EvIT: Event-Oriented Instruction Tuning for Event Reasoning

Zhengwei Tao[1,2] **Xiancai Chen**[1,2] **Zhi Jin**[1,2*] **Xiaoying Bai**[3*] **Haiyan Zhao**[1,2] **Yiwei Lou**[2]
[1]Key Laboratory of High Confidence Software Technologies (PKU), MOE, China
[2]School of Computer Science, Peking University
[3]Advanced Institute of Big Data
{tttzw, xiancaich}@stu.pku.edu.cn, baixy@aibd.ac.cn
{zhijin, zhhy.sei}@pku.edu.cn

Abstract

Events refer to specific occurrences, incidents, or happenings that take place under a particular background. Event reasoning aims to infer events according to certain relations and predict future events. The cutting-edge techniques for event reasoning play a crucial role in various natural language processing applications. Large language models (LLMs) have made significant advancements in event reasoning owing to their wealth of knowledge and reasoning capabilities. However, smaller instruction-tuned models currently in use do not consistently demonstrate exceptional proficiency in managing these tasks. This discrepancy arises from the absence of explicit modeling of events and the interconnections of them within their instruction data. Consequently, these models face challenges in comprehending event structures and semantics while struggling to bridge the gap between their interpretations and human understanding of events. Additionally, their limitations in grasping event relations lead to constrained event reasoning abilities to effectively deduce and incorporate pertinent event knowledge. In this paper, we propose Event-Oriented Instruction Tuning to train our LLM named EvIT specializing in event reasoning tasks. Specifically, we first propose a novel structure named event quadruple which contains the structure and semantics of events and is complete in the event representation. We then design event-relation learning based on the structures. We encapsulate the learning into the instruction-tuning formulation to better stimulate the event reasoning capacity of our model. We design a heuristic unsupervised method to mine event quadruple from a large-scale corpus. At last, we finetune a Llama model on our Event-Oriented Instruction Tuning. We conduct extensive experiments on event reasoning tasks on several datasets. Automatic and human evaluations demonstrate EvIT achieves competitive performances on event reasoning.

**Corresponding authors.

1 Introduction

Events are instances or occurrences that form the basic semantic building units encompassing the meanings of Activities, Accomplishments, Achievements, and States (Vendler, 1957). By employing advanced techniques and models, event reasoning aims to enable machines to comprehend the mechanism of real-world event evolution (Tao et al., 2023a). Under this ultimate goal, event reasoning consists of several key sub-objectives, including the understanding and reasoning about a diverse range of event inter-relations, and predicting events pertaining to certain relations. Reasoning events forms the foundation of sorts of NLP applications such as recommendation systems (Yang et al., 2020), and question answering (Souza Costa et al., 2020).

In recent times, substantial research efforts are dedicated to instructing-tuning language models to acquire the abilities for zero-shot inference such as Flan-T5 Alpaca (Taori et al., 2023), Vicuna (Chiang et al., 2023), WizardLM (Xu et al., 2023), and Dolly (Conover et al., 2023). These models have shown the potential to enhance the language models with versatile instruction-following capabilities through fine-tuning various instruction datasets. Nonetheless, in the training of these models, the instruction-tuning data involved did not explicitly model events and their inter-relations. Consequently, these models perform inferiorly on most event reasoning tasks. The limitations observed in the instruction-tuned models stem from several fundamental factors. Firstly, these models display an inadequate understanding of event structures and semantics and show discrepancies between the model's interpretation and human comprehension of events. Secondly, the models exhibit deficiencies in comprehending the relations between events, resulting in insufficient event reasoning capabilities and the inability to effectively infer and integrate relevant event knowledge. Based on the perfor-

Findings of the Association for Computational Linguistics: ACL 2024, pages 8966–8979
August 11-16, 2024 ©2024 Association for Computational Linguistics

mances, instruction-tuning smaller language models exhibit poorer performance when contrasted with large language models (LLMs) such as ChatGPT and Bloomz-175B (Muennighoff et al., 2022).

To address these obstacles, we present EvIT which is trained on our novel **Ev**ent-oriented **I**nstruction **T**uning. In our method, we incorporate explicit event modeling and event relation comprehension. Specifically, to enhance the comprehension of the structure and semantics of events, we first design a novel structure named event quadruple. This event-centric structure contains two events, their relation, and the background information where the fact holds. The event quadruple covering contextualized events and their inter-relation knowledge would improve the model's conceptions of events. Based on the event quadruple, we develop an event-relation learning paradigm. We train EvIT to predict the tail events of event quadruple in both generation and discrimination manners. We further encapsulate this training process into instruction tuning with generated instruction templates. It can better stimulate the model's abilities to conduct event-related reasoning and associate event knowledge. To implement our training, we construct event quadruple from a large-scale textual corpus. We design a heuristic negative events mining algorithm to construct candidate events for discriminative event-relation training. We finetune Llama by our event-oriented instruction tuning.

We conduct extensive experiments to testify to the effectiveness of EvIT. We first evaluate the performance of EvIT across 8 tasks of event reasoning which are not seen during training. Among these tasks, four are held-in tasks, involving relations explicitly handled during training, while the remaining four tasks are held-out tasks. Results of automatic and human evaluations show that EvIT outperforms other instruction-tuned models.

We summarize our contributions:

- We propose a novel event-oriented instruction tuning paradigm that may also shed light on other event-oriented training. We first design an event-centric structure named event quadruple. Based on event quadruple, we develop the event-relation learning. We then encapsulate the objectives into instruction-tuning.

- We construct an event-oriented instruction-tuning dataset encompassing integrated and diversified data of events in terms of both syntax and semantics with rich relation knowledge.

- We conduct extensive experiments on 8 datasets for testing. Results show the effectiveness of EvIT. Code and Dataset are available on https://github.com/TZWwww/EvIT.

2 Preliminaries

2.1 Event Definition

An event is something that happens involving participants (Mitchell, 2005), which may have correlations with others. Formally, let $\mathcal{E}$ be an event consisting of several participants or arguments. Two events $\mathcal{E}_u$ and $\mathcal{E}_v$ can have a relation $\mathcal{R} \in \mathbb{S}^{\mathcal{R}}$. $\mathbb{S}^{\mathcal{R}}$ is the universe set of event inter-relation which could cover abundant relation types such as temporality, causality, condition, prerequisites, and counterfactual (Zhang et al., 2020).

2.2 Event Reasoning

Event reasoning aims to comprehend, deduce interrelated events, or anticipate forthcoming occurrences (Tao et al., 2023a). It requires to process of queries to deduce events pertaining to specific relations (Han et al., 2021). These relations encompass causality, temporality, counterfactual scenarios, and intent. Distinct interconnections between events demand diverse reasoning proficiencies.

Building upon relational reasoning, the advanced objective of event reasoning revolves around predicting future events (Zhao, 2021). This intricate task mandates the model to grasp events and their relations, possess substantial event-related knowledge and an understanding of event-evolution mechanisms, and ultimately integrate these aspects to prognosticate future events.

3 EvIT Methodology

3.1 Overview

Our primary aim is to achieve an improved model EvIT that excels in event reasoning tasks. An overview of the EvIT training and evaluation process is illustrated in Figure 1. To accomplish this objective, we begin by proposing Event-Oriented Instruction Tuning. Within this training framework, we introduce an event-centric structure denoted as event quadruple along with event-relation learning. This learning is then integrated into the instruction-tuning process. Subsequently, we establish the method of construction of the event quadruple and the training dataset to execute our novel training approach outlined.

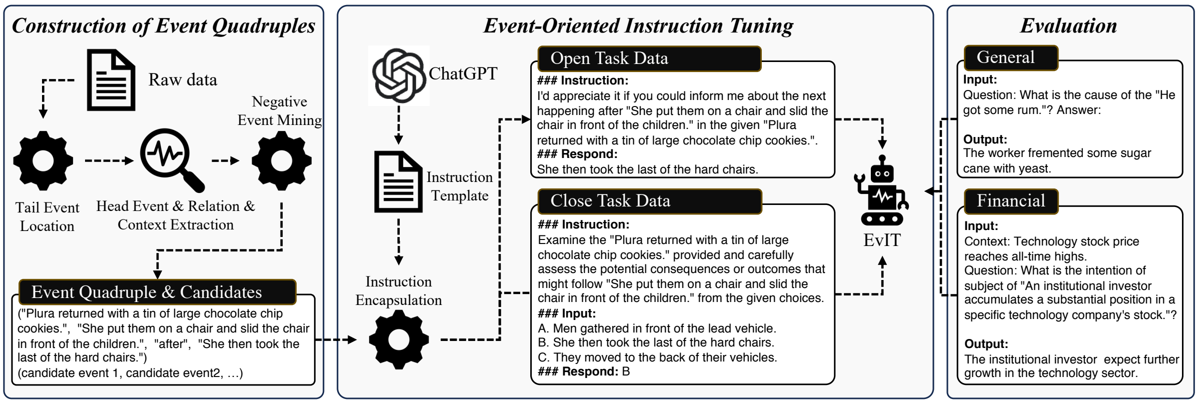

Figure 1: Overview of training process and evaluation of EvIT. The training process encompasses Event-Oriented Instruction Tuning and Construction of event quadruple.

3.2 Event-Oriented Instruction Tuning

Large language models are first pre-trained on enormous unsupervised data and then fine-tuned on supervised data with instructions (Taori et al., 2023; Chiang et al., 2023; Xu et al., 2023; Conover et al., 2023; OpenAI, 2023). However, during all stages of training, existing LLMs are not explicitly trained to understand events and their inter-relations. This leads to several deficiencies. First, they exhibit a lack of comprehension of the structure and semantics of events. This makes a difference between the conceptualization of these models and human understanding. Second, they exhibit deficient apprehension of relations between events. When executing event reasoning, they prove unable to adequately ascertain and integrate knowledge pertaining to the events in question. This shows that these LLMs may not be able to achieve good performance in event reasoning.

In an endeavor to mitigate these limitations, we initially introduce a novel structure referred to as event quadruple, which encompasses comprehensive event knowledge and their inter-relations. Subsequently, we establish event-relation learning based on this framework, ultimately encapsulating this approach into instruction tuning.

Event Quadruple An event quadruple $\mathcal{Q}$ is:

$$\mathcal{Q} = (\mathcal{C}, \mathcal{E}^h, \mathcal{R}, \mathcal{E}^t), \tag{1}$$

in which $\mathcal{E}^h$ is the head event, $\mathcal{E}^t$ is the tail event, and $\mathcal{R}$ is the relation between them. $\mathcal{C}$ is a paragraph of context describing the background information of both events. The event quadruple $\mathcal{Q}$ entails rich semantic and syntactic information of events since each $\mathcal{E}$ describes an event occurring unit that aligns with human understanding. Besides, $\mathcal{Q}$ is rich in event relational and structural knowledge since it precisely captures event inter-relations. Finally, $\mathcal{Q}$ extracts the necessary information for the above events from the context. Contextual information is important for an accurate understanding of an event, because, in the absence of contextual information, the understanding of the event is prone to ambiguity. In summary, using the event quadruple $\mathcal{Q}$ to capture different aspects of events may reduce the risk of event misunderstanding and enhance the conceptions of structure and semantics of the events, thereby improving the accuracy of achieving event reasoning.

Event-Relation Learning Our next objective is to leverage the event quadruple to stimulate the event reasoning abilities of LLMs. The motivation is to enhance the model's understanding of event semantics, event composition, and the interpretation of event relations. We require the model learns to generate the tail event $\mathcal{E}^t$ based on the head event $\mathcal{E}^h$, the context $\mathcal{C}$ and according to the relation $\mathcal{R}$:

$$\mathcal{E}^t = \mathrm{M}\,(\mathcal{E}^h, \mathcal{R}, \mathcal{C}). \tag{2}$$

M is the model to be trained. Through learning to generate events, the model's comprehension of event semantics and structure was stimulated, enabling it to accomplish event reasoning tasks in a manner more aligned with human understanding. Concurrently, this process necessitated the model's apprehension of inter-event relationships, empowering it to associate pertinent event knowledge in order to conduct event relational inference. Moreover, the model learns to draw proper information from the context to answer event reasoning questions more precisely.

In order to enhance the model's event understanding capability and reduce instances of hallucination, we introduce an additional step involving multiple-choice discrimination:

$$\mathcal{E}^t = \mathrm{M}\left(\mathcal{E}^h, \mathcal{R}, \mathcal{C} \mid \mathbb{D}\right). \tag{3}$$

$\mathbb{D}$ is the set of candidate events including the ground-truth tail event $\mathcal{E}^t$ and also several negative candidates. This learning process further reinforced the model's comprehension of events and their interrelationships, enhancing the model's discriminative capabilities of event knowledge.

Instruction-Tuning Encapsulation Incorporating event-relation learning into equations Eq. (2) and (3) can be approached by a basic method of merging the two training procedures into generation training (Tao et al., 2023b). However, this approach does not successfully capture the human strategies employed in these tasks, resulting in an absence of unsupervised event reasoning abilities. In contrast, instruction-tuning techniques achieve alignment and knowledge enhancement (Taori et al., 2023; Chiang et al., 2023). Thus, we integrate event-relation learning into instruction tuning as our means to attain the desired goal.

In instruction-tuning, each dataset includes an instruction, an input, and a response. Our method involves encapsulating the input notation $\mathcal{Q}$ within an instruction, adhering to a predefined template. Initially, we derive instruction templates by querying ChatGPT. Our exploration of event-relation learning encompasses $|\mathbb{S}^{\mathcal{R}}|$ relations, approached through two distinct formulations: generation and discrimination. Furthermore, we account for situations in which the context $\mathcal{C}$ might be absent. Consequently, we require total amounts to $|\mathbb{S}^{\mathcal{R}}| \times 2 \times 2$ variations of instruction templates. For each kind, we ask ChatGPT to list 100 prompts with the query. We depict a query for discrimination instruction templates of $\mathcal{R}$ = `Before` with context $\mathcal{C}$ in Figure 2 (a). More queries are in the Appendix E. We then query ChatGPT to generate instruction templates. The generation examples are in Figure 2 (b). More generated templates are in the Appendix F.

After that, we obtain an encapsulated instruction by changing the placeholder [event] by the head event $\mathcal{E}^h$ and the placeholder [context] by the context $\mathcal{C}$ (if exists). To encapsulate the candidates when in discrimination training, we formulate the choices as a multiple-choice question as shown in Figure 2 (c). Based on the acquired encapsulated instruction and event candidates, following Alpaca (Taori et al., 2023), the final inputs are shown in Figure 2 (d).

(a) Input for ChatGPT

Give me 100 instructions that aim to choose the most possible previous event from given choices of a given event based on a given context. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event and the context in the generated instructions and mention them as [event] and [context]. Don't generate double quotation marks.

(b) Encapsulated instruction templates

- Considering the [context] provided, carefully analyze the potential antecedents for [event] among the given choices, prioritizing those that closely align with the context's temporal and causal factors.
- Examine the sequence of events in the [context] to identify the most probable event preceding [event] from the options provided.
- In the given [context], determine the chronological order of events, and select the choice that logically precedes [event].

(c) Event candidates template

Choices:
A. [event 1] B. [event 2] C. [event 3]

(d) Input for training

Below is an instruction that describes a task.
Write a response that appropriately completes the request.

Instruction: [Encapsulated Instruction]
Input: [Encapsulated Candidates or None]

Response:

Figure 2: (a) ChatGPT input prompt of `Before` relation of discrimination learning with context. (b) The ChatGPT generation examples of query in (a). [event] and [context] are placeholders for the head event $\mathcal{E}^h$ and context $\mathcal{C}$. (c) Template for encapsulating event candidates. (d) The final input for our event-relation training.

3.3 Counstruction of Event Quadruples

In this section, we elaborate on the detail of constructing the event quadruple. We extract event quadruple from BookCorpus (Zhu et al., 2015).

Initially, we locate tail events which may have associated head events linked by a specific relation. Drawing inspiration from Zhou et al. (2022a), we identify explicit relation connectives within the PDTB (Prasad et al., 2008). For each identified connective, we proceed to locate its child nodes. If any of these child nodes possess a VERB part-of-speech tag, we consider it as the triggering term for the tail event. Subsequently, we traverse the dependency tree originating from the trigger term, capturing a subsection of the tree. Given the sequential nature of the dependency tree, the resultant verb-rooted subsection can be correlated with a span of words, thereby forming a recognized tail event denoted as $\mathcal{E}^t$.

Next, we proceed to extract the head event $\mathcal{E}^h$, relation $\mathcal{R}$, and the contextual information $\mathcal{C}$ for event quadruple. It is important to note that obtaining $\mathcal{E}^h$ is notably more complicated than locating the tail event. This increased complexity arises from the fact that establishing a direct link between the trigger of the head event and the relational connective is often challenging through dependency tree analysis since there may be other nodes intermediately. Rather than relying on linguistic rules for extraction, we employ an end-to-end relation parser similar to the one utilized in ASER (Zhang et al., 2020). The function of this relation parser is to dissect a given text where the tail event is. Then extract the head event with a series of relations connecting these two events [1]. The parsed relation is denoted as $\mathcal{R}$. Within this work, our focus is on the following set of relations:

$$\mathcal{R} \in \mathbb{S}^{\mathcal{R}} = \{\texttt{Cause}, \texttt{Effect}, \texttt{After}, \texttt{Before}, \texttt{isCond}, \texttt{hasCond}\}. \quad (4)$$

We only keep relations $\mathcal{R} \in \mathbb{S}^{\mathcal{R}}$. We concatenate sentences before the sentence of $\mathcal{E}^h$ as the context $\mathcal{C}$. Thus far, we have accomplished the construction of event quadruple $\mathcal{Q}$.

We follow Zhou et al. (2022a) to retrieve the negative events to create the candidate event set $\mathbb{D}$. We build a pool of events from the whole corpus and then retrieve negative events by three heuristic rules. Specifically, given a tail event $\mathcal{E}^t$, we build its negative events, in light of lexicon-based, PoS-based, or in-domain retrieval. Then we sample two events from all negative events and form the candidate event set $\mathbb{D}$ with the gold tail event $\mathcal{E}^t$.

4 Experiment

4.1 Evaluation Dataset

We follow Tao et al. (2023a) to incorporate `ECARE` (Du et al., 2022), `MCTACO` (Zhou et al., 2019), `SocialIQA` (Sap et al., 2019), and `SCT` (Mostafazadeh et al., 2016) to evaluate models' capabilities. These datasets assess the abilities of causal, temporal, intentional event reasoning, and event prediction respectively. For each dataset, we evaluate both CLOSE and OPEN forms of task. In CLOSE form we provide candidates while in OPEN form we don't. All datasets are the same with Tao et al. (2023a). We finally have 8 tasks for test. Note that `ECARE` and `MCTACO` are held-in datasets since we explicitly incorporate causal and temporal relations in our event-relational learning. On the contrary, `SocialIQA` and `SCT` are held-out tasks.

[1]We only consider $\mathcal{E}^h$ occurring before the tail event.

4.2 Baselines

We introduce Alpaca-7B (Taori et al., 2023), Vicuna-7B (Chiang et al., 2023), WizardLM-7B (Xu et al., 2023), Dolly-v2-7B (Conover et al., 2023), ChatGPT, and InstructGPT (Ouyang et al., 2022) as our baselines. Details are in Appendix B.

4.3 Implementation Settings

EVIT undergoes fine-tuning using academic resources. Precisely, we utilize 4 $\times$ NVIDIA A100 GPUs to train the Llama-7B for 3 epochs. We use the DeepSpeed training framework [2], and ZERO-2 strategy along with mixed-precision training (fp16) using the standard AdamW optimizer. The maximum sequence length is set to 512, and the batch size is configured as 32. We use gradient checkpointing. The entire fine-tuning process is completed within a duration of 3 hours.

We use Spacy[3] for all linguistic extraction. We utilize event quadruple instances where both $\mathcal{E}^h$ and $\mathcal{E}^t$ have lengths in 2 to 10 words. We exclude data whose context length falls outside the range of 10 to 50 words. For each event quadruple instance, we equally consider training it as either generation or discrimination in event-relational learning.

In our pilot experiments, we test multiple input prompts for each model to search for the optimum prompt for evaluation tasks. We observe minimal fluctuations in the results despite prompt variations. To mitigate the impact of other variables, we ensure consistency by employing the same prompt for all models when they undertake the same task. We turn the CLOSE tasks into multiple-choice questions and require the model to answer by the label of choice. All prompts can be found in the Appendix D.

We find ChatGPT and Vicuna don't generate well-formed events in the zero-shot setting. They generate answers in narrative sentences with explanations leading to difficulty in evaluation. Therefore, we use two-shot in-context learning for them. Other models are in the zero-shot setting.

4.4 Evaluation Metrics

Automatic Evaluation We follow Tao et al. (2023a) to evaluate all models on automatic metrics. For CLOSE tasks, we use accuracy. In OPEN tasks, we use ROUGE-L (Lin, 2004), and BERTSCORE (Zhang et al., 2019) metrics for evaluation.

[2]https://www.deepspeed.ai
[3]https://spacy.io

♣ CLOSE	HELD-IN		HELD-OUT		AVG		
	ECARE	MCTACO	SocialIQA	SCT	HELD-IN	HELD-OUT	ALL
LARGE-SCALE MODELS							
ChatGPT	82.36	90.24	69.68	95.88	86.30	82.78	84.54
Text-Davinci-002	76.08	90.64	73.10	95.99	83.36	84.54	83.95
7B MODELS							
Alpaca (Taori et al., 2023)	67.73	82.49	53.43	81.77	75.11	67.60	71.35
Vicuna (Chiang et al., 2023)	49.86	49.20	33.21	55.16	49.53	44.18	46.85
WizardLM (Xu et al., 2023)	54.32	68.21	34.30	53.13	61.26	43.71	52.48
Dolly-v2 (Conover et al., 2023)	49.06	44.57	33.57	49.71	46.81	41.64	44.22
EvIT (Ours)	**77.06**	**82.80**	**55.60**	**87.33**	**79.93**	**71.46**	**75.69**

Table 1: Automatic evaluation results on CLOSE tasks. The metric for CLOSE tasks is accuracy. Bold numbers stand for the best scores of 7B models.

♣ OPEN	HELD-IN		HELD-OUT		AVG		
	ECARE	MCTACO	SocialIQA	SCT	HELD-IN	HELD-OUT	ALL
LARGE-SCALE MODELS							
ChatGPT	13.34 / 32.95	21.55 / 41.90	12.90 / 34.67	16.38 / 25.13	37.42	29.99	33.70
Text-Davinci-002	7.53 / 22.71	13.50 / 22.29	9.00 / 13.79	12.04 / 19.43	22.50	16.61	19.55
7B MODELS							
Alpaca (Taori et al., 2023)	10.48 / 17.04	13.25 / 26.33	**7.72** / 19.48	**15.98** / 25.67	21.68	22.57	22.12
Vicuna (Chiang et al., 2023)	10.50 / 15.97	8.47 / 1.97	6.64 / 17.28	8.92 / 5.67	8.97	11.47	10.22
WizardLM (Xu et al., 2023)	7.50 / 6.01	7.85 / 13.66	4.31 / 7.45	7.72 / 5.68	9.83	6.56	8.19
Dolly-v2 (Conover et al., 2023)	10.80 / 15.02	12.87 / 23.91	7.08 / 19.79	14.64 / 16.52	19.46	18.15	18.80
EvIT (Ours)	**10.54 / 28.97**	**15.60 / 34.93**	5.12 / **27.02**	13.23 / **27.60**	**31.95**	**27.31**	**29.63**

Table 2: Automatic evaluation results on OPEN tasks in general domain. The metrics for OPEN tasks are ROUGE-L, and BERT-SCORE. Bold numbers stand for the the best scores of 7B models. AVG for OPEN task is the average BERT-SCORE.

For CLOSE tasks, some models won't directly generate the label as the answer. We design the following decode protocol to parse the output answers and obtain the final prediction for all models. We show this protocol in the Appendix A.

Human Evaluation One difficulty in automatically evaluating the OPEN tasks is that the answers for OPEN tasks may not be unique. Therefore, we also conduct the human evaluation for OPEN causality, intentional, and prediction tasks. In our evaluation, we focus on two main aspects. Firstly, we assess the content, which involves checking the correctness, reasonableness, and specificity of the generated events. A higher-quality event should accurately align with the queried relation, exhibiting logical coherence and minimal hallucination. Secondly, we examine the format, ensuring that the generated content adheres to the proper structure and completeness expected in an event. We give a score range from 1 to 5 for each aspect and report the average score of the well-educated human evaluators for each data.

4.5 Results

CLOSE Tasks We show evaluation results of **Close** tasks in Table 1. We first find EvIT performs well in HELD-IN tasks. EvIT outperforms all other instruction-tuning models both in HELD-IN and HELD-OUT. EvIT obtains 75.69 overall average CLOSE score which is 4.34 higher than the second best Alpaca. In ECARE dataset, EvIT event achieves better results than Text-Davinci-002. The results demonstrate the effectiveness of our event-oriented instruction tuning. EvIT can better associate event knowledge to distinguish the correct event from event candidates.

We also find EvIT performs well in HELD-OUT tasks. EvIT outperforms all other instruction-tuning models both in SocialIQA and SCT and obtains a 71.46 average score which is 3.86 higher than the second-best Alpaca. The results demonstrate that EvIT can transfer event knowledge to other event reasoning tasks or event relations.

	Content				Format			
	`Causal`	`Intentional`	`Prediction`	`Avg`	`Causal`	`Intentional`	`Prediction`	`Avg`
Alpaca (Taori et al., 2023)	3.9	**3.7**	3.2	3.60	3.2	3.1	3.3	2.86
WizardLM (Xu et al., 2023)	3.0	3.2	1.8	2.66	2.9	2.4	1.4	2.23
EvIT (Ours)	**4.6**	3.1	**3.5**	**3.73**	**4.7**	**3.8**	**3.8**	**4.10**

Table 3: Human Evaluation results. Bold numbers stand for the best scores.

Pattern	Example
subj-verb-obj	Erika slept part of the trip.
subj-verb-prep	Morgan ran down the hallway.
subj-verb-xcomp	They want to cast me out.
subj-aux-verb-obj	Pierce was taking legal action.
subj-verb-ccomp	He smiled that he had survived.
subj-verb	A riot of questions surged.
subj-verb-obj-prep	I see them through a ripple of smoke.
verb-obj	Adopt an outlook on all affairs.

Table 4: Top frequent event patterns.

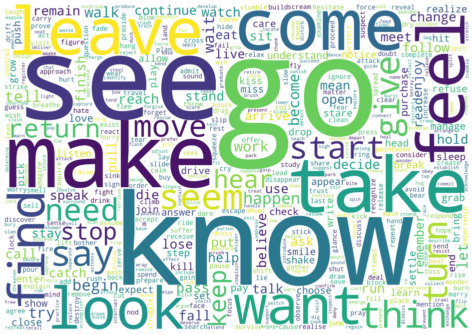

Figure 3: Wordcloud of verbs of events.

Open Tasks We report automatic evaluation of Open Tasks in Table 2. We find EvIT performs well in ROUGE-L and Bert-Score. The average Open Bert-Score of our model is 7.51 higher than the second-best Alpaca. This result shows that EvIT can understand the event semantics more and generate better structures and semantics.

Human Evaluation We conduct a human evaluation of three Open tasks. We assess Content and Format aspects for all tasks. We find this human evaluation is consistent with automatic evaluation. EvIT achieves highest scores in both Content and Format. These results further demonstrate the effectiveness of our model. Our model can answer the event relational reasoning tasks in a way that human favors more. It can generate more precisely and concisely. The generations are more readable and understandable by humans. The events generated are more complete than others. The results also indicate that EvIT can generate more confidently without extra guesses by generally trained models. We find, in the `intentional` task, EvIT falls behind Alpaca in Content. This result may be due to the training relation we choose. Since `intentional` is a held-out relation, there may exist a misalignment of generations of intention content. Overall, EvIT performs the best under human evaluation on average.

4.6 Case Analysis

Event Structure We show the top frequent event structure patterns in Table 4. We obtain the pattern by extracting the root verb and its direct children of an event according to dependency parsing results. We find our event quadruple maintains the completeness of events and covers stereotypical patterns. We also show the length distribution of events in the Appendix C. We notice the events are diversified in patterns and lengths.

Event Distribution In Figure 3, we showcase a word cloud of verbs of event quadruple. We find our curated event quadruple covers a spectrum of event types. This is the main reason that EvIT is able to integrate event knowledge and reason events in various domains.

Evaluation Cases We showcase several cases of Open task generations of three models in Table 5. In the first example, EvIT can generate precise cause intent of the head event. The generation is also as concise as the label. Alpaca could generate the correct intent of the subject, however, the generation is lengthy. The WizardLM fails to output the answer. In the second example, EvIT also answers correctly about the cause of the head event. Alpaca and WizardLM make predictions with excessive association and hallucination. These cases further demonstrate that EvIT can associate correct event knowledge and maintain the completeness of generated events. Furthermore, EvIT can make concise inferences among all models.

5 Related Work

QUESTION	LABEL	EVIT	ALPACA	WIZARDLM
Context: Sasha gave the dog some pills to help it feel better.Question: What will Sasha want to do? Answer:	Hope that it gets well.	She could take him to the vet.	Sasha will likely want to monitor the dog's progress to make sure the pills are working and that the dog is feeling better.	Based on the context, it is not clear what Sasha wants to do next. Can you provide more information or context?
What is the cause of the "Lysosomes have less acidic property than the cytosol."? Answer:	A higher ph indicates less acidic properties.	They have a higher pH.	Lysosomes are membrane-bound organelles that contain enzymes that are able to break down cell waste products.	The statement that describes the difference in pH levels between the lysosomes and the cytosol of a cell.

Table 5: Generation cases of different models on OPEN tasks.

Event Reasoning Event relational reasoning infers events of certain inter-relations. Du et al. (2022) aims to select the accurate cause or effect event from candidates. Zhou et al. (2019) serves as a dataset for event temporal reasoning. Current works presented a scenario for current language understanding and generation systems by incorporating the need for counterfactual reasoning (Qin et al., 2019, 2020). In addition to single-event relation reasoning, existing works also reason events according to diversified event relations (Poria et al., 2021; Han et al., 2021; Yang et al., 2022). Tao et al. (2023b) further unifies datasets of several event-inter relations to transfer event relational knowledge to unseen tasks.

Predicting events necessitates the model to anticipate forthcoming occurrences grounded in the present context (Zhao, 2021). Mostafazadeh et al. (2016) employs a multiple-choice framework to predict future events by encompassing a diverse range of common-sense connections among events. Guan et al. (2019) establish a dataset oriented towards capturing event logic, enabling the generative prediction of future incidents.

Tao et al. (2023a) present the Event Semantic Processing including the event understanding, reasoning, and prediction of event semantics.

Instruction Tuning Instruction tuning refers to the process of fine-tuning a large language model based on specific instructions or guidance provided during training. Chung et al. (2022) finetunes on T5 with a scaling number of datasets which achieves strong few-shot performance even compared to much larger models. Taori et al. (2023) is trained by fine-tuning the LLaMA (Touvron et al., 2023) model using a dataset consisting instructions generated by text-davinci-003. Chiang et al. (2023) is an open-source chatbot created by fine-tuning LLaMA using user-shared conversations gathered from ShareGPT. Xu et al. (2023) extends the previous model by evolve-instruct algorithms to improve the model. Conover et al. (2023) leverages data on the Databricks platform.

In another line of research, instruction tuning is used to make a language model more focused and specialized in certain abilities or domains. Zhang et al. (2023a) trains a medical conversation model with different sources of datasets with instructions. Cui et al. (2023) propose a legal LLM named ChatLaw by legal domain dataset and mitigate hallucination of the model. Zhang et al. (2023b) train an LLM specialized for information extraction with data adapted from a knowledge graph. Yang et al. (2023) design an automatic data curation pipeline and in building financial open-source LLM. Tang et al. (2023) propose a dataset to improve the tool manipulating ability of LLMs. Our work lies in this ability enhancement line of research.

Event-Aware Pretraining Considering both the pre-training and fine-tuning strategies, researchers are dedicated to improving event processing through fine-tuning techniques that incorporate events. In their study, Yu et al. (2020) inject intricate commonsense knowledge about events into pre-trained language models. Similarly, Zhou et al. (2022a,b) enhance language models by focusing on event-related tasks through event masking prediction and generation. However, these works struggle to effectively perform zero-shot reasoning.

6 Conclusion

In this study, we introduce Event-Oriented Instruction Tuning to enhance event reasoning capabilities and train our model EVIT. We first introduce a novel structure called event quadruple as a foundational structure. Building upon this, we establish event relation learning through instruction tuning using generated prompts. We create an instruction-tuning dataset focused on events, encompassing comprehensive and diversified event data both in syntax and semantics. Subsequently, we fine-tune Llama to create the EvIT model. We conduct experiments on both CLOSE and OPEN task settings and compare with several strong cutting-edge instruction-tuned models. Through extensive experiments on 8 datasets, the outcomes demonstrate the efficacy of our proposed approach.

Limitations

In this paper, we only achieve a model that excels in textual event reasoning. However, the event can be represented in other modalities such as visual data. Images would contain more information beyond sentences of events. Leveraging data from other modalities to improve performance remains challenging. We leave it to future work.

Acknowledgement

This work is supported by the National Natural Science Foundation of China under Grant No. 62192731.

References

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.

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 arXiv:2210.11416*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *arXiv preprint arXiv:2306.16092*.

Li Du, Xiao Ding, Kai Xiong, Ting Liu, and Bing Qin. 2022. e-care: a new dataset for exploring explainable causal reasoning. *arXiv preprint arXiv:2205.05849*.

Jian Guan, Yansen Wang, and Minlie Huang. 2019. Story ending generation with incremental encoding and commonsense knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6473–6480.

Rujun Han, I-Hung Hsu, Jiao Sun, Julia Baylon, Qiang Ning, Dan Roth, and Nanyun Peng. 2021. Ester: A machine reading comprehension dataset for reasoning about event semantic relations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7543–7559.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Alexis Mitchell. 2005. The automatic content extraction (ace) program-tasks, data, and evaluation.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

R OpenAI. 2023. Gpt-4 technical report. *arXiv*, pages 2303–08774.

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.

Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Deepanway Ghosal, Rishabh Bhardwaj, Samson Yu Bai Jian, Pengfei Hong, Romila Ghosh, Abhinaba Roy, Niyati Chhaya, et al. 2021. Recognizing emotion cause in conversations. *Cognitive Computation*, 13:1317–1332.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The penn discourse treebank 2.0. In *LREC*.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. 2019. Counterfactual story reasoning and generation. *arXiv preprint arXiv:1909.04076*.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. *arXiv preprint arXiv:2010.05906*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*.

Tarcísio Souza Costa, Simon Gottschalk, and Elena Demidova. 2020. Event-qa: A dataset for event-centric question answering over knowledge graphs. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 3157–3164.

Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, and Le Sun. 2023. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases. *arXiv preprint arXiv:2306.05301*.

Zhengwei Tao, Zhi Jin, Xiaoying Bai, Haiyan Zhao, Yanlin Feng, Jia Li, and Wenpeng Hu. 2023a. Eveval: A comprehensive evaluation of event semantics for large language models. *arXiv preprint arXiv:2305.15268*.

Zhengwei Tao, Zhi Jin, Haiyan Zhao, Chengfeng Dou, Yongqiang Zhao, Tao Shen, and Chongyang Tao. 2023b. Unievent: Unified generative model with multi-dimensional prefix for zero-shot event-relational reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7088–7102.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

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

Zeno Vendler. 1957. Verbs and times. *The philosophical review*, pages 143–160.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Chengbiao Yang, Weizhuo Li, Xiaoping Zhang, Runshun Zhang, and Guilin Qi. 2020. A temporal semantic search system for traditional chinese medicine based on temporal knowledge graphs. In *Semantic Technology: 9th Joint International Conference, JIST 2019, Hangzhou, China, November 25–27, 2019, Revised Selected Papers 9*, pages 13–20. Springer.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*.

Linyi Yang, Zhen Wang, Yuxiang Wu, Jie Yang, and Yue Zhang. 2022. Towards fine-grained causal reasoning and qa. *arXiv preprint arXiv:2204.07408*.

Changlong Yu, Hongming Zhang, Yangqiu Song, and Wilfred Ng. 2020. Cocolm: Complex commonsense enhanced language model with discourse relations. *arXiv preprint arXiv:2012.15643*.

Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Jianquan Li, Guiming Chen, Xiangbo Wu, Zhiyi Zhang, Qingying Xiao, et al. 2023a. Huatuogpt, towards taming language model to be a doctor. *arXiv preprint arXiv:2305.15075*.

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. 2020. Aser: A large-scale eventuality knowledge graph. In *Proceedings of the web conference 2020*, pages 201–211.

Ningyu Zhang, Jintian Zhang, Xiaohan Wang, Honghao Gui, Yinuo Jiang, Xiang Chen, Shengyu Mao, Shuofei Qiao, Zhen Bi, Jing Chen, Xiaozhuan Liang, Yixin Ou, Ruinan Fang, Zekun Xi, Xin Xu, Liankuan Tao, Lei Li, Peng Wang, Zhoubo Li, Guozhou Zheng, and Huajun Chen. 2023b. Deepke-llm: A large language model based knowledge extraction toolkit.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Liang Zhao. 2021. Event prediction in the big data era: A systematic survey. *ACM Computing Surveys (CSUR)*, 54(5):1–37.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. " going on a vacation" takes longer than" going for a walk": A study of temporal commonsense understanding. *arXiv preprint arXiv:1909.03065*.

Yucheng Zhou, Xiubo Geng, Tao Shen, Guodong Long, and Daxin Jiang. 2022a. Eventbert: A pre-trained model for event correlation reasoning. In *Proceedings of the ACM Web Conference 2022*, pages 850–859.

Yucheng Zhou, Tao Shen, Xiubo Geng, Guodong Long, and Daxin Jiang. 2022b. Claret: Pre-training a correlation-aware context-to-event transformer for event-centric generation and classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2559–2575.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Decoding Protocol

We show our decoding protocol for extracting answers of CLOSE tasks as follows:

```
Input  : Prediction 𝒫, candidate set 𝔻.
Output : Answer 𝒜.
pattern =
  "the(?: correct)? (?:option|answer) is[\ s:]+([A-H])"
if 𝒫.startsWithAlphabet() then
  | 𝒜 = starts_alphabet
else if re.match(pattern, 𝒫) then
  | 𝒜 = re.extract(𝒫, patten)
else
  | 𝒜=argmax_{c∈𝔻}(WordOverlap(c, 𝒫))
return 𝒜
```

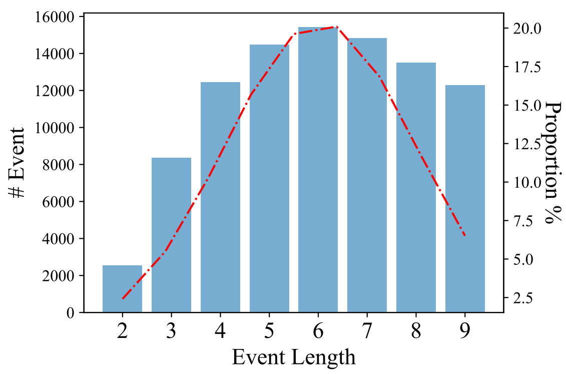

Figure 4: Statistic of the length of events.

B Baselines

Alpaca Vicuna, an open-source chatbot, is developed by fine-tuning LLaMA using user-shared conversations collected from ShareGPT. Preliminary evaluations with GPT-4 as the evaluator reveal that Vicuna achieves more than 90% quality when compared to ChatGPT.

Vicuna This particular model undergoes training through fine-tuning the LLaMA 7B model with a dataset containing 52,000 demonstrations accompanied by instructions generated using Text-Davinci-003.

WizardLM WizardLM is trained on instruction-tuning data generated by the Evol-Instruct algorithm. It demonstrates remarkable performance on complex tasks and remains competitive across various metrics.

Dolly-v2 Databricks' Dolly-v2 7B is a sizable language model designed for instruction-following, trained using 15,000 instruction/response fine-tuning records created by Databricks employees. These records cover various capability domains, encompassing classification, closed QA, generation, information extraction, open QA, and summarization.

ChatGPT An extensive language model developed by OpenAI[4]. The model undergoes fine-tuning, employing a combination of supervised and reinforcement learning techniques to enhance its performance.

InstructGPT We assess two InstructGPT models, specifically Text-Davinci-002.

C Event Length

We show the length distribution of events in Figure 4.

D Event Reasoning Evaluation Prompts

We show prompts for evaluation on all tasks for all models in Figure 5.

E Input for ChatGPT

We show ChatGPT input for generating instruction templates in Figure 6.

F Examples of Instruction Templates

We showcase examples of instruction templates in Figure 7.

[4] https://chat.openai.com/

ECARE Close

Input:
Answer the question by selecting A, B.

Question:
What is the cause of "He got some rum."?
Choices:
A. The worker fremented some sugar cane with yeast.
B. Tom went out and want to hunt some cottontails.
The answer is:
Output:
A

ECARE Open

Input:

Question:
What is the cause of the "He got some rum."?

Answer :

Output:
The worker fremented some sugar cane with yeast.

MCTACO Close

Input:
Answer the question by selecting A, B.

Context:
Durer's father died in 1502, and his mother died in 1513.
Question:
What happened after Durer's father died?
Choices:
A. Durer took care of his mother.
B. He got a new job.
The answer is:
Output:
A

MCTACO Open

Input:

Context:
Durer's father died in 1502, and his mother died in 1513.
Question:
What happened after Durer's father died?

Answer:

Output:
Durer took care of his mother.

SocialIQA Close

Input:
Answer the question by returning A, B or C.

Context:
Due to his car breaking down, Robin decided to ride with Jan's friends to school.
Question:
What will Robin want to do?
Choices:
A. Fix his car.
B. Avoid missing class.
C. Arrive on time to school.
The answer is:
Output:
A

SocialIQA Open

Input:

Context:
Due to his car breaking down, Robin decided to ride with Jan's friends to school.
Question:
What will Robin want to do?

Answer:

Output:
Fix his car..

STC Close

Input:
Answer the question by returning A or B.

Context:
John was writing lyrics for his new album. He started experiencing writer's block. He tried to force himself to write but it wouldn't do anything. He took a walk, hung out with some friends, and looked at nature.
Question:
What is the next event?
Choices:
A. He felt inspiration and then went back home to write.
B. John then got an idea for his painting.
The answer is:
Output:
A

STC Open

Input:

Context:
John was writing lyrics for his new album. He started experiencing writer's block. He tried to force himself to write but it wouldn't do anything. He took a walk, hung out with some friends, and looked at nature.
Question: What is the next event?

Answer:

Output:
He felt inspiration and then went back home to write.

Figure 5: Evaluation prompts for all models.

Input prompts for ChatGPT

	Relations	W/ Choice	W/O Choice
W/ Context	After	Give me 100 instructions that aim to choose the most possible next event from given choices of a given event based on a given context. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event and the context in the generated instructions and mention them as [event] and [context]. Don't generate double quotation marks.	Give me 100 instructions that aim to ask for the next event of a given event based on a given context. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event and the context in the generated instructions and mention them as [event] and [context]. Don't generate double quotation marks.
	Before	Give me 100 instructions that aim to choose the most possible previous event from given choices of a given event based on a given context. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event and the context in the generated instructions and mention them as [event] and [context]. Don't generate double quotation marks.	Give me 100 instructions that aim to ask for the previous event of a given event based on a given context. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event and the context in the generated instructions and mention them as [event] and [context]. Don't generate double quotation marks.
	Cause	Give me 100 instructions that aim to choose the most possible cause event from given choices of a given event based on a given context. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event and the context in the generated instructions and mention them as [event] and [context]. Don't generate double quotation marks.	Give me 100 instructions that aim to ask for the cause event of a given event based on a given context. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event and the context in the generated instructions and mention them as [event] and [context]. Don't generate double quotation marks.
	Effect	Give me 100 instructions that aim to choose the most possible result event from given choices of a given event based on a given context. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event and the context in the generated instructions and mention them as [event] and [context]. Don't generate double quotation marks.	Give me 100 instructions that aim to ask for the result event of a given event based on a given context. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event and the context in the generated instructions and mention them as [event] and [context]. Don't generate double quotation marks.
	hasCond	Give me 100 instructions that aim to select from given choices an event for which a given event can be a precondition based on a given context. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event and the context in the generated instructions and mention them as [event] and [context]. Don't generate double quotation marks.	Give me 100 instructions that aim to ask what the given event might be a precondition for what event based on a given context. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event and the context in the generated instructions and mention them as [event] and [context]. Don't generate double quotation marks.
	isCond	Give me 100 instructions that aim to choose the most possible precondition event from given choices of a given event based on a given context. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event and the context in the generated instructions and mention them as [event] and [context]. Don't generate double quotation marks.	Give me 100 instructions that aim to ask for the prerequisite event of a given event based on a given context. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event and the context in the generated instructions and mention them as [event] and [context]. Don't generate double quotation marks.
W/ Context	After	Give me 100 instructions that aim to choose the most possible next event from given choices of a given event. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event in the generated instructions and mention it as [event]. Don't generate double quotation marks.	Give me 100 instructions that aim to ask for the next event of a given event. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event in the generated instructions and mention it as [event]. Don't generate double quotation marks.
	Before	Give me 100 instructions that aim to choose the most possible previous event from given choices of a given event. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event in the generated instructions and mention it as [event]. Don't generate double quotation marks.	Give me 100 instructions that aim to ask for the previous event of a given event. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event in the generated instructions and mention it as [event]. Don't generate double quotation marks.
	Cause	Give me 100 instructions that aim to choose the most possible cause event from given choices of a given event. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event in the generated instructions and mention it as [event]. Don't generate double quotation marks.	Give me 100 instructions that aim to ask for the cause event of a given event. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event in the generated instructions and mention it as [event]. Don't generate double quotation marks.
	Effect	Give me 100 instructions that aim to choose the most possible result event from given choices of a given event. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event in the generated instructions and mention it as [event]. Don't generate double quotation marks.	Give me 100 instructions that aim to ask for the result event of a given event. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event in the generated instructions and mention it as [event]. Don't generate double quotation marks.
	hasCond	Give me 100 instructions that ask to select from a given candidate an event for which the given event can be a precondition. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event in the generated instructions and mention it as [event]. Don't generate double quotation marks.	Give me 100 instructions to answer what event a given event can be a prerequisite for. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event in the generated instructions and mention it as <event>. Don't generate double quotation marks.
	isCond	Give me 100 instructions that aim to choose the most possible precondition event from given choices of a given event. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event in the generated instructions and mention it as [event]. Don't generate double quotation marks.	Give me 100 instructions asking what is the precondition of the given event. The generated instructions should be as rich as possible in syntax, semantics, and form, covering various task difficulties. Include the event in the generated instructions and mention it as [event]. Don't generate double quotation marks.

Figure 6: Input for ChatGPT to generate instruction-tuning templates.

Instruction Templates

	Relations	W/ Choice	W/O Choice
W/ Context	After	Examine the [context] provided and carefully assess the potential consequences or outcomes that might follow [event] from the given choices.	I'd appreciate it if you could inform me about the next happening after [event] in the given [context].
	Before	Evaluate the potential roles of fate or destiny within the [context] to infer the event that may have been predestined, leading to [event].	Could you please provide the event that is related to [event] and happened before it within the context of [context]?
	Cause	Examine the logical progression of events in the [context] to determine the event that is the most logical causal to [event].	Can you share the series of events that occurred prior to [event] and played a role in its cause within the given [context]?
	Effect	Based on the information provided in the [context], choose the event that represents the most immediate and direct effect of [event].	I'm curious about the events that followed or were influenced by [event] in the given [context]. What can be identified as the results?
	hasCond	Evaluate the potential chain of events leading from [event] to the given choices to identify the one that is directly conditioned on [event].	Please provide insights into the cause-and-effect relationship that links [event] as a precondition to what event in the context of [context].
	isCond	Examine the logical progression of events in the [context] to determine the event that is a condition to [event].	Can you share the series of events that need to happen before [event] and act as its prerequisites within the given [context]?
W/O Context	After	Utilize causal reinforcement learning to identify the optimal sequence of choices leading to [event].	I'm curious about the upcoming occurrence following [event]. Could you elaborate?
	Before	Consider the potential for omitted variable bias in the analysis of each previous event's impact on [event].	I'd appreciate it if you could let me know what happened before [event].
	Cause	Can you provide a detailed chronological explanation of the events that caused [event]?	Utilize causal impulse response functions to explore the dynamic effects of each cause event on [event] over time.
	Effect	Consider the potential impact of each choice on employee morale and productivity concerning [event].	I'd like to know what happened next in the sequence after [event] came to an end.
	hasCond	Select the event from the list for which [event] can serve as a necessary condition.	I'm interested in knowing the events that rely on [event] as a fundamental step. Explain them.
	isCond	Assess the potential role of subconscious desires or psychological motives in the [context] to infer the event that follows from these internal factors, acting as the precondition for [event].	Tell me about the prerequisites that must be fulfilled for the successful execution of [event].

Figure 7: Examples of instruction templates generated by ChatGPT.