

Improving Large Language Models in Event Relation Logical Prediction

Meiqi Chen^{1*}, Yubo Ma², Kaitao Song^{3†}, Yixin Cao⁴, Yan Zhang^{1†}, Dongsheng Li³

¹ Peking University ² Nanyang Technological University ³ Microsoft Research Asia

⁴ School of Computer Science, Fudan University

meiqichen@stu.pku.edu.cn, yubo001@e.ntu.edu.sg

{kitaosong, dongshengli}@microsoft.com,

caoyixin2011@gmail.com, zhyzhy001@pku.edu.cn

Abstract

Event relations are crucial for narrative understanding and reasoning. Governed by nuanced logic, event relation extraction (ERE) is a challenging task that demands thorough semantic understanding and rigorous logical reasoning. In this paper, we conduct an in-depth investigation to systematically explore the capability of LLMs in understanding and applying event relation logic. More in detail, we first investigate the deficiencies of LLMs in logical reasoning across different tasks. Our study reveals that LLMs are not logically consistent reasoners, which results in their suboptimal performance on tasks that need rigorous reasoning. To address this, we explore three different approaches to endow LLMs with event relation logic, and thus enable them to generate more coherent answers across various scenarios. Based on our approach, we also contribute a synthesized dataset (LLM-ERL) involving high-order reasoning for evaluation and fine-tuning. Extensive quantitative and qualitative analyses on different tasks also validate the effectiveness of our approaches and provide insights for solving practical tasks with LLMs in future work. Codes are available at <https://github.com/chenmeiqii/Teach-LLM-LR>.

1 Introduction

Understanding the relationships between events is fundamental to effective communication and reasoning, a challenge central to the field of Event Relation Extraction (ERE). ERE tasks, which involve identifying coreference, temporal, causal, and subevent relationships, demand not only semantic comprehension but also rigorous logical reasoning. Despite recent advances in Large Language Models (LLMs) such as ChatGPT (Ouyang

*This work was done during her internship at Microsoft Research Asia.

†Corresponding author.

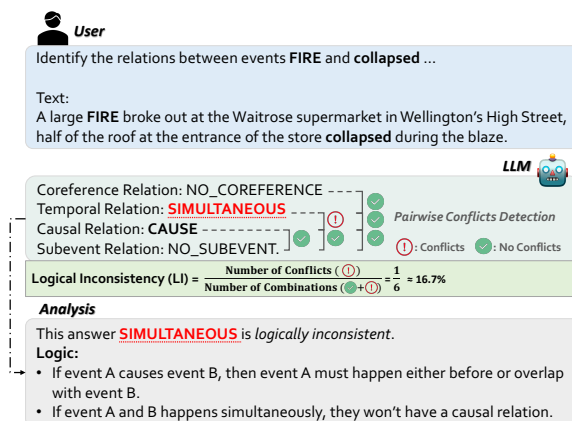


Figure 1: An example of LLM in generating logically inconsistent answers. We let an LLM (e.g., ChatGPT) predict the relations between events “FIRE” and “collapsed” from the given passage. We can find that LLM predicts an incorrect answer (i.e., SIMULTANEOUS) because it ignores some prior logic in this scenario.

et al., 2022) and Llama2 (Touvron et al., 2023), these models struggle to fully grasp the complexities of event relation logic, often failing to apply it accurately in ERE tasks.

As showcased in Figure 1, ChatGPT incorrectly predicts the temporal and causal relations between events “FIRE” and “collapsed” as “simultaneous” and “cause”, respectively. According to the prior logical constraints, we could readily claim the predictions are not fully correct even before reading the context. Some works (Xu et al., 2023; Pan et al., 2023; Lyu et al., 2023) attribute this gap in logic adherence to LLMs’ inherent deficiencies (e.g., hallucination, unfaithfulness). However, how to disentangle and improve the capability of LLMs in these tasks is still an open problem.

To deeply understand the deficiencies of LLMs in logical reasoning and explore the corresponding solutions, in this paper, we conduct an in-depth investigation of LLMs in solving reasoning tasks from multiple dimensions. Our experimental re-

sults show that: **1)** Even the cutting-edge LLMs still generate large amounts of inconsistent answers, e.g., over 60% of the answers from ChatGPT on the MAVEN-ERE (Wang et al., 2022a) dataset are logically inconsistent as shown in Figure 2; **2)** Providing relevant logic to LLMs improves performance, but injecting irrelevant logic introduces fluctuations in results. Therefore, how to obtain the relevant logic and inject its information into LLMs is a non-trivial problem, deserving further exploration.

Based on these findings, we put forward a series of solutions to endow LLMs with event relation logic and generate more coherent answers. Here, we propose three different kinds of approaches according to the ways of logic acquisition: **1) Generative-based approach**, which encourages LLMs to generate rationale themselves, inspired by CoT prompting (Wei et al., 2022b). In this paradigm, we find that incorporating logical constraints into LLM instruction will bring substantial improvements, but the uncertainty of the generated rationales may also bring some biases, leading to an incorrect subsequent answer; **2) Retrieval-based approach**, which collects constraints from realistic data, then retrieves relevant contents and adds them to the LLM instruction. This kind of approach ensures the correctness of logic and significantly improves performance, but requires some hand-crafted engineering; **3) Finetuning-based approach**, which first constructs a high-order event relation logical prediction dataset (LLM-ERL), then uses it to fine-tune specialized LLMs. The finetuning dataset consists of multi-hop event relation logical prediction instances. This strategy encodes logic in model parameters inherently, making them more suitable for white-box LLMs. Therefore, how to choose the most suitable strategy can be a trade-off based on the practical scenario.

Furthermore, based on the above framework, we also conduct extensive quantitative and qualitative analyses to validate the effectiveness of the proposed approaches and provide insights for future work: **1)** Directly using CoT to infer ERE tasks is limited by the inherent issues of LLMs, but incorporating logical constraints in the reasoning process can be beneficial; **2)** Retrieval-based approaches can significantly reduce inconsistencies in LLM responses. Stronger models like GPT-4 can effectively perform retrievals by themselves, whereas weaker models require assistance in filtering relevant information. Besides, directly conveying constraints to LLMs is more effective than adding

post-processing operations based on the results; **3)** When fine-tuned on LLM-ERL, LLMs such as Llama2-13B (Touvron et al., 2023) can achieve better performance, which validates the effectiveness of our proposed approaches.

Overall, the contributions of our paper can be summarized as follows:

- We provide an in-depth investigation of the logical inconsistency issue of current LLMs, highlighting their challenges in understanding event relation logic.
- We propose several solutions to endow LLMs with event relation logic and generate more coherent answers. Based on our approach, we construct a synthesized dataset (LLM-ERL) involving high-order reasoning to enhance LLMs.
- Experimental results on different tasks with quantitative and qualitative analyses further verify the effectiveness of our approach in endowing LLMs with event relation logic.

2 Event Relation Logic

2.1 Event Relations

In this subsection, we introduce four common types of event relations that are crucial for narrative comprehension and reasoning. *Coreference relations*: identify whether two event mentions refer to the same occurrence. *Temporal relations*: establish the chronological order of events. *Causal relations*: identify causality between events. *Subevent relations*: identify whether one event is a subcomponent of another. More descriptions of these event relations can be found in Appendix A.

Based on these four relations, event relation extraction (ERE) can be formulated as a multi-label classification problem, assigning one label for each relation type. Compared with other common tasks, ERE tasks should take more considerations about the logical constraints between event relations (e.g., as shown in Figure 1), and guarantee the predictions should conform to these constraints to avoid counterfactuals. Therefore, we need to rigorously consider the logical constraints between each event pair during prediction. To better measure the capability of LLMs on the ERE task, we formulate the logical consistency metric.

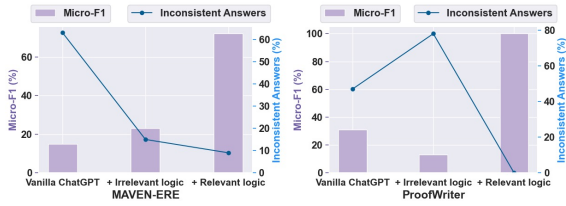


Figure 2: Performance of ChatGPT in the pilot study.

2.2 Logical Consistency Between Event Relations

Logical consistency plays a crucial role in accurate event relation prediction. In this paper, we consider a comprehensive set including 11 logical constraints applicable to all possible relations between two events, which are derived from realistic data and are detailed in Appendix B. To quantify LLMs’ adherence to these constraints, we introduce a metric called *Logical Inconsistency (LI)*. This metric is calculated as the proportion of conflicts (i.e., the answers that conflict with the known logical constraints) to the total possible relation combinations (i.e., all combinations between any two relation types).

To better illustrate the computation of LI, here we introduce an example (as shown in Figure 1): if an LLM outputs the relations between two events as “NO_COREFERENCE, SIMULTANEOUS, CAUSE, NO_SUBEVENT”. Among these, “SIMULTANEOUS” and “CAUSE” are identified as conflicting with each other based on the logical constraints we have defined, creating an inconsistency. Considering there are four relation types to assess for each event pair, the total number of relation combinations is determined by the formula: $C_4^2 = 6$. Thus in this example, with one identified conflict, LI is computed as 1/6 (or approximately 16.7%). Based on the logical constraints, an algorithm can be designed to automatically detect conflicts and calculate the value of LI. Intuitively, the smaller the value of LI is, the more coherent and reasonable answer that LLM can produce.

3 Unveiling LLMs in Logical Reasoning

Considering the rigorous logical reasoning required by ERE tasks, in this section, we conduct a pilot study to investigate how current LLMs exhibit reasoning tasks and how logic benefits LLMs.

3.1 Data Source

We conduct a manual evaluation on MAVEN-ERE (Wang et al., 2022a) and ProofWriter (Tafjord et al., 2021). MAVEN-ERE is a unified large-scale dataset for the ERE task, which needs to identify four types of relations. ProofWriter is a commonly used dataset for deductive reasoning, where each example is a pair of (problem, goal) and the label is selected from {Proved, Disproved, Unknown}. To employ our investigation, we randomly choose 100 samples (50 from MAVEN-ERE and 50 from ProofWriter).

3.2 Experimental Setup

Our experiments are conducted in a zero-shot fashion. Given a task input (X), we also write a prompt (T) describing the task, and let LLM generate output (Y) by answering the given query. We also add “Let’s think step by step” before each answer for prediction generation, which is a simple but effective trick to improve zero-shot reasoning for LLMs (Kojima et al., 2022). We adopt ChatGPT as the backbone and manually check its generated rationales under the following three settings:

- Vanilla LLM (i.e., ChatGPT) without any additional information;
- LLM (i.e., ChatGPT) plus the most relevant (i.e., ground truth) logic;
- LLM (i.e., ChatGPT) plus irrelevant logical constraints.

The latter two use a multi-turn conversational way based on the initial prediction from LLMs, so as to leverage LLM’s interaction ability. The process of determining constraints for each way and the corresponding prompt examples can be found in Appendix J.1.

3.3 Analysis

As shown in Figure 2, we visualize the micro-F1 values and the proportion of logically inconsistent answers generated by ChatGPT. We find that no matter whether on MAVEN-ERE or ProofWriter, Vanilla ChatGPT always achieves a bad result with low micro-F1 performance and high inconsistency values (e.g., 15% micro-F1 and 63% inconsistent answers on MAVEN-ERE), which indicates the deficiencies of LLM in solving complex reasoning tasks. To investigate this issue in depth, we conduct analyses from the following two aspects.

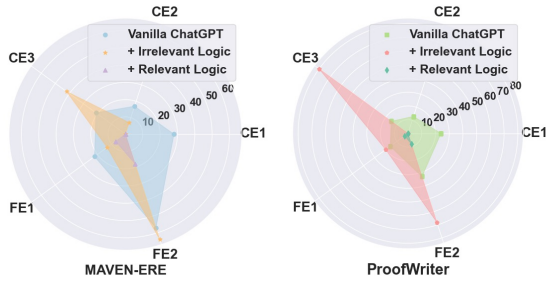


Figure 3: Error analysis of ChatGPT in the pilot study by human evaluation. CE and FE denote incorrectness and unfaithfulness errors, respectively.

What is the Relation Between Logical Consistency and Model Performance? From Figure 2, we find that: 1) The model directly receives significant improvements on both MAVEN-ERE and ProofWriter when adding relevant logic; 2) When adding some irrelevant logic, the results show some fluctuations (exaltation in MAVEN-ERE and degeneration in ProofWriter). That means directly adding logic without any constraints will bring some uncertainty; 3) Typically, a higher logical inconsistency corresponds to a poorer micro-F1. However, rectifying logical inconsistency does not necessarily lead to the same degree of increase in micro-F1. Generally, an intuitive observation is that incorporating relevant logic into the LLM instruction will be very helpful in solving reasoning tasks. Therefore, the challenges are how to obtain these relevant logic and how to utilize them for LLMs.

What Types of Errors Does LLM Usually Make?

To delve into a deep understanding of the failures that vanilla LLM encounters in logical reasoning, we also conduct a detailed error analysis. Here, we divide the error types into two aspects: 1) *Incorrectness to the Constraint (CE)*: whether the rationale generated by LLM is wrong (CE1), incomplete (CE2), or redundant (CE3) compared with the true logical constraints. 2) *Unfaithfulness to the Reasoning Process (FE)*: where LLM does not correctly use the constraints. We define two types of errors upon FE, i.e., i) Wrong start, LLM begins with an irrelevant fact or focuses on an improper perspective for the correct answer (FE1). ii) Wrong process, LLM starts from a proper point, but makes mistakes during the reasoning process (FE2). Annotators are asked to review 100 predictions generated by ChatGPT and mark the error types. Results in Figure 3 show that: 1) The quality of constraints produced by the vanilla ChatGPT

is not high enough, which limits its subsequent reasoning ability. 2) Incorporating relevant logical constraints could guarantee the correctness of constraints and thus greatly improve the generation quality of ChatGPT in faithfulness.

4 Teaching LLMs to Predict Event Relation Logic

From the above analysis, the main reason for the failure of LLMs stems from their lack of logical reasoning abilities. In this section, we expect to explore how to augment LLMs with the capability to comprehend and apply event relation logic. Specifically, we first introduce the instruction-following technique used in Section 4.1 and then propose three different approaches to instruct LLMs to generate answers with better logical consistency, including generative-based, retrieval-based, and finetuning-based approaches (Section 4.2 to 4.4). We illustrate these three approaches in Figure 4.

4.1 In-Context Learning for LLMs

We deploy LLMs for event relation logical prediction via in-context learning (ICL, (Brown et al., 2020; Ouyang et al., 2022)). Given a task input (X), we write a prompt (T) describing the task, then further provide several demonstrations $D = \{D_i\}_{i=1}^{|D|}$, where $D_i = (X_i, Y_i)$ are used for few-shot learning. Then, the LLM generates the output (Y) by completing the prompt ($Y = \mathcal{M}(T, D, X)$), where \mathcal{M} denotes the LLM. In such a setting, the LLM can follow the structure of the provided demonstrations to output answers in the expected format for subsequent automatic evaluation. Additionally, the whole process does not require any gradient update, allowing LLMs to generate predictions without massive training data.

4.2 Generative-based Approaches

Generative-based approaches involve letting LLMs generate logic by using a form of few-shot ICL. Here, we study three variants:

Vanilla ICL: which utilizes the common prompts consisting of the task description, the demonstration, and the input case.

Vanilla CoT: which first bootstraps rationales by using chain-of-thought as intermediate reasoning steps following the style of the given demonstration, then output answers. Rationales here do not involve the content of logical constraints.

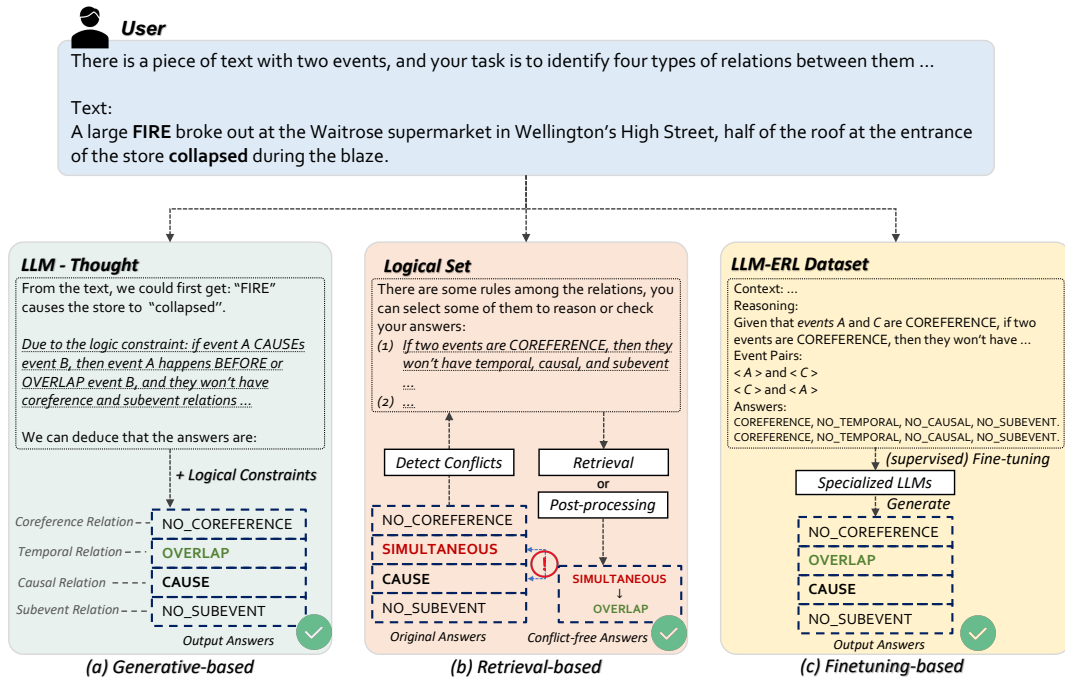


Figure 4: Incorporate logical constraints into LLMs by using generative, retrieval, and finetuning-based approaches. The dashed boxes indicate answers outputted by LLMs, and the underlined texts indicate the logical constraints.

CoT with self-generated logical constraints: which teaches LLMs to generate and utilize logical constraints based on CoT (shown in Figure 4 (a)). Specifically, it first extracts the obvious relations/facts and generates relevant logical constraints accordingly. LLMs are then prompted to infer the remaining relations and facts using these constraints along with the known information. An example prompt is provided in Appendix J.2.

4.3 Retrieval-based Approaches

Although generative-based approaches enable models to automatically generate and utilize logic, the generated rationales may be uncertain and inaccurate. Therefore, we also provide retrieval-based approaches, which aim to obtain relevant logic from our predefined logical set and add it to LLM instruction (shown in Figure 4 (b)). Specifically, we take all the constraints defined in Section 2.2 as the retrieval set, and our solutions include:

with all logical constraints: which directly adds all the text of logical constraints in the set.

with retrieved logical constraints: which means that we first detect logically inconsistent answers based on the prediction of LLMs, and then retrieve the corresponding information if we find any conflicts. Finally, we add the retrieved text to the LLM instruction and let LLMs regenerate the answers.

Details can be found in Appendix B.1.

with post-processing: which first obtains the answers of LLMs, then automatically generates some logically consistent candidates according to the known constraints, and randomly selects one of them as the final answer. This approach ensures that there are no logical conflicts ($LI = 0\%$). Details can be found in Appendix B.2.

4.4 Finetuning-based Approach

Although the retrieval-based approach guarantees the correctness of logical constraints, it still needs to interact with an external logical set constantly. Therefore, we provide a finetuning-based approach to embed the logical constraints into LLMs themselves. Specifically, we first construct a high-order event relation logical prediction dataset LLM-ERL, then fine-tune specialized models on it, and finally use the fine-tuned models to conduct prediction.

To construct LLM-ERL, we initiate with a foundational set of logical constraints for relations between two events that have been defined in Section 2.2, and expand it to include additional constraints for high-order relations among three or more events based on *transitive dependency* (Allen, 1983; Gerevini and Schubert, 1995), i.e., one event may affect another through an intermediate event. The full transitivity rules are detailed in Appendix C (Table 6).

Dataset Construction Once obtaining the constraint set, the process of dataset construction becomes inferring new relations within a sequence of events based on any given relations. From there, we combine an initial relation with any other given relation to form a multi-hop query. This query aims to predict the logical outcome of a complex event interaction that spans multiple steps, leveraging the established logical constraints as a guide. For instance, if we have an initial relation “BEFORE(A , B)”, and we combine this with another two relations such as “SIMULTANEOUS(B , C)” and “OVERLAP(C , D)”, we are faced with a 3-hop query that seeks to deduce the relation between event A and event D . Given the logical constraints, such as the transitivity rule that combines “BEFORE” and “SIMULTANEOUS” relations to infer new relations, we can deduce a logical outcome “BEFORE(A , D)”. The corresponding pseudocode can be found in Appendix C.1.

The process of deducing the answer to these multi-hop queries is automated by employing logic programming (Lloyd, 2012; Frederiksen, 2008), specifically using forward- and backward-chaining methods in Prolog (Clocksin and Mellish, 2003). This allows for the automatic inference of new relations based on the established set of logical constraints and the known relations among events. The outcome of this process can not only serve as the benchmark for evaluating or enhancing the reasoning capabilities of LLMs, but also act as a versatile platform for validating combinations of event relations across any number of hops.

Fine-tuning on LLM-ERL To fine-tune LLMs on LLM-ERL, we use the generated 2 to 5-hop reasoning data. We do not adopt longer hop data here considering the computation complexity and the length limitation of LLMs. We translate the symbolic representations of event relations into natural language descriptions to formulate queries, aligning with the ERE task setup. This process resulted in a total of 6,776 instances. The dataset statistics are in Appendix D and an illustrative example of such a prompt is depicted in Figure 4 (c). These queries not only promote LLMs’ understanding of the logical constraints governing event sequences but also enhance their ability to apply these constraints in predicting the relations among events that are not explicitly given. Finally, we could conduct inference with the fine-tuned LLMs.

5 Experiments

5.1 Experimental Setup

Compared Models We choose several limited-access LLMs (*gpt-3.5-turbo*, *text-davinci-003*, and *gpt-4*), and open-source LLMs (Vicuna-13B (v1.3) (Chiang et al., 2023) and Llama2-13B (Touvron et al., 2023)) as the main experimental LLMs for evaluation. We also provide two fine-tuning RoBERTa-large (Liu et al., 2019) baselines (one-shot and fully fine-tuned) for comparison, the fine-tuning details can be found in Appendix F.

Dataset Construction Our main experiments are evaluated on two ERE datasets, MAVEN-ERE (Wang et al., 2020) and Causal-TimeBank (Mirza et al., 2014). All experiments are conducted in a one-shot fashion. Further details can be found in Appendix E.

Fine-tuning Details For the finetuning-based approach, we adopt Vicuna-13B (v1.3) and Llama2-13B as the base models and employ the LoRA (Hu et al., 2022) technique. During fine-tuning, only LoRA parameters are optimized. The finetuned models are named Vicuna-FT and Llama2-FT, respectively. Further details can be found in Appendix G.

Evaluation Metrics We adopt the averaged micro-F1 score as the evaluation metric and also report the logical inconsistency metric LI (defined in Section 2.2) on ERE datasets. The reported value is averaged by the results of three runs to reduce random fluctuation.

5.2 Main Results

From Table 1, We could observe that:

Generative-based Approaches 1) Compared with a smaller language model RoBERTa-large, the generalization ability of vanilla LLMs under the one-shot setting is remarkable, but there is still a gap with the fully-finetuned baseline.

2) Directly using CoT to infer logic does not help much for ERE tasks, a possible reason is that the inherent issues of LLMs may cause them to fail in generating precise rationales (i.e., a high ratio of logical inconsistency).

3) When using generative-based approaches to encourage LLMs to produce logical constraints in the reasoning process, LLMs can significantly improve their performance on ERE tasks (e.g., 7.3%

Model		MAVEN-ERE		Causal-TimeBank	
		Micro-F1 (%)	LI (%) ↓	Micro-F1 (%)	LI (%) ↓
RoBERTa-Large (fully fine-tuned)		56.8	6.4	22.2	36.2
RoBERTa-Large (one-shot)		17.4	54.8	-	-
Turbo	vanilla ICL	18.0	53.3	19.0	54.0
	vanilla CoT	18.8	49.3	17.0	30.3
	CoT w. logical constraints	25.3	37.9	27.0	12.8
	w. all logical constraints	20.8	30.9	20.0	36.8
	w. retrieved logical constraints	22.3	30.2	22.0	11.3
	w. post-processing	14.0	0	15.0	0
Davinci	vanilla ICL	21.6	49.1	18.0	58.8
	vanilla CoT	20.5	60.5	21.0	64.7
	CoT w. logical constraints	24.8	5.5	23.0	39.2
	w. all logical constraints	27.0	25.6	31.0	21.8
	w. retrieved logical constraints	27.8	30.8	22.0	40.5
	w. post-processing	14.8	0	19.0	0
GPT-4	vanilla ICL	29.3	50.7	22.5	30.5
	vanilla CoT	30.3	36.7	23.0	35.0
	CoT w. logical constraints	32.3	13.7	24.5	24.0
	w. all logical constraints	37.3	8.3	26.0	20.0
	w. retrieved logical constraints	33.5	28.8	24.0	13.5
	w. post-processing	17.0	0	19.0	0
Vicuna	vanilla ICL	13.8	25.4	4.5	84.1
	vanilla CoT	11.6	47.4	6.0	57.6
	CoT w. logical constraints	14.9	21.7	8.0	33.1
	w. all logical constraints	15.2	37.6	11.0	23.5
	w. retrieved logical constraints	15.7	33.2	10.0	26.7
	w. post-processing	9.8	0	9.0	0
Llama2	vanilla ICL	17.0	54.6	11.5	26.7
	vanilla CoT	17.8	58.4	10.5	33.6
	CoT w. logical constraints	21.5	18.9	13.0	18.1
	w. all logical constraints	19.5	34.6	10.0	23.5
	w. retrieved logical constraints	18.3	38.2	9.5	26.7
	w. post-processing	12.0	0	9.5	0
Vicuna-FT	vanilla ICL	15.3	21.2	8.0	35.5
	vanilla CoT	15.8	17.8	7.5	52.5
	CoT w. logical constraints	18.0	6.0	8.5	2.0
	w. all logical constraints	16.3	8.7	12.1	0
	w. retrieved logical constraints	16.1	19.0	10.7	9.5
	w. post-processing	11.0	0	8.0	0
Llama2-FT	vanilla ICL	19.0	45.8	12.0	22.7
	vanilla CoT	22.1	42.9	11.5	3.0
	CoT w. logical constraints	26.4	15.7	13.3	13.0
	w. all logical constraints	20.2	28.7	12.0	23.0
	w. retrieved logical constraints	18.7	34.2	11.0	19.4
	w. post-processing	11.0	0	11.0	0

Table 1: Proprietary LLMs (*gpt-3.5-turbo*, *text-davinci-003*, and *gpt-4*), Vicuna-13B, Llama2-13B’s performance on MAVEN-ERE and Causal-TimeBank. “PT” denotes after fine-tuning on LLM-ERE. For each dataset, the best result of each LLM is in **bold**. RoBERTa-Large (one-shot) fails to output any correct answers on Causal-TimeBank. The highlighted colors denote generative-based, retrieval-based, and finetuning-based approaches, respectively.

F1 performance gains from 18.0% to 25.3% of *gpt-3.5-turbo* on MAVEN-ERE). We give a case study for the generative-based approach in Appendix I.1, which shows how LLMs perform when generating logical constraints by themselves.

Retrieval-based Approaches 1) When using retrieval-based approaches to obtain logic constraints and incorporate them into LLM instruction, the logical inconsistency of LLMs’ answers is greatly reduced and the overall performance is fur-

ther improved (e.g., 6.2% F1 performance gains from 21.6% to 27.8%, and 18.3% LI decrease from 49.1% to 30.8% of *text-davinci-003* on the MAVEN-ERE dataset).

2) Among all the limited-access models, we find that only *gpt-4* perform better under the “*w. all logical constraints*” setting compared with the “*w. retrieved logical constraints*” setting. We hypothesize that this is due to the superior language understanding and retrieval capabilities of *gpt-4*, enabling it to identify some useful logical constraints to derive the answers accurately. In contrast, earlier models may struggle to filter out irrelevant information and therefore still require our assistance in retrieval to screen the necessary information.

3) Although the post-processing baseline guarantees the absence of logical conflicts (resulting in LI of 0%), it may severely affect the quality of the whole generation. On one hand, the semantics of the post-processing answer may be far from the ground truth due to the random selection. On the other hand, the size of the candidate set for each case will also affect the performance. It may also need more operations at the post-processing stage, which we leave as future work. We also conduct ablation studies on the number of demonstration samples and iterative retrievals in Section 5.3.

Finetuning-based Approach 1) Once fine-tuned on LLM-ERL, the performance of Llama2-FT and Vicuna-FT improves greatly compared with vanilla Llama2 and Vicuna, especially on the baselines without logical constraints.

2) The performance of Llama2-FT (i.e., 26.4% F1 score on MAVEN-ERE) could even surpass that of some greater LLMs (e.g., vanilla *gpt-3.5-turbo*, 25.3%), which further validates the importance of teaching LLM with event relation logic in solving ERE tasks. We also conduct a case study comparing the output answers of Llama2 and Llama2-FT in Appendix I.2.

5.3 Ablation Study

We conduct an ablation study using ChatGPT (*gpt-3.5-turbo*) in this section.

Demonstrations Following previous experiences (Brown et al., 2020), we also append demonstrations into the prompt to investigate how logical constraints will affect when combined with different numbers of demonstrations. Here, we select different numbers of demonstration samples K from {1, 5, 10, 20}. The experiments are tested

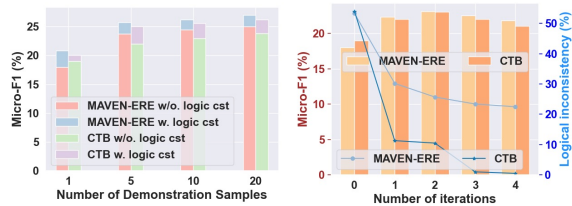


Figure 5: Ablation Study of ChatGPT for demonstrations and iterative retrieval, where “logic cst” denotes the event relation logical constraints.

on the “*w. all logical constraints*” settings, and we choose the “*vanilla ICL*” baseline for comparison. From Figure 5 (left), we can observe that: 1) When the number of demonstrations increases from 1 to 5, there is an evident performance improvement, but the subsequent improvements are limited when continue to increase the number of demonstrations (e.g., ≥ 10); 2) Adding logical constraints into LLM instructions can provide stable improvements, especially with more demonstrations; 3) The performance of incorporating logical constraints with a smaller number of demonstrations can even surpass that of prompts with only a larger number of demonstrations (e.g., the F1 performance of using 5 demonstrations on MAVEN-ERE w. logical constraints, 25.7%, surpasses that of 10 demonstrations w/o. logical constraints, 24.5%). This indicates that it is important to tell LLMs both “What” (demonstrations) and “How” (logical constraints). Overall, these studies further confirm the merits of using event relation logic in solving ERE tasks.

Iterative Retrieval Considering the outstanding ability of LLMs in interaction, we further explore whether we can introduce logical constraints into the multi-turn conversation (for the prompt design, please see Appendix J.3). Here, we adopt a retrieval-based approach to incorporate retrieved logical constraints iteratively and the results are shown in Figure 5 (right). We find that the logical inconsistency of answers will gradually decrease with the increase of iterations, but the overall micro-F1 score seems relatively stable. We guess the main reason for this phenomenon is the overthinking of LLMs, as although it can bring more reasoning rationale, it possibly produces correct but more useless or abundant information when inferring multiple iterations. Overall, instructing LLM with logic is beneficial for conversation, but how to support longer information is still challenging.

6 Related Work

6.1 Large Language Models (LLMs)

We are fortunate to witness the surging development of Large Language Models (LLMs (Brown et al., 2020; Ouyang et al., 2022; Chowdhery et al., 2022; Chung et al., 2022)), and a series of work aiming to leverage the reasoning abilities of LLMs such as chain-of-thought prompting (Wei et al., 2022a; Kojima et al., 2022; Zhang et al., 2022), self verification (Wang et al., 2022c; Jung et al., 2022), self learning (Zelikman et al., 2022; Huang et al., 2022), etc. However, recent studies show LLMs still stumble in generating hallucination and logic inconsistency (Golovneva et al., 2022; Jang and Lukasiewicz, 2023; Bang et al., 2023; Liu et al., 2023; Jiao et al., 2023). To solve such challenges, our work explores teaching LLMs logical reasoning through various approaches.

6.2 Event Relation Extraction (ERE)

Events play crucial roles in comprehending narratives, and understanding the complex relationships between events is essential to understanding the text (Sundheim, 1991). Thus ERE tasks are fundamental information extraction (IE) tasks and support various downstream applications (Chaturvedi et al., 2017; Zhang et al., 2020). Extensive studies have been carried out on ERE tasks, including different kinds of relations such as coreference relations (Lu and Ng, 2021; Lu et al., 2022), temporal relations (Ning et al., 2018; Wang et al., 2020; Han et al., 2019; Zhou et al., 2021), causal relations (Caselli and Vossen, 2017; Chen et al., 2022, 2023), and subevent relations (Aldawsari and Finlayson, 2019; Wang et al., 2021).

There also have been some recent explorations on how to leverage the power of LLMs on event-related information extraction tasks (Wang et al., 2022b; Gao et al., 2023; Ma et al., 2023; Qiu et al., 2023; Yuan et al., 2024). To the best of our knowledge, however, our work is the first to 1) design elaborate experiments to evaluate the performance of LLMs on the ERE task, including coreference, temporal, causal, and subevent relations, 2) delve into the high-order logical constraints between these event relations, and (3) analyze the logical reasoning abilities of LLMs using ERE as an intermediate task.

7 Conclusion

In this paper, we conduct a detailed investigation on how to enhance LLMs with event relation logic. Specifically, we first investigate the existing issues of current LLMs in event relation logical prediction. Then, we study multiple strategies to obtain and utilize logic for LLMs, including generative-based, retrieval-based, and finetuning-based approaches. Based on our approaches, we also contribute a synthesized dataset (LLM-ERL) involving multi-hop reasoning for evaluation and fine-tuning. We show that LLMs are not logically consistent reasoners, but their performance could be improved if we explicitly teach them the logical constraints. Comprehensive quantitative and qualitative analyses have been conducted to further provide insights.

Limitations

Although we have explored a series of approaches in detail to enhance LLMs to generate more logically consistent answers and greatly improve their performance, we find that there is still a certain gap between this and the ideal situation (i.e., incorporating the most relevant logical constraints in Section 3). In view of the LLMs’ potential to understand logical constraints and make more rigorous reasoning, we believe that further exploration of how to make better use of logical constraints will help us understand the reasoning ability of LLMs, and we will take this as our future work.

Acknowledgments

We thank all the anonymous reviewers for their valuable feedback throughout the review process. This work is also supported by Ucap Cloud.

References

- Mohammed Aldawsari and Mark Finlayson. 2019. *Detecting subevents using discourse and narrative features*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4780–4790, Florence, Italy. Association for Computational Linguistics.
- James F Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.
- Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. *A multi-task, multilingual, multimodal evaluation of chatgpt*

- on reasoning, hallucination, and interactivity. *ArXiv preprint*, abs/2302.04023.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- 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*, pages 77–86, Vancouver, Canada. 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, Yixin Cao, Yan Zhang, and Zhiwei Liu. 2023. [Cheer: Centrality-aware high-order event reasoning network for document-level event causality identification](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 10804–10816.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. [Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality](#).
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. [Palm: Scaling language modeling with pathways](#). *ArXiv preprint*, abs/2204.02311.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. [Scaling instruction-finetuned language models](#). *ArXiv preprint*, abs/2210.11416.
- William F Clocksin and Christopher S Mellish. 2003. *Programming in PROLOG*. Springer Science & Business Media.
- Bruce Frederiksen. 2008. Applying expert system technology to code reuse with pyke. *PyCon: Chicago*.
- Jun Gao, Huan Zhao, Changlong Yu, and Ruifeng Xu. 2023. [Exploring the feasibility of chatgpt for event extraction](#).
- Alfonso Gerevini and Lenhart Schubert. 1995. Efficient algorithms for qualitative reasoning about time. *Artificial intelligence*, 74(2):207–248.
- Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2022. [Roscoe: A suite of metrics for scoring step-by-step reasoning](#).
- Rujun Han, I-Hung 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, Hong Kong, China. Association for Computational Linguistics.
- Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, et al. 2022. [Folio: Natural language reasoning with first-order logic](#). *ArXiv preprint*, abs/2209.00840.
- 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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. [Large language models can self-improve](#).
- Myeongjun Jang and Thomas Lukasiewicz. 2023. [Consistency analysis of chatgpt](#).
- Fangkai Jiao, Zhiyang Teng, Shafiq Joty, Bosheng Ding, Aixin Sun, Zhengyuan Liu, and Nancy F Chen. 2023. [Logicllm: Exploring self-supervised logic-enhanced training for large language models](#). *ArXiv preprint*, abs/2305.13718.
- Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. [Maieutic prompting: Logically consistent reasoning with recursive explanations](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1266–1279, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). *ArXiv preprint*, abs/2205.11916.
- Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023. [Evaluating the logical reasoning ability of chatgpt and gpt-4](#). *ArXiv preprint*, abs/2304.03439.
- 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](#). *ArXiv preprint*, abs/1907.11692.
- John W Lloyd. 2012. *Foundations of logic programming*. Springer Science & Business Media.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Jing Lu and Vincent Ng. 2021. [Conundrums in event coreference resolution: Making sense of the state of the art](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1368–1380, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yaojie Lu, Hongyu Lin, Jialong Tang, Xianpei Han, and Le Sun. 2022. End-to-end neural event coreference resolution. *Artificial Intelligence*, 303:103632.
- Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. [Faithful chain-of-thought reasoning](#). *ArXiv preprint*, abs/2301.13379.
- Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. 2023. [Large language model is not a good few-shot information extractor, but a good reranker for hard samples!](#)
- Paramita Mirza, Rachele Sprugnoli, Sara Tonelli, and Manuela Speranza. 2014. [Annotating causality in the TempEval-3 corpus](#). In *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)*, pages 10–19, Gothenburg, Sweden. Association for Computational Linguistics.
- Paramita Mirza and Sara Tonelli. 2014. [An analysis of causality between events and its relation to temporal information](#). In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2097–2106, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.
- Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018. [Joint reasoning for temporal and causal relations](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2278–2288, Melbourne, Australia. Association for Computational Linguistics.
- Tim O’Gorman, Kristin Wright-Bettner, and Martha Palmer. 2016. [Richer event description: Integrating event coreference with temporal, causal and bridging annotation](#). In *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*, pages 47–56, Austin, Texas. Association for Computational Linguistics.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. [Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning](#). *ArXiv preprint*, abs/2305.12295.
- Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo M Ponti, and Shay B Cohen. 2023. [Are large language models temporally grounded?](#) *ArXiv preprint*, abs/2311.08398.
- Beth M. Sundheim. 1991. [Evaluating text understanding systems](#). In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*.
- Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. [ProofWriter: Generating implications, proofs, and abductive statements over natural language](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online. Association for Computational Linguistics.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *ArXiv preprint*, abs/2307.09288.
- 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)*, pages 696–706, Online. Association for Computational Linguistics.
- Haoyu Wang, Hongming Zhang, Muhao Chen, and Dan Roth. 2021. [Learning constraints and descriptive segmentation for subevent detection](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5216–5226, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

- 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. 2022a. [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.
- Xingyao Wang, Sha Li, and Heng Ji. 2022b. [Code4struct: Code generation for few-shot structured prediction from natural language](#).
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022c. [Self-consistency improves chain of thought reasoning in language models](#). *ArXiv preprint*, abs/2203.11171.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022a. [Chain of thought prompting elicits reasoning in large language models](#). *ArXiv preprint*, abs/2201.11903.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. [Chain-of-thought prompting elicits reasoning in large language models](#). In *NeurIPS*.
- Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2023. [Are large language models really good logical reasoners? a comprehensive evaluation from deductive, inductive and abductive views](#). *ArXiv preprint*, abs/2306.09841.
- Chenhan Yuan, Qianqian Xie, Jimin Huang, and Sophia Ananiadou. 2024. [Back to the future: Towards explainable temporal reasoning with large language models](#). In *Proceedings of the ACM on Web Conference 2024*, pages 1963–1974.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. [STar: Bootstrapping reasoning with reasoning](#). In *Advances in Neural Information Processing Systems*.
- Hongming Zhang, Daniel Khashabi, Yangqiu Song, and Dan Roth. 2020. [Transomcs: From linguistic graphs to commonsense knowledge](#). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4004–4010. ijcai.org.
- Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. [Automatic chain of thought prompting in large language models](#). *ArXiv preprint*, abs/2210.03493.
- Yichao Zhou, Yu Yan, Rujun Han, J. Harry Caufield, Kai-Wei Chang, Yizhou Sun, Peipei Ping, and Wei Wang. 2021. [Clinical temporal relation extraction with probabilistic soft logic regularization and global inference](#). In *AAAI 2021, IAAI 2021, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14647–14655. AAAI Press.

A Understanding Event Relations

There are four kinds of widely-used event relations: coreference, temporal, causal, and subevent relations (O’Gorman et al., 2016; Wang et al., 2022a).

1. *Coreference relations* between events occur when multiple event mentions in a text refer to the same underlying event. We call these event mentions *cluster*.
2. *Temporal relations* refer to the temporal ordering of events based on their occurrence in time. In this paper, we consider seven different types of temporal relations:

- **NO_TEMPORAL**: if there is no clear temporal relation between event *A* and *B*.
- **BEFORE**: if event *A* happened completely before event *B*.
- **OVERLAP**: if event *A* has an overlap with event *B*.
- **CONTAINS**: if event *A*’s time contains event *B*’s time.
- **SIMULTANEOUS**: if events *A* and *B* happen at the same time.
- **ENDS-ON**: if event *A* ends when event *B* starts.
- **BEGINS-ON**: if event *A* and event *B* start at the same time, but end at different times.

In Figure 6, we list all the types of temporal relations and illustrate their distinctions on a unified timeline. Note that in our study, we adhere to a unidirectional perspective where the start time of event *A* precedes that of event *B*. Consequently, our framework does not encompass symmetrical relationships, such as the inverse of “AFTER” being “BEFORE”. To illustrate, if event *A* is considered “AFTER” event *B*, this would correspond to event *B* being “BEFORE” event *A* in our defined context.

3. *Causal relations* refer to that one event (the cause) brings about or influences the occurrence of another event (the effect). They can be classified into two different types: *CAUSE* relation where the tail event is inevitable given the head event, and *PRECONDITION* where the tail event would not have happened if the head event had not happened.
4. *Subevent relations* refer to that one event (the subevent) is a component or a smaller part of

Diagram	Relation(A, B)
	BEFORE
	OVERLAP
	CONTAINS
	SIMULTANEOUS
	ENDS-ON
	BEGINS-ON

Figure 6: Interpretations of the temporal relation between two events *A* and *B*. Brackets represent time intervals along the time axis.

another event (the main event). Identifying and understanding subevent relations helps to reveal the underlying hierarchy and organizational structure of events in a given text.

Event Relation Extraction Event Relation Extraction (ERE) includes identifying coreference, temporal, causal, and subevent relations between every two events in the text. We formulate ERE as a multi-label classification problem, determining one label (relation) for each of these four relation types. For coreference relations, the labels $\in \{\text{NO_COREFERENCE}, \text{COREFERENCE}\}$; for temporal relations, the labels $\in \{\text{NO_TEMPORAL}, \text{BEFORE}, \text{OVERLAP}, \text{CONTAINS}, \text{SIMULTANEOUS}, \text{ENDS-ON}, \text{BEGINS-ON}\}$; for causal relations, the labels $\in \{\text{NO_CAUSAL}, \text{PRECONDITION}, \text{CAUSE}\}$; for subevent relations, the labels $\in \{\text{NO_SUBEVENT}, \text{SUBEVENT}\}$.

B Logical Constraints Between Two Events

In Table 2, we provide a comprehensive set of logical constraints for the relations between two events to assess their logical consistency. We also manually design description text for each constraint to let LLMs follow the prompt. As shown in Table 5, $\text{COREFERENCE}(A, B) \rightarrow \neg \text{TEMPORAL}(A, B)$, $\neg \text{CAUSAL}(A, B)$, $\neg \text{SUBEVENT}(A, B)$ indicates that "if event *A* and event *B* have a coreference relation, they will not have temporal, causal, and subevent relations".

B.1 An Example of Detecting Conflicts and Retrieving Relevant Constraints

As described above, for the ERE task, we meticulously collect 11 logical constraints covering all

If Relation(A, B)	Then Relation (A, B)	Then Relation (B, A)
COREFERENCE	\neg TEMPORAL, \neg CAUSAL, \neg SUBEVENT	COREFERENCE
\neg TEMPORAL	\neg CAUSAL, \neg SUBEVENT	/
BEFORE	\neg COREFERENCE, \neg SUBEVENT	\neg TEMPORAL
OVERLAP	\neg COREFERENCE, \neg SUBEVENT	\neg TEMPORAL
CONTAINS	\neg COREFERENCE, \neg CAUSAL	\neg TEMPORAL
SIMULTANEOUS	\neg COREFERENCE, \neg CAUSAL, \neg SUBEVENT	SIMULTANEOUS
ENDS-ON	\neg COREFERENCE, \neg CAUSAL, \neg SUBEVENT	\neg TEMPORAL
BEGINS-ON	\neg COREFERENCE, \neg CAUSAL, \neg SUBEVENT	BEGINS-ON
CAUSE	\neg COREFERENCE, BEFORE \vee OVERLAP, \neg SUBEVENT	\neg TEMPORAL
PRECONDITION	\neg COREFERENCE, BEFORE \vee OVERLAP, \neg SUBEVENT	\neg TEMPORAL
SUBEVENT	\neg COREFERENCE, CONTAINS \neg CAUSAL	\neg TEMPORAL

Table 2: Logical Constraints of relations between two events, where \neg denotes "NOT", \vee denotes "OR".

relations between two events. These constraints serve as our benchmark to identify inconsistencies in the predictions made by LLMs.

Let us consider an illustrative example. If LLM produces an answer such as "NO_COREFERENCE, SIMULTANEOUS, CAUSE, NO_SUBEVENT" (refer to Figure 1 and Figure 4), we could detect the inconsistency between "SIMULTANEOUS" and "CAUSE", as shown in Table 2:

- A "SIMULTANEOUS" relation implies a "NO_CAUSAL" (\neg CAUSAL) relation.
- Conversely, a "CAUSE" relation suggests the presence of either a "BEFORE" or an "OVERLAP" relation.

Given this, "SIMULTANEOUS" and "CAUSE" are inherently contradictory, and they cannot coexist in a consistent prediction. To rectify this, we retrieve the associated textual description from Table 5. Specifically, the statements "If event A CAUSES event B, then event A happens BEFORE or OVERLAP event B ..." and "If event A and event B happen SIMULTANEOUSly, then they won't have coreference, causal, and subevent relations ..." are integrated into the LLM's instruction.

B.2 An Example of Post-processing

As shown in Figure 4, if LLMs predict the relations between two events as "NO_COREFERENCE, SIMULTANEOUS, CAUSE, NO_SUBEVENT", we can detect that "SIMULTANEOUS" and "CAUSE" conflict according to the logical constraints. In order to eliminate conflicts, one relation can be fixed first, and then the other relation can be randomly decided by the candidates that do not conflict with the current relation. For example, when the fixed temporal relation is "SIMULTANEOUS",

the causal relations can only be "NO_CAUSAL", while when the fixed causal relation is "CAUSE", the temporal relation can be either "BEFORE" or "OVERLAP". We also add a negative option "NO_COREFERENCE, NO_TEMPORAL, NO_CAUSAL, NO_SUBEVENT" to the candidate set because it is possible that neither relation exists. Finally, we randomly select one option from:

- NO_COREFERENCE, SIMULTANEOUS, NO_CAUSAL, NO_SUBEVENT
- NO_COREFERENCE, OVERLAP, CAUSE, NO_SUBEVENT
- NO_COREFERENCE, BEFORE, CAUSE, NO_SUBEVENT
- NO_COREFERENCE, NO_TEMPORAL, NO_CAUSAL, NO_SUBEVENT

as the ultimate answer, thus ensuring that the results must be logically consistent (i.e., $LI = 0$).

C Transitivity Rules Among Events

We provide a comprehensive set of 39 logical constraints for the transitivity rules among three events in Table 6. We also manually design prompts for each constraint, as shown in Table 7.

C.1 Pseudo Code of Logic Programming

Once obtaining 11 constraints between two events and 39 constraints among three events, we apply logic programming to automatically reason new event relations by inputting the known constraints and relations. The pseudo-code mentioned in the main text is shown in Algorithm 1.

D Statistics of the Fine-tuning Dataset

As shown in Table 3, we provide the statistics of the fine-tuning dataset originating from LLM-ERL.

Hop	# Count
2	39
3	179
4	945
5	5613

Table 3: Statistics of the fine-tuning dataset.

E Dataset Construction

MAVEN-ERE contains 4,480 documents, 103,193 events coreference chains, 1,216,217 temporal relations, 57,992 causal relations, and 15,841 subevent relations, which is larger than existing datasets of all the ERE tasks by at least an order of magnitude (Wang et al., 2022a). MAVEN-ERE has released the train and valid set, but does not release the ground-truth test set, so we randomly split its train set into train/valid sets with a ratio of 8:2, and then use its original valid set as the new test set.

Causal-TimeBank contains 184 documents, 6,813 events, and 7,608 event pairs (Mirza and Tonelli, 2014). Among them, 318 and 6,115 event pairs are annotated with causal and temporal relations, respectively. Due to Causal-TimeBank does not split train/valid/test sets, we randomly split it to train/valid/test sets with a ratio of 6:1:3. We do not evaluate coreference and subevent relations in Causal-TimeBank since there are no annotations for these two relation types.

For ERE tasks, We conduct sampling at the sentence level. The samples of the two events that do not have any relations will be excluded. Note that Causal-TimeBank inherently contains fewer event relations compared to MAVEN-ERE. After processing and dividing the data split, its test set comprises only 139 samples. Therefore, we randomly sample 500 examples from the test set of MAVEN-ERE and 100 examples from the test set of Causal-TimeBank as our testbed.

ProofWriter is a commonly used dataset for deductive reasoning (Tafjord et al., 2021). We use the OWA subset of it, which is divided into five parts, each part requiring 0, 1, 2, 3, and 5 hops of

reasoning, respectively. We evaluate the hardest 5-hop subset. To reduce the computation cost, we randomly sample 200 examples in the test set and ensure a balanced label distribution.

FOLIO is a challenging expert-written dataset for logical reasoning (Tafjord et al., 2021), whose questions require complex first-order logic reasoning to solve. We use its entire test set for evaluation, consisting of 204 examples.

Algorithm 1 An Example of 3-hop Reasoning

Initialize the knowledge base with facts and rules

Knowledge Base:

Fact: BEFORE(A, B)

Fact: SIMULTANEOUS(B, C)

Fact: OVERLAP(C, D)

Rule: BEFORE \leftarrow BEFORE \wedge SIMULTANEOUS

Rule: OVERLAP \leftarrow SIMULTANEOUS \wedge OVERLAP

Rule: BEFORE \leftarrow BEFORE \wedge OVERLAP

Initialize the logic engine with the query

Query: BEFORE(A, D)?

while obtain new facts **do**

for each rule r of the Knowledge Base **do**

if r 's premise is satisfied by the current known facts **then**

Add r 's conclusion to the knowledge base

end if

end for

end while

Query result: BEFORE(A, D) is satisfied with BEFORE(A, C) and OVERLAP(B, D)

F Training Details of RoBERTa-large On Two Tasks

Our experiments include two settings. (1) fully fine-tuned: we fine-tune smaller language models (SLMs) with complete and abundant samples. This setting is for reference to see the performance limit of SLMs. (2) one-shot: we sample only one example for each label and construct a tiny training set. This setting is for direct comparison with our experiments on LLMs (similar training/demonstration sample number).

Model (%)		ProofWriter	FOLIO
Vicuna	vanilla ICL	37 / 38	40 / 43
	vanilla CoT	40 / 42	38 / 40
	CoT w. logic.	42 / 44	42 / 45
Llama2	vanilla ICL	29 / 33	42 / 45
	vanilla CoT	31 / 37	44 / 46
	CoT w. logic.	40 / 42	46 / 48

Table 4: Vicuna and Llama2’s performance on ProofWriter and FOLIO before and after fine-tuning on LLM-ERL (split by “/”).

We implement vanilla fine-tuning approaches on two datasets and use RoBERTa-Large as backbones. We run each experiment on a single NVIDIA V100 GPU. We adopt the AdamW (Loshchilov and Hutter, 2019) optimizer with a linear scheduler and 0.1 warm-up steps. We set the weight-decay coefficient as $1e-5$ and maximum gradient norms as 1.0. We set the batch size as 16 with 20 or 50 epochs. We set the maximum input length as 256 and the learning rate as $2e-5$.

G Implementation Details of Finetuning-based Approach

We set the rank of LoRA modules to be 64. Our model is optimized with a learning rate of $2e-4$ and a linear warm-up for the first 3% steps. We clip the gradients of model parameters to a max norm of 0.3. All the LoRA parameters are fine-tuned on an NVIDIA A100 GPU with 80GB memory.

H Generalization to Logical Reasoning

In this section, we verify whether LLMs enhanced by LLM-ERL can be generalized to other tasks that need logical reasoning. We translate the symbolic representations of event relations into a form of deductive reasoning (i.e., containing facts, rules, and queries) to maintain consistency in task settings. The prompt example can be found in Appendix J.4.

Dataset Construction We conduct experiments on two datasets: ProofWriter (Tafjord et al., 2021) and FOLIO (Han et al., 2022). Details of the datasets can be found in Appendix E.

Results As shown in Table 4, we are surprised to find that models fine-tuned on LLM-ERL (e.g., Llama2-FT) can also bring performance improvement on other logical reasoning datasets, even though LLM-ERL focuses on event relation logic. This shows that the logical reasoning ability acquired by LLMs in the fine-tuning process can be

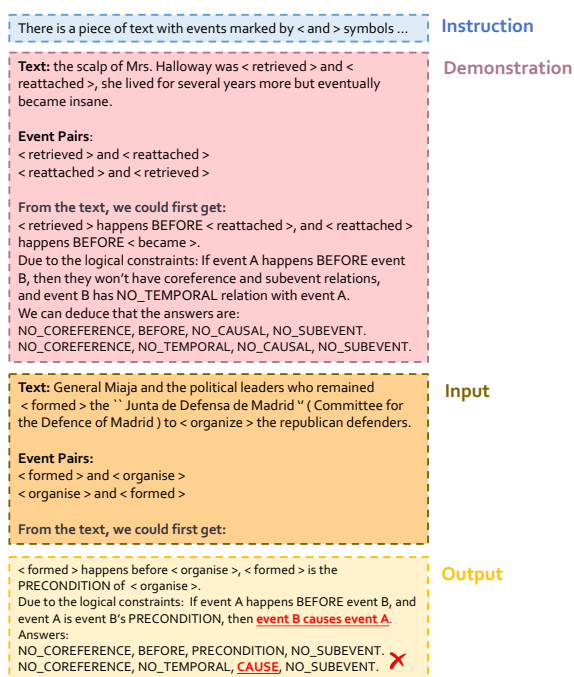


Figure 7: A case study that ChatGPT generates inaccurate logical constraints.

generalized to other domains. We intend to explore this intriguing aspect in future work.

I Case Study

I.1 Case Study on Self-generated Logical Constraints

In the main context, we have found that directly using CoT to infer logic does not help much for ERE tasks. One possible reason is that the inherent issues may lead to the failure of LLM in the precise rationale generation. To further illustrate an intuitive impression, we conduct a case study on MAVEN-ERE and find that the logical constraints generated by LLMs themselves are often inaccurate in content. As shown in Figure 7, ChatGPT could follow the logical constraint provided in the demonstration to a certain extent. However, it wrongly applies this to other relations — knowing that event A is event B ’s precondition, it is wrong to think that event B will cause event A . Actually, according to the logical constraints in Table 2, the relations between (B , A) should be “NO_COREFERENCE, NO_TEMPORAL, NO_CAUSAL, NO_SUBEVENT”.

I.2 Case Study on Llama2 and Llama2-FT

In Figure 8, We conduct a case study of Llama2-13B’s answers to the same input before and af-

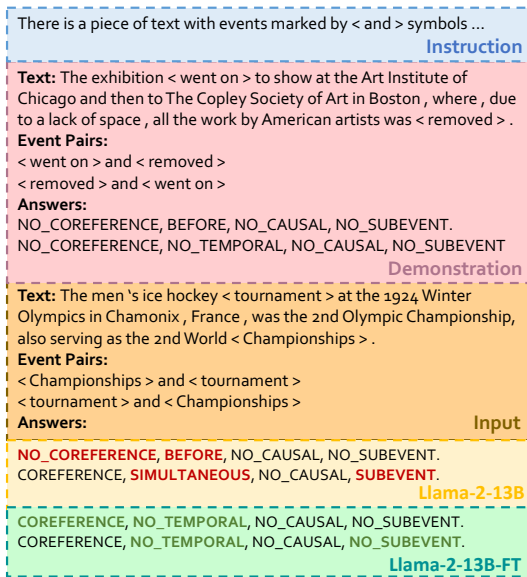


Figure 8: Case study on Llama-2-13B before and after fine-tuning (FT).

ter fine-tuning. From Figure 8 we can see that Llama2-FT could output the correct answers after fine-tuning on LLM-ERL, which validates the effectiveness of our fine-tuning approach.

If Relation(A, B)	Prompt Text
COREFERENCE	If event <i>A</i> and event <i>B</i> are COREFERENCE, then they won't have temporal, causal, and subevent relations, and COREFERENCE relation is bidirectional.
NO_TEMPORAL	If event <i>A</i> and event <i>B</i> do not have a temporal relation, then they won't have causal and subevent relations.
BEFORE	If event <i>A</i> happens BEFORE event <i>B</i> , then they won't have coreference and subevent relations, and event <i>B</i> has NO_TEMPORAL relation with event <i>A</i> .
OVERLAP	If event <i>A</i> happens OVERLAP with event <i>B</i> , then they won't have coreference and subevent relations, and event <i>B</i> has NO_TEMPORAL relation with event <i>A</i> .
CONTAINS	If event <i>A</i> 's time CONTAINS event <i>B</i> 's time, then they won't have coreference and causal relations, and event <i>B</i> has NO_TEMPORAL relation with event <i>A</i> .
SIMULTANEOUS	If event <i>A</i> and event <i>B</i> happen SIMULTANEOUSly, then they won't have coreference, causal, and subevent relations, and SIMULTANEOUS relation is bidirectional.
ENDS-ON	If event <i>A</i> ENDS-ON event <i>B</i> , then they won't have coreference, causal and subevent relations, and event <i>B</i> has NO_TEMPORAL relation with event <i>A</i> .
BEGINS-ON	If event <i>A</i> BEGINS-ON event <i>B</i> , then they won't have coreference, causal and subevent relations and BEGINS-ON relation is bidirectional.
CAUSE	If event <i>A</i> CAUSEs event <i>B</i> , then event <i>A</i> happens BEFORE or OVERLAP event <i>B</i> , and they won't have coreference and subevent relations, and event <i>B</i> has NO_TEMPORAL relation with event <i>A</i> .
PRECONDITION	If event <i>A</i> is event <i>B</i> 's PRECONDITION, then event <i>A</i> happens BEFORE or OVERLAP event <i>B</i> , and they won't have coreference and subevent relations, and event <i>B</i> has NO_TEMPORAL relation with event <i>A</i> .
SUBEVENT	If event <i>B</i> is a SUBEVENT of event <i>A</i> , then they won't have coreference and causal relations, and event <i>A</i> 's time should CONTAINS event <i>B</i> 's time, and event <i>B</i> has NO_TEMPORAL relation with event <i>A</i> .

Table 5: Prompt text of relations between two events.

If Relation(A, B) \wedge Relation(B, C)	Then Relation (A, C)
COREFERENCE \wedge COREFERENCE	COREFERENCE, \neg TEMPORAL, \neg CAUSAL, \neg SUBEVENT
COREFERENCE \wedge BEFORE	BEFORE, \neg COREFERENCE, \neg SUBEVENT
COREFERENCE \wedge OVERLAP	OVERLAP, \neg COREFERENCE, \neg SUBEVENT
COREFERENCE \wedge CONTAINS	CONTAINS, \neg COREFERENCE, \neg CAUSAL
COREFERENCE \wedge SIMULTANEOUS	SIMULTANEOUS, \neg COREFERENCE, \neg CAUSAL, \neg SUBEVENT
COREFERENCE \wedge ENDS-ON	ENDS-ON, \neg COREFERENCE, \neg CAUSAL, \neg SUBEVENT
COREFERENCE \wedge BEGINS-ON	BEGINS-ON, \neg COREFERENCE, \neg CAUSAL, \neg SUBEVENT
COREFERENCE \wedge CAUSE	CAUSE, \neg COREFERENCE, BEFORE \vee OVERLAP, \neg SUBEVENT
COREFERENCE \wedge PRECONDITION	PRECONDITION, \neg COREFERENCE, BEFORE \vee OVERLAP, \neg SUBEVENT
COREFERENCE \wedge SUBEVENT	SUBEVENT, \neg COREFERENCE, CONTAINS \neg CAUSAL
BEFORE \wedge BEFORE	BEFORE, \neg COREFERENCE, \neg SUBEVENT
BEFORE \wedge OVERLAP	BEFORE, \neg COREFERENCE, \neg SUBEVENT
BEFORE \wedge CONTAINS	BEFORE, \neg COREFERENCE, \neg SUBEVENT
BEFORE \wedge SIMULTANEOUS	BEFORE, \neg COREFERENCE, \neg SUBEVENT
BEFORE \wedge ENDS-ON	BEFORE, \neg COREFERENCE, \neg SUBEVENT
BEFORE \wedge BEGINS-ON	BEFORE, \neg COREFERENCE, \neg SUBEVENT
OVERLAP \wedge BEFORE	BEFORE, \neg COREFERENCE, \neg SUBEVENT
OVERLAP \wedge SIMULTANEOUS	OVERLAP, \neg COREFERENCE, \neg SUBEVENT
CONTAINS \wedge CONTAINS	CONTAINS, \neg COREFERENCE, \neg CAUSAL
CONTAINS \wedge SIMULTANEOUS	CONTAINS, \neg COREFERENCE, \neg CAUSAL
SIMULTANEOUS \wedge BEFORE	BEFORE, \neg COREFERENCE, \neg SUBEVENT
SIMULTANEOUS \wedge OVERLAP	OVERLAP, \neg COREFERENCE, \neg SUBEVENT
SIMULTANEOUS \wedge CONTAINS	CONTAINS, \neg COREFERENCE, \neg CAUSAL
SIMULTANEOUS \wedge SIMULTANEOUS	SIMULTANEOUS, \neg COREFERENCE, \neg CAUSAL, \neg SUBEVENT
SIMULTANEOUS \wedge ENDS-ON	ENDS-ON, \neg COREFERENCE, \neg SUBEVENT
SIMULTANEOUS \wedge BEGINS-ON	BEGINS-ON, \neg COREFERENCE, \neg SUBEVENT
SIMULTANEOUS \wedge COREFERENCE	SIMULTANEOUS, \neg COREFERENCE, \neg CAUSAL, \neg SUBEVENT
ENDS-ON \wedge CONTAINS	BEFORE, \neg COREFERENCE, \neg SUBEVENT
ENDS-ON \wedge BEGINS-ON	ENDS-ON, \neg COREFERENCE, \neg CAUSAL, \neg SUBEVENT
ENDS-ON \wedge SIMULTANEOUS	ENDS-ON, \neg COREFERENCE, \neg CAUSAL, \neg SUBEVENT
BEGINS-ON \wedge SIMULTANEOUS	BEGINS-ON, \neg COREFERENCE, \neg CAUSAL, \neg SUBEVENT
BEGINS-ON \wedge BEGINS-ON	BEGINS-ON, \neg COREFERENCE, \neg CAUSAL, \neg SUBEVENT
BEGINS-ON \wedge COREFERENCE	BEGINS-ON, \neg COREFERENCE, \neg CAUSAL, \neg SUBEVENT
CAUSE \wedge CAUSE	CAUSE, \neg COREFERENCE, BEFORE \vee OVERLAP, \neg SUBEVENT
CAUSE \wedge SUBEVENT	CAUSE, \neg COREFERENCE, BEFORE \vee OVERLAP, \neg SUBEVENT
PRECONDITION \wedge CAUSE	CAUSE, \neg COREFERENCE, BEFORE \vee OVERLAP, \neg SUBEVENT
PRECONDITION \wedge PRECONDITION	PRECONDITION, \neg COREFERENCE, BEFORE \vee OVERLAP, \neg SUBEVENT
PRECONDITION \wedge SUBEVENT	PRECONDITION, \neg COREFERENCE, BEFORE \vee OVERLAP, \neg SUBEVENT
SUBEVENT \wedge SUBEVENT	SUBEVENT, \neg COREFERENCE, CONTAINS \neg CAUSAL

Table 6: Logical Constraints for the transitivity rules among three events, where \wedge denotes "AND", \neg denotes "NOT", \vee denotes "OR".

If Relation(A, B) \wedge Relation(B, C)	Prompt Text
COREFERENCE \wedge COREFERENCE COREFERENCE \wedge BEFORE COREFERENCE \wedge OVERLAP COREFERENCE \wedge CONTAINS COREFERENCE \wedge SIMULTANEOUS COREFERENCE \wedge ENDS-ON COREFERENCE \wedge BEGINS-ON COREFERENCE \wedge CAUSE COREFERENCE \wedge PRECONDITION COREFERENCE \wedge SUBEVENT	If event A and event B are COREFERENCE, then the relations between event B and event C should be the same as that between event A and event C .
BEFORE \wedge BEFORE BEFORE \wedge OVERLAP BEFORE \wedge CONTAINS BEFORE \wedge SIMULTANEOUS BEFORE \wedge ENDS-ON BEFORE \wedge BEGINS-ON	If event A happens BEFORE event B , and Relation(B, C), then event A happens BEFORE event C .
OVERLAP \wedge BEFORE	If event A happens OVERLAP with event B , and event B happens BEFORE event C , then event A happens BEFORE event C .
OVERLAP \wedge SIMULTANEOUS	If event A happens OVERLAP with event B , and event B and event C happen SIMULTANEOUSly, then event A happens BEFORE event C .
CONTAINS \wedge CONTAINS	If event A 's time CONTAINS event B 's time, and event B 's time CONTAINS event C 's time, then event A 's time CONTAINS event C 's time.
CONTAINS \wedge SIMULTANEOUS	If event A 's time CONTAINS event B 's time, and event B and event C happen SIMULTANEOUSly, then event A 's time CONTAINS event C 's time.
SIMULTANEOUS \wedge BEFORE SIMULTANEOUS \wedge OVERLAP SIMULTANEOUS \wedge CONTAINS SIMULTANEOUS \wedge SIMULTANEOUS SIMULTANEOUS \wedge ENDS-ON SIMULTANEOUS \wedge BEGINS-ON	If events A and B happen SIMULTANEOUSly, and Relation(B, C), then event A 's time CONTAINS event C 's time.
ENDS-ON \wedge CONTAINS	If event A ENDS-ON event B , and event B 's time CONTAINS event C 's time, then event A happens BEFORE event C .
ENDS-ON \wedge BEGINS-ON ENDS-ON \wedge SIMULTANEOUS	If event A ENDS-ON event B , and Relation(B, C), then event A ENDS-ON event C .
BEGINS-ON \wedge SIMULTANEOUS BEGINS-ON \wedge BEGINS-ON	If event A BEGINS-ON event B , and Relation(B, C), then event A BEGINS-ON event C .
CAUSE \wedge CAUSE	If event A CAUSEs event B , and event B CAUSEs event C , then event A CAUSEs event C .
CAUSE \wedge PRECONDITION	If event A CAUSEs event B , and event B is event C 's PRECONDITION, then event A is event C 's PRECONDITION.
CAUSE \wedge SUBEVENT	If event A CAUSEs event B , and event C is a SUBEVENT of event B , then event A CAUSEs event C .
PRECONDITION \wedge PRECONDITION	If event A is event B 's PRECONDITION, and event B is event C 's PRECONDITION, then event A is event C 's PRECONDITION.
PRECONDITION \wedge SUBEVENT	If event A is event B 's PRECONDITION, and event C is a SUBEVENT of event B , then event A is event C 's PRECONDITION.
SUBEVENT \wedge SUBEVENT	If event B is a SUBEVENT of event A , and event C is a SUBEVENT of event B , then event C is a SUBEVENT of event A .

Table 7: Prompt text of relations among three events.

J Prompt Examples

In this section, we provide examples of prompts used for each task and approach.

J.1 Pilot Case Study

In the context of our paper, “relevant logical constraints” refer to the necessary knowledge or requirements for processing the current sample. They are accurately defined and closely related to the case in question. On the other hand, “irrelevant logical constraints” denote logic that, while possibly correct in content, does not directly pertain to the specific sample at hand. This distinction is crucial to maintain the focus and relevance of our analysis.

Process of Determining Relevant Logic

- For MAVEN-ERE: we have presented the critical importance of ensuring the logical consistency of answers generated by LLMs. Therefore, we implement a rigorous manual check of the LLM outputs. During this process, we specifically identify and rectify any logical inconsistencies. We guide LLM by incorporating the most relevant logical constraints from Table 5 into the LLM’s instruction, thereby facilitating the refinement and accuracy of its responses.
- For ProofWriter: we have observed that the context often contains some facts and rules that are not directly pertinent to the current question. Therefore, we start by analyzing the question at hand and the initial answers provided by the LLM. Based on this, we selectively introduce rules and facts that are specifically relevant to the current scenario. This method allows us to provide the LLM with focused guidance, enabling it to refine its answers more effectively and accurately.

Process of Determining Irrelevant Logic

- For MAVEN-ERE: We randomly sample 1-2 constraints from the entire set removing those relevant logical constraints and construct the prompts based on each sample.
- For ProofWriter: We artificially select irrelevant logical constraints from each sample’s content, thereby introducing a form of “noise” or “distraction” to the LLM’s judgment process.

Prompt Examples

- MAVEN-ERE w. relevant logic constraints (Figure 9);
- MAVEN-ERE w. irrelevant logic constraints (Figure 10);
- ProofWriter w. relevant logic constraints (Figure 11);
- ProofWriter w. irrelevant logic constraints (Figure 12).

MAVEN-ERE w. relevant logic constraints (💡)



Task Description:

There is a piece of text with two events marked by < and > symbols, and your task is to identify four types of relations between them (... more context here ...)

Text:

The 2013 Lahad Datu < standoff > (also known as the Lahad Datu incursion) was a military < conflict > that started on 11 February 2013 and fully ended on 24 March 2013 .

Event Pairs:

< standoff > and < conflict >
< conflict > and < standoff >

Answers:

Event Pair 1: < standoff > and < conflict >

(1) Coreference relation : COREFERENCE (2) Temporal relation: **CONTAINS** ✗
(3) Causal relation: NO_CAUSAL (4) Subevent relation: NO_SUBEVENT

Event Pair 2: < conflict > and < standoff >

(1) Coreference relation: **NO_COREFERENCE** ✗ (2) Temporal relation: NO_TEMPORAL
(3) Causal relation: NO_CAUSAL (4) Subevent relation: NO_SUBEVENT



Let's think step by step. You have identified that < standoff > and < conflict > are coreferences. *The coreference should be bidirectional. And if two events are COREFERENCE, they won't have temporal relations.* So please modify your Coreference and Temporal answers. 💡

Apologies for the confusion. Based on the revised information, here are the corrected answers:

Event Pair 1: < standoff > and < conflict >

(1) Coreference relation : COREFERENCE (2) Temporal relation: **NO_TEMPORAL** ✓
(3) Causal relation: NO_CAUSAL (4) Subevent relation: NO_SUBEVENT

Event Pair 2: < conflict > and < standoff >

(1) Coreference relation: **COREFERENCE** ✓ (2) Temporal relation: NO_TEMPORAL
(3) Causal relation: NO_CAUSAL (4) Subevent relation: NO_SUBEVENT



Figure 9: MAVEN-ERE w. relevant logic constraints

MAVEN-ERE w. irrelevant logic constraints (💡)



Task Description:

There is a piece of text with two events marked by < and > symbols, and your task is to identify four types of relations between them (... more context here ...)

Text:

The 2013 Lahad Datu < standoff > (also known as the Lahad Datu incursion) was a military < conflict > that started on 11 February 2013 and fully ended on 24 March 2013 .

Event Pairs:

< standoff > and < conflict >
< conflict > and < standoff >

Answers:

Event Pair 1: < standoff > and < conflict >

(1) Coreference relation : COREFERENCE (2) Temporal relation: **CONTAINS** ✗
(3) Causal relation: NO_CAUSAL (4) Subevent relation: NO_SUBEVENT

Event Pair 2: < conflict > and < standoff >

(1) Coreference relation: **NO_COREFERENCE** ✗ (2) Temporal relation: NO_TEMPORAL
(3) Causal relation: NO_CAUSAL (4) Subevent relation: NO_SUBEVENT



ChatGPT



Let's think step by step. *If < standoff > happens BEFORE < conflict >, then they won't have coreference and subevent relations, and < conflict > will have NO_TEMPORAL relation with < standoff > .* (💡)

Yes, you are correct. Based on the information, here are the answers:

Event Pair 1: < standoff > and < conflict >

(1) Coreference relation : **NO_COREFERENCE** ✗ (2) Temporal relation: **BEFORE** ✗
(3) Causal relation: NO_CAUSAL (4) Subevent relation: NO_SUBEVENT

Event Pair 2: < conflict > and < standoff >

(1) Coreference relation: **NO_COREFERENCE** ✗ (2) Temporal relation: NO_TEMPORAL
(3) Causal relation: NO_CAUSAL (4) Subevent relation: NO_SUBEVENT



ChatGPT

Figure 10: MAVEN-ERE w. irrelevant logic constraints

ProofWriter w. relevant logic constraints (💡)



Task Description:

Given a problem statement as contexts, the task is to answer a logical reasoning question.

Text:

Bob is smart. Bob is young. Charlie is cold. Charlie is round. Charlie is white. Charlie is young. Gary is smart. Gary is white. Harry is rough. Harry is smart. Harry is white. All red people are round. Round, smart people are cold. If someone is cold and young then they are white. Red people are round. All round, young people are red. If someone is smart then they are rough. Rough, young people are red. If Bob is round then Bob is smart. If Charlie is white and Charlie is rough then Charlie is young.

Question:

Based on the above information, is the following statement true, false, or unknown? Harry is not round.

Based on the given information, the statement "Harry is not round" is **true**. ❌
The information states that all red people are round, and Harry is described as white, smart, and rough, **but not round**. Therefore, Harry does not fit the criteria of being round, and the statement is true.



Let's think step by step. From the text, we could first get relevant facts:

- Harry is smart.
- Harry is white.

And *the relevant rule*:

- *All red people are round.*
- *Round, smart people are cold.*
- *All round, young people are red.*



Given these statements, we can conclude the following:

Harry is smart and white, but there is no direct statement about his being red.

Apologies for the confusion. You are correct, there is no direct statement in the given information that explicitly mentions whether Harry is round or not. Therefore, the statement "Harry is not round" is **unknown** based solely on the provided information. ✓



Figure 11: ProofWriter w. relevant logic constraints

ProofWriter w. irrelevant logic constraints (🧠)



User

Task Description:

Given a problem statement as contexts, the task is to answer a logical reasoning question.

Text:

Bob is smart. Bob is young. Charlie is cold. Charlie is round. Charlie is white. Charlie is young. Gary is smart. Gary is white. Harry is rough. Harry is smart. Harry is white. All red people are round. Round, smart people are cold. If someone is cold and young then they are white. Red people are round. All round, young people are red. If someone is smart then they are rough. Rough, young people are red. If Bob is round then Bob is smart. If Charlie is white and Charlie is rough then Charlie is young.

Question:

Based on the above information, is the following statement true, false, or unknown? Harry is not round.

Based on the given information, the statement "Harry is not round" is **true**. ❌
The information states that all red people are round, and Harry is described as white, smart, and rough, **but not round**. Therefore, Harry does not fit the criteria of being round, and the statement is true.



ChatGPT



User

Let's think step by step. From the text, we could get some facts:

- Harry is smart.
- Harry is white.

And some rules:

- *If someone is cold and young then they are white.*
- *If someone is smart then they are rough.*
- *Rough, young people are red.*



Given these statements, please output your answers.

Apologies for the confusion. Based on the information, the statement "Harry is not round" is **False**. ❌



ChatGPT

Figure 12: ProofWriter w. irrelevant logic constraints

J.2 Incorporating Logical Constraints

The highlighted parts represent the content generated by LLMs. We omit the demonstration here for clarity.

Vanilla ICL

Task Description:

There is a piece of text with two events marked by < and > symbols, and your task is to identify four types of relations between them (... more context here ...)

Text:

Before her death , Todd posted a video on YouTube in which she used a series of flash cards to < tell > her experience of being blackmailed into exposing her breasts via webcam , and of being < bullied > and physically assaulted .

Event Pairs:

< tell > and < bullied >
< bullied > and < tell >

Answers:

NO_COREFERENCE, NO_TEMPORAL, NO_CAUSAL, NO_SUBEVENT.
NO_COREFERENCE, NO_TEMPORAL, PRECONDITION, NO_SUBEVENT.

Vanilla ICL plus CoT

Task Description:

There is a piece of text with two events marked by < and > symbols, and your task is to identify four types of relations between them (... more context here ...)

Text:

Before her death , Todd posted a video on YouTube in which she used a series of flash cards to < tell > her experience of being blackmailed into exposing her breasts via webcam , and of being < bullied > and physically assaulted .

Event Pairs:

< tell > and < bullied >
< bullied > and < tell >

Reasoning:

< bullied > happens before < tell >, and being < bullied > leads her to < tell >.

Answers:

NO_COREFERENCE, BEFORE, NO_CAUSAL, NO_SUBEVENT.
NO_COREFERENCE, BEFORE, PRECONDITION, NO_SUBEVENT.

CoT w. self-generated logical constraints

Task Description:

There is a piece of text with two events marked by < and > symbols, and your task is to identify four types of relations between them (... more context here ...)

Text:

Before her death , Todd posted a video on YouTube in which she used a series of flash cards to < tell > her experience of being blackmailed into exposing her breasts via webcam , and of being < bullied > and physically assaulted .

Event Pairs:

< tell > and < bullied >
< bullied > and < tell >

From the text, we could first get:

< bullied > happens before < tell >, and being < bullied > leads her to < tell >

Due to the logical constraint: If event A is event B's PRECONDITION, then event A happens BEFORE or OVERLAP event B, and they won't have coreference and subevent relations, and event B has NO_TEMPORAL relation with event A.

We can deduce that the answers are:

NO_COREFERENCE, NO_TEMPORAL, NO_CAUSAL, NO_SUBEVENT.
NO_COREFERENCE, BEFORE, PRECONDITION, NO_SUBEVENT.

w. all logical constraints

Task Description:

There is a piece of text with two events marked by < and > symbols, and your task is to identify four types of relations between them (... more context here ...)

Logical constraints:

There are some rules among the relations, you can select some of them to reason or check your answers:

(1) If two events are COREFERENCE, then they won't have temporal, causal, and subevent relations, and COREFERENCE relation is bidirectional.

(... more context here ...)

(11) If event B is a SUBEVENT of event A, then they won't have coreference and causal relations, and event A's time should CONTAINS event B's time, and event B has NO_TEMPORAL relation with event A.

Text:

Before her death , Todd posted a video on YouTube in which she used a series of flash cards to < tell > her experience of being blackmailed into exposing her breasts via webcam , and of being < bullied > and physically assaulted .

Event Pairs:

< tell > and < bullied >
< bullied > and < tell >

Answers:

NO_COREFERENCE, NO_TEMPORAL, NO_CAUSAL, NO_SUBEVENT.
NO_COREFERENCE, BEFORE, PRECONDITION, NO_SUBEVENT.

w. retrieved logical constraints

Task Description:

There is a piece of text with two events marked by < and > symbols, and your task is to identify four types of relations between them (... more context here ...)

Logical constraints:

There are some rules among the relations, you can select some of them to reason or check your answers:

(1) If event A is event B's PRECONDITION, then event A happens BEFORE or OVERLAP event B, and they won't have coreference and subevent relations, and event B has NO_TEMPORAL relation with event A.

(2) If event A and event B do not have a temporal relation, then they won't have causal and subevent relations.

Text:

Before her death , Todd posted a video on YouTube in which she used a series of flash cards to < tell > her experience of being blackmailed into exposing her breasts via webcam , and of being < bullied > and physically assaulted .

Event Pairs:

< tell > and < bullied >
< bullied > and < tell >

Answers:

NO_COREFERENCE, NO_TEMPORAL, NO_CAUSAL, NO_SUBEVENT.
NO_COREFERENCE, BEFORE, PRECONDITION, NO_SUBEVENT.

J.3 Iterative Retrievals

In this section, we present a prompt example used in Section 5.3. As shown in Figure 13, with iterative prompting, ChatGPT finally outputs the correct answers.

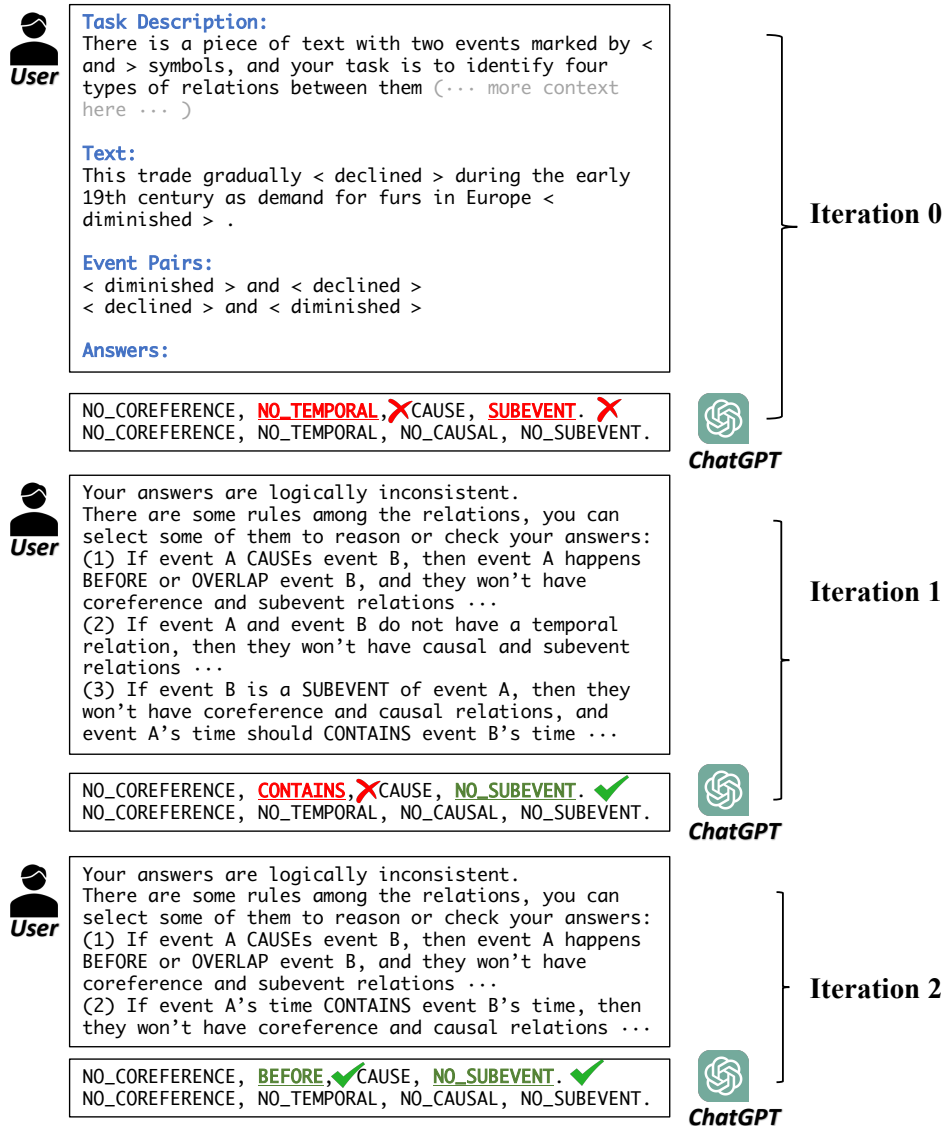


Figure 13: Multi-turn conversation with ChatGPT. We retrieve relevant logical constraints and provide them to ChatGPT.

J.4 Deductive Reasoning

The highlighted parts represent the content generated by LLMs. We omit the demonstration here for clarity.

Vanilla ICL

Task Description:

Given a problem statement as contexts, the task is to answer a logical reasoning question. For each question, choose from the following options: A) True, B) False, or C) Unknown.

Context:

The cow likes the lion. The cow visits the tiger.
(... more context here ...)

Question:

Based on the above information, is the following statement true, false, or unknown? The tiger is not young.

The correct option is:

C.

Vanilla ICL plus CoT

Task Description:

Given a problem statement as contexts, the task is to answer a logical reasoning question. For each question, choose from the following options: A) True, B) False, or C) Unknown.

Context:

The cow likes the lion. The cow visits the tiger.
(... more context here ...)

Question:

Based on the above information, is the following statement true, false, or unknown? The tiger is not young.

Reasoning:

The tiger likes the cow. The tiger likes the squirrel. If something likes the squirrel and it likes the cow, then it visits the tiger. So the tiger visits the tiger. If something visits the tiger then it is nice. So the tiger is nice. If something is nice and it sees the tiger then it is young. So the tiger is young.
The correct option is: B.

CoT w. self-generated logical constraints

Task Description:

Given a problem statement as contexts, the task is to answer a logical reasoning question. For each question, choose from the following options: A) True, B) False, or C) Unknown.

Context:

The cow likes the lion. The cow visits the tiger.
(... more context here ...)

Question:

Based on the above information, is the following statement true, false, or unknown? The tiger is not young.

From the text, we could first get facts:

The tiger likes the cow. The tiger likes the squirrel.

Due to the mentioned rules: if something likes the squirrel and it likes the cow, then it visits the tiger.

We can deduce the facts: the tiger visits the tiger.

Due to the mentioned rules: if something visits the tiger then it is nice.

We can deduce the facts: the tiger is nice.

Due to the mentioned rules: if something is nice and it sees the tiger then it is young.

We can deduce the facts: the tiger is young.

Therefore, the correct option is: B.