Schema-Guided Event Reasoning: A Plug-and-Play Event Reasoning Framework Based on Large Language Models

Yuying Liu[1], Xuechen Zhao[2], Yanyi Huang[3], Ye Wang[1], Xin Song[1], Yue Zhang[1], Haiyan Liu[1], Bin Zhou[1] *

[1] National University of Defense Technology, College of Computer Science and Technology, Changsha, China
[2] Shandong Women's University, School of Data and Computer Science, Jinan, China
[3] State Grid Fujian Information & Telecommunication Company, Fuzhou, China
liuyuying23@nudt.edu.cn, zxcayumi@gmail.com, hhz1668372523@gmail.com, ye.wang@nudt.edu.cn, songxin@nudt.edu.cn, zhangyue@nudt.edu.cn, haiyan_liu@nudt.edu.cn, binzhou@nudt.edu.cn

Abstract

Recent advancements in Large Language Models have increasingly demonstrated their potential for event reasoning. However, LLMs still struggle with this task due to inadequate modeling of event structures. Although introducing schema knowledge has been shown to improve event reasoning performance, existing methods rely on predefined schema library, compromising their scalability and lightweight deployment. To address these challenges, we propose SGER, a plug-and-play **S**chema-**G**uided **E**vent **R**easoning framework. In the schema extraction stage, the model maps event descriptions with diverse surface forms to potential semantic structure representations, achieving an abstract transformation from instances to schemas. The schema prediction stage captures the potential associations between historical event schemas to make forward-looking inferences about possible future event schemas. In the event reasoning stage, we integrate historical events and predicted schemas into prompts to guide LLMs in generating specific, contextually consistent predicted events. Experimental evaluations demonstrate that our framework significantly improves event reasoning performance of LLMs.

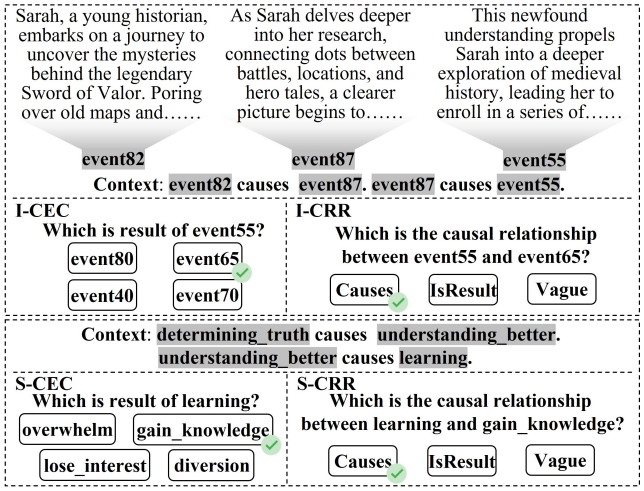

Figure 1: The example of event reasoning involving CEC and CRR tasks. "I-" and "S-" denote instance-level task and schema-level task respectively.

Introduction

Events are fundamental semantic units that capture various types of occurrences, including activities, accomplishments, achievements, and states (Doddington et al. 2004). By employing advanced technologies and models, event reasoning seeks to enable machines to comprehend the mechanisms underlying the evolution of real-world events (Tao et al. 2023). In fields such as law (Rumi, Deng, and Salim 2018), finance (Yang et al. 2019), and healthcare (Dempsey et al. 2017), event reasoning technology helps identify potential risks, enhancing safety and risk management capabilities through logical analysis. Figure 1 shows the example of event reasoning, where each event instance corresponds to an event type. "I-" and "S-" denote instance-level task and schema-level task respectively. These events and their interrelationships form the foundation of the analysis. This study focuses on two core tasks in event reasoning: inferring another event based on known event relations (Contextual Event Classification, CEC) and parsing unknown relations between events (Contextual Relation Reasoning, CRR) (Tao et al. 2025).

In recent years, large language models (LLMs) (Achiam et al. 2023) have achieved significant breakthroughs in the field of natural language processing (NLP). Through large-scale pre-training and instruction fine-tuning, these models are now capable of performing complex reasoning tasks (Achiam et al. 2023; GLM et al. 2024), providing new technical pathways for event reasoning.

Although LLMs have demonstrated powerful performance across various NLP tasks, their performance in event reasoning tasks still needs improvement. A key limitation stems from their relatively weak ability to explicitly model event structure, which hinders adequate understanding of events and their relations. Existing research indicates that introducing structured event schemas helps enhance models' understanding of the semantic relations between events, thereby improving the effectiveness of reasoning (Du et al.

*Corresponding author

2022; Tao et al. 2025). However, these approaches depend on curated schema library, compromising their scalability and lightweight deployment. Therefore, we propose a plug-and-play event reasoning framework named SGER. This framework does not rely on predefined schema libraries but automatically identifies and extracts high-quality event schemas during the reasoning process. Then we construct a schema-guided reasoning mechanism to assist LLMs in performing more accurate event reasoning.

First, we design a schema extractor that transforms concrete historical events into abstract event types in a generative manner. This process resembles the human cognitive process of abstracting specific facts into conceptual models, effectively filtering out non-essential information and highlighting essential event characteristics. Second, we construct a schema predictor that predicts event types or relations capable of answering given questions based on historical contextual schema information. This component focuses on capturing high-order regularities in event evolution. Finally, we integrate historical events and predicted event types or relations into carefully designed prompt templates, guiding LLMs to generate specific predicted events that conform to semantic and contextual logic under schema constraints.

Event schemas serve as crucial intermediate representations, preserving the core semantic structure of events while providing regularities and insights at an abstract level, enabling LLMs to perform reasoning at a higher cognitive level.

The main contributions of this study can be summarized as follows:

- We design a schema-guided event reasoning framework that effectively decouples event understanding and event reasoning tasks through a plug-and-play schema abstraction layer, enabling the system to capture event evolution patterns at a higher level of abstraction.
- We develop the schema extractor and schema predictor that achieve effective transformation from concrete events to abstract schemas, providing schema-level predictive guidance for LLMs.
- The results demonstrate that our lightweight schema-guided framework can significantly improve the event reasoning performance of LLMs.

Problem Formulation

This study focuses on two core tasks in event reasoning: the first is Contextual Event Classification (CEC), which aims to infer another event based on known event relations; the second is Contextual Relation Reasoning (CRR), which is used to determine the semantic relations between events. In both tasks, we consider totally six relation types, namely $R \in$ {Causes, IsResult, Before, After, IsSubevent, HasSubevent} (Tao et al. 2025).

CEC The CEC task evaluates the model's understanding of event semantics and structure. This task is divided into two levels: instance-level (I-CEC) and schema-level (S-CEC). Either instance- or schema- level, given the Instruction, the Context, the Question (containing query event E_q and target relation R), and candidate event set CE, the model M identify event $E_{answer} \in CE$ such that $R(E_q, E_{answer})$ holds:

$$E_{answer} = M(Instruction, Context, E_q, R, CE). \quad (1)$$

CRR The CRR task evaluates the model's understanding of event relations, and is similarly divided into two levels: instance-level (I-CRR) and schema-level (S-CRR). Either instance- or schema- level, given the Instruction, the Context, the Question (containing two query event E_{q_i} and E_{q_j}), and the candidate relation set CR, the model M identify the relation $R_{answer} \in CR$ such that $R_{answer}(E_{q_i}, E_{q_j})$ holds:

$$R_{answer} = M(Instruction, Context, E_{q_i}, E_{q_j}, CR). \quad (2)$$

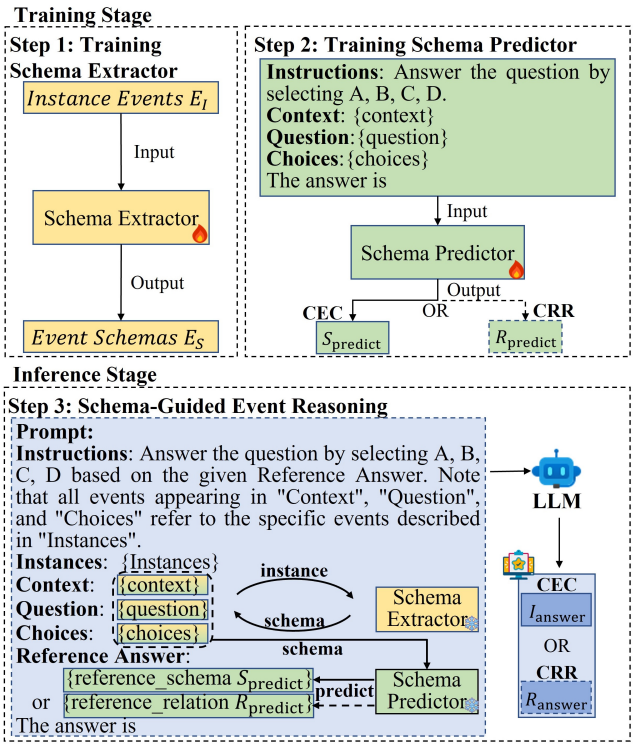

Figure 2: Overview of training and inference process of SGER framework.

Methodology

Our research objective is to develop an enhanced framework that improves the performance of LLMs in event reasoning. For the instance-level event reasoning task, we introduce intermediate representations, namely reference schemas $S_{predict}$ (or reference relations $R_{predict}$), thereby decomposing instance-level reasoning tasks into three consecutive subtasks, as shown in Figure 2.

Under this framework, we train schema extractor and schema predictor separately. During the inference stage, the schema extractor is responsible for transforming concrete event descriptions into abstract event types and providing them as input to the schema predictor. Subsequently, the schema predictor provides reference event type or relation to assist LLMs in event reasoning. The overall framework consists of the following three core components:

- Schema Extractor: This component is responsible for abstracting instance-level event descriptions (E_I) into corresponding schema-level event types (E_S), thereby achieving unified modeling and semantic alignment across instances.
- Schema Predictor: Based on event schemas in the context, this component generates task-relevant reference event types $S_{predict}$ (or relations $R_{predict}$) as contextual anchor points for the reasoning process, guiding the model toward more precise logical inference.
- Event Reasoner: This component selects appropriate reasoning paradigms according to task types and integrates intermediate reference results from the schema predictor to enhance the performance of LLMs in event reasoning tasks.

  The schema-level event reasoning tasks do not require schema extraction and can directly utilize the schema predictor to assist the reasoning process.

  The above components constitute a structured and scalable event reasoning framework. In the following sections, we will provide detailed descriptions of the design and implementation of each component.

Schema Extractor

Each concrete event description corresponds to an abstract event type. For example, "Emma felt her awareness on the topic had expanded significantly" might be abstracted as "knowledge increase." This event type abstracts away specific situational details while preserving the essential characteristics of behaviors, which helps in identifying behavioral patterns. We formalize this process as a sequence-to-sequence transformation task and define the mapping function as: $f_{extract} : E_I \rightarrow E_S$, where E_I represents the textual description of concrete event, and E_S represents its corresponding abstract event type. We can formalize this process as:

$$E_S = \text{Decoder}(\text{Encoder}(E_I)). \tag{3}$$

We train the schema extractor using paired instance-schema data. Given training samples (E_I, E_S), the model optimizes parameters by minimizing the cross-entropy loss function, which is defined as follows:

$$L_{CE} = \frac{1}{N}\sum_{i=1}^{N} CE_{loss}\left(E_{S_i}|f_{extractor}\left(E_{I_i}, \theta_{extractor}\right)\right), \tag{4}$$

where N represents the total number of training samples, $\theta_{extractor}$ represents the trainable parameters of the schema extractor, and $f_{extractor}$ represents the forward computation function of the model.

Schema Predictor

The schema prediction module aims to generate relevant event types or relations that can guide reasoning for specific questions. This module operates at the schema level and predicts which event type or relation is most appropriate for answering given queries. Its core task is to generate reference event type $S_{predict}$ or reference relation $R_{predict}$ based on event schema information in the context. We formalize this mapping process as a function: $f_{predict} : Input_S \rightarrow (S_{predict} \vee R_{predict})$, where $Input_S$ represents the input contextual event schemas, and the output is a relevant event type or relation.

To enhance the discriminative capability of the model, we employ contrastive learning (Du et al. 2024). By introducing negative samples during training, the model minimizes the distance between predictions and positive samples while maximizing separation from negative samples. As a result, the model learns more robust and distinguishable schema representations. $Input_{S_i}$ comprises $Instruction_{S_i}$, $Context_{S_i}$, $Question_{S_i}$, and $Choices_{S_i}$, where $Choices_{S_i} \in \{CS_i, CR_i\}$, $Question_{S_i} \in \{(S_{q_i}, R_i), (S_{q_i}, S_{q_j})\}$, formally represented as:

$$Input_{S_i} = (Instruction_{S_i}, Context_{S_i}, Question_{S_i}, Choices_{S_i}), \tag{5}$$

where CS and CR represent the candidate event type and relation set, respectively. Let $(Input_i, p_{S_i})$ and $(Input_i, n_{S_i})$ denote positive and negative sample pairs respectively. We formalize the contrastive loss function as:

$$L_{CL} = \frac{1}{N}\sum_{i=1}^{N} \frac{CE_{loss}(p_{S_i}|f_{predictor}(Input_{S_i}, \theta_{predictor}))}{CE_{loss}(n_{S_i}|f_{predictor}(Input_{S_i}, \theta_{predictor}))}, \tag{6}$$

where N denotes the number of samples, p_{S_i} is the positive sample representation, and n_{S_i} is a negative sample representation randomly selected from the choices excluding the positive sample. CE_{loss} denotes the cross-entropy loss, $p_{S_i}, n_{S_i} \in \{S_{label_i}, R_{label_i}\}$, where S_{label_i} and R_{label_i} represent the event type and relation indicated by the labels, respectively. This training approach enables the model to not only generate correct event types and relations but also distinguish their semantic differences, thereby enhancing its ability to discriminate between similar yet distinct schemas.

Event Reasoner

In the final stage, we enhance the reasoning capability of LLMs by integrating outputs from the preceding modules. For instance-level tasks, the reasoning process comprises three key steps: First, the schema extraction module abstracts concrete event descriptions into event types, extracting their latent semantic structures. Second, the schema prediction module embeds the extracted event types with contextual information to generate task-relevant reference types $S_{predict}$ or relations $R_{predict}$, serving as reasoning anchors. Finally, we integrate the above information into a structured prompt template as input to the LLMs, guiding practical reasoning.

For schema-level tasks, since the inputs are already abstract event schemas, the schema extraction module is not required. Instead, the schema predictor directly generates reference event types or relations to assist reasoning.

The event reasoner aims to select the correct answer, either S_{answer} or R_{answer}, from the candidate set. This selection is based on contextual information, instance event descriptions, and the reference event type $S_{predict}$ or relation $R_{predict}$ predicted by the schema predictor. To fully leverage the generative capabilities of LLMs, we design a structured prompt template to guide them in generating events or relations that conform to the predicted schemas. This prompt design comprises three key components:

I-CEC		I-CRR		S-CEC		S-CRR	
T-E	Test	T-E	Test	T-P	Test	T-P	Test
98	393	147	588	98	388	146	584

Table 1: Dataset partitioning. T-E denotes the training set for the schema extractor, and T-P denotes the training set for the schema predictor.

- Task Instruction: Explicitly requires the model to generate events or relations related to the predicted schema and instructs it to make selections from the given candidate set to output the final result.
- Contextual Information: Provides the model with comprehensive background descriptions, enabling it to understand the context in which the current event occurs accurately.
- Schema Guidance: Introduces the reference schema $S_{predict}$ or relation $R_{predict}$ generated by the schema predictor as guidance for the reasoning process.

This design decomposes complex event reasoning tasks into multiple subtasks, reducing the difficulty of reasoning while effectively maintaining the coherence and accuracy of overall predictions.

Experiments

This section presents the experimental setup, results, and case analysis. We first validate the effectiveness of the proposed SGER framework on event reasoning tasks. We then conduct ablation studies and analytical experiments to examine the contribution of each component to the model's overall performance. Finally, we present representative cases to intuitively demonstrate the model's behavior and decision-making process in event reasoning.

Experimental Settings

Datasets We employ the EV^2 dataset proposed by (Tao et al. 2025) to evaluate event reasoning performance. EV^2 offers a unified and comprehensive evaluation standard. "S" and "I" denote schema-level and instance-level data, respectively, "CEC" and "CRR" correspond to Contextual Event Classification and Contextual Relation Reasoning.

We extract 20% of samples from each of the two instance-level task datasets (I-CEC and I-CRR) for training the schema extractor, and 20% from each of the two schema-level task datasets (S-CEC and S-CRR) for training the schema predictor. More details on data partitioning are shown in Table 1.

To evaluate the model's generalization capability, we employ the MCNC dataset (Li, Ding, and Liu 2018) for experiments. This dataset constitutes a five-choice task that requires the model to predict possible ending events based on a given sequence of preceding event contexts. Unlike the datasets used during training, we use MCNC to test the model's reasoning performance on unseen, out-of-distribution events, thereby evaluating its cross-distribution generalization capability.

Training and Implementation Details The proposed framework in this study includes sequential training of the schema extractor and schema predictor. Both models employ the Adam optimizer for parameter updates. All experiments are conducted on Ubuntu servers equipped with an Nvidia A800 GPU, with the software environment built on Python 3.10, PyTorch 2.6, and CUDA 12.1.

For the schema extractor, we adopt the BART-large sequence-to-sequence model as the base model. BART (Lewis et al. 2019) is a pre-trained encoder-decoder model that has demonstrated excellent performance across various natural language generation tasks. Compared to decoder-only models, BART's encoder-decoder architecture is better suited for text transformation tasks, enabling more effective modeling of input event semantics and generation of corresponding event types. Specifically, the model training learning rate is set to 5e-5 with a linear decay schedule, and the training batch size is set to 8.

For the schema predictor, we fine-tune based on the Llama3.2-3B model (Touvron et al. 2023) using LoRA (Hu et al. 2022). Although this is a relatively small model, since our task has been simplified to schema-level prediction, the model possesses sufficient capability to capture inter-schema associations. Schema abstraction reduces task complexity, enabling small models to achieve good performance. Specifically, the LoRA configuration parameters are as follows: rank is set to 16, and scaling factor is set to 32. During training, the learning rate is set to 5e-5 with a linear decay schedule, and the training batch size is 8.

The event reasoning stage employs various LLMs as event reasoner. They adopt greedy decoding strategy with max new tokens set to 512. Accuracy serves as the evaluation metric in the experiments.

Baselines We include both closed-source and open-source LLMs as baselines in our experiments. Specifically, we evaluate closed-source models GPT-4o and GPT-3.5, along with open-source models: Qwen2-7B (Team 2024), Baichuan2-7B (Yang et al. 2023), Orca2-7B (Mitra et al. 2023), chatglm2-6B (GLM et al. 2024), Interlm2-7B (Cai et al. 2024), Llama2-7B (Touvron et al. 2023), Vicuna-7B (Chiang et al. 2023), and Llama3.2-3B (Touvron et al. 2023). To ensure fair comparison, all LLMs use identical parameters during inference.

Main Results

To validate the effectiveness of the proposed SGER framework, we conduct extensive experiments and compare it with various closed-source and open-source LLMs. The experiments encompassed four evaluation dimensions: S-CEC, I-CEC, S-CRR, and I-CRR. Table 2 presents the performance of each model on these tasks.

As shown in Table 2, our SGER framework achieves significant performance improvements for closed-source LLMs, such as GPT-4o and GPT-3.5, in most tasks. Specifically, in the S-CEC task, GPT-4o's accuracy increases by 3.61%, while GPT-3.5 improves by 8.25%. Additionally, in the S-CRR and I-CRR tasks, the two models achieve accuracy gains of 2.91% and 8.22% (S-CRR), and 1.19% and

Model	S-CEC			I-CEC			S-CRR			I-CRR		
	Van	SGER	Δ	Van	SGER	Δ	Van	SGER	Δ	Van	SGER	Δ
Closed-Source LLMs												
GPT4o	68.81	72.42	↑**3.61**	70.99	71.37	↑**0.38**	63.36	66.27	↑**2.91**	62.93	64.12	↑**1.19**
GPT3.5	63.4	71.65	↑**8.25**	49.35	47.86	↓1.49	53.6	61.82	↑**8.22**	36.56	52.89	↑**16.33**
Open-Source LLMs												
Qwen2-7B	61.08	71.13	↑**10.05**	58.78	60.52	↑**1.74**	51.03	64.21	↑**13.18**	47.79	49.15	↑**1.36**
Baichuan2-7B	51.29	70.36	↑**19.07**	37.15	56.36	↑**19.21**	46.4	62.33	↑**15.93**	40.48	59.52	↑**19.04**
Orca2-7B	54.64	61.34	↑**6.7**	57	52.47	↓4.53	42.47	54.79	↑**12.32**	45.41	48.64	↑**3.23**
Chatglm2-6B	53.09	72.16	↑**19.07**	33.08	47.27	↑**14.19**	48.97	66.27	↑**17.3**	48.64	63.1	↑**14.46**
Interlm2-7B	63.92	71.13	↑**7.21**	63.1	66.75	↑**3.65**	53.08	62.84	↑**9.76**	56.46	62.76	↑**6.3**
Llama2-7B	39.69	66.24	↑**26.55**	33.33	54.29	↑**20.96**	40.92	64.73	↑**23.81**	41.5	61.9	↑**20.4**
Vicuna-7B	34.54	43.3	↑**8.76**	28.5	32.99	↑**4.49**	47.95	53.94	↑**5.99**	53.23	61.22	↑**7.99**
Llama3.2-3B	55.67	71.65	↑**15.98**	58.78	65.19	↑**6.41**	54.97	65.24	↑**10.27**	50.68	61.05	↑**10.37**

Table 2: Performance comparison of various models on the test set. "Van" represents the original LLMs. Upward arrows indicate increases while downward arrows indicate decreases in metric values.

16.33% (I-CRR), respectively. However, in the I-CEC task, although GPT-4o achieves a slight increase, GPT-3.5's accuracy decreases by 1.49%. This discrepancy may stem from restrictions on GPT-3.5's compliance. The model fails to effectively utilize the correct guidance from the schema predictor, leading to predictions that diverge from the ground truth. Future work will explore robust instruction alignment methods.

Regarding open-source LLMs, the SGER framework similarly demonstrates its superiority, with all models showing varying degrees of performance improvement across different datasets. For example, Llama2-7B achieves an accuracy gain of 26.55% in the S-CEC task and 20.4% in the I-CRR task. This suggests that for models with relatively low baseline performance, our framework can substantially improve their performance in complex event reasoning tasks. Notably, compared to these significant improvements, the overall enhancement on GPT-4o is relatively modest, possibly because this model already possesses strong capabilities, thus limiting the space for further optimization.

Overall, whether on closed-source or open-source LLMs, our SGER framework effectively improves model performance in both schema-level and instance-level event reasoning tasks. By introducing structured prompts and schema-guided mechanisms, models can better understand and generate events or relations, thereby achieving superior performance in complex event reasoning tasks.

Ablation Experiments

To further validate the effectiveness of our proposed framework, we conduct detailed ablation experiments on the Llama2-7B model. The experimental results are shown in Table 3, covering the same four evaluation dimensions: S-CEC, I-CEC, S-CRR, and I-CRR.

We present the performance of the original model (Van) on each task as the baseline for ablation experiments. Building upon this, we evaluate a configuration using only the schema extractor and event reasoner (E+R). In this configuration, we directly inject the event types generated by the schema extractor and their relations into the prompt template of the event reasoner. The experimental results show that accuracy drops to 29.09% and 37.41% for the I-CEC and I-CRR tasks, respectively. Indicating that directly adding structured schema information not only fails to enhance the model's reasoning capability but may also interfere with its judgment.

Method	S-CEC	I-CEC	S-CRR	I-CRR
Van	39.69	33.33	47.95	53.23
E+R	–	29.09	–	37.41
RA+R	64.69	**65.58**	62.63	**64.12**
SGER (Ours)	**66.24**	54.29	**64.73**	61.9

Table 3: Ablation experiments on EV^2. The "-" indicates that the method does not apply to the dataset. "Van" represents the original LLMs. "E+R" represents removing the schema predictor and directly injecting the event types generated by the schema extractor and their relations into the prompt template of the event reasoner. "RA+R" represents removing both the schema extractor and schema predictor, and instead training a Reasoning Assistant using a mixed dataset containing both instances and schemas to assist the event reasoner.

We further evaluate an alternative approach: removing both the schema extractor and schema predictor, and instead training a Reasoning Assistant using a mixed dataset containing both instances and schemas (RA+R). This configuration possesses the capability to handle both input types, accepting either instance-level or schema-level information, and generating corresponding reference event type or relation to assist LLMs in event reasoning. This tests if hybrid training can replace stepwise schema abstract and predict. The experimental results show that this method underperforms our proposed framework on both S-CEC and S-CRR tasks. While the model performed well on the I-

Model	MCNC		
	Vanilla	SGER	RA+R
Qwen2-7B	26.63	↑2.52	↓2.51
Baichuan2-7B	21.11	↑5.52	↑0.5
Orca2-7B	25.13	↑3.01	↓1.51
Chatglm2-6B	20.1	↑6.53	↑4.52
Interlm2-7B	29.65	↓2.01	↓5.03
Llama2-7B	20.5	↑5.63	↑3.62
Vicuna-7B	21.11	↑2.01	↓1.01
Llama3.2-3B	25.63	↑1.51	↓1.01

Table 4: Generalization experiment on MCNC dataset in the ablation experiment.

Module	S-CEC	I-CEC	S-CRR	I-CRR
Extractor	–	55.04	–	62.93
Predictor	71.91	–	64.89	–
SGER (Our)	66.24	54.29	64.73	61.9

Table 5: Analytical experiments on EV^2. The "-" indicates that the method does not apply to the dataset.

CEC and I-CRR tasks (65.58% and 64.12%, respectively), its accuracy dropped significantly on the out-of-distribution MCNC dataset, as shown in Table 4. This indicates that the model merely fitted the surface patterns of the instance-level training data, rather than truly improving its event reasoning ability. This further validates the importance of explicitly extracting event schemas and conducting reasoning at the schema level in our framework. This design not only enhances the model's event reasoning capability but also strengthens its overall generalization and adaptability.

Analytical Experiments

Since errors in the early stages of our framework may cascade through the pipeline, potentially amplifying mistakes in the final output, we conduct analytical experiments to test the performance ceiling of the schema extractor and schema predictor separately. The modular design enables independent optimization of each component, facilitating targeted improvements and simplifying maintenance.

We first evaluated the schema extractor's performance on instance-level data. Since the generative task is challenging to evaluate, we use gold event types (instead of the extractor's outputs) as input for the schema predictor, with subsequent steps following the same procedure as SGER. As shown in Table 5, this method achieves 55.04% and 62.93% accuracy on I-CEC and I-CRR tasks, respectively, comparable to the performance of the SGER framework. These results demonstrate that the trained schema extractor effectively identifies and represents high-quality event schemas, confirming its effectiveness and reliability.

We also evaluate the performance of the schema predictor on schema-level data. In this setup, we retain only the schema predictor to examine its independent capability

Module	Training Time	Inference Time (Avg)	GPU Memory
Vanilla	0	3.18 s	26.57 GB
Extractor	20 min	0.51 s	1.38 GB
Predictor	40 min	0.17 s	5.97 GB
SGER (Ours)	60 min	4.12 s	34.40 GB

Table 6: Comparison of computational costs

on schema-level tasks. As shown in Table 5, the schema predictor achieves accuracy rates of 71.91% and 64.89% on S-CEC and S-CRR tasks, respectively. However, when embedded into the overall framework, its performance on the S-CEC dataset decreases slightly to 66.24%. This phenomenon shows that although the schema predictor can generate high-quality schema information, the reasoner incompletely adopts predictor's correct suggestions. This discrepancy may stem from restrictions on LLM's compliance rather than predictor inadequacy, we will explore optimization strategies in the future.

Computational Complexity and Scalability Analysis

- **Computational Complexity:** Table 6 lists the computational costs of each component during the training and inference stages. We conduct experiments on inference tasks using the Llama2-7B model. Results demonstrate that the two proposed plug-and-play modules consume relatively little time in both training and inference stages, with low incremental memory consumption. Compared to the magnitude of performance improvement, the time overhead and computational resource consumption introduced by these modules are within acceptable ranges.
- **Scalability Advantages:** The modular design of SGER endows the system with superior scalability, as its plug-and-play architecture enables the seamless integration of the schema extractor and schema predictor into any closed-source or open-source LLMs without modifying the underlying model structure.

Case Study

This case analysis is based on the Llama2-7B model and explores the SGER framework process using the I-CEC dataset.

First, the schema extractor is responsible for converting instance-level event descriptions into corresponding event types. Then, these event types are integrated into the schema predictor's input to generate reference event types or relations. Finally, the event reasoner conducts a comprehensive evaluation based on the candidate reference answers generated from the previous stage to derive the final result. The specific input and output details for each stage are shown in Figure 3. This process demonstrates a systematic approach from event description to schema recognition and ultimately to final judgment, reflecting the framework's capability in handling complex event classification tasks.

Schema Extractor

Input:
Instances: {"event82": "Sarah, a young historian, embarks on a journey to uncover the mysteries behind the legendary Sword of Valor. Pouring over old maps and faded letters, she pieces together clues from various sources, determined to shine light on historical truths."}
Output:
event type: {"event82": "cogitate"}

Schema Predictor

Input:
Instructions: Answer the question by selecting A, B, C, D.
Context: "cogitate" causes "understanding". "understanding" causes "learning".
Question: Which is result of "learning"?
Choices:
A. overwhelm
B. lose_interest
C. focus_on_work
D. becoming_informed
The answer is
Output:
The correct answer is D. becoming_informed.
Here's the reasoning:
1. "Cogitate" causes "understanding" (A → B)
2. "Understanding" causes "learning" (B → C)
3. "Learning" causes "becoming informed" (C → D)
So, the correct sequence is: Cogitate → Understanding → Learning → Becoming informed.

Event Reasoner

Input:
Instructions: Answer the question by selecting A, B, C, D based on the given reference event type. If the reference event type is correct, please answer according to it; otherwise, please provide the correct answer and explain the reason. Note that all events appearing in "Context" "Question", and "Choices" refer to the specific events described in "Instances".
Instances: {"event82": "Sarah, a young historian, embarks on a journey to uncover the mysteries behind the legendary Sword of Valor. Pouring over old maps and faded letters, she pieces together clues from various sources, determined to shine light on historical truths.", ...}
Context: "event82" causes "event87". "event87" causes "event55".
Question: Which is result of "event55"?
Choices:
A. event80
B. event40
C. event70
D. event65
Reference event type: becoming_informed
The answer is
Output: D.event65

Figure 3: Case study of event reasoning. Single and double underlines indicate the content output by the schema extractor and the schema predictor, respectively.

Related Work

With the development of LLMs, researchers have begun to focus on their performance in event reasoning. Tao et al. (Tao et al. 2024) proposed a novel event-oriented instruction tuning method called EVIT, which enhances model understanding of event structure and semantics through designing event quadruple structures and event relation learning, and encapsulates these into an instruction tuning framework to fine-tune the model, thereby improving the model's ability to predict event relations and reason about future events. Subsequently, Tao et al. (Tao et al. 2025) introduced the concept of event semantic processing, encompassing two aspects: event understanding and event reasoning. To evaluate the capabilities of LLMs in event reasoning, they introduced a novel benchmark test called EV^2, which comprehensively evaluates event relations and reasoning paradigms from both schema and instance perspectives. Through extensive experiments, the research revealed the capabilities and limitations of LLMs in event reasoning. The study found that while LLMs possess event schema knowledge, they differ from humans in their utilization of this knowledge for event reasoning. This research provides new directions and insights for future research and applications of LLMs in the field of event reasoning. It is of great significance for advancing the field's development.

Inspired by the above research work, we propose a plug-and-play integrated framework designed to achieve an organic combination of schema extraction, schema prediction, and event reasoning, fully leveraging the generative advantages of LLMs. This design effectively improves the accuracy of reasoning results and provides a more reliable solution for event-based intelligent reasoning.

Conclusion

In this study, we introduce a novel plug-and-play event reasoning framework that enhances LLMs' understanding and reasoning capabilities for complex event structures through an explicit schema-instance bridging mechanism. The framework systematically integrates schema extraction, schema prediction, and schema-guided reasoning modules, effectively bridging the semantic gap between concrete event descriptions and abstract event schemas. Extensive experiments on the EV^2 dataset demonstrate the effectiveness of our framework, validating the crucial role of explicit schema modeling in event reasoning. The modular design of the framework ensures robust performance while enhancing structural flexibility. Through ablation experiments and analytical experiments, we further validate the contribution of each component to overall performance, demonstrating the indispensability of each module in the reasoning process.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (Grant No.2023YFC3306100).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Cai, Z.; Cao, M.; Chen, H.; Chen, K.; Chen, K.; Chen, X.; Chen, X.; Chen, Z.; Chen, Z.; Chu, P.; et al. 2024. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3): 6.

Dempsey, W. H.; Moreno, A.; Scott, C. K.; Dennis, M. L.; Gustafson, D. H.; Murphy, S. A.; and Rehg, J. M. 2017. iSurvive: An interpretable, event-time prediction model for mHealth. In *International Conference on Machine Learning*, 970–979. PMLR.

Doddington, G. R.; Mitchell, A.; Przybocki, M. A.; Ramshaw, L. A.; Strassel, S. M.; and Weischedel, R. M. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, 837–840. Citeseer.

Du, X.; Zhang, Z.; Li, S.; Yu, P.; Wang, H.; Lai, T.; Lin, X.; Wang, Z.; Liu, I.; Zhou, B.; et al. 2022. RESIN-11: Schema-guided event prediction for 11 newsworthy scenarios. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*, 54–63.

Du, Y.; Zhao, S.; Zhao, D.; Ma, M.; Chen, Y.; Huo, L.; Yang, Q.; Xu, D.; and Qin, B. 2024. Mogu: A framework for enhancing safety of open-sourced llms while preserving their usability. *arXiv preprint arXiv:2405.14488*.

GLM, T.; Zeng, A.; Xu, B.; Wang, B.; Zhang, C.; Yin, D.; Zhang, D.; Rojas, D.; Feng, G.; Zhao, H.; et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Li, Z.; Ding, X.; and Liu, T. 2018. Constructing narrative event evolutionary graph for script event prediction. *arXiv preprint arXiv:1805.05081*.

Mitra, A.; Del Corro, L.; Mahajan, S.; Codas, A.; Simoes, C.; Agarwal, S.; Chen, X.; Razdaibiedina, A.; Jones, E.; Aggarwal, K.; et al. 2023. Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*.

Rumi, S. K.; Deng, K.; and Salim, F. D. 2018. Theft prediction with individual risk factor of visitors. In *Proceedings of the 26th ACM SIGSPATIAL international conference on advances in geographic information systems*, 552–555.

Tao, Z.; Chen, X.; Jin, Z.; Bai, X.; Zhao, H.; and Lou, Y. 2024. EVIT: Event-Oriented Instruction Tuning for Event Reasoning. *arXiv preprint arXiv:2404.11978*.

Tao, Z.; Jin, Z.; Bai, X.; Zhao, H.; Feng, Y.; Li, J.; and Hu, W. 2023. Eveval: A comprehensive evaluation of event semantics for large language models. *arXiv preprint arXiv:2305.15268*.

Tao, Z.; Jin, Z.; Zhang, Y.; Chen, X.; Zhao, H.; Li, J.; Liang, B.; Tao, C.; Liu, Q.; and Wong, K.-F. 2025. A comprehensive evaluation on event reasoning of large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 25273–25281.

Team, Q. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Yang, A.; Xiao, B.; Wang, B.; Zhang, B.; Bian, C.; Yin, C.; Lv, C.; Pan, D.; Wang, D.; Yan, D.; et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Yang, Y.; Wei, Z.; Chen, Q.; and Wu, L. 2019. Using external knowledge for financial event prediction based on graph neural networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 2161–2164.