Improving LLM-based Unified Event Relation Extraction via Multiple Answer Questions

Anonymous ACL submission

Abstract

Extracting event relations that deviate from known schemas has proven challenging for previous methods based on multi-class classification, MASK prediction, or prototype matching. While the LLM-based method can devise diverse instructions to alleviate these issues, it is also accompanied by certain limitations: the need to create a large number of training and inference samples, heightened sensitivity to the sequence of event relation generation, and difficulties in extracting scattered event relations. To tackle these challenges, we present an improved unified event relation extraction framework based on LLM named MAQERE. Firstly, we transform the pair-based extraction issue in LLM-based methods into a multiple answer question problem, which reduces the number of samples required for training and inference. Additionally, by incorporating a bipartite matching loss, we have reduced the dependency of the LLM-based method on the generation sequence. Then, we employ Parse-CoT to extract structured information for enhancing the connections between event mentions. Our experimental results demonstrate that MAQERE can significantly improve the performance of the LLM-based method in the task of event relation extraction.

1 Introduction

Event Relation Extraction (ERE) is the task of predicting relations between event mentions in unstructured text. Take the text "Last year, more than 3,000 civilians were killed and another 4,500 were injured in Afghanistan, with roughly a 5% increase from 2010" as an example. The goal of ERE is to identify all relevant event mention pairs (<killed, sub-event, increase>) from the given event mentions ("killed", "injured", and "increase"). ERE tasks are highly diversified due to their varying sub-tasks (coreference, temporal, causal, sub-event, etc.) and complex relations (symmetrical, asymmetrical, cross, etc.) (Han et al., 2019, 2020; Min et al., 2020; Wen and Ji, 2021; Tang et al., 2021; Hu et al., 2023b).

Most previous studies (Nguyen et al., 2022a; Wang et al., 2023a; Yuan et al., 2023; Caselli and Vossen, 2017; Xu et al., 2022; Nguyen et al., 2022b) have primarily focused on optimizing a specific sub-task, making it difficult to migrate model structures, optimization strategies, specialized knowledge sources, and domain data between different sub-tasks. While some studies (Wang et al., 2022; Hu et al., 2023b) employ multi-head classification or prototype matching to tackle multiple subtasks simultaneously, these methods rely on pre-defined relation schemas and are unable to effectively handle newly introduced, modified, or upgraded relation schemas. While large language models such as ChatGPT and LLAMA demonstrate exceptional semantic understanding and zero-shot learning capabilities, the LLM-based method, which can devise diverse instructions to address these issues, also faces certain limitations such as the need for a large number of training samples, high sensitivity to the generated sequence, and difficulty in extracting scattered event relations.

Classification Based

[CLS] ***battle*** [SEP] ***attacking*** [SEP]The Battle of Sultanabad occurred …[SEP]
[CLS] ***Battle of Sultanabad*** [SEP] ***attacking*** [SEP]The Battle of Sultan…[SEP]

LLM Based

instruction: What kind of event relation is ***battle*** and ***attacking***? The candidate event relations are: *effect*, *cause*, *coreference*, *parent*, *child*, *contains*, …
input: The *Battle of Sultanabad* occurred on *Feb. 13, 1812*. … The Persians won the *battle* by moving faster than the Russians and *attacking* …
output: *contains*, *child*

Multiple Answers Question Based

instruction: List the *child* event of ***attacking***?
input: The <0x64>*Battle of Sultanabad* occurred on <0x65>*Feb. 13, 1812*. … The Persians won the <0x66>*battle* by moving faster than the Russians and <0x67>*attacking* …
output: *<0x64>Battle of Sultanabad, <0x66>battle*

Figure 1: Different ERE methods. The special, individual, unused character <0x64>-<0xFF> in LLAMA is used to indicate candidate event mentions.

For a more intuitive comparison, we present the different methods in Figure 1. The classification-based method utilizes one-hot embedding to represent the event relation labels, which overlooks the semantic information of the labels. The LLM-based method employs candidate event mention pairs and all event relations as the instruction, utilizing the large language model to generate all event relations. Obviously, the LLM-based method has some significant drawbacks. Firstly, it involves a substantial amount of training and inference samples, reaching $n \times n$, where n represents the number of event mentions. Secondly, the model is heavily influenced by the sequence of generation when multiple relations are produced. Using the LLM-based method shown in Figure 1 as an example, the model generates $p(contains|child)$ and $p(child|contains)$ with varying probabilities. However, in the event relation extraction task, the sequence of generation should not affect the event relation between event mentions.

To reduce the training and inference samples of the LLM-based model, we draw inspiration from multi-span extraction and multi-choice reading comprehension (Hu et al., 2019; Yang et al., 2021; Segal et al., 2020).

 # Multi-Choice Reading Comprehension
Context: *I wanted to plant a tree. I went to the home and garden store and picked a nice oak. Afterwards, I planted it in my garden.*

Question: *When did he plant the tree?*
A. *after watering it* B. *after taking it home*

Answers: B

 # Multi-Span Extraction Reading Comprehension
Context: *Salary. The average salary range for a zoologist in the initial stages of his or her career is \$30,000 to \$45,000 per year. After five years of work experience, the range is \$40,000 to \$55,000 per year.*

Question: *zoology salary*

Answers: *\$30,000 to \$45,000, \$40,000 to \$55,000*

By integrating multi-span extraction and multi-choice techniques, we incorporate special characters into the text to indicate candidate event mentions. This approach enables the large language model to select from them during generation. For specific examples, please refer to the multiple answer question based method in Figure 1. In the event relation extraction task, the number of event relation types $k \ll n$. Therefore, for the multiple answer question based model, the training and inference samples are reduced from $n \times n$ to $k \times n$.

To reduce the effect of generated sequences on LLM-based methods, we introduce a bipartite matching loss. As shown in Figure 2, the LLM-based method employs cross-entropy loss to guarantee an accurate sequence of generation. Nonetheless, for the task of event relation extraction, the sequence of generation does not affect the final result. This makes the bipartite matching loss a better fit for such tasks. The example in Figure 2 demonstrates that using the cross-entropy loss results in 2 mistakes, while the bipartite matching loss yields 1 correct answer and 1 mistake.

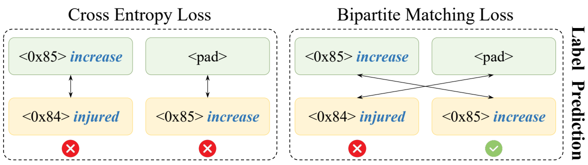

Figure 2: Comparison of cross-entropy loss and bipartite matching loss.

Additionally, event mentions are short phrases or single words, providing limited details. Furthermore, the relations between event mentions are extremely scattered, with pairs that have relations making up less than 5%. Despite this, the LLM-based method typically utilizes uni-directional transformers, which are especially prone to the issue of long-distance forgetting. To address this challenge, we have implemented Parse-CoT as a strategy to decelerate this problem, which is depicted in Figure 3. For example, in the text "Last year, more than 3,000 civilians were <0x83> killed and another 4,500 <0x84> injured in Afghanistan, with a roughly 5% <0x85> increase compared to 2010", where "increase" is the direct object related to "killed", and "injured" is linked as a conjunction with "killed"[1]. By integrating information from Parse-CoT, the model is able to improve its ability to extract scattered event relations.

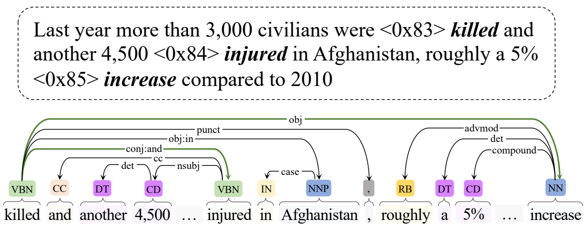

Figure 3: Dependency parsing tree of the input context.

In summary, the main contributions of this paper are:

1) We propose a unified event relation extraction framework (MAQERE) based on multiple answer

[1] The composition and meaning of dependent edges refer to https://stanfordnlp.github.io/CoreNLP/

questions. Compared with the LLM-based method, our method reduces the training and inference samples from $n \times n$ to $k \times n$.

2) In the MAQERE framework, we incorporate a bipartite matching loss to reduce the dependency of the LLM-based method on the generation sequence, making it more suitable for event relation extraction tasks.

3) We propose a Parse-CoT that enhances the capability of LLM-based methods in extracting scattered event relations.

2 Related Work

Previous existing methods (Man et al., 2022; Hwang et al., 2022; Huang et al., 2023; Barhom et al., 2019; Hu et al., 2023a; Wang et al., 2022; Tan et al., 2023) for event relation extraction primarily utilize multi-class classification, MASK prediction, or prototype matching, which focus on addressing specific sub-tasks such as coreference, temporal, causal, or sub-event relations. In the classification-based approach (Huang et al., 2023; Lu and Ng, 2021; Tran et al., 2021; Zeng et al., 2020; Wang et al., 2020; Barhom et al., 2019), event mentions are paired together, and then additional features are incorporated, such as prototypes, logical rules, graph convolutional networks, or prompts. MASK prediction based methods (Xiang et al., 2023; Shen et al., 2022; Cui et al., 2022) train a masked language model to predict the relation. The prototype matching based method (Hu et al., 2023b) manually selects instances to serve as prototypes for each relation. Then, new instances are matched against these prototypes. Segal et al. (2020) and Hu et al. (2019) each proposed a reading comprehension model based on multi-choice and multi-span, respectively, which allows the model to select the correct answer from the candidate options or to generate multiple answers simultaneously. Simultaneously, there are many entity relation extraction methods based on LLMs (Wang et al., 2023b; Xu et al., 2024; Xiao et al., 2024), which directly prompt large language models to generate relations between pairs of entities. In this task, these methods have many drawbacks. Therefore, we have designed a series of improvement measures to address these identified deficiencies.

3 Methodology

The architecture of our framework is illustrated in Figure 4. Our model mainly consists of three parts. Firstly, the event relation extraction samples are constructed based on multiple answer questions. Secondly, we constructed Parse-CoT using the Core NLP Dependency Parser in the Stanford NLP toolkit. Finally, we introduce a loss function for multiple answer questions to reduce reliance on the generated sequences.

3.1 Sample Construction

The training and inference samples of our framework are constructed as follows:

Instruction: To unify the various inputs for different event relation extraction sub-tasks, we have developed various instructions, as demonstrated in Table 1. Each instruction contains an event relation and a candidate event mention, where <0x64>-<0xFF> is a special, individual, unused character in LLAMA, which we use to indicate the candidate event mention.

	Instruction
Coref.	List the ***coreference*** event of <0x85> ***ruled*** ?
Temp.	List the... ***earlier than*** <0x72> ***said*** ?
	List the... ***later than*** <0x72> ***said*** ?
	List the... ***the same time as*** <0x72> ***said*** ?
	List the... ***inconsistent with***...<0x72> ***said*** ?
Causal	List the ***cause*** event of <0x64> ***keep*** ?
	List the ***effect*** event of <0x64> ***keep*** ?
Sub.	List the ***parent*** event of <0x83> ***killed*** ?
	List the ***child*** event of <0x83> ***killed*** ?

Table 1: Various instructions for different event relation extraction sub-tasks.

Context: In the event relation extraction task, all candidate event mentions are provided. We insert a marker (<0x64>-<0xFF>) sequentially in the text where the candidate events appear, with the first candidate event mention receiving <0x64>, the second <0x65>, and so on. These markers signal the large language model to confine its generation results to only the specified contents.

Label: The output is divided into two parts: Parse-CoT and Multiple Answers, separated by a colon. The construction of Parse-CoT is according to section 3.2. Similar to before, markers will also be inserted in the Parse-CoT and Multiple Answers part to uniquely identify the event mentions. If there are multiple answers, they are listed in the order they appear in the text, separated by commas. For those without associated event mentions, the Multiple Answers part is set to none.

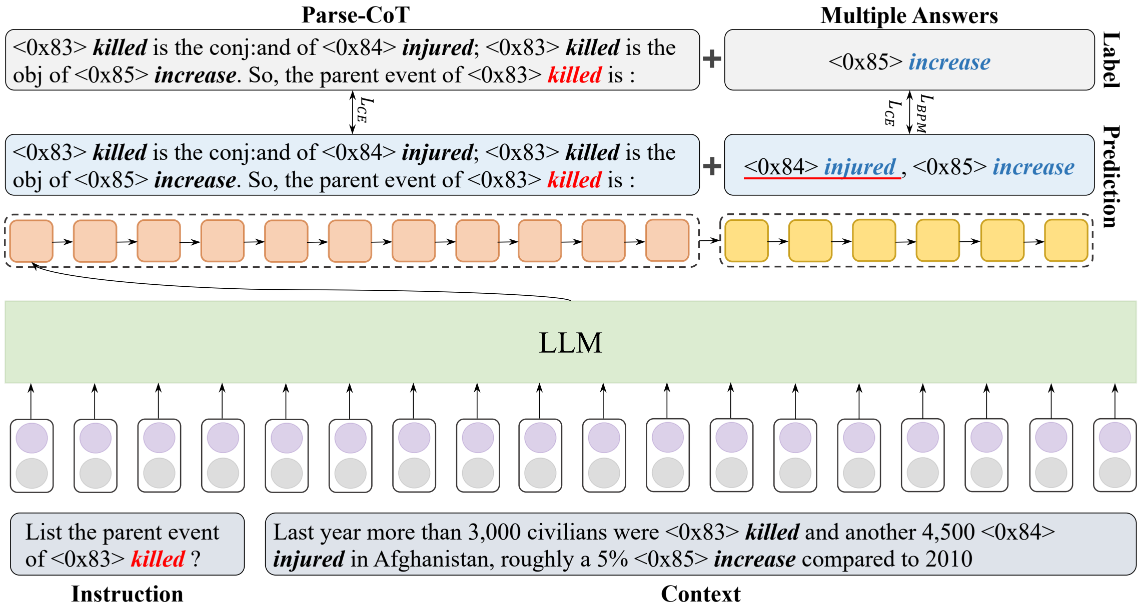

Figure 4: The overview of the MAQERE framework. The input includes instructions and context, and the special characters <0x64>-<0xFF> in LLAMA are used to indicate candidate event mentions. The output includes Parse-CoT and Multiple Answers.

However, in event relation extraction tasks, there are a large number of event mentions, but the relations between event mentions are extremely scattered, with pairs that have relations making up less than 5%. As a result, whether using the LLM-based or MAQ-based approach, a large number of negative samples are created (the Multiple Answers part is none), making training the model challenging. To tackle this challenge, we utilized positive sample expansion and negative sample downsampling techniques. For specific implementation details, refer to Appendix A.

3.2 Parse-CoT Construction

We employ the Core NLP Dependency Parser from the Stanford NLP toolkit to derive the dependency parse tree of the context. As shown in Figure 3, after parsing the context for dependencies, numerous dependency edges are generated. The meaning of each type of edge can be found in the official documentation of the Stanford NLP toolkit. In event relation extraction tasks, we only focus on the edges between event mentions. Therefore, we retain only the minimum number of nodes and edges necessary to connect all the event mentions. In cases where the number of nodes and edges is the same, we retain them based on the order in which the nodes appear. As shown in Figure 5, both <r_1, r_2, r_4> and <r_3, r_2, r_4> are valid paths, but we only retain the first one that appears, <r_1, r_2, r_4>. It is crucial to mention that since the dependency parser functions at the sentence level, we substitute "." with ";" to ensure the generation of the required Parse-CoT.

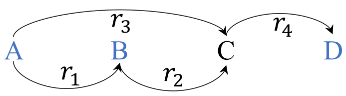

Figure 5: A, B, D represent event mentions, while C denotes other words. r_1, r_2, r_3, r_4 represent different dependency relations.

3.3 Multiple Answer Questions Loss

The generated sequence significantly affects the effectiveness of text generation, as supported by relevant research (Ye et al., 2021; Cao and Zhang, 2022). However, in the task of event relation extraction, the sequence of generating the answer does not affect the final result. To mitigate the impact of generation sequence, we calculate distinct losses for Parse-CoT and Multiple Answers. The loss of Parse-CoT and Multiple Answers is defined as follows:

$$\mathcal{L}_{CE} = \frac{1}{N} \sum_{i=0}^{N} CE(y_i, p(y_i|x)) \tag{1}$$

where $N = N_1 + N_2$, N_1 represents the length of Parse-CoT and N_2 represents the length of Multiple Answers. CE is the cross-entropy loss. As illustrated in Figure 2, the sequence of generation does not impact the multiple answers. The loss of Multiple Answers is calculated as follows:

(a) First, use the Hungarian Algorithm to find the optimal match.

$$\hat{\theta} = \underset{\theta \in \Psi_{N_2}}{\arg\min} \sum_{i=0}^{N_2} 1 - \log \hat{p}_{\theta(i)}(c_i) \tag{2}$$

(b) After optimal allocation, the loss function for Multiple Answers is:

$$\mathcal{L}_{BPM} = \sum_{i=0}^{N_2} 1 - \log \hat{p}_{\hat{\theta}(i)}(c_i) \tag{3}$$

(c) Finally, the total loss is as follows:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{BPM} \tag{4}$$

where Ψ_{N_2} denotes a permutation of N_2. θ is one of the permutations. $\theta(i)$ is the i-th element in permutation θ. c_i represents the target vocabulary id of the i-th element. The probability of the i-th element in the permutation θ belonging to the target vocabulary id is denoted by $\hat{p}_{\theta(i)}(c_i)$. $\hat{\theta}$ stands for the optimal permutation. The weight parameter is represented by λ.

4 Experimental Settings

Dataset. Our experiments are conducted on four widely-used datasets (cf. Table 2), including MAVEN-ERE (Wang et al., 2022) for coreference relation extraction and unified event relation extraction, HiEve (Glavas et al., 2014) for sub-event relation extraction, MATRES (Ning et al., 2018) for temporal relation extraction, and MECI (Lai et al., 2022) for causal relation extraction. For a fair comparison, we divided the data into the same training, validation, and test sets as in previous studies (Wang et al., 2022; Man et al., 2022; Zhou et al., 2022; Lai et al., 2022). In particular, since the training and test sets are not divided, consistent with previous works, HiEve selects 80 documents for training (0.4 probability for down-sampling of negative examples) and 20 documents for testing. Since MAVEN-ERE does not have an open test set, we have chosen to use the validation set for testing.

Datasets	#Docs	#Mentions	#Links
MAVEN-ERE	4,480	112,276	103,193
HiEve	100	3, 185	3,648
MATRES	275	11,861	13,573
MECI	438	8,732	2,050

Table 2: Dataset Statistics. "#" denotes the amount. "Mentions" represents the potential events. "Links" means the event relations.

Evaluation Metric. Based on previous research on event relation extraction (Choubey and Huang, 2017; Nguyen et al., 2022a; Wang et al., 2023a; Yuan et al., 2023; Caselli and Vossen, 2017; Xu et al., 2022; Nguyen et al., 2022b), we adopt MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), $CEAF_e$ (Luo, 2005) and BLANC (RECASENS and HOVY, 2011) metrics for event coreference relation. For the other three subtasks, we adopt the standard micro-averaged precision, recall, and F-1 metrics. In particular, in the subevent relation extraction task, PC and CP represent the F1 scores for parent-child and child-parent relations, respectively. For more details, please refer to Appendix B.

Implementation Details. For MAQERE, we have chosen the llama-2-chat[2] as the backbone network. Our training is conducted on 4×A100-80G. The input sequence length is 1536, and the output sequence length is 512. The weight for the bipartite matching loss, denoted as λ, is set to 0.2. We use a learning rate of 5e-4, a batch size of 16, and a gradient accumulation of 2. The learning rate scheduler follows a cosine function, and the model is trained for 20 epochs. The results reported in the experiment are the averages of 5 different random seeds (0,1,2,3,4). For other hyper-parameters and details, please refer to Appendix C.

5 Experimental Results

5.1 Comparison Methods

The baseline model of MAVEN-ERE (Wang et al., 2022) utilizes joint learning to incorporate relation interactions. In the case of HiEve, the baseline model (Man et al., 2022) involves selecting the optimal context sentence for event-event relation extraction. Meanwhile, the baseline model (Zhou et al., 2022) in MATRES involves constructing a graph based on syntax and semantics to extract relational structures. Lastly, the baseline approach (Lai et al., 2022) in MECI uses a graph-based model to construct interaction graphs that depict crucial connections among important entities. This enables the identification of event causality at the document level. **BertERE** employs a RoBERTa-based multi-class classification method to extract event

[2]https://huggingface.co/hfl/chinese-alpaca-2-7b

Method	MAVEN-ERE				HiEve			MATRES			MECI		
	B^3	$CEAF_e$	MUC	BLANC	PC	CP	Avg	P	R	F1	P	R	F1
Baselines	97.9	97.6	79.7	88.4	68.7	63.2	65.9	82.2	85.8	84.0	48.1	69.5	56.8
BertERE	94.5	95.1	77.4	87.2	65.7	61.5	63.4	80.2	82.4	81.3	50.7	54.2	52.4
BertERE$_{joint}$	95.5	94.8	77.1	85.3	64.9	60.8	62.8	79.4	79.6	79.5	48.1	51.4	49.7
LLM-based	93.5	93.4	74.1	85.4	65.5	63.5	64.5	80.3	79.5	79.9	57.8	54.7	56.2
LLM-based$_{joint}$	91.2	91.5	72.6	83.2	64.2	60.8	62.5	79.9	78.5	79.2	56.3	55.5	55.8
MAQERE	**98.1**	**97.8**	**79.9**	**88.7**	67.8	68.5	**68.1**	85.5	83.9	**84.7**	62.9	61.6	**62.3**
MAQERE$_{joint}$	97.4	96.5	78.8	87.2	67.2	67.0	67.1	82.3	83.5	82.9	59.7	60.5	60.1

Table 3: The comprehensive performance of MAQERE across various datasets.

Models	COREFERENCE				TEMPORAL			CAUSAL			SUBEVENT		
	B^3	$CEAF_e$	MUC	BLANC	P	R	F1	P	R	F1	P	R	F1
BertERE$_{joint}$	97.8	97.6	79.8	88.3	50.9	53.4	52.1	31.3	30.5	30.9	24.6	22.9	23.7
LLM-based$_{joint}$	94.2	93.5	73.3	84.7	48.5	51.0	49.7	28.6	28.0	28.3	20.9	21.7	21.3
MAQERE$_{joint}$	**98.1**	**97.9**	**80.2**	**88.9**	53.3	54.3	**53.8**	33.4	31.6	**32.5**	25.8	24.6	**25.2**

Table 4: The performance of various unified event relation extraction models on the unified dataset MAVEN-ERE.

relations for event pairs consisting of all event mentions. **BertERE$_{joint}$** encodes the whole document using RoBERTa, then sets an additional classification head that takes the contextualized representations at the positions of different event pairs. Afterward, it fine-tunes the model to classify relation labels. **LLM-based** method employs candidate event mention pairs and event relations as the instruction, leveraging the large language model's capability to generate comprehensive event relations. **MAQERE** stands for event relation extraction based on multiple answer questions, which enhances the effectiveness of LLM-based methods through the integration of bipartite matching loss and Parse-CoT. **MAQERE$_{joint}$** and **LLM-based$_{joint}$** represent the joint training of various diverse subtask datasets. For more implementation details and hyper-parameters of the compared methods, please refer to Appendix D.

5.2 Overall Results

Separate Training. The model is trained on a subtask dataset. As shown in Table 3, we evaluate our framework on four widely-used event relation extraction datasets independently. As observed, MAQERE outperforms the previous advanced baseline model by 3.34%, 0.83%, and 9.68% in F1 score in the HiEve, MATRES, and MECI datasets, respectively. Simultaneously, our method shows a slight improvement over the baseline method in coreference relation extraction. There are two main reasons: (1) MAQERE reduces the number of training and inference samples from $n \times n$ to $k \times n$, resulting in denser relations between event mentions that are easier to train; (2) MAQERE overcomes the length limitations present in baseline models, making it easier to extract long-distance event relations. Furthermore, within the realm of generative models, our approach outperforms the LLM-based method, and our method achieves an average improvement of 5.22% on the MAVEN-ERE dataset. In terms of F1 score, MAQERE shows improvements of 5.58%, 6.01%, and 10.85% on the HiEve, MATRES, and MECI datasets, respectively. The primary reason is that MAQERE leverages the superior semantic understanding capability of large language models to integrate structured information of event mentions, and uses bipartite matching loss to mitigate the impact of sequence generation on generative models.

Joint Training. The model is simultaneously trained on multiple subtasks datasets. To construct a unified event relation extraction model, joint training is primarily conducted with two sets of data. For the first group, the coreference dataset from MAVEN-ERE is jointly trained with HiEve, MATRES, and MECI. The second group involved joint training of the coreference, temporal, causal, and sub-event datasets within MAVEN-ERE. As shown in Table 3, joint training with data from different sources resulted in performance that is lower than that of separate training. The primary reason for this is that datasets from different sources have conflicting definitions of relations, resulting in the introduction of noise during joint extraction. As indicated in Table 4, when data from the same source is used for joint training, the performance of the joint training model is better than that of separate training. Analysis has found that relations defined consistently from the same source can be effectively enhanced across multiple joint extraction models. Overall, compared to BertERE$_{joint}$ and LLM-based$_{joint}$, MAQERE$_{joint}$ also demonstrated excellent performance in joint training.

5.3 Model Ablation Studies

We ablate each component of our model on MATRES and MECI, as shown in Table 5. First, without the marker (<0x64>-<0xFF>), we observe performance drops of 2.48% on MATRES and 5.14% on MECI, which verifies the usefulness of the prefix marker. In cases where multiple answers consist only of markers, such as "<0x84>, <0x85>" instead of "<0x84> injured, <0x85> increase", that will lead to a slight decrease in effectiveness. There is a possibility that these markers may not contain complete semantic information. By removing positive sample expansion and negative sample downsampling, the performance drop is equally significant. Furthermore, after removing Parse-CoT, the performance decrease is most significant. The main reason is that Parse-CoT improves its ability to extract scattered event relations by leveraging structured information. When the bipartite matching loss function is removed, the model effect drops seriously, which indicates that the bipartite matching loss is more appropriate for scenarios where the sequence of generated results is not predetermined.

Method	MATRES			MECI		
	P	R	F1	P	R	F1
MAQERE	85.5	83.9	84.7	62.9	61.6	62.3
w/o Marker	81.2	84.1	82.6	58.9	59.3	59.1
only Marker	84.6	83.8	84.2	62.2	61.8	62.0
w/o Expansion	82.5	83.1	82.8	61.7	58.8	60.2
w/o Sampling	83.5	83.3	83.4	60.2	62.6	61.4
w/o Parse-CoT	82.3	80.5	81.4	57.5	59.3	58.4
w/o $\mathcal{L}_{BPM}$	81.4	83.6	82.5	61.1	61.5	61.3

Table 5: Model ablation studies. Marker refers to the identifier that precedes a event mention, e.g., "<0x8F>".

5.4 Bipartite Matching Loss Analysis

The performance of a generative model is greatly affected by the generation sequence. According to Table 6, when the bipartite matching loss is not considered, random answer sequences perform the worst, with a reduction of 4.00% and 3.92% compared to ordered sequences in MATRES and MECI, respectively. However, after incorporating the bipartite matching loss, MAQERE is capable of effectively generating the correct results with any answer sequence used. Therefore, this evidence indicates that the bipartite matching loss is especially suitable for tasks where the generated sequence is not crucial. For sensitivity analysis of bipartite matching loss, as shown in Figure 6, the results indicate that the model achieves optimal performance when the weight λ assigned to the bipartite matching loss is 0.2. As λ increases, the model's performance will decrease, and it may even perform worse than when bipartite matching loss is not utilized. The main reason is that an increase in

Method		MATRES			MECI		
		P	R	F1	P	R	F1
w/o $\mathcal{L}_{BPM}$	Random	80.8	77.7	79.2	59.4	58.4	58.9
	Sequence	81.4	83.6	**82.5**	61.1	61.5	**61.3**
	Reverse	80.1	80.7	80.4	61.3	59.9	60.6
	Distance	81.5	82.7	82.1	60.7	61.1	60.9
	Dict	78.9	81.8	80.3	60.1	58.5	59.3
w/ $\mathcal{L}_{BPM}$	Random	82.2	84.6	83.4	60.8	61.4	61.1
	Sequence	85.5	83.9	**84.7**	62.9	61.6	**62.3**
	Reverse	83.7	84.5	84.1	61.2	62.4	61.8
	Distance	83.5	85.1	84.3	61.7	62.7	62.2
	Dict	83.2	83.8	83.5	62.5	60.3	61.4

Table 6: The performance of different answer sequences. "Random" indicates that the answers are in a random sequence, "Sequence" represents the sequence in which they appear in the text, "Reverse" indicates the reverse sequence of their appearance, "Distance" means the answers are sorted by distance from the query mention, and "Dict" sorts them from A to Z.

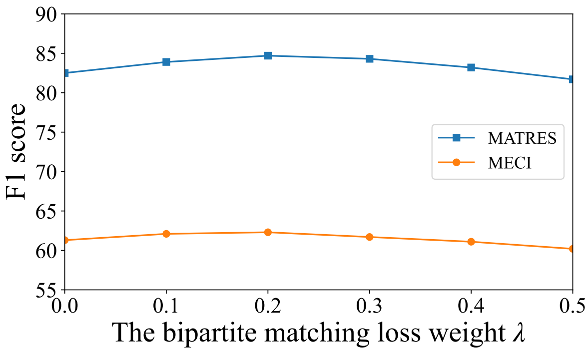

Figure 6: The impact of the bipartite matching loss weight λ on MAQERE.

bipartite matching loss leads to a reduction in CE loss, causing the model to neglect the optimization of Parse-CoT, resulting in inaccuracies in structured information, thereby affecting the generation of the final results.

5.5 Parse-CoT Analysis

Document-level event relation extraction usually involves extracting relations among event mentions that are scattered throughout the text. The utilization of structured information, such as dependency parse trees, can enhance the associations between event mentions. For example, Figure 3 shows how a dependency parse tree connects the event mentions "kill," "injured," and "increase" more closely. However, integrating this structured information effectively into MAQERE is not straightforward. Previously, the primary approach involved directly integrating dependency parse data into the input. As shown in Table 7, incorporating structured information at the input can indeed lead to performance enhancements compared to not providing dependency parse. However, since parser information can be overly complex and not always relevant, selectively utilizing only those segments of the structure that relate to the specific event mentions can reduce unnecessary noise, thus improving the performance of MAQERE. Incorporating structured information into the input will weaken the generation results as the length of the text increases. To address this issue, we integrate parsing information into the output of the model. Table 7 demonstrates that integrating structured information into the output can significantly enhance the performance of MAQERE.

Method	MATRES			MECI		
	P	R	F1	P	R	F1
w/o parser	82.3	80.5	81.4	57.5	59.3	58.4
input-all	81.6	83.2	82.4	60.9	60.1	60.5
input-shortest	82.9	83.7	83.3	61.4	60.8	61.1
output-all	83.7	82.5	83.1	62.8	60.3	61.5
output-shortest	85.5	83.9	84.7	62.9	61.6	62.3

Table 7: The impact of dependency parsing on MAQERE. "all" indicates that the path includes all edges, whether they are event mentions or non-event mentions. "shortest" refers to incorporating only the shortest path that includes edges associated with all event mentions.

5.6 Case Study

To conduct a qualitative analysis of extracting multiple answers, we provide two examples of event temporal relation extraction, as depicted in Figure 7. The first example demonstrates the correct extraction during the inference process. Generating Parse-CoT provides helpful prompts for producing the final result. In Figure 7, we also present an incorrect example that illustrates two issues: missing the recall of event <0x6A> and incorrectly recalling event <0x71>. This can be attributed to the complexity of the generated Parse-CoT, which diminishes the relevant structural information and results in missed recall. Additionally, irrelevant structural information is introduced, leading to inaccurate recall.

A prediction with two correct event relations

Instruction: List events that may have occurred later than <0x65> ***prevented***?
Context: Embassy in Tanzania was so powerful that it <0x64> ***shook*** furniture in homes a mile away But the solid construction of the Israeli-built embassy may have <0x65> ***prevented*** the carnage of a similar bomb in Kenya, an official <0x66> ***said*** Saturday.
Predict: <0x64> ***shook*** is the ccomp of ***powerful***; ***powerful*** is the parataxis of <0x66> ***said***; <0x66> ***said*** is the ccomp of <0x65> ***prevented***. So, the event may have occurred later than <0x65> ***prevented*** is : <0x64> ***shook***, <0x66> ***said***
Label: <0x64> ***shook***, <0x66> ***said***

A prediction with some incorrect event relations

Instruction: List events that may have occurred later than <0x6F> ***pressing***?
Context: Cuban exiles in Miami will now <0x6A> ***proceed*** with all their resources to <0x6B> ***impede*** or <0x6C> ***delay*** the decision by the Immigration and Naturalization Service. The government <0x6D> ***exhorted*** citizens to <0x6E> ***keep*** up mass protests <0x6F> ***pressing*** for Elian's return until the boy <0x70> ***comes*** home. We cannot <0x71> ***stop*** <0x72> ***mobilizing***!
Predict: <0x6A> ***proceed*** is the parataxis of <0x6D> ***exhorted***; <0x6D> ***exhorted*** is the xcomp of <0x6E> **keep**; <0x6E> keep is the obj of ***protests***; ***protests*** is the acl of <0x6F> ***pressing***; <0x6F> ***pressing*** is the obl:for of ***return***; ***return*** is the dep of <0x70> ***comes***; <0x70> ***comes*** is the parataxis of <0x71> ***stop***; <0x71> ***stop*** is the obj of <0x72> ***mobilizing***; So, the event may have occurred later than <0x6F> ***pressing*** is : <0x6D> ***exhorted***, <0x70> ***comes***, <0x71> ***stop***, <0x72> ***mobilizing***
Label: <0x6A> ***proceed***, <0x6D> ***exhorted***, <0x70> ***comes***, <0x72> ***mobilizing***

Figure 7: Two examples demonstrating the use of MAQERE in extracting temporal relations.

6 Conclusion

In this study, we present a unified framework called MAQERE, aiming to improve LLM-based methods via multiple answer questions, effectively extracting various event relations through different types of instructions. Upon the LLM-based method, MAQERE significantly improves the performance of this model by introducing strategies such as multiple answer questions, parser-cot, and bipartite matching loss. Our extensive ablation studies demonstrate that our strategies effectively address the issues present in the LLM-based method. Beyond event relation extraction, our work may provide insights into other relation prediction tasks.

Limitations

Nonetheless, these results must be interpreted with caution, and several limitations should be kept in mind. Firstly, even though the number of inference samples has been reduced from $n \times n$ to $k \times n$ ($k \ll n$) by using a MAQ-based event relation extraction method, the inference speed of MAQERE is still slower than that of the BERT-based classification model. But the benefits of MAQERE will become more pronounced as the quantity of event mentions increases. Secondly, MAQERE is sensitive to instructions and markers. For more details, please refer to Appendix F and G. Achieving optimal results requires empirical adjustments through multiple experiments, as it cannot be determined solely by theoretical analysis. Finally, although MAQERE has the ability to train a larger unified event relation extraction model, the development of a larger unified MAQ-based event relation extraction model has been hindered by constraints such as the availability of training data and GPU resources.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566.

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4179–4189. Association for Computational Linguistics.

Jie Cao and Yin Zhang. 2022. Otseq2set: An optimal transport enhanced sequence-to-set model for extreme multi-label text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5588–5597. Association for Computational Linguistics.

Tommaso Caselli and Piek Vossen. 2017. The event storyline corpus: A new benchmark for causal and temporal relation extraction. In *Proceedings of the Events and Stories in the News Workshop@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 77–86. Association for Computational Linguistics.

Prafulla Kumar Choubey and Ruihong Huang. 2017. Event coreference resolution by iteratively unfolding inter-dependencies among events. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2124–2133. Association for Computational Linguistics.

Shiyao Cui, Jiawei Sheng, Xin Cong, Quangang Li, Tingwen Liu, and Jinqiao Shi. 2022. Event causality extraction with event argument correlations. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 2300–2312. International Committee on Computational Linguistics.

Goran Glavas, Jan Snajder, Marie-Francine Moens, and Parisa Kordjamshidi. 2014. Hieve: A corpus for extracting event hierarchies from news stories. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 3678–3683. European Language Resources Association (ELRA).

Rujun Han, Qiang Ning, and Nanyun Peng. 2019. Joint event and temporal relation extraction with shared representations and structured prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 434–444. Association for Computational Linguistics.

Rujun Han, Yichao Zhou, and Nanyun Peng. 2020. Domain knowledge empowered structured neural net for end-to-end event temporal relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5717–5729. Association for Computational Linguistics.

Minghao Hu, Yuxing Peng, Zhen Huang, and Dongsheng Li. 2019. A multi-type multi-span network for reading comprehension that requires discrete reasoning. 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 1596–1606, Hong Kong, China. Association for Computational Linguistics.

Zhilei Hu, Zixuan Li, Xiaolong Jin, Long Bai, Saiping Guan, Jiafeng Guo, and Xueqi Cheng. 2023a. Semantic structure enhanced event causality identification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10901–10913. Association for Computational Linguistics.

Zhilei Hu, Zixuan Li, Daozhu Xu, Long Bai, Cheng Jin, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng.

2023b. Protoem: A prototype-enhanced matching framework for event relation extraction.

Quzhe Huang, Yutong Hu, Shengqi Zhu, Yansong Feng, Chang Liu, and Dongyan Zhao. 2023. More than classification: A unified framework for event temporal relation extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9631–9646. Association for Computational Linguistics.

EunJeong Hwang, Jay-Yoon Lee, Tianyi Yang, Dhruvesh Patel, Dongxu Zhang, and Andrew McCallum. 2022. Event-event relation extraction using probabilistic box embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 235–244, Dublin, Ireland. Association for Computational Linguistics.

Viet Dac Lai, Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, and Thien Huu Nguyen. 2022. MECI: A multilingual dataset for event causality identification. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 2346–2356. International Committee on Computational Linguistics.

Jing Lu and Vincent Ng. 2021. Constrained multi-task learning for event coreference resolution. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 4504–4514. Association for Computational Linguistics.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 25–32, Morristown, NJ, USA. 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 *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11058–11066. AAAI Press.

Bonan Min, Manaj Srivastava, Haoling Qiu, Prasannakumar Muthukumar, and Joshua Fasching. 2020. Learnit: On-demand rapid customization for event-event relation extraction. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 13630–13631. AAAI Press.

Minh Van Nguyen, Bonan Min, Franck Dernoncourt, and Thien Nguyen. 2022a. Learning cross-task dependencies for joint extraction of entities, events, event arguments, and relations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 9349–9360. Association for Computational Linguistics.

Minh Van Nguyen, Bonan Min, Franck Dernoncourt, and Thien Huu Nguyen. 2022b. Joint extraction of entities, relations, and events via modeling inter-instance and inter-label dependencies. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 4363–4374. 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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1318–1328. 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.

Elad Segal, Avia Efrat, Mor Shoham, Amir Globerson, and Jonathan Berant. 2020. A simple and effective model for answering multi-span questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3074–3080. Association for Computational Linguistics.

Shirong Shen, Heng Zhou, Tongtong Wu, and Guilin Qi. 2022. Event causality identification via derivative prompt joint learning. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 2288–2299. International Committee on Computational Linguistics.

Xingwei Tan, Gabriele Pergola, and Yulan He. 2023. Event temporal relation extraction with bayesian translational model. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 1117–1130. Association for Computational Linguistics.

Jialong Tang, Hongyu Lin, Meng Liao, Yaojie Lu, Xianpei Han, Le Sun, Weijian Xie, and Jin Xu. 2021. From discourse to narrative: Knowledge projection for event 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 732–742. Association for Computational Linguistics.

Hieu Minh Tran, Duy Phung, and Thien Huu Nguyen. 2021. Exploiting document structures and cluster consistencies for event coreference resolution. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4840–4850. Association for Computational Linguistics.

Marc B. Vilain, John D. Burger, John S. Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Message Understanding Conference*.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020. Joint constrained learning for event-event relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 696–706. Association for Computational Linguistics.

Haoyu Wang, Hongming Zhang, Yuqian Deng, Jacob R. Gardner, Dan Roth, and Muhao Chen. 2023a. Extracting or guessing? improving faithfulness of event temporal relation extraction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 541–553. Association for Computational Linguistics.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023b. Instructuie: Multi-task instruction tuning for unified information extraction.

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.

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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10431–10437. Association for Computational Linguistics.

Wei Xiang, Chuanhong Zhan, and Bang Wang. 2023. Daprompt: Deterministic assumption prompt learning for event causality identification. *CoRR*, abs/2307.09813.

Xinglin Xiao, Yijie Wang, Nan Xu, Yuqi Wang, Hanxuan Yang, Minzheng Wang, Yin Luo, Lei Wang, Wenji Mao, and Daniel Zeng. 2024. Yayi-uie: A chat-enhanced instruction tuning framework for universal information extraction.

Jun Xu, Mengshu Sun, Zhiqiang Zhang, and Jun Zhou. 2024. Chatuie: Exploring chat-based unified information extraction using large language models.

Jun Xu, Weidi Xu, Mengshu Sun, Taifeng Wang, and Wei Chu. 2022. Extracting trigger-sharing events via an event matrix. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1189–1201, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Junjie Yang, Zhuosheng Zhang, and Hai Zhao. 2021. Multi-span style extraction for generative reading comprehension. In *Proceedings of the Workshop on Scientific Document Understanding co-located with 35th AAAI Conference on Artificial Inteligence, SDU@AAAI 2021, Virtual Event, February 9, 2021*, volume 2831 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Packed levitated marker for entity and relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4904–4917, Dublin, Ireland. Association for Computational Linguistics.

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021. One2set: Generating diverse keyphrases as a set. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4598–4608. Association for Computational Linguistics.

Changsen Yuan, Heyan Huang, Yixin Cao, and Yonggang Wen. 2023. Discriminative reasoning with sparse event representation for document-level event-event relation extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 16222–16234. Association for Computational Linguistics.

Yutao Zeng, Xiaolong Jin, Saiping Guan, Jiafeng Guo, and Xueqi Cheng. 2020. Event coreference resolution with their paraphrases and argument-aware embeddings. In *Proceedings of the 28th International*

Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020, pages 3084–3094. International Committee on 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, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 2001–2010. International Committee on Computational Linguistics.

A Expansion and Downsampling

The specific approach is outlined as follows:

Positive sample expansion: To expand the number of positive samples, we employ two strategies: (1) randomly replacing non-event mention words or phrases with synonyms, and (2) using the Mask-then-Fill strategy. The Mask-then-Fill strategy involves generating an instruction for filling the [MASK] token. Meanwhile, non-event mentioned words or phrases in positive samples are randomly replaced with the [MASK] token. Then, ChatGPT is used to predict the content of the [MASK] token. In this way, a new positive sample is produced. Finally, each positive sample is expanded to create three additional positive samples.

Input:
3,000 civilians were killed and another 4,500 injured. . .

Mask:
[MASK] were killed and [MASK] injured. . .

Fill:
ten soldiers were killed and twenty injured. . .

Negative sample downsampling: The large number of negative samples presents a challenge for training an effective model. To tackle this problem, we decided to decrease the number of negative samples through downsampling. Our key strategies are two-fold: first, we randomly remove the marker (<0x**>) from specific invalid event mentions; second, we utilize llama-2-chat to extract and predict event relations in texts that lack any relations, and subsequently randomly remove samples without event relations. It is important to note that these techniques are specifically applied to the training dataset, ensuring that the integrity of the test set remains intact.

B Evaluation Details

Coreference relations are distinguished by their transitive nature, unlike other types of event relations. Therefore, we will continue to use the evaluation metrics B^3 (Bagga and Baldwin, 1998), $CEAF_e$ (Luo, 2005), MUC (Vilain et al., 1995) and BLANC (RECASENS and HOVY, 2011), as established by the previous method. The essence of B^3 lies in considering the contribution of each individual event mention. The system calculates the precision and recall for each coreference event mention and then averages these across all event mentions. This means that every event mentioned impacts the overall score equally, regardless of the size of the chain it belongs to. $CEAF_e$ takes into account the alignment between coreferent event mentions and chains. The system matches the coreference chains generated with the gold-standard chains and evaluates accuracy based on the best alignment. MUC focuses on merging coreference chains with a minimal number of operations. The performance is evaluated based on the minimum number of merge operations required to align the system's identified chains with the answer key chains. This method is usually very sensitive to missing or incorrect links. BLANC is a relatively new metric designed to assess the accuracy of both coreferent and non-coreferent decisions. It considers not only the correctly linked entities but also the accurate identification of entities that are not linked. Therefore, BLANC provides a more comprehensive perspective on coreference resolution performance. Finally, we use precision (P), recall (R), and F1 measure as the evaluation metrics for other event relation extraction tasks.

C Implementation Details

We utilize the llama-2-chat as the textual encoder, which consists of 32 layers, 4096 hidden units, and 32 attention heads. We train the model using an Adam optimizer with weight decay, and the weight decay rate is 1e-4. The warm-up proportion for the learning rate is 0.1, and the dropout rate is 0.1. The temperature used to adjust the probabilities of the next token is set to 0.01, and the smallest set of the most probable tokens with probabilities top_p that add up to 0.9. In the output, we use ":" (token id 584) as a delimiter to distinguish the Parse-CoT from the Multiple Answers.

D Comparison Methods Details

In this section, we provide more implementation details of the baselines. For a fair comparison, all of these models are implemented using PyTorch and tested on the NVIDIA TESLA A100 GPU. **BertERE** treats event relation extraction as a multiclass classification problem. The various types of relations between events form the label set for the classification model. For **BertERE**$_{joint}$, we utilize RoBERTa as the backbone network, setting the learning rate for the Transformer at 2e-5 and for the classification multilayer perceptron at 5e-4. When providing text input, the system selects the longest text containing the event pair, with a maximum length limit of 512. **LLM-based** method treats event relation extraction as a text generation task, and its backbone network, pre-trained models, and training parameters are consistent with those of MAQERE.

E Expansion and Downsampling Analysis

There are a large number of event mentions, but the proportion of event mention pairs that actually have a relation is comparatively small, as indicated by the data ($\frac{Links}{Mentions \times Mentions}$) in Table 2. Regardless of the approach employed (classification, LLM, or MAQ), the model struggles to assimilate valuable information when trained on all event mention pairs. To tackle this issue, it is necessary to increase the number of positive samples and decrease the number of negative samples. Importantly, to ensure consistency in evaluation, data augmentation and sampling techniques are only applied to the training dataset. For positive sample expansion, as shown in Table 8, we employ a LLM with a Mask-then-Fill technique, which has been found to be more effective than simply replacing words with their synonyms. However, there are cases where the LLM fails to generate a sufficiently diverse range of samples. In such cases, using synonyms can be a more suitable approach. When downsampling negative samples, randomly removing markers from event mentions can effectively improve the performance of the model. Additionally, leveraging the LLM for zero-shot predictions helps preserve the more challenging samples.

Method		MATRES			MECI		
		P	R	F1	P	R	F1
Expan.	Synonym	84.9	82.7	83.8	62.2	60.6	61.4
	M & F	83.5	81.1	84.3	63.8	60.5	62.1
	Mixed	85.5	83.9	84.7	62.9	61.6	62.3
Samp.	Random	83.1	82.1	82.6	60.5	61.3	60.9
	LLM Pred	84.7	83.1	83.9	61.3	62.1	61.7
	Mixed	85.5	83.9	84.7	62.9	61.6	62.3

Table 8: The impact of positive sample expansion and negative sample downsampling on the model.

F Different Instructions Analysis

The event relation extraction model based on LLM is greatly affected by instructions. We conducted experiments to validate different sets of instructions and found that, for fixed tasks, shorter and more concise instructions tend to be more effective. Simultaneously, we conducted several tests, as presented in Table 9. Firstly, providing all potential event mentions in the instruction resulted in a slight drop in the F1 score. Secondly, when the model is allowed to directly generate event relations based on event mentions, its performance significantly decreases due to the large number of event mention pairs generating relations labeled as NoRel. When multiple different relations are generated simultaneously, the model's performance is at its worst.

Instruction	MECI
List the ***cause*** event of <0x85> ***earthquake*** ?	62.3
Find the ***cause*** event of <0x85> ***earthquake*** from the event mentions <0x71> ***scorched***, . . . ?	61.7
What's the event relation between <0x85> ***earthquake*** and <0x71> ***scorched***, <0x72> ***deny***, . . . ?	60.4
List the ***cause*** and ***effect*** event of <0x85> ***earthquake*** ?	56.6

Table 9: The F1 score of MAQERE on MECI varies among different instructions.

G Different Markers Analysis

In our study, we use various markers to prompt event mentions, building on previous research (Lu et al., 2022; Ye et al., 2022). The experiments are divided into three groups, as outlined in Table 10. The first set of experiments utilizes special tokens already present in llama-2-chat as markers, such as <0x**>. This method produced the best results compared to the other sets of experiments. Additionally, we observed that adding the special end character after the event mention does not improve performance. This is primarily due to the lack of actual semantic information and the use of multiple tokens, which compromises the original semantic

Marker	Tokenizer	MECI
<0x64>	[103]	62.3
<0x64> </>	[103] [1533, 29958]	62.1
<No64>	[529, 3782, 29953, 29946, 29958]	59.5
<No64> </>	[529, 3782, 29953, 29946, 29958] [1533, 29958]	59.3
<Strong>	[529, 1110, 29958]	60.2
<Strong> </>	[529, 1110, 29958] [1533, 29958]	59.7

Table 10: The F1 score of MAQERE on MECI varies among different markers.

coherence. In the second set of experiments, we replaced <0x**> with <No**> and observed a significant drop in the model's effectiveness. As in the previous case, the insertion of too many tokens results in semantic incoherence. In the third set of experiments, all event mentions are inserted into the same marker, resulting in a noticeably worse effect.