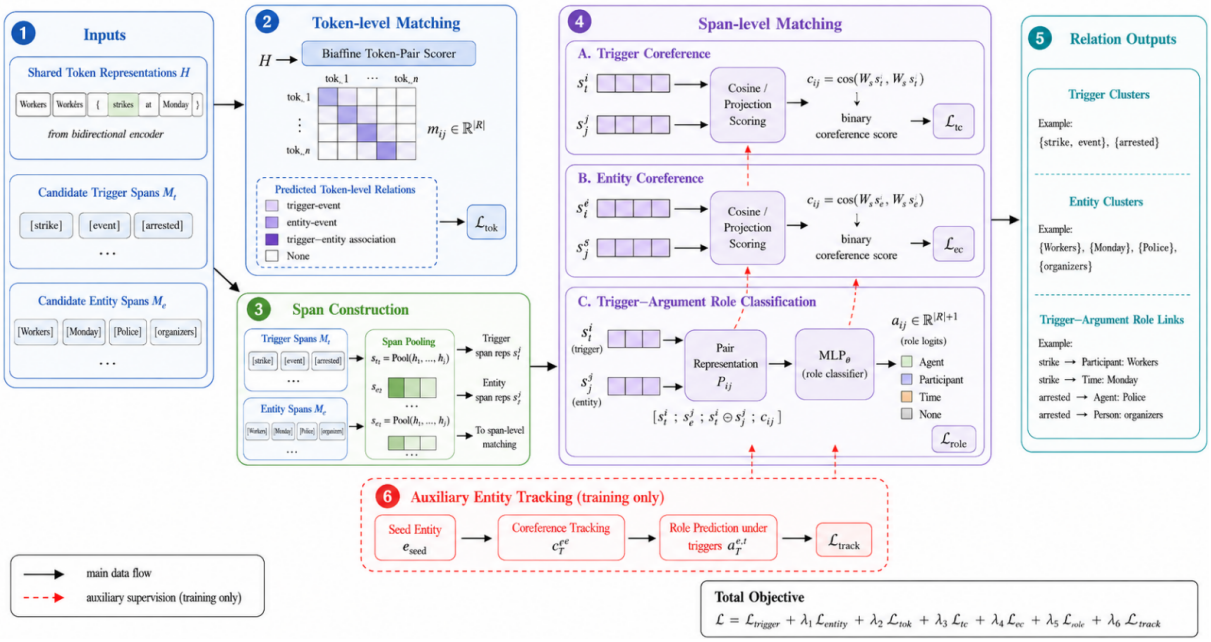

Figure 3: Architecture of our LLM-based document level event extraction framework

4.3 Relation Matching

Given the candidate trigger spans and entity spans extracted by the sequence labeling heads, the relation matching module predicts trigger coreference, entity coreference, and trigger–argument role relations.

Token-level matching. We first apply a biaffine scorer to token representations:

$$\mathbf{m}_{ij} = \mathrm{Biaff}(\mathbf{h}_i, \mathbf{h}_j), \quad \mathbf{m}_{ij} \in \mathbb{R}^{|\mathcal{R}|}, \tag{4}$$

where $\mathcal{R}$ is the token-level relation label set, including a NONE label.

Let $\mathcal{P}_{tok}$ denote the valid token-pair set and $\hat{r}_{ij}$ denote the gold token-level relation label. The token-level matching loss is:

$$\mathcal{L}_{tok} = \frac{1}{|\mathcal{P}_{tok}|} \sum_{(i,j)\in\mathcal{P}_{tok}} \ell_{\mathrm{ce}}(\mathbf{m}_{ij}, \hat{r}_{ij}). \tag{5}$$

Span representations. For each candidate span $s_i = (l_i, r_i)$, its representation is obtained by pooling the encoder states inside the span:

$$\mathbf{s}_i = \mathrm{Pool}(\mathbf{h}_{l_i}, \ldots, \mathbf{h}_{r_i}). \tag{6}$$

Span-level coreference. For each pair of candidate spans (s_i, s_j) of the same type, we compute a binary coreference logit:

$$c_{ij} = \tau \cdot \cos(\mathbf{W}_c\mathbf{s}_i, \mathbf{W}_c\mathbf{s}_j), \tag{7}$$

where $\mathbf{W}_c$ is a learnable projection matrix and τ is a scaling factor.

Let $\mathcal{P}_c^t$ and $\mathcal{P}_c^e$ denote trigger-span and entity-span coreference pairs, and let $\hat{c}_{ij} \in \{0, 1\}$ be the gold coreference label. The span-level coreference losses are:

$$\begin{aligned} \mathcal{L}_{tc} &= \frac{1}{|\mathcal{P}_c^t|} \sum_{(i,j)\in\mathcal{P}_c^t} \ell_{\mathrm{bce}}(c_{ij}, \hat{c}_{ij}), \\ \mathcal{L}_{ec} &= \frac{1}{|\mathcal{P}_c^e|} \sum_{(i,j)\in\mathcal{P}_c^e} \ell_{\mathrm{bce}}(c_{ij}, \hat{c}_{ij}). \end{aligned} \tag{8}$$

Trigger–argument role classification. For each candidate trigger span $\mathbf{s}_i^t$ and entity span $\mathbf{s}_j^e$, we construct a pair representation:

$$\mathbf{p}_{ij} = \mathrm{MLP}_p([\mathbf{s}_i^t; \mathbf{s}_j^e; \mathbf{s}_i^t \odot \mathbf{s}_j^e; c_{ij}]). \tag{9}$$

The role logits are computed as:

$$\mathbf{a}_{ij} = \mathrm{MLP}_r(\mathbf{p}_{ij}), \quad \mathbf{a}_{ij} \in \mathbb{R}^{|\mathcal{Q}|+1}, \tag{10}$$

where $\mathcal{Q}$ is the argument role set and the additional dimension corresponds to NONE.

Let $\mathcal{P}_r$ denote the candidate trigger–entity span pairs and $\hat{a}_{ij}$ denote the gold role label. The role classification loss is:

$$\mathcal{L}_{role} = \frac{1}{|\mathcal{P}_r|} \sum_{(i,j)\in\mathcal{P}_r} \ell_{\mathrm{ce}}(\mathbf{a}_{ij}, \hat{a}_{ij}). \tag{11}$$

4.4 Entity Tracking

Given a seed entity span s_{seed} wrapped by special tokens <SEED> and </SEED>, the entity tracking task identifies its coreferential mentions and predicts its semantic roles under different triggers.

Let $\mathbf{e}_{seed}$ denote the pooled representation of the seed entity. For each candidate entity span $\mathbf{s}_i^e$, we compute a seed-entity coreference logit:

$$c_i^{tr} = \tau \cdot \cos(\mathbf{W}_c \mathbf{e}_{seed}, \mathbf{W}_c \mathbf{s}_i^e). \tag{12}$$

For each candidate trigger span $\mathbf{s}_j^t$, we compute the role logits of the seed entity:

$$\mathbf{a}_j^{tr} = \mathrm{MLP}_r(\mathrm{MLP}_p([\mathbf{e}_{seed}; \mathbf{s}_j^t; \mathbf{e}_{seed} \odot \mathbf{s}_j^t])). \tag{13}$$

Let $\hat{c}_i^{tr} \in \{0, 1\}$ indicate whether $\mathbf{s}_i^e$ is coreferential with the seed entity, and let $\hat{a}_j^{tr}$ denote the gold role label of the seed entity under trigger $\mathbf{s}_j^t$. The entity tracking loss is:

$$\begin{aligned}\mathcal{L}_{track} =& \frac{1}{|\mathcal{P}_e^{tr}|} \sum_{i \in \mathcal{P}_e^{tr}} \ell_{\mathrm{bce}}(c_i^{tr}, \hat{c}_i^{tr}) \\ &+ \frac{1}{|\mathcal{P}_t^{tr}|} \sum_{j \in \mathcal{P}_t^{tr}} \ell_{\mathrm{ce}}(\mathbf{a}_j^{tr}, \hat{a}_j^{tr}).\end{aligned} \tag{14}$$

Total Training Loss. The final training objective is:

$$\begin{aligned}\mathcal{L} =& \mathcal{L}_{trigger} + \lambda_1 \mathcal{L}_{entity} + \lambda_2 \mathcal{L}_{tok} + \lambda_3 \mathcal{L}_{tc} \\ &+ \lambda_4 \mathcal{L}_{ec} + \lambda_5 \mathcal{L}_{role} + \lambda_6 \mathcal{L}_{track}.\end{aligned} \tag{15}$$

5 Experiment

5.1 Settings

We conduct experiments on the WikiEvents and MAVEN-Arg datasets. Following standard practice, we use the official train/dev/test splits when available and evaluate all methods under two parallel protocols: non-topic-centric and topic-centric settings. The two settings are not intended for direct comparison with each other; instead, methods are compared within the same protocol. In the non-topic-centric setting, the model extracts all event types from the document, while in the topic-centric setting, a target event type set is provided in advance and extraction is restricted to these specified types.

We report three evaluation metrics: Trigger Classification F1 (TC-F1), which measures trigger detection and type classification; Argument Classification F1 (AC-F1), which evaluates argument identification and role classification with respect to event triggers; and Argument Coreference F1 (ACo-F1), which measures whether arguments referring to the same entity or concept are correctly linked across the document.

5.2 Baselines

We compare Joint-DEE with representative baselines in the main experiments. First, we include end-to-end structured extraction models, including **AMR-IE** (Zhang and Ji, 2021) and **One-IE** (Wadden et al., 2019). Second, we compare with QA-based and generation-based models, including **RCEE** (Liu et al., 2020), **EEQA** (Hsu et al., 2023a), and **DEGREE** (Hsu et al., 2022). Additional comparisons with LLM-based baselines and cascaded Tag-Prime-based pipelines, including **Tag-Prime+Tag-Prime** (Hsu et al., 2023a) and **Tag-Prime+PAIE** (Ma et al., 2022), are reported in Appendix D. Detailed descriptions, task adaptations, and implementation details are provided in Appendix C.

5.3 Main Results

The main results are shown in Tables 1 and 2, where we compare Joint-DEE with representative end-to-end baselines. Additional comparisons with LLM-based and cascaded Tag-Prime baselines are reported in Appendix D. Within each evaluation protocol, Joint-DEE achieves the best overall performance on both datasets. In the non-topic-centric setting, it improves over the strongest baselines by 2.7/5.9/5.7 points on WikiEvents and 0.8/5.0/5.2 points on MAVEN-Arg in TC-F1, AC-F1, and ACo-F1, respectively. In the topic-centric setting, where target event types are provided in advance, Joint-DEE also obtains gains of 4.1/3.5/3.5 points on WikiEvents and 1.2/2.0/8.1 points on MAVEN-Arg.

These results show that Joint-DEE consistently benefits document-level event extraction, especially on argument extraction and argument coreference. The improvements suggest that jointly modeling trigger/entity candidates, coreference links, and trigger–argument relations provides more effective document-level structural reasoning than treating these subtasks separately.

Comparison with Cascaded Frameworks under Controlled Trigger Extraction. To better examine the effect of joint relation modeling, we compare Joint-DEE with a cascaded counterpart that

Methods	WikiEvents			MAVEN-Arg		
	TC-F1	AC-F1	ACo-F1	TC-F1	AC-F1	ACo-F1
AMR-IE	43.5	24.8	27.4	57.2	16.4	19.1
One-IE	44.7	22.5	24.2	65.5	15.0	17.7
RCEE	38.6	36.2	38.8	55.0	19.4	21.5
EEQA	43.3	35.4	38.2	63.8	17.0	18.5
DEGREE	47.7	29.0	30.4	65.1	12.6	13.8
Joint-DEE(Ours)	**50.4**	**42.1**	**44.5**	**66.3**	**24.4**	**26.7**

Table 1: Event extraction performances on WikiEvents and MAVEN-Arg datasets under the **non-topic-centric** setting.

Methods	WikiEvents			MAVEN-Arg		
	TC-F1	AC-F1	ACo-F1	TC-F1	AC-F1	ACo-F1
AMR-IE	56.5	35.0	37.8	60.6	19.3	20.7
One-IE	54.4	40.5	44.5	71.3	19.5	21.4
RCEE	49.6	45.4	47.6	62.2	25.2	21.5
EEQA	52.4	44.8	47.4	67.4	18.8	20.0
DEGREE	57.1	43.0	34.4	68.4	14.0	16.3
Joint-DEE(Ours)	**61.2**	**48.9**	**51.1**	**72.5**	**27.2**	**29.6**

Table 2: Event extraction performances on WikiEvents and MAVEN-Arg datasets under the **topic-centric** setting.

uses the same bidirectional LLM encoder for upstream trigger extraction. The downstream argument extractor is implemented with PAIE, whose architecture does not support direct replacement with our LLM encoder at the same scale. Therefore, this comparison primarily controls the trigger extraction stage and contrasts cascaded trigger-conditioned argument extraction with our unified relation matching design. As shown in Table 3, Joint-DEE achieves consistent improvements on argument-related metrics under both non-topic-centric and topic-centric settings, while maintaining comparable TC-F1. This suggests that the argument-related gains are mainly associated with improved document-level relation modeling rather than stronger trigger detection. By jointly modeling trigger coreference, entity coreference, and trigger–argument relations, Joint-DEE can organize candidate spans into more coherent event structures than the cascaded framework based on hard upstream decisions.

Candidate Span Recall and Coverage. We further analyze candidate span recall and coverage in Appendix F, where the results show that Joint-DEE preserves more gold-relevant trigger and argument structures before relation matching.

5.4 Ablation and Analysis

We conduct ablation studies to examine the contribution of each component under both non-topic-centric and topic-centric settings. As shown in Table 4, removing any key component generally leads to performance degradation, especially on argument-related metrics. The removal of bidirectional attention causes consistent drops on both datasets, confirming the importance of full document context for argument extraction and coreference modeling. In addition, ablating trigger coreference, entity coreference, or span-level matching weakens AC-F1 and ACo-F1, showing that these relation matching components provide complementary signals for organizing trigger and entity candidates. Entity tracking also brings stable improvements, particularly for argument extraction and argument coreference. Overall, these results demonstrate that Joint-DEE benefits from the combination of bidirectional document encoding, unified relation matching, and entity tracking. Detailed analysis is provided in Appendix G.

We further analyze the training behavior of Joint-DEE from two perspectives: the interaction between trigger and entity extraction, and the consistency between token-level and span-level relation supervision. The results show that trigger and entity extraction remain stable during joint training, while token-level supervision provides useful com-

Setting	Methods	WikiEvents			MAVEN-Arg		
		TC-F1	AC-F1	ACo-F1	TC-F1	AC-F1	ACo-F1
Non-topic-centric	Cascaded-DEE	**50.2**	41.7	43.8	66.1	22.4	24.1
	Joint-DEE	50.4	**42.1**	**44.5**	**66.3**	**24.4**	**26.7**
	Δ	+0.2	+0.4	+0.7	+0.2	+2.0	+2.6
Topic-centric	Cascaded-DEE	61.2	47.4	49.9	72.5	26.4	28.7
	Joint-DEE	61.2	48.9	51.1	72.5	27.2	29.6
	Δ	+0.0	+1.5	+1.2	+0.0	+0.8	+0.9

Table 3: Controlled comparison between cascaded and joint frameworks under both non-topic-centric and topic-centric settings. The cascaded baseline uses Tag-Prime with the same bidirectional LLM encoder as the upstream trigger extractor, followed by PAIE for downstream argument extraction. Due to architectural constraints, PAIE is kept in its original backbone configuration; therefore, this comparison controls the upstream trigger extraction stage rather than enforcing identical backbones across the entire pipeline.

Setting	Method	WikiEvents			MAVEN-Arg		
		TC-F1	AC-F1	ACo-F1	TC-F1	AC-F1	ACo-F1
Non-Topic-Centric	Joint-DEE	**50.4**	**42.1**	**44.5**	**66.3**	**24.4**	**26.7**
	w/o bi-attention	48.7	39.7	42.0	64.4	23.7	26.2
	w/o span match	50.2	41.7	43.8	65.6	23.9	26.0
	w/o trigger coref	49.8	41.9	44.2	65.3	22.5	25.2
	w/o entity coref	49.6	41.4	43.1	66.2	22.9	24.6
	w/o entity tracking	49.2	41.4	43.6	65.6	23.2	25.5
Topic-Centric	Joint-DEE	**61.2**	**48.9**	**51.1**	**72.5**	**27.2**	**29.6**
	w/o bi-attention	60.4	47.3	50.3	71.3	25.7	27.8
	w/o span match	61.0	48.2	50.7	72.5	26.4	27.7
	w/o trigger coref	60.6	48.7	50.7	72.2	26.4	27.9
	w/o entity coref	60.9	48.4	50.5	72.5	25.4	28.5
	w/o entity tracking	61.2	47.4	49.9	72.5	26.2	28.8

Table 4: Ablation study of Joint-DEE on WikiEvents and MAVEN-Arg under both non-topic-centric and topic-centric settings.

plementary signals for span-level relation learning. Detailed curves and analysis are provided in Appendix H.

6 Conclusion

In this paper, we propose Joint-DEE, an end-to-end framework for document-level event extraction that jointly models trigger extraction, entity extraction, coreference resolution, and argument association within a unified architecture. The proposed relation matching module integrates token-level and span-level interaction modeling, while the auxiliary entity tracking task improves document-level contextual aggregation and argument consistency. Experimental results on WikiEvents and MAVEN-Arg demonstrate the effectiveness of our framework under both topic-centric and non-topic-centric settings. Future work includes incorporating richer event relations, such as temporal and causal dependencies, and improving generalization under low-resource settings.

Limitations

While our approach demonstrates promising results, there are several aspects that warrant further exploration. First, the current framework primarily models coreference and trigger–argument relations, and does not explicitly incorporate richer event relations such as temporal, causal, or hierarchical dependencies between events. Extending the model to cover a wider spectrum of event relations may further enhance document-level event understanding. Second, although our argument extraction module benefits from unified relation modeling, its performance is not yet fully comparable to dedicated Event Argument Extraction (EAE) methods that rely on specialized architectures and supervision, suggesting potential gains from more refined argument modeling strategies. Finally, despite leveraging pre-trained language models for semantic representation, our method does not fully exploit their few-shot learning capabilities, leaving room for future work on improving generalization under limited annotation settings.

References

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 Lin-*

guistics, pages 8057–8077, Online. Association for Computational Linguistics.

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)*, pages 12542–12556, Toronto, Canada. Association for Computational Linguistics.

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

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)*, pages 12917–12932, Toronto, Canada. 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)*, pages 10976–10993, Toronto, Canada. Association for Computational Linguistics.

Heyan Huang, Xiao Liu, Ge Shi, and Qian Liu. 2023. Event extraction with dynamic prefix tuning and relevance retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 35(10):9946–9958.

Kuan-Hao Huang, I-Hung Hsu, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. Multilingual generative language models for zero-shot cross-lingual event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4633–4646, Dublin, Ireland. Association for Computational Linguistics.

Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West. 2022. GenIE: Generative information extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4626–4643, Seattle, United States. Association for Computational Linguistics.

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

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online. Association for Computational Linguistics.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

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

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

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

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 365–371, Beijing, China. Association for Computational Linguistics.

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

Omar Sharif, Joseph Gatto, Madhusudan Basak, and Sarah M. Preum. 2024. Explicit, implicit, and scattered: Revisiting event extraction to capture complex arguments. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12061–12081, Miami, Florida, USA. Association for Computational Linguistics.

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)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022. DeepStruct: Pretraining of language models for structure prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland. Association for Computational Linguistics.

Xiaozhi Wang, Hao Peng, Yong Guan, Kaisheng Zeng, Jianhui Chen, Lei Hou, Xu Han, Yankai Lin, Zhiyuan Liu, Ruobing Xie, Jie Zhou, and Juanzi Li. 2024. MAVEN-ARG: Completing the puzzle of all-in-one event understanding dataset with event argument annotation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4072–4091, Bangkok, Thailand. Association for Computational Linguistics.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online. Association for Computational Linguistics.

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

Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021. A unified generative framework for aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2416–2429, Online. Association for Computational Linguistics.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, Florence, Italy. Association for Computational Linguistics.

Guangjun Zhang, Hu Zhang, YuJie Wang, Ru Li, Hongye Tan, and Jiye Liang. 2024. Hyperspherical multi-prototype with optimal transport for event argument extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9271–9284, Bangkok, Thailand. Association for Computational Linguistics.

Zixuan Zhang and Heng Ji. 2021. Abstract Meaning Representation guided graph encoding and decoding for joint information extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–49, Online. Association for Computational Linguistics.

Tong Zhu, Junfei Ren, Zijian Yu, Mengsong Wu, Guoliang Zhang, Xiaoye Qu, Wenliang Chen, Zhefeng Wang, Baoxing Huai, and Min Zhang. 2023. Mirror: A universal framework for various information extraction tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8861–8876, Singapore. Association for Computational Linguistics.

Yuxin Zuo, Wenxuan Jiang, Wenxuan Liu, Zixuan Li, Long Bai, Hanbin Wang, Yutao Zeng, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2025. KnowCoder-X: Boosting multilingual information extraction via code. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14486–14509, Vienna, Austria. Association for Computational Linguistics.

A Dataset

The datasets used in this study comprise two publicly available datasets for event extraction, which are split into training and test sets. Table 5 and Table 6 present relevant statistics of the datasets.

	WikiEvents-train	WikiEvents-test
Num_Docs	481	84
Num_Triggers	3222	710
Num_Arguments	4507	994
Num_Entities	27978	4929

Table 5: Dataset statistics of WikiEvents.

	MAVEN-Arg train	MAVEN-Arg test
Num_Docs	2913	710
Num_Triggers	70775	16996
Num_Arguments	190479	46458
Num_Entities	142390	31733

Table 6: Dataset statistics of MAVEN-Arg.

B Model Settings and Hyperparameters

Component	Setting
Base Model	Llama-3-8B-Instruct
Position Encoding	Retain original RoPE
Attention Backend	Flash Attention 2
Max Sequence Length	1024 tokens
LoRA Rank r	64
LoRA Alpha α	128
o_proj	
LoRA Dropout	0.05
Trainable Parameters	41.9M ($\approx$ 0.52% of total)
Optimizer	AdamW
LoRA Learning Rate	1×10^{-5}
MLP Learning Rate	2×10^{-4}
LR Scheduler	Cosine, 10% warmup steps
Batch Size	4
Training Epochs	20
λ_1 for entity loss	1.0
λ_2 for token-level matching loss	1.0
λ_3 for trigger coreference loss	0.2
λ_4 for entity coreference loss	0.8
λ_5 for role classification loss	1.0
λ_6 for entity tracking loss	1.0
τ 5.0	

Table 7: Configuration of the bidirectional Llama-3 encoder adaptation, LoRA-based fine-tuning, and loss weights.

C Baseline Details

This appendix briefly describes the baseline systems used in our experiments. For all baselines, we adapt their output schema to the event types and argument roles of WikiEvents and MAVEN-Arg, and normalize their predictions into the same trigger–argument format for evaluation.

AMR-IE parses the input into an AMR graph and extracts event structures from semantic subgraphs. We adopt the official code and adapt its event and argument output space to the schema of each dataset.

One-IE jointly models entities, relations, and events with graph-based structured inference. We adopt the official code adapt its output heads and label space to our event types and argument roles.

RCEE formulates trigger detection and argument filling as a reading-comprehension process. We follow its query-based extraction formulation and convert the predicted spans into event records.

EEQA performs event extraction through span-based question answering. We adopt the official codes and adapt its event-type and role-specific questions to the schemas used in our evaluation.

DEGREE uses a BART-based generator to fill event templates and decode event records as natural-language strings. We normalize its generated outputs into trigger–argument structures.

PAIE extracts event arguments with prompt-based argument interaction. Since it assumes given triggers, we mainly use the PAIE-large as a downstream argument extractor in cascaded settings.

Tag-Prime is a discriminative framework for relational structure extraction with task-specific tag prompts. We implement it with the same bidirectional LLM-Encoder backbone and adapt its tag space for trigger-oriented extraction and argument extraction under our event schema.

Tag-Prime+Tag-Prime is a cascaded baseline that uses Tag-Prime for both trigger extraction and argument extraction. The second stage performs argument extraction based on upstream trigger predictions.

Tag-Prime+PAIE is a cascaded baseline that uses Tag-Prime for trigger extraction and PAIE for downstream argument extraction. It combines discriminative trigger prediction with prompt-based argument extraction. Since PAIE is a prompt-based argument extractor with its own architectural assumptions, we do not replace its backbone with the bidirectional LLM encoder. Instead, we control the upstream trigger extractor by using the same Tag-Prime implementation and encoder scale as Joint-DEE, and use PAIE as the downstream argument extraction module in the cascaded pipeline.

Llama3-8B-SFT is a direct generative baseline based on Llama3-8B-Instruct. Given a document and event schema, it is supervised to generate structured event records.

GPT-4-QA is a closed-source LLM-based QA baseline. It uses event-type and role-specific questions to extract triggers and arguments step by step.

Evaluation protocol All baselines are evaluated under the same non-topic-centric and topic-centric settings as Joint-DEE. Since they differ in backbone, decoding strategy, and task formulation, we use them as benchmark-level comparisons rather than strictly controlled architectural counterparts.

D Additional Comparison with Cascaded and LLM-based Baselines

Table 8 reports additional comparisons with representative cascaded and LLM-based baselines. Overall, Joint-DEE achieves the best performance across both datasets and both settings, especially on argument-related metrics. Compared with the cascaded baselines, `Joint-DEE` consistently improves AC-F1 and ACo-F1, showing that even strong cascaded extractors such as `Tag-Prime+Tag-Prime` and `Tag-Prime+PAIE` still suffer from the separation between upstream trigger extraction and downstream argument extraction. In contrast, Joint-DEE jointly models trigger candidates, entity candidates, coreference links, and trigger–argument relations, leading to more coherent document-level event structures.

The results also show that directly applying LLM-based extraction methods is less effective in this setting. Although `Llama3-8B-SFT` is supervised to generate structured event records, its performance remains clearly lower than discriminative extraction methods, suggesting that direct generation struggles with dense document-level event structures and fine-grained argument-role assignment. Similarly, `GPT-4-QA` performs poorly compared with Joint-DEE, indicating that closed-source LLMs with QA-style prompting are still limited for this task, especially when complete trigger detection, argument extraction, and argument coreference must be handled jointly. These results further support the necessity of explicit relation matching for document-level event extraction.

E Case Study

Joint-DEE instead models candidate triggers and entities in a unified relation space. By linking "strike" and "event" as coreferential trigger mentions, the model can transfer argument evidence from the local context of "strike" to the event-level output. The entity tracking module further aggregates document-level entity mentions and their relations to different triggers, helping the model produce a more coherent event structure.

F Span Coverage Analysis

The controlled comparison above shows that the main gains of Joint-DEE come from argument extraction and argument coreference rather than local trigger detection. To further analyze where the gains come from, we evaluate whether the candidate span space preserves enough gold structures for the relation matcher.

We define three coverage metrics. Given the candidate trigger spans $\tilde{\mathcal{T}}$, candidate entity spans $\tilde{\mathcal{E}}$, gold trigger spans $\mathcal{T}^*$, gold entity spans $\mathcal{E}^*$, and gold trigger–argument pairs $\mathcal{A}^*$, trigger coverage, entity coverage, and argument-pair coverage are computed as:

$$\mathrm{TriCov} = \frac{|\tilde{\mathcal{T}} \cap \mathcal{T}^*|}{|\mathcal{T}^*|}, \quad \mathrm{EntCov} = \frac{|\tilde{\mathcal{E}} \cap \mathcal{E}^*|}{|\mathcal{E}^*|},$$

$$\mathrm{PairCov} = \frac{|\{(t,e) \in \mathcal{A}^* : t \in \tilde{\mathcal{T}}, e \in \tilde{\mathcal{E}}\}|}{|\mathcal{A}^*|}.$$

Among them, PairCov is the most important indicator, since a gold argument link can only be recovered by the relation matcher when both its trigger and entity are included in the candidate space.

As shown in Table 9, Joint-DEE consistently preserves a broader candidate space than the cascaded hard-output framework. On WikiEvents, Joint-DEE improves trigger coverage from 66.8 to 75.4 under the non-topic-centric setting and from 73.5 to 77.5 under the topic-centric setting. Its entity coverage is also high, reaching 93.6 and 93.7 in the two settings. More importantly, although Joint-DEE's PairCov is a stricter metric than Cascaded-DEE's ArgReachCov because it requires both the trigger and the entity to be retained, it still achieves higher coverage: 54.9 vs. 50.6 in the non-topic-centric setting and 58.3 vs. 57.8 in the topic-centric setting.

A similar trend is observed on MAVEN-Arg. Joint-DEE improves trigger coverage from 68.3 to 73.6 under the non-topic-centric setting and from 75.1 to 76.8 under the topic-centric setting. Compared with WikiEvents, entity coverage and pair coverage are lower on MAVEN-Arg, reflecting the denser entity mentions and more fine-grained argument structures in this dataset. Nevertheless, Joint-DEE still improves the argument-level coverage from 36.7 to 41.2 in the non-topic-centric setting and from 41.5 to 44.2 in the topic-centric setting.

These results suggest that Joint-DEE mainly mitigates, rather than eliminates, the pruning effect of upstream extraction. Since spans that are not included in the candidate set cannot be recovered by the relation matcher, the benefit of Joint-DEE lies in preserving more potentially useful trigger and entity candidates before relation modeling. The improved coverage indicates that the relation matcher

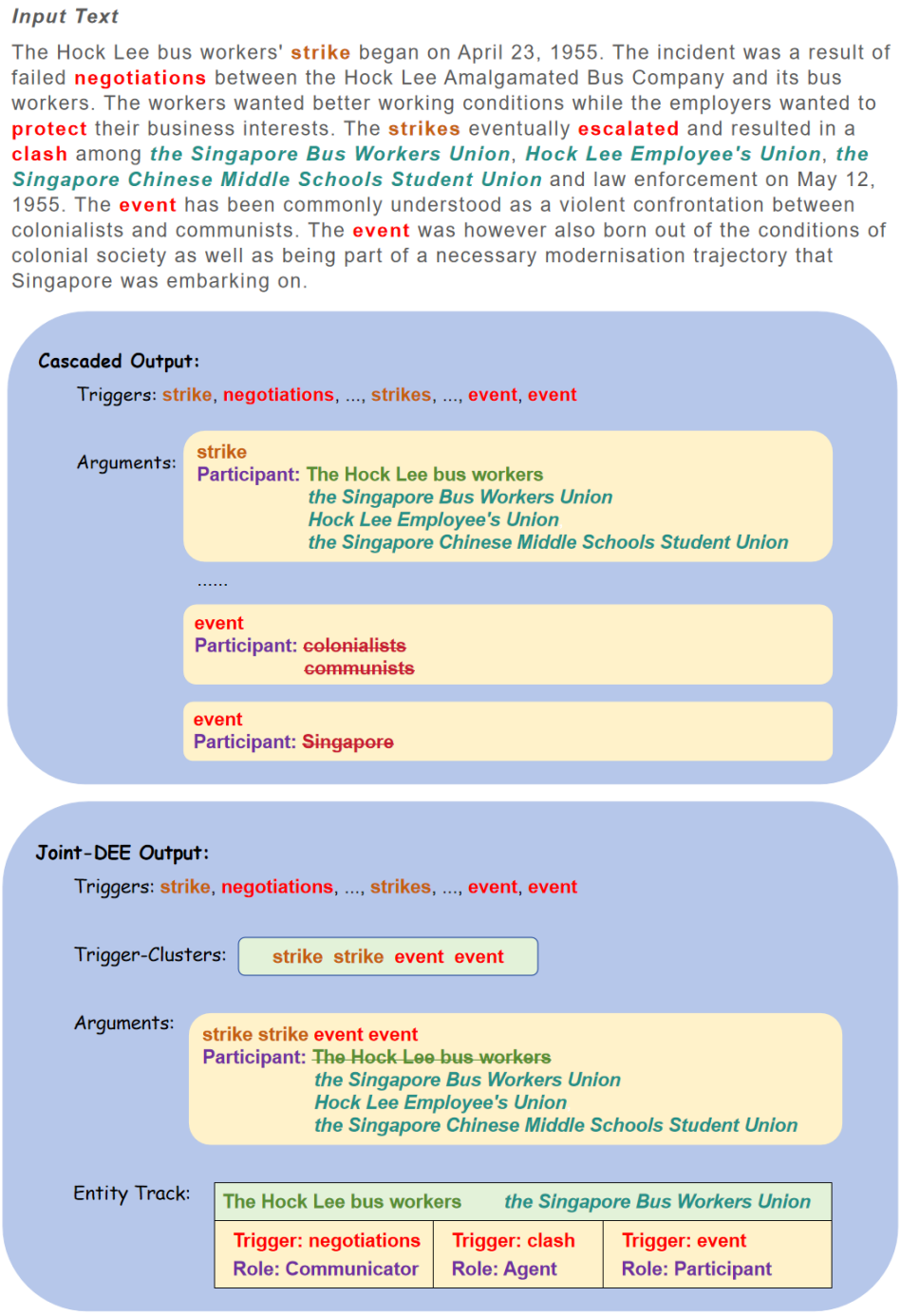

Figure 4: Case study comparing cascaded extraction and Joint-DEE. In the cascaded framework, the trigger "strike" is not correctly classified, and the downstream argument extractor continues to extract arguments independently based on this noisy upstream prediction. Meanwhile, the generic trigger "event" fails to capture cross-sentence arguments. In contrast, Joint-DEE links "strike" and "event" through trigger coreference, allowing arguments located near "strike" to be aggregated into the event-level output. The entity tracking module further aggregates document-level entity mentions and their global relations with triggers, producing a more coherent event structure.

Setting	Method	WikiEvents			MAVEN-Arg		
		TC-F1	AC-F1	ACo-F1	TC-F1	AC-F1	ACo-F1
Non-topic-centric	Tag-Prime+Tag-Prime	50.2	23.2	24.6	66.1	21.9	23.0
	Tag-Prime+PAIE	50.2	41.7	43.8	66.1	22.4	24.1
	Llama3-8B-SFT	40.5	20.5	21.3	58.3	17.2	18.4
	GPT-4-QA	35.6	18.8	20.2	42.4	14.7	16.3
	Joint-DEE(Ours)	**50.4**	**42.1**	**44.5**	**66.3**	**24.4**	**26.7**
Topic-centric	Tag-Prime+Tag-Prime	61.2	44.9	45.4	72.5	25.7	26.6
	Tag-Prime+PAIE	61.2	47.7	49.5	72.5	26.4	28.7
	Llama3-8B-SFT	45.2	31.0	38.4	60.6	17.5	20.3
	GPT-4-QA	38.5	20.6	22.7	47.8	17.9	19.2
	Joint-DEE(Ours)	**61.2**	**48.9**	**51.1**	**72.5**	**27.2**	**29.6**

Table 8: Additional comparison with cascaded and LLM-based baselines under non-topic-centric and topic-centric settings.

Framework	Setting	WikiEvents			MAVEN-Arg		
		TriCov	EntCov	Arg/Pair Cov	TriCov	EntCov	Arg/Pair Cov
Cascaded-DEE	Non-topic-centric	66.8	–	50.6	68.3	–	36.7
	Topic-centric	73.5	–	57.8	75.1	–	41.5
Joint-DEE	Non-topic-centric	75.4	93.6	54.9	73.6	66.4	41.2
	Topic-centric	77.5	93.7	58.3	76.8	66.4	44.2

Table 9: Candidate coverage analysis on WikiEvents and MAVEN-Arg. TriCov measures the coverage of gold trigger spans. EntCov measures the coverage of gold entity spans and is not reported for Cascaded-DEE because it does not explicitly produce entity candidates. For Cascaded-DEE, Arg/Pair Cov denotes ArgReachCov, i.e., the proportion of gold trigger–argument pairs whose triggers are preserved by upstream extraction. For Joint-DEE, Arg/Pair Cov denotes PairCov, i.e., the proportion of gold trigger–argument pairs for which both the trigger and the argument entity are retained as candidates.

is exposed to a slightly broader set of gold-relevant structures, which provides better conditions for modeling trigger coreference, entity coreference, and trigger–argument relations.

G Detailed Ablation Analysis

We provide a detailed analysis of the ablation results in Table 4. Across both datasets and both evaluation settings, removing core components consistently weakens the performance of Joint-DEE, with the most visible drops appearing in argument-related metrics.

First, bidirectional attention plays an important role in document-level contextual modeling. Under the non-topic-centric setting, removing bidirectional attention decreases AC-F1 by 2.4 points on WikiEvents and 0.7 points on MAVEN-Arg. Under the topic-centric setting, the same ablation also reduces AC-F1 by 1.6 points on WikiEvents and 1.5 points on MAVEN-Arg. These results indicate that access to both left and right contexts is important for identifying argument structures in long documents.

Second, the relation matching components provide complementary benefits. Removing span-level matching leads to consistent drops in both AC-F1 and ACo-F1, suggesting that span-level interaction is useful for capturing structured relations between candidate triggers and entities. Trigger coreference matching is particularly helpful for maintaining event-level consistency. For example, on MAVEN-Arg, removing trigger coreference reduces AC-F1 and ACo-F1 under both settings, indicating that linking coreferential trigger mentions helps argument assignment in dense event structures. Entity coreference matching also contributes substantially to argument consistency, especially on MAVEN-Arg, where removing it causes clear drops in AC-F1 and ACo-F1. This shows that aggregating entity mentions is important for assigning consistent roles across the document.

Third, entity tracking improves the model by encouraging entity-centered document-level aggregation. Removing entity tracking consistently hurts AC-F1 and ACo-F1. In the topic-centric setting, for instance, removing entity tracking reduces AC-F1 from 48.9 to 47.4 and ACo-F1 from 51.1 to 49.9 on WikiEvents, and reduces AC-F1 from 27.2 to 26.2

and ACo-F1 from 29.6 to 28.8 on MAVEN-Arg. These results suggest that entity tracking helps the model maintain more stable argument representations and improves role assignment across events.

Overall, the ablation results show that the three main designs of Joint-DEE are complementary. Bidirectional attention improves document-level representation, relation matching organizes trigger and entity candidates into structured event relations, and entity tracking strengthens entity-centered aggregation. Their combination leads to the best performance across both topic-centric and non-topic-centric settings.

H Training Dynamics Analysis

Trigger–Entity Training Dynamics. We first analyze the training behavior of trigger and entity extraction under joint and separate training on WikiEvents dataset. As shown in Figure 5, both tasks improve steadily during training. The separately trained curves show that trigger and entity extraction can be optimized stably with their own task-specific supervision, while the jointly trained curves remain comparable and do not show clear mutual interference. This suggests that introducing parallel trigger and entity extraction does not destabilize either task, providing a reliable candidate span space for subsequent relation matching.

Span–Token Consistency. Joint supervision at the span and token levels enhances relation learning by providing complementary signals. As shown in Figure 6, span-level trigger coreference F1 steadily improves on MAVEN-Arg dataset when token-level supervision is included, whereas span-only training yields lower performance, especially in later epochs. Token-level F1 also benefits from joint learning. These trends indicate that token-level supervision improves the consistency and effectiveness of span-level relation learning.

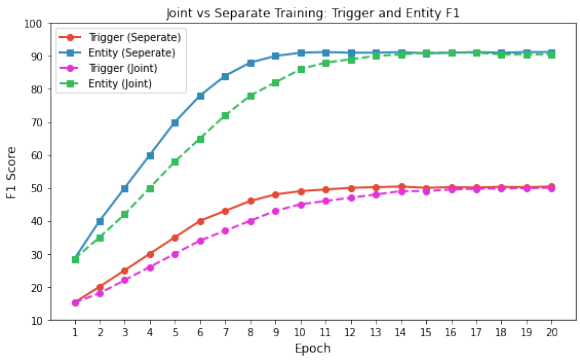

Figure 5: Scores of trigger and entity extraction during joint and separate training. Solid lines represent joint training, while dashed lines represent separate training.

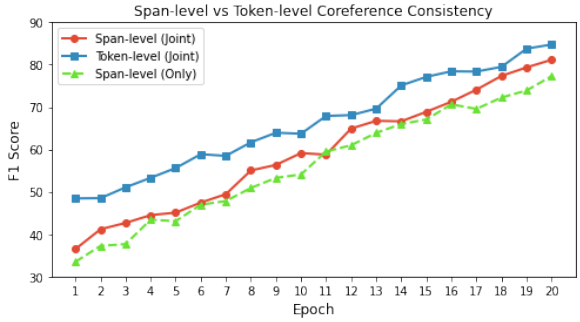

Figure 6: Training curves of span-level and token-level coreference F1. Solid lines show joint span-token supervision, while the dashed line shows span-level training alone.

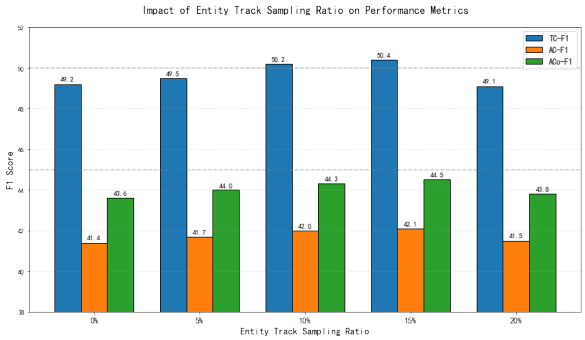

Figure 7: Performance comparison of event extraction metrics across different entity tracking sampling ratios. The best overall performance is achieved at a 15% sampling ratio.

Impact of Entity Tracking Sampling Ratio. Entity tracking is effective when applied with a moderate sampling ratio. As shown in Figure 7, increasing the sampling ratio from 0% to 15% on WikiEvents empirically improves TC-F1, AC-F1, and ACo-F1, indicating that entity tracking provides useful contextual signals for trigger identification, argument extraction, and argument coreference. However, further increasing the ratio leads to performance degradation. This suggests that excessive entity tracking may introduce redundant or noisy supervision. We therefore use a sampling ratio of 15%, which provides the best balance between auxiliary supervision and training noise.