Large Language Model-Based Event Relation Extraction with Rationales

Zhilei Hu, Zixuan Li, Xiaolong Jin*, Long Bai, Jiafeng Guo and Xueqi Cheng
CAS Key Laboratory of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences;
School of Computer Science and Technology, University of Chinese Academy of Sciences
{huzhilei19b, lizixuan, jinxiaolong, bailong}@ict.ac.cn
{guojiafeng, cxq}@ict.ac.cn

Abstract

Event Relation Extraction (ERE) aims to extract various types of relations between different events within texts. Although Large Language Models (LLMs) have demonstrated impressive capabilities in many natural language processing tasks, existing ERE methods based on LLMs still face three key challenges: (1) **Time Inefficiency**: The existing pairwise method of combining events and determining their relations is time-consuming for LLMs. (2) **Low Coverage**: When dealing with numerous events in a document, the limited generation length of fine-tuned LLMs restricts the coverage of their extraction results. (3) **Lack of Rationale**: Essential rationales concerning the results that could enhance the reasoning ability of the model are overlooked. To address these challenges, we propose LLMERE, an LLM-based approach with rationales for the ERE task. LLMERE transforms ERE into a question-and-answer task that may have multiple answers. By extracting all events related to a specified event at once, LLMERE reduces time complexity from $O(n^2)$ to $O(n)$, compared to the pairwise method. Subsequently, LLMERE enhances the coverage of extraction results by employing a partitioning strategy that highlights only a portion of the events in the document at a time. In addition to the extracted results, LLMERE is also required to generate corresponding rationales/reasons behind them, in terms of event coreference information or transitive chains of event relations. Experimental results on three widely used datasets show that LLMERE achieves significant improvements over baseline methods[1].

1 Introduction

Event Relation Extraction (ERE) is a crucial task in natural language processing and information extraction, which aims to identify the types of relations between events in texts, such as temporal, causal, subevent, and coreference. For example, in Figure 1, an ERE model needs to identify all existing event relations within a document. ERE plays a significant role in establishing extensive connections among events, which benefits various practical applications, including event prediction (Chaturvedi et al., 2017; Bai et al., 2021), reading comprehension (Berant et al., 2014), and question answering (Oh et al., 2017).

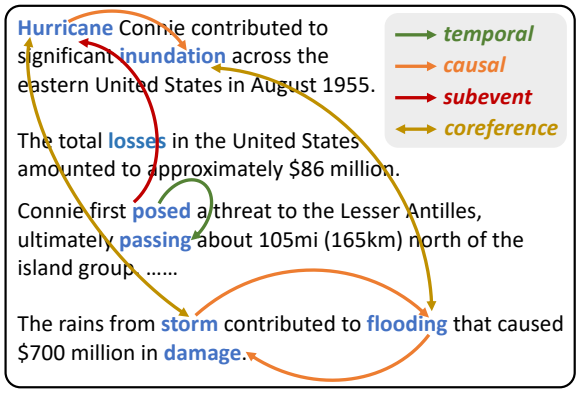


Figure 1: An example of the ERE task. Lines indicate the relations between events.

Previous methods (Wen and Ji, 2021; Chen et al., 2022) primarily rely on Pre-trained Language Models (PLMs) to encode documents. Due to the length limitation of the input sequence (e.g., BERT has a maximum encoding length of 512 tokens), PLMs are unable to encode an entire document at once, leading to challenges in capturing long-range semantic dependencies. Currently, Large Language Models (LLMs) have demonstrated impressive capabilities in understanding the semantics of long texts (Peng et al., 2023; Dong et al., 2024). Therefore, some methods (Yuan et al., 2023; Yu et al., 2023) attempt to utilize LLMs to accomplish the ERE task.

However, existing LLM-based ERE methods still face three key challenges: (1) **Time Inef-**

*Corresponding author.

[1]The source code is available at https://github.com/HerbertHu/LLMERE

Proceedings of the 31st International Conference on Computational Linguistics, pages 7484–7496
January 19–24, 2025.

ficiency: Reasoning with LLMs is highly time-consuming, as determining the relations between all event pairs requires significant computational resources. (2) **Low Coverage**: The limited generation length of fine-tuned LLMs restricts the coverage of their extraction results. As shown in Figure 2, when the number of events in a document is large, LLMs can only extract a portion of the event relations, resulting in limited coverage. (3) **Lack of Rationale**: Essential rationales concerning the results that could enhance the reasoning ability of the model are overlooked.

To address the aforementioned challenges, this paper proposes an LLM-based ERE method with rationales, called LLMERE. For the first challenge, LLMERE transforms ERE into a question-and-answer task with multiple possible answers, where LLMs answer the question of which events are related to a specific event within a given document. By doing so, LLMERE significantly reduces the time complexity of training and inference from $O(n^2)$ to $O(n)$, compared to the existing pairwise methods. For the second challenge, we propose a partitioning strategy to reduce the generation length of the output for each sample. Specifically, the document is duplicated, with each copy highlighting a subset of the candidate events. The model focuses on generating answers from a different portion of events each time. For the third challenge, in addition to the extracted event relation results, we also require the model to generate corresponding rationales behind the relations, such as event coreference information and event relation transitive chains that adhere to logical rules. By learning from these rationales, the model can develop reasoning abilities that enable it to infer more accurate results.

In general, the main contributions of this paper can be summarized as follows:

- We propose an LLM-based method for ERE (LLMERE). It utilizes a question-and-answer format that enumerates multiple answers to reduce the task complexity from $O(n^2)$ to $O(n)$.
- A partitioning strategy is proposed to address the problem of limited coverage in extraction results when the number of events in the document is large.
- LLMERE is required to generate rationales that could enhance its reasoning capabilities, including event coreference information and event relation transitive chains.
- According to experimental results on three widely used datasets, LLMERE achieves significant improvements in the F1 score compared to the state-of-the-art baselines.

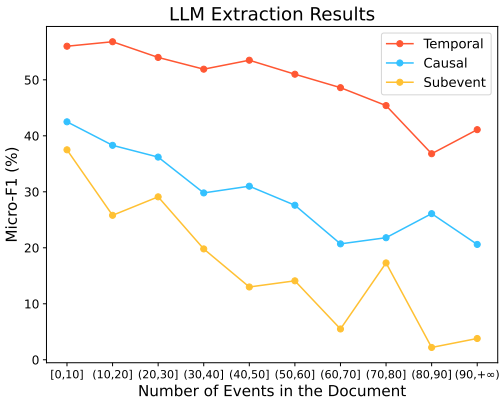


Figure 2: The ERE results of a fine-tuned LLM on the MAVEN-ERE dataset, across documents with varying numbers of events.

2 Related Work

Since the fundamental role of event relations in natural language processing, extracting event relations has attracted extensive attention in the past few years. Early methods (Riaz and Girju, 2013, 2014) rely on lexical and syntactic features to determine event relations. Subsequently, some methods (Wen and Ji, 2021; Zhou et al., 2022) leverage PLMs to encode text, obtaining semantic representations of event pairs for classification. Some methods (Phu and Nguyen, 2021; Fan et al., 2022) construct event graphs, treating events within the document as nodes and modeling the interactions among them. In addition, since PLMs are unable to process entire documents at once, some methods (Man et al., 2022; Guan et al., 2024) attempt to select important contexts.

Recently, LLMs have demonstrated impressive performance in understanding the semantics of long documents. To investigate the performance of LLMs in various ERE tasks, several studies (Yuan et al., 2023; Yu et al., 2023; Chen et al., 2024; Zhang et al., 2024) conduct evaluations. The results of these studies indicate that, under few-shot settings, directly using LLMs does not effectively identify relations between events. Moreover, these

methods adopt a pairwise approach to determine relations between each pair of events, resulting in excessively high task complexity. Therefore, we propose an ERE method tailored for LLMs, which aims to reduce task complexity while improving extraction performance.

Furthermore, some studies (Wiegreffe et al., 2021; Wei et al., 2022) have demonstrated that rationales can improve LLMs by generating high-quality reasoning steps to explain their predictions. In the ERE task, event coreference information and event relation transitive chains can provide detailed explanations for the extracted event relations. Therefore, we leverage them as rationales to enhance the reasoning capabilities of the model.

3 The LLMERE Model

In this section, we introduce the proposed LLMERE model. First, we construct input-output training data using the method described below and then utilize these data to train the model. Figure 3 illustrates the process of constructing the training data. In the inference phase, only the inputs need to be given and the model will generate the corresponding outputs. Next, we sequentially introduce the Document Partitioning, Context Input, Model Output, and Instruction Tuning of the model.

3.1 Document Partitioning

In the preliminary experimental results, we observe that as the number of events in the document increases, the performance of the LLM gradually deteriorates. This is because the limited generation length of fine-tuned LLMs restricts the coverage of their extraction results. To alleviate this problem, we propose a partitioning strategy. Specifically, we count the number of events n in the document and set a threshold k for the maximum number of annotated events allowed per document. After applying the partitioning strategy, the number of document replications m is calculated as follows: $m = \lceil \frac{n-1}{k} \rceil$, where $\lceil \rceil$ denotes the ceiling function. Subsequently, as shown on the left side of Figure 3, we specify one event and then uniformly and randomly annotate other events across m documents. Uniform partitioning helps avoid issues with uneven event distribution in the partitioned documents. Random partitioning can enhance the ability of the model to recognize events located at different positions within the document, preventing it from overly focusing on a single text segment. It is worth noting that the partitioning strategy is applied during both the training and testing phases. During the testing phase, the extraction results from multiple samples are merged to obtain all events related to the specified event.

3.2 Context Input

The input part of the data provides the necessary contextual information, which primarily consists of three components: Task description, Document, and Instruction. As shown in the middle section of Figure 3, these three components are concatenated.

Task Description. Since LLMs are significantly influenced by the content of the prompts, we provide a detailed task description to thoroughly explain the task being performed. We treat the four types of event relations (temporal, causal, subevent, and coreference) as distinct subtasks, each with its own task description. The model adaptively performs the corresponding subtask based on the provided task description each time. For each subtask, event relations are further divided into multiple subtypes. Existing multi-turn question-answering methods for relation extraction (Li et al., 2019) require specifying a head entity first, and then extracting the results for each subtype during the multi-turn process. In contrast, LLMERE extracts the results for all subtypes of the task in a single step. This approach has two advantages: first, it increases extraction efficiency; second, the model is able to make global reasoning, avoiding the identification of multiple conflicting relations for the event pair (e_A, e_B). Subsequently, to make it easier for the model to identify event locations, the method for annotating events in the document is provided. To facilitate the subsequent analysis of the output results, we specify a detailed expected output format. A well-defined task description enables the LLM to better understand the task being performed, thereby enhancing its performance. Detailed task descriptions can be found in Appendix A.

Document. To better understand the global semantics of the document, the entire content of the document is utilized as input. Additionally, we construct samples using the partitioned documents. To more accurately detect the location of events and indicate them in the extraction results, special symbols are used to annotate the events. Specifically, we use angle brackets to enclose event mentions and label them with sequential numbers, for example, *<e0 Hurricane> <e1 inundation>*.

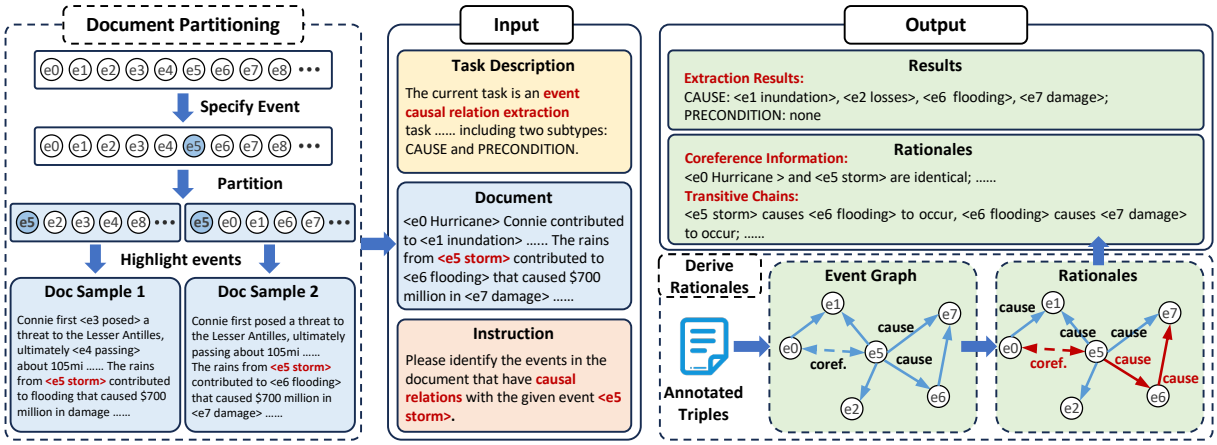


Figure 3: The construction process of LLMERE training data. Training data includes input and output.

Instruction. To reduce the complexity of using LLMs for ERE, we specify a particular event and prompt the model to enumerate all other events that are related to it.

3.3 Model Output

In the output section, the model is required not only to generate the extraction results regarding event relations but also to provide corresponding rationales for these results.

3.3.1 Extraction Result

In this section, since a single event from the sampled document partition is specified in the prompt, the model only needs to identify other events that are related to it. Furthermore, for each ERE subtask, the model extracts results for all relation types in a single inference process. If none of the events in the document are related to the specified event, the output should be "none". For example, in the event causal relation extraction subtask, as shown on the right side of Figure 3, the extraction result is "CAUSE: <e1 inundation>, <e2 losses>, <e6 flooding>, <e7 damage>; PRECONDITION: none". "CAUSE" and "PRECONDITION" represent two subtypes of the causal relation.

3.3.2 Rationales

The rationales regarding the results helps the model better understand the relations between events. These rationales primarily include event coreference information and event relation transitive chains that adhere to logical rules. By enabling the model to learn these rationales while extracting results, its reasoning capabilities can be enhanced, leading to more accurate extraction results. All rationales are derived from existing datasets. Event coreference information is contained within the dataset. The event relation transitive chains are derived from the original annotated data using logical rules.

Event Coreference. When multiple event mentions in the text refer to the same event, these mentions exhibit coreference relations. Event coreference relations connect scattered events throughout the document. Identifying coreference relations between events helps in comprehensively understanding the content of the entire document. We represent event coreference relations in natural language, for example, "<e5 storm> and <e0 Hurricane> are identical." For the event coreference relation extraction subtask, the model only outputs the extraction results.

Transitive Chains. When determining the relations between a specified event and other events, some relations are difficult to directly infer and require leveraging transitive chains of event relations for assistance. Therefore, we design a method to automatically generate transitive chains and utilize logical rules to verify the validity of these chains. First, we extract all event relation triples from the annotated data and then convert these triples into an event graph. After that, we designate the specified event as the starting node of the path and set the first-order neighbors of the specified event as the ending nodes of the path. We then search for paths between two nodes in the graph, excluding those paths where the two nodes are directly connected. If the relation between the starting and ending nodes inferred through the transitive chains matches the relation annotated in the original data, we consider such a path to be one that

satisfies the logical rules. For example, the transitive chain, <e5 *storm*> $\xrightarrow{cause}$ <e6 *flooding*> $\xrightarrow{cause}$ <e7 *damage*> is a path that satisfies the logical rules, as it yields a result consistent with the labeled data, namely <e5 *storm*> $\xrightarrow{cause}$ <e7 *damage*>. If multiple paths exist between two nodes, we randomly retain one to avoid information redundancy. Next, we convert all transitive paths into textual descriptions to facilitate the understanding of the model. Finally, we train the model to learn from the transitive chains, thereby enhancing its reasoning capabilities. Detailed transitivity logic rules can be found in Appendix B. We did not introduce transitive chains for temporal data, because the temporal graph is dense, and doing so would introduce additional noise.

3.4 Instruction Tuning and Evaluation

After constructing the training dataset, we use instruction tuning to train the LLMERE. The LLaMA-Factory (Zheng et al., 2024) framework is employed to train the model. The base models used for fine-tuning are primarily the LLaMA2-7B (Touvron et al., 2023) and LLaMA3-8B (Grattafiori et al., 2024) series. The Lora (Hu et al., 2022) technique is adopted for parameter-efficient fine-tuning. The model training utilizes the cross-entropy loss function.

Multi-task Joint Training. As mentioned in Section 3.2, we treat the four types of event relations as four distinct subtasks, resulting in four separate sets of training data. To enable an LLM to extract multiple types of event relations, we combine these four separate sets of training data and perform joint training.

Negative Sample Sampling. Due to the sparsity of relations between events, many events within a document exist independently and have no connection to other events. We consider such instances as negative samples, while others are treated as positive samples. This leads to an imbalance between positive and negative samples. To mitigate this issue, we employ a negative sample sampling strategy, retaining only a portion of the negative samples.

Evaluation. During the testing phase, to evaluate LLMERE on unseen documents, these input documents only include event trigger annotations without event relations, and thus do not contain any rationales about these relations.

Dataset	#Doc.	#Events	#Events/Doc.
MAVEN-ERE	4,480	112,276	25.1
MATRES	275	11,861	43.1
HiEve	100	3,185	31.9

Table 1: Statistics of the datasets.

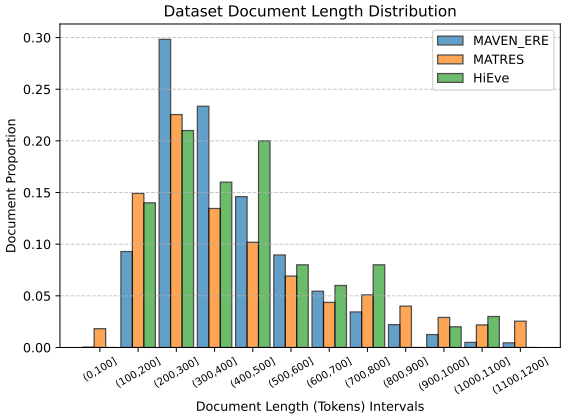


Figure 4: Document length distribution in the datasets.

4 Experiments

4.1 Datasets and Metrics

Datasets. We utilize three commonly used benchmark datasets to evaluate the proposed method: MAVEN-ERE, MATRES, and HiEve. The statistics of these datasets are presented in Table 1. **MAVEN-ERE** (Wang et al., 2022) is a unified large-scale dataset for the ERE task, annotated on English Wikipedia documents in the general domain. This dataset is annotated with four major types of event relations, i.e., temporal, causal, subevent, and coreference. It contains 103,193 events coreference chains, 1,216,217 temporal relations, 57,992 causal relations, and 15,841 subevent relations. The annotation scale of this dataset is at least an order of magnitude larger than that of any existing datasets for the ERE task. **MATRES** (Ning et al., 2018) is a commonly used dataset for evaluating event temporal relation extraction. The documents in the MATRES primarily originate from news reports, with annotations limited to verb events. It contains 13,573 temporal relations. **HiEve** (Glavaš et al., 2014) is a commonly used dataset for evaluating subevent relation extraction. The documents in the HiEve primarily consist of news reports, with manually annotated events that have actually occurred. It contains 3,648 subevent relations. The distribution of document lengths (tokens) in these datasets is presented

in Figure 4. All datasets are publicly accessible[2]. The details of the dataset partitioning are presented in Appendix C.

Metrics. For the tasks of temporal, causal, and subevent relation extraction, we adopt the standard micro-averaged Precision (P), Recall (R), and F1-score as evaluation metrics. For event coreference resolution, following previous works (Wang et al., 2022), we adopt MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), $CEAF_e$ (Luo, 2005) and BLANC (Recasens and Hovy, 2011) metrics. Due to space limitations, we only report the average F1-score for all coreference metrics. We conduct significance testing at the level of 0.05 for all of our experiments.

4.2 Experimental Setup

Implementation Details. LLMERE is a method suitable for generative LLMs, and its backbone can be replaced. The ratios of positive to negative samples for temporal, causal, subevent, and coreference relations are set at 4:1, 1:1, 2:3, and 2:3, respectively. The Lora rank is set to 64, and the maximum sequence length is set to 2048. The threshold k for the number of annotated events per document is set to 30. The model is optimized with a learning rate of 2e-4, and the learning rate scheduler employs a cosine function. The model is trained for 3/10/10 epochs on MAVEN-ERE/MATRES/Hieve, respectively. All the LoRA parameters are trained on an NVIDIA A100 GPU with 40GB memory.

Baseline Methods. In prior research, event relation extraction tasks about different types are regarded as distinct tasks, leading to inconsistent baselines across different datasets. We compare two types of methods: one is classification-based, while the other is generation-based.

Classification-based: For MAVEN-ERE, **ERGO** (Chen et al., 2022) builds a relational graph to model interactions between event pairs; **Joint** (Wang et al., 2022) is a method that performs joint training for multiple types of event relations, while **Split** trains each type of event relations separately; **ProtoERE** (Hu et al., 2023) models the prototypes of event relation types. The detailed baselines for the MATRES and HiEve datasets are presented in Appendix D.

Generation-based: Yuan et al. (2023) and Wei et al. (2024) conduct experiments on the MATRES and MAVEN-ERE datasets using the official API of OpenAI[3]. ChatGPT (Zero-shot) and ChatGPT (CoT) denote reasoning methods utilizing zero-shot and chain-of-thought (Wei et al., 2022), respectively. Doc-SFT refers to fine-tuning LLMs to directly extract all event relation triples from the given document.

4.3 Experimental Results

Table 2 presents the experimental results on the MAVEN-ERE dataset. Overall, our method outperforms both classification-based and generation-based approaches.

Compared to the classification-based SOTA method ProtoERE, LLMERE improves the overall F1 score by 1.7%, achieving superior results. Additionally, LLMERE exhibits outstanding performance in causal relation extraction, with a 4.2% improvement over the SOTA method. This indicates that LLMs contain rich causal knowledge, and after fine-tuning, they can more easily determine causal relations between events.

For generation-based methods, we can observe that even advanced models like GPT-4, under a 5-shot setting, fail to recognize event relations effectively. This underscores the necessity of fine-tuning LLMs. After fine-tuning, the ability of the model to recognize event relations improves significantly. The overall F1 score of LLMERE is 29.2% higher than that of GPT-4 and 13.1% higher than that of Doc-SFT. Additionally, compared to pairwise methods, LLMERE reduces the time complexity of the task. This indicates that our method achieves a favorable balance between task complexity and extraction performance.

To investigate the impact of language model backbones on experimental results, we conduct experiments using various backbones. As shown in Table 2, using the base versions of the models (LLaMA2-7B-base, LLaMA3-8B-base) yields better performance compared to using the fine-tuned models (LLaMA2-7B-chat, LLaMA3-8B-instruct). This is because their fine-tuning formats differ significantly from the format of our ERE task. Additionally, using language models with better foundational capabilities yields better results (e.g., LLaMA2-7B-base vs. LLaMA3-8B-base).

Tables 3 and 4 present the experimental results on the MATRES and HiEve datasets, respectively. Compared to the SOTA model, LLMERE improves the F1 score by 1.5% on the MATRES dataset and

[2] https://github.com/THU-KEG/MAVEN-ERE

[3] https://platform.openai.com/docs/models

Model	Language Model	Temporal			Causal			Subevent			Coref.	Overall
		P	R	F1	P	R	F1	P	R	F1	F1	F1
Classification-based methods												
ERGO (Chen et al., 2022)	RoBERTa-base	50.3	52.1	51.2	31.5	25.2	28.0	26.9	18.4	21.8	89.3	47.6
Joint (Wang et al., 2022)	RoBERTa-base	49.4	56.0	52.5	32.8	27.5	29.9	27.3	19.6	22.7	90.4	48.8
Split (Wang et al., 2022)	RoBERTa-base	49.5	55.6	52.4	32.7	26.8	29.4	26.7	21.8	23.9	90.4	49.0
ProtoERE (Hu et al., 2023)	RoBERTa-base	48.9	59.9	53.8	32.5	31.3	31.8	26.2	29.7	27.9	89.8	50.8
Generation-based methods												
Llama2 (5-shot) (Wei et al., 2024)	Llama2-7b-chat	13.5	2.7	4.5	0.0	0.0	0.0	0.0	0.0	0.0	60.8	16.3
ChatGPT (5-shot) (Wei et al., 2024)	gpt-3.5-turbo	16.3	8.0	10.8	3.9	4.8	4.3	0.0	0.0	0.0	60.9	19.0
GPT4 (5-shot) (Wei et al., 2024)	gpt-4	21.6	11.6	15.1	10.4	6.0	7.6	1.9	2.4	2.1	68.4	23.3
Doc-SFT	Llama2-7b-base	35.9	18.5	24.4	25.8	30.0	27.7	19.8	27.2	23.0	82.5	39.4
LLMERE (Ours)	Llama2-7b-chat	50.1	57.9	53.7	34.1	35.0	34.6	23.6	26.6	25.0	**91.1**	51.1
LLMERE (Ours)	Llama2-7b-base	51.0	58.1	54.3	34.7	36.4	35.6	24.2	30.6	27.0	90.7	51.9
LLMERE (Ours)	Llama3-8b-instruct	50.5	59.9	**54.8**	34.9	36.1	35.5	25.4	26.0	25.7	90.7	51.7
LLMERE (Ours)	Llama3-8b-base	50.1	60.2	54.7	35.0	37.2	**36.0**	26.0	30.8	**28.2**	90.9	**52.5**

Table 2: Experimental results (%) on the MAVEN-ERE dataset. Precision, Recall, and F1-score are denoted by P, R, and F1, respectively. The best results are highlighted in **bold**, and the second-best results are underlined.

Method	P	R	F1
CSE+ILP (Ning et al., 2019)	71.3	82.1	76.3
Deep (Han et al., 2019)	77.4	86.4	81.7
Stack-P (Wen and Ji, 2021)	78.4	85.2	81.7
TIMERS (Mathur et al., 2021)	81.1	84.6	82.3
SCS-EERE (Man et al., 2022)	78.8	88.5	83.4
RSGT (Zhou et al., 2022)	82.2	85.8	84.0
ChatGPT (Zero-shot)	26.4	24.3	25.3
ChatGPT (CoT)	48.0	57.7	52.4
LLMERE (Llama2-7b)	**82.9**	87.6	85.2
LLMERE (Llama3-8b)	82.6	**88.7**	**85.5**

Table 3: Experimental results (%) on MATRES.

Method	P	R	F1
BERT (Devlin et al., 2019)	19.8	15.2	16.3
RoBERTa (Liu et al., 2019)	20.2	16.1	17.8
Hierarchical (Adhikari et al., 2019)	21.4	17.3	16.7
SIEF (Xu et al., 2022)	21.8	17.4	18.6
SCS-EERE (Man et al., 2022)	20.6	19.7	19.2
TacoERE (Guan et al., 2024)	**22.6**	19.5	20.8
LLMERE (Llama2-7b)	18.2	30.9	22.9
LLMERE (Llama3-8b)	20.0	**35.6**	**25.6**

Table 4: Experimental results (%) on HiEve.

4.8% on the HiEve dataset, with the recall score increasing by 2.9% and 6.1%, respectively. This indicates that LLMERE possesses more background knowledge, allowing it to identify more potential event relations. Compared to ChatGPT (CoT), LLMERE achieves a 33.1% improvement in F1 score on the MATRES dataset, further demonstrating that existing general-purpose LLMs lack the ability to accurately infer event relations and require domain-specific fine-tuning.

4.4 Ablation Studies

To demonstrate the impact of each component on the experimental results, we conduct ablation experiments on the MAVEN-ERE dataset. The experimental results are shown in Table 5. *coref.* represents the coreference information, *trans.* indicates the event relation transitive chains, *partition* represents the partitioning strategy. *Multitask-once* indicates that LLMs extract all four types of event relations in a single step, rather than extracting them separately.

Impact of the Rationales. Comparing *-coref.* with LLMERE, the F1 scores for temporal, causal, and subevent relations decrease by 0.5%, 1.0%, and 1.2%, respectively. This suggests that enabling the model to learn coreference relations between events helps improve the extraction of other event relations. Comparing *-trans.* with LLMERE, the F1 scores for causal and subevent relations drop by 0.7% and 1.7%, respectively. This indicates that the model can enhance its reasoning abilities by learning the transitive chains of event relations, leading to more accurate results. Moreover, the two kinds of rationales are complementary, and learning them together can achieve better results.

Impact of the Partitioning Strategy. Comparing *−Rationale & Partition* with *−Rationale(all)*, the F1 scores for temporal, causal, and subevent relations decrease by 3.4%, 1.6%, and 2.9%, respectively. This indicates that the partitioning strategy is an effective method for enhancing the extraction performance of LLMs.

Impact of the Multi-task Training. Comparing *Multitask-once* with *−Rationale & Partition*,

Model	Temporal			Causal			Subevent			Coref.
	P	R	F1	P	R	F1	P	R	F1	F1
LLMERE	51.0	58.1	54.3	34.7	36.4	35.6	24.2	30.6	27.0	90.7
-Rationale (coref.)	50.8	57.1	53.8 (-0.5)	35.5	33.6	34.6 (-1.0)	23.4	28.7	25.8 (-1.2)	90.9 (+0.2)
-Rationale (trans.)	51.2	58.0	54.4 (+0.1)	33.9	36.0	34.9 (-0.7)	22.2	29.4	25.3 (-1.7)	90.7 (-0.0)
-Rationale (all)	51.1	56.5	53.7 (-0.6)	34.4	33.6	34.0 (-1.6)	21.8	29.1	24.9 (-2.1)	90.7 (-0.0)
-Rationale & Partition	52.0	48.7	50.3 (-4.0)	31.0	34.0	32.4 (-3.2)	30.1	17.3	22.0 (-5.0)	90.7 (-0.0)
Multitask-once	51.9	41.1	45.9 (-8.4)	33.6	26.8	29.8 (-5.8)	24.7	13.3	17.3 (-9.7)	90.2 (-0.5)

Table 5: Ablation results (%) on the MAVEN-ERE dataset. The values in parentheses indicate the changes in F1 scores relative to LLMERE.

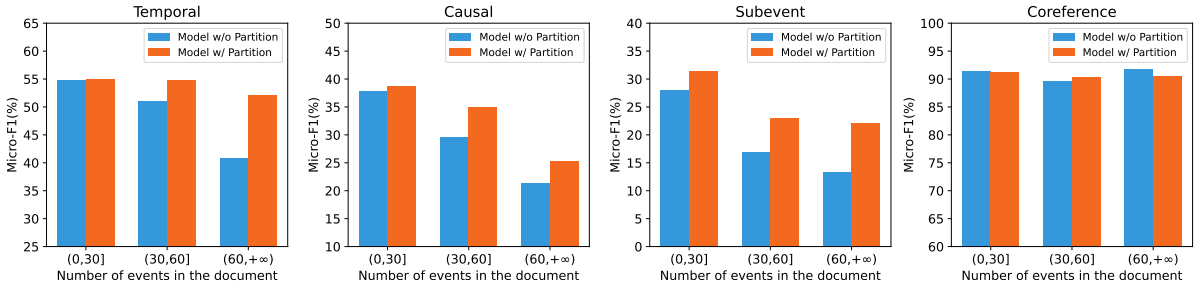


Figure 5: Experimental results (%) on MAVEN-ERE for documents with different numbers of events.

Model	Temporal	Causal	Subevent	Coref.
	F1	F1	F1	F1
No Rationales	53.7	34.0	24.9	90.7
Before	54.3	33.4	25.1	90.3
After	54.3	35.6	27.0	90.7

Table 6: Experimental results (%) under different locations of rationales on MAVEN-ERE. "Before" indicates that the rationales appear before the extracted results in the output list, whereas "After" indicates the opposite.

the F1 scores for temporal, causal, and subevent relationships decrease by 4.4%, 2.6%, and 4.7%, respectively. This indicates that for LLMs, extracting all four types of event relations simultaneously is challenging. Performing each ERE subtask separately yields better results.

4.5 Time Complexity Analysis

To demonstrate the effectiveness of LLMERE, we compare its time complexity with that of the pairwise method. Assuming there are n events in the document and t ERE subtasks to be performed. The number of inferences for the pairwise method is $t \times n(n-1)$, and its time complexity is $O(t \cdot n^2)$. In contrast, the LLMERE requires $t \times m \times n$ inferences, where m represents the number of partitioned documents and $m \ll n$. Therefore, the time complexity of LLMERE is $O(t \cdot n)$.

We conduct experiments on 50 documents, with the average inference times per document for the pairwise method and our method being 90.3 seconds and 14.4 seconds, respectively, which demonstrates the effectiveness of our method.

4.6 Detail Analysis

Partitioning Strategy. To investigate the impact of the partitioning strategy on documents with varying numbers of events, we categorize the documents into three sections based on the number of events. As shown in Figure 5, for temporal, causal, and subevent relations, when the number of events in a document is large, employing the partitioning strategy can significantly improve extraction results. This further demonstrates that the partitioning strategy can effectively address the issue of incomplete extraction results in LLMs. Regarding coreference relations, due to the relatively small variation in results, the overall F1 score remains unchanged at 90.7%.

Location of the Rationales. To investigate the impact of the location of rationales, we place it either before or after the extracted results. The "Before" format is a method resembling the CoT, where the model initially provides some reasoning cues and then summarizes conclusions based on this information. In the "After" format, the model first presents the answer and then provides the necessary explanations and justifications for

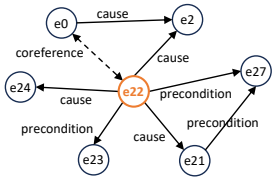
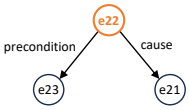
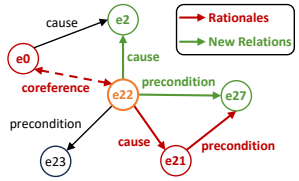
Document and Events	Golden	LLMERE w/o Rationales	LLMERE w/ Rationales
The <e0 storm> 2011 halloween nor'easter, was a large low pressure area that <e1 produced> unusually early <e2 snowfall> across the northeastern United States and the Canadian Maritimes. ..... In all, 3.2 million U.S. residences and businesses in 12 states experienced power <e21 outages>, with the <e22 storm> also <e23 impacting> three Canadian provinces, causing property <e24 damage>. Some customers in Connecticut did not <e25 get> power back until early November; Many communities <e26 chose> to <e27 postpone> celebrations. Please identify the events in the document that have causal relations with the given event <e22 storm>.	CAUSE: <e2 snowfall>, <e21 outages>, <e24 damage>; PRECONDITION: <e23 impacting>, <e27 postpone> Coreference information: <e0 storm> and <e22 storm> are identical Relevant reasoning information: <e22 storm> causes <e21 outages> to occur, <e21 outages> is a precondition for <e27 postpone>	CAUSE: <e21 outages>; PRECONDITION: <e23 impacting>	CAUSE: <e2 snowfall>, <e21 outages>; PRECONDITION: <e23 impacting>, <e27 postpone> **Coreference information**: <e0 storm> and <e22 storm> are identical **Transitive Chains**: <e22 storm> causes <e21 outages> to occur, <e21 outages> is a precondition for <e27 postpone>
			

Figure 6: Results of the case study where <e22 storm> is the specified event.

these answers. The experimental results are shown in Table 6. The performance of "Before" declines compared to "After". One possible reason for this is that some of the reasoning cues generated by the model are incorrect, which will introduce additional noise into the subsequent inference process.

4.7 Case Study

To illustrate how LLMERE improves ERE, a case is studied. Figure 6 shows the specific document content along with the events within the document. Here, LLMERE directly identifies other events that have relations with the specified event *<e22 storm>*. The model sequentially enumerates multiple answers, significantly reducing the complexity of the task from $O(n^2)$ to $O(n)$. In this scenario, some event relations are relatively difficult to determine. For example, *<e22 storm>* and *<e2 snowfall>* are distant from each other in the document, leading the model to assume there is no relation between them. However, (<e22 storm>, coreference, <e0 storm>) and (<e0 storm>, cause, <e2 snowfall>) are easier to detect, allowing the model to infer the "cause" relation between *e22* and *e2*. Moreover, after recognizing the relations (<e22 storm>, cause, <e21 outages>) and (<e21 outages>, precondition, <e27 postpone>), the model is able to infer the relation between *e22* and *e27*. Overall, using rationales forces the model to engage in deeper reasoning, enhancing its inferential capabilities and allowing it to identify relations more accurately.

5 Conclusions

In this paper, we proposed an LLM-based method with rationales for ERE (LLMERE). It utilized a question-and-answer format that enumerates multiple answers to reduce the task complexity from $O(n^2)$ to $O(n)$. Subsequently, a partitioning strategy was introduced to improve the coverage of extraction results. Finally, the model was required to generate rationales beneficial for ERE, further enhancing its extraction capabilities. Experimental results on three widely used datasets demonstrate that LLMERE can effectively perform ERE.

Limitations

For LLMERE, the following limitations exist: (1) LLMERE treats the extraction tasks for different types of event relations as separate subtasks, without considering the interactions between these subtasks. Further exploration is required to enable LLMs to understand the connections between different types of event relations. (2) Currently, LLMERE directly inputs all the content from a document into the LLM, which may introduce noise and redundant information. How to filter the information within the document to facilitate understanding by the LLM is also an important problem. (3) The datasets include only English documents, and the event relations are all within the document.

Acknowledgments

This work is funded by the Lenovo-CAS Joint Lab Youth Scientist Project, the project under Grants No. JCKY2022130C039, and the Strategic Priority Research Program of the CAS under Grants No. XDB0680102, and the National Natural Science Foundation of China under grant 62306299. We thank anonymous reviewers for their insightful comments and suggestions.

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Docbert: Bert for document classi-

fication. *Preprint*, arXiv:1904.08398.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Long Bai, Saiping Guan, Jiafeng Guo, Zixuan Li, Xiaolong Jin, and Xueqi Cheng. 2021. Integrating deep event-level and script-level information for script event prediction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9869–9878, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D. Manning. 2014. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1499–1510, Doha, Qatar. Association for Computational Linguistics.

Snigdha Chaturvedi, Haoruo Peng, and Dan Roth. 2017. Story comprehension for predicting what happens next. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1603–1614, Copenhagen, Denmark. Association for Computational Linguistics.

Meiqi Chen, Yixin Cao, Kunquan Deng, Mukai Li, Kun Wang, Jing Shao, and Yan Zhang. 2022. Ergo: Event relational graph transformer for document-level event causality identification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2118–2128, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Meiqi Chen, Yubo Ma, Kaitao Song, Yixin Cao, Yan Zhang, and Dongsheng Li. 2024. Improving large language models in event relation logical prediction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9451–9478, Bangkok, Thailand. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zican Dong, Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2024. Bamboo: A comprehensive benchmark for evaluating long text modeling capacities of large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2086–2099, Torino, Italia. ELRA and ICCL.

Chuang Fan, Daoxing Liu, Libo Qin, Yue Zhang, and Ruifeng Xu. 2022. Towards event-level causal relation identification. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, pages 1828–1833, New York, NY, USA. Association for Computing Machinery.

Goran Glavaš, Jan Šnajder, Parisa Kordjamshidi, and Marie-Francine Moens. 2014. Hieve: A corpus for extracting event hierarchies from news stories. In *Proceedings of 9th Language Resources and Evaluation Conference*, pages 3678–3683. ELRA.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yong Guan, Xiaozhi Wang, Lei Hou, Juanzi Li, Jeff Z. Pan, Jiaoyan Chen, and Freddy Lecue. 2024. Tacoere: Cluster-aware compression for event relation extraction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15511–15521, Torino, Italy. ELRA and ICCL.

Rujun Han, I. Hsu, Mu Yang, Aram Galstyan, Ralph Weischedel, and Nanyun Peng. 2019. Deep structured neural network for event temporal relation extraction. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 666–106, Stroudsburg. ACL Press.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Zhilei Hu, Zixuan Li, Daozhu Xu, Long Bai, Cheng Jin, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2023. Protoem: A prototype-enhanced matching framework for event relation extraction. *Preprint*, arXiv:2309.12892.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350, Florence, Italy. Association for Computational Linguistics.

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

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Hieu Man, Nghia Trung Ngo, Linh Ngo Van, and Thien Huu Nguyen. 2022. Selecting optimal context sentences for event-event relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11058–11066.

Puneet Mathur, Rajiv Jain, Franck Dernoncourt, Vlad Morariu, Quan Hung Tran, and Dinesh Manocha. 2021. Timers: Document-level temporal relation 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 2: Short Papers)*, pages 524–533, Online. Association for Computational Linguistics.

Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. An improved neural baseline for temporal relation extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6203–6209, Hong Kong, China. Association for Computational Linguistics.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1318–1328, Melbourne, Australia. Association for Computational Linguistics.

Jong-Hoon Oh, Kentaro Torisawa, Canasai Kruengkrai, Ryu Iida, and Julien Kloetzer. 2017. Multi-column convolutional neural networks with causality-attention for why-question answering. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 415–424, New York, NY, USA. Association for Computing Machinery.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. Yarn: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*.

Minh Tran Phu and Thien Huu Nguyen. 2021. Graph convolutional networks for event causality identification with rich document-level structures. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3480–3490, Online. Association for Computational Linguistics.

M. Recasens and E. Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 17(4):485–510.

Mehwish Riaz and Roxana Girju. 2013. Toward a better understanding of causality between verbal events: Extraction and analysis of the causal power of verb-verb associations. In *Proceedings of the SIGDIAL 2013 Conference*, pages 21–30, Metz, France. Association for Computational Linguistics.

Mehwish Riaz and Roxana Girju. 2014. Recognizing causality in verb-noun pairs via noun and verb semantics. In *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)*, pages 48–57, Gothenburg, Sweden. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, and et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Xiaozhi Wang, Yulin Chen, Ning Ding, Hao Peng, Zimu Wang, Yankai Lin, Xu Han, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022. Maven-ere: A unified large-scale dataset for event coreference, temporal, causal, and subevent relation extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 926–941, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Kangda Wei, Aayush Gautam, and Ruihong Huang. 2024. Are llms good annotators for discourse-level event relation extraction? *Preprint*, arXiv:2407.19568.

Haoyang Wen and Heng Ji. 2021. Utilizing relative event time to enhance event-event temporal relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10431–10437, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2021. Measuring association between labels and free-text rationales. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10266–10284, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wang Xu, Kehai Chen, Lili Mou, and Tiejun Zhao. 2022. Document-level relation extraction with sentences importance estimation and focusing. In *Proceedings*

of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2920–2929, Seattle, United States. Association for Computational Linguistics.

Jifan Yu, Xiaozhi Wang, Shangqing Tu, and et al. 2023. Kola: Carefully benchmarking world knowledge of large language models. In *The Twelfth International Conference on Learning Representations*.

Chenhan Yuan, Qianqian Xie, and Sophia Ananiadou. 2023. Zero-shot temporal relation extraction with chatgpt. *Preprint*, arXiv:2304.05454.

Baiyan Zhang, Qin Chen, Jie Zhou, Jian Jin, and Liang He. 2024. Enhancing event causality identification with rationale and structure-aware causal question answering. *Preprint*, arXiv:2403.11129.

Yaowei Zheng, Richong Zhang, Junhao Zhang, YeYanhan YeYanhan, and Zheyan Luo. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

Jie Zhou, Shenpo Dong, Hongkui Tu, Xiaodong Wang, and Yong Dou. 2022. Rsgt: Relational structure guided temporal relation extraction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2001–2010, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

A Task Descriptions

Detailed task descriptions are presented in Figure 7.

B Transitivity Logical Rules

Chen et al. (2024) summarize the transitivity logical rules for event relations, as shown in Table 7.

C Dataset Partitioning

For **MAVEN-ERE**, its test set is not publicly available. Following Chen et al. (2024), we split its original training set into training/validation sets with a ratio of 8:2, and then utilize its original valid set as the new test set. For **MATRES**, following previous works (Ning et al., 2019; Wang et al., 2022), the training, validation, and test sets contain 182, 73, and 20 documents, respectively. For **HiEve**, following previous works (Wang et al., 2022; Guan et al., 2024), we split the 100 documents in a 60/20/20 ratio for training, validation, and testing, respectively.

If Relation(A, B) $\wedge$ **Relation**(B, C)	**Then Relation** (A, C)
BEFORE $\wedge$ BEFORE	BEFORE
BEFORE $\wedge$ OVERLAP	BEFORE
BEFORE $\wedge$ CONTAINS	BEFORE
BEFORE $\wedge$ SIMULTANEOUS	BEFORE
BEFORE $\wedge$ ENDS-ON	BEFORE
BEFORE $\wedge$ BEGINS-ON	BEFORE
OVERLAP $\wedge$ BEFORE	BEFORE
OVERLAP $\wedge$ SIMULTANEOUS	OVERLAP
CONTAINS $\wedge$ CONTAINS	CONTAINS
CONTAINS $\wedge$ SIMULTANEOUS	CONTAINS
SIMULTANEOUS $\wedge$ BEFORE	BEFORE
SIMULTANEOUS $\wedge$ OVERLAP	OVERLAP
SIMULTANEOUS $\wedge$ CONTAINS	CONTAINS
SIMULTANEOUS $\wedge$ SIMULTANEOUS	SIMULTANEOUS
SIMULTANEOUS $\wedge$ ENDS-ON	ENDS-ON
SIMULTANEOUS $\wedge$ BEGINS-ON	BEGINS-ON
ENDS-ON $\wedge$ CONTAINS	BEFORE
ENDS-ON $\wedge$ BEGINS-ON	ENDS-ON
ENDS-ON $\wedge$ SIMULTANEOUS	ENDS-ON
BEGINS-ON $\wedge$ SIMULTANEOUS	BEGINS-ON
BEGINS-ON $\wedge$ BEGINS-ON	BEGINS-ON
CAUSE $\wedge$ CAUSE	CAUSE
CAUSE $\wedge$ PRECONDITION	PRECONDITION
PRECONDITION $\wedge$ CAUSE	PRECONDITION
PRECONDITION $\wedge$ PRECONDITION	PRECONDITION
SUBEVENT $\wedge$ SUBEVENT	SUBEVENT

Table 7: Logical Constraints for the transitivity rules among three events, where $\wedge$ denotes "AND".

D Detailed Baselines

For **MATRES**, CSE+ILP (Ning et al., 2019) conducts global inference via Integer Linear Programming (ILP); Deep (Han et al., 2019) is a deep structured support vector machine model; Stack-P (Wen and Ji, 2021) is a Stack-Propagation framework; TIMERS (Mathur et al., 2021) utilizes temporal, rhetorical, and syntactic information; SCS-EERE (Man et al., 2022) selects the optimized sentences for event relation inference; RSGT (Zhou et al., 2022) models the syntactic and semantic graphs.

For **HiEve**, Hierarchical (Adhikari et al., 2019) encodes different chunks of the document and aggregates their representations; SIEF (Xu et al., 2022) randomly removes the useless sentences for prediction; TacoERE (Guan et al., 2024) utilizes document clustering and cluster summarization to spotlight important text.

Event temporal relation extraction

The current task is **an event temporal relation extraction task**, which aims to identify temporal relations among events in texts. The temporal relation between events refers to the chronological order in which they occur, involving six subtypes, namely, **SIMULTANEOUS, ENDS-ON, BEGINS-ON, OVERLAP, CONTAINS, and BEFORE.** In the provided document, event trigger words are annotated within angle brackets (<>). The desired outcome is a list of events in the document that have temporal relations with the given event. The prescribed output format should follow this structure: 'relation1: event1, event2; relation2: event3, event4'. The output 'relation: none' indicates that the given event lacks this particular type of relation with other events.

Event causal relation extraction

The current task is **an event causal relation extraction task,** which aims to identify causal relations among events in texts. The causal relation between events denotes that the occurrence of the first event precipitates the happening of the second event, delineated into two subtypes: **CAUSE and PRECONDITION**. In the provided document, event trigger words are annotated within angle brackets (<>). The desired outcome is a list of events in the document that have causal relations with the given event. The prescribed output format should follow this structure: 'relation1: event1, event2; relation2: event3, event4'. The output 'relation: none' indicates that the given event lacks this particular type of relation with other events.

Subevent relation extraction

The current task is **a subevent relation extraction task**, which aims to identify subevent relations among events in texts. The subevent relation, labeled as **SUBEVENT**, denotes a hierarchical relation where the first event is contained by the second. In the provided document, event trigger words are annotated within angle brackets (<>). The desired outcome is a list of events in the document that have a subevent relation with the given event. The prescribed output format should follow this structure: 'relation1: event1, event2; relation2: event3, event4'. The output 'relation: none' indicates that the given event lacks this particular type of relation with other events.

Event coreference relation extraction

The current task is **an event coreference relation extraction task**, which aims to identify coreference relations among events in texts. The coreference relation, labeled as **COREFERENCE**, denotes that two events are the same one. In the provided document, event trigger words are annotated within angle brackets (<>). The desired outcome is a list of events in the document that have a coreference relation with the given event. The prescribed output format should follow this structure: 'relation1: event1, event2; relation2: event3, event4'. The output 'relation: none' indicates that the given event lacks this particular type of relation with other events.

Figure 7: Detailed task descriptions on the MAVEN-ERE dataset.