ODL-TempLLM: Ontology-Guided and Description Logic-Reasoned Temporal Reasoning with LLMs

Jinshuo Liu[1†], Cheng Bi[1†], Meng Wang[1*], Juan Deng[2], Donghong Ji[1], Jeff Z. Pan[3],
[1]School of Cyber Science and Engineering, Wuhan University, China
[2]School of Computer Science, Wuhan University, China
[3]Knowledge Graph Group, Alan Turing Institute, The University of Edinburgh, UK
{liujinshuo, bicheng, wang_meng, dengjuan, dhji}@whu.edu.cn, j.z.pan@ed.ac.uk

Abstract

Temporal reasoning is crucial for large language models (LLMs) to understand event concurrency and complex temporal interactions in natural language. Recent approaches rely on the LLM to infer temporal relations between events and largely overlook the inherent structural nature of temporal relationships. In this work, we propose **ODL-TempLLM** (**O**ntology-Guided and **D**escription **L**ogic–Constrained **Temp**oral Reasoning with **LLM**s), a novel paradigm for temporal reasoning with LLMs that shifts focus from internal inference to the explicit modeling of temporal structure. ODL-TempLLM leverages ontology learning to explicitly construct structured temporal knowledge, employs a symbolic reasoner to deductively reason about temporal relations and uses logic-constrained retrieval augmentation to obtain relevant facts. Experiments results evaluated across three datasets via various LLM backbones show that our method outperforms state-of-the-art methods by 2.07–31.83 F1 points and 1.00–30.73 EM points, exhibiting strong generalization and highlighting the potential of explicit temporal reasoning.

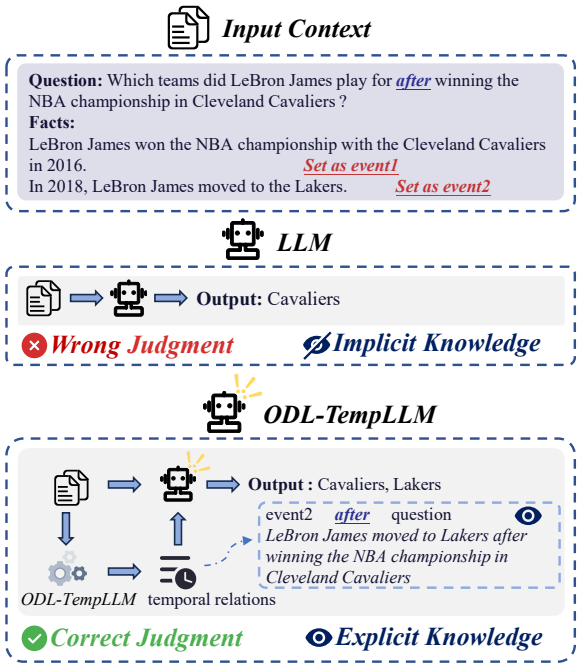

Figure 1: Current LLMs often fail to capture complete temporal relations due to reliance on implicit knowledge. ODL-TempLLM enables explicit learning of temporal relations by constructing a time-event ontology, improving reasoning accuracy.

1 Introduction

Temporal reasoning—the ability to understand and infer relationships among events over time—is a fundamental capability for artificial intelligence, underpinning the comprehension of the dynamic, temporally structured world and progress toward general intelligence(Wang and Zhao, 2024; Xiong et al., 2024). While LLMs have demonstrated impressive performance across many NLP tasks, they often perform poorly on complex temporal reasoning(Chu et al., 2024; Jain et al., 2023; Tang and Belle, 2024). This is due to LLMs' limited ability to integrate multiple temporal facts into coherent and consistent reasoning chains, especially in complex or ambiguous contexts(Fatemi et al., 2024; Wallat et al., 2025).

Traditional approaches,including pre-training (Tan et al., 2023; Wang et al., 2023) and fine-tuning (Kimura et al., 2021, 2022), primarily aim to enhance the temporal reasoning capabilities of LLMs by focusing on improving their internal inference mechanisms. While these methods have demonstrated effectiveness in various settings, they largely concentrate on the model's internal reasoning processes, without fully accounting for the inherent structural characteristics of temporal reasoning. Recent work (Xiong et al., 2024; Bazaga et al., 2025) advocates structured temporal representations over implicit context-based reasoning.

*Corresponding Author. †Equal Contribution.

However, these methods still do not expose explicit temporal relations, leaving models to infer them—and risking incomplete or uncertain predictions. For example, in Figure 1, standard LLMs struggle to infer complete temporal relations due to their reliance on implicit knowledge embedded in the input context. In contrast, Our approach improves model performance by offloading temporal reasoning to an external module that directly yields explicit temporal relations.

Therefore, to enhance LLMs' temporal reasoning, we propose explicitly modeling temporal knowledge as a structured representation. To achieve this goal, we need to overcome the following challenges: **C1: Absence of explicit temporal structure** – temporal relations are implicit and unstructured, hindering systematic temporal reasoning (Lin et al., 2016; Han et al., 2019); **C2: Poor temporal reasoning capability** – LLMs lack an effective mechanism for temporal relation computation and often produce invalid inferences (Yuan et al., 2023, 2024); **C3: Distraction by irrelevant relations** – Irrelevant temporal relations mislead the model's reasoning, resulting in incorrect inferences (Shi et al., 2023; Liu et al., 2024).

To address these challenges, we propose ODL-TempLLM, a framework that replaces implicit temporal reasoning with explicit temporal modeling through ontology learning. It operates in three stages: **for C1:Event-Temporal Ontology Learning** – learns an event-temporal ontology via LLM function calling to explicitly model time; **for C2: Deductive Symbolic Temporal Reasoning** – performs deductive symbolic inference to derive reliable implicit temporal relations; **for C3: Logic-Constrained Retrieval-Augmented Generation** – retrieves only semantic-aligned temporal relations via logical constraints.

Our work highlights the importance of structured knowledge representation in enhancing both the accuracy and interpretability of LLMs. By making temporal reasoning explicit, traceable, and controllable, ODL-TempLLM offers a promising direction for building more reliable and transparent AI systems for complex temporal understanding.

To summarize, our contributions are as follows:

- We propose **ODL-TempLLM**, a neuro-symbolic framework that enhances temporal reasoning in LLMs by integrating explicit event-temporal ontology learning with formal symbolic deduction.
- We design an ontology-based deductive reasoning mechanism grounded in extended *OWL-Time* and Allen's interval algebra, enabling automatic discovery of implicit temporal relations through deductive symbolic method.
- We conduct extensive experiments on CoTempQA, MentaQA and TOT, demonstrating significant performance gains across multiple LLMs, with ablation studies validating the effectiveness of each component.

2 Related Work

In recent years, increasing research has focused on enhancing the temporal reasoning capabilities of LLMs. Current approaches can be broadly categorized into four types: prompt engineering, fine-tuning, pre-training, and integration of mathematical reasoning modules.

Prompt engineering: Zhang et al. (2024) proposes a novel prompting technique specifically designed for temporal reasoning, guiding the model to generate a temporal graph. Bazaga et al. (2025) guides the model to perform more accurate temporal reasoning through temporal self-reflective prompting.

Fine-tuning:Kimura et al. (2022, 2021) employ multi-stage fine-tuning to improve the performance and robustness of LLMs on temporal question answering tasks, especially in low-data regimes. Yuan et al. (2024) develops TimeLlama through instruction tuning, endowing the model with the ability to perform temporal reasoning and generate interpretable explanations.

Pre-training domain: Tan et al. (2023) enhances temporal understanding via temporal span extraction pre-Training. Wang et al. (2023) injects timestamp and temporal signals during continued pre-training to enhance performance on time-related tasks.Rosin et al. (2022) proposes a simple pre-training modification that encourages the model to acquire temporal knowledge through time masking.

Mathematical reasoning modules: Su et al. (2024a) highlights the critical role of mathematical reasoning in understanding co-temporal events. Su et al. (2024b) improves temporal reasoning by improving the mathematical reasoning capabilities of the model.

Moreover, Wei et al. (2023) reveals that existing LLMs overly rely on explicit temporal cues and lack deep modeling of temporal relations. Therefore, this work proposes explicitly modeling tem-

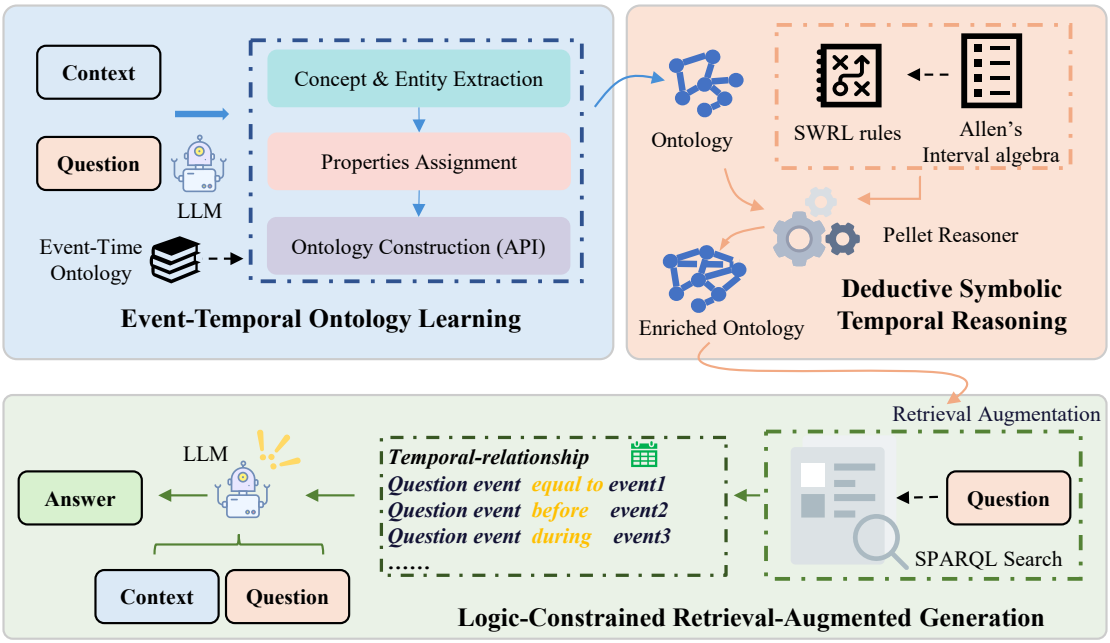

Figure 2: The overview of ODL-TempLLM

poral knowledge as a structured representation to enhance the model's temporal reasoning and understanding capabilities.

3 Method

3.1 Problem Formulation

Given a temporal question-answering dataset $\mathcal{D} = \{(c_i, q_i, a_i)\}_{i=1}^{N}$, where each instance consists of a textual context c_i, a temporal question q_i, and a ground-truth answer a_i, the goal is to generate a_i by reasoning over explicit and implicit temporal relations among events mentioned in c_i and q_i.

We formalize the solution as a three-stage neuro-symbolic pipeline:

- **Event-Temporal Ontology Learning:** Given a question-context pair (c_i, q_i), a set of prompting templates $\mathcal{P}_{\text{OC}} = \{\mathcal{P}_{\text{e}}, \mathcal{P}_{\text{o}}, \mathcal{P}_{\text{d}}\}$ (see Appendix D for prompt details) for entity construction, object property assignment, and data property assignment, and a predefined ontology manipulation interface I, this stage produces a structured event-temporal ontology o_i via LLM-guided function calling.

- **Deductive Symbolic Temporal Reasoning:** Given the event-temporal ontology o_i and a predefined set of SWRL rules R, this stage applies a description logic reasoner to perform deductive inference, producing an enriched ontology o_i'.

- **Logic-Constrained Retrieval-Augmented Generation:** Given the enriched ontology o_i' and the query q_i, this stage removes irrelevant relations through semantic filtering, yielding a refined prompt p_i that is semantically aligned with the q_i.

3.2 Event-Temporal Ontology Learning

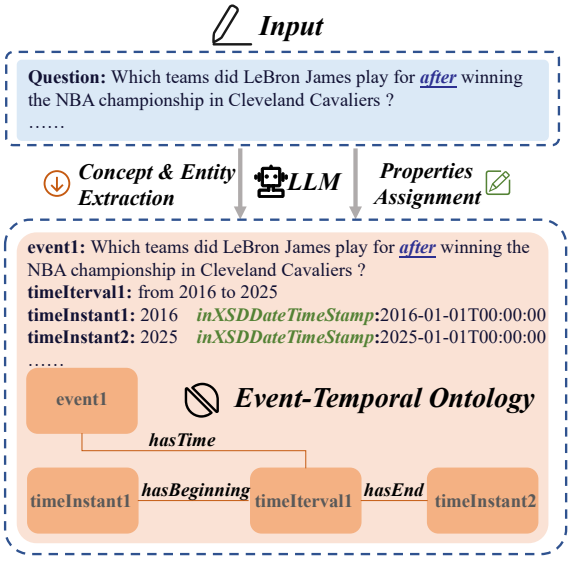

Figure 3: Event-Temporal Ontology Construction Diagram.

We extend OWL-Time (Pan and Hobbs, 2006) (a W3C ontology for time instants, intervals, and their relations) with *Event* classes and the *EventTime*

object property to construct an event-temporal ontology that explicitly models temporal relationships among events.

This module is designed to learn an event-temporal ontology from unstructured temporal question-answering texts by leveraging a LLM to process unstructured text, progressively transforming natural language content into executable ontology manipulation instructions.

3.2.1 Concept Extraction and Entity Construction

For each question–context pair, we prompt an LLM to extract event and temporal entities. The extracted entities are denoted as $E = \{e_i\}_{i=1}^{N}$, and are generated via:

$$e_i = LLM(c_i \circ q_i, \mathcal{P}_{\mathrm{e}}) \tag{1}$$

where LLM represents the LLM, $\circ$ denotes the concatenation operator.

3.2.2 Object Property Linking and Data Property Assignment

Next, we enrich the extracted entities by assigning object properties and data properties, thereby establishing semantic relationships between events and temporal expressions mentioned in the context and question. The object properties and data properties are denoted as $P_{obj} = \{p_{obj,i}\}_{i=1}^{N}$ and $P_{data} = \{p_{data,i}\}_{i=1}^{N}$, respectively. This step is formalized as:

$$p_{obj,i} = LLM(c_i \circ q_i, \mathcal{P}_{\mathrm{o}}) \tag{2}$$

$$p_{data,i} = LLM(c_i \circ q_i, \mathcal{P}_{\mathrm{d}}) \tag{3}$$

This process enables structured semantic grounding of entities within the knowledge construction pipeline. To provide sufficient temporal information for downstream reasoning, we define four object properties and one data property:

$$\forall P_{obj} \in \{eventTime, hasTime, hasBeginning, hasEnd\} \tag{4}$$

$$\forall P_{data} \in \{inXSDDateTimeStamp\} \tag{5}$$

3.2.3 Ontology Construction Based on Function Calling

To bridge LLM outputs with formal ontology construction, we employ a function calling mechanism that maps structured outputs directly to executable ontology operations. We have defined three operation functions, each corresponding to entity creation, data property assignment, and object property linking.

$$\forall I \in \{add\_individual, add\_data\_pro, add\_obj\_pro\} \tag{6}$$

All functions are defined as a structured schema in the prompt, guiding the LLM to generate compliant outputs.

Parsed function calls are executed by the ontology API, producing a constructed ontology. The construct ontology is denoted as $O = \{o_i\}_{i=1}^{N}$, and is generated via:

$$o_i = f_{construct\_ontology}(e_i, P_{obj,i}, P_{data,i}, I) \tag{7}$$

The defined functions are as follows:
$add\_individual$: Creates an individual of a specified category and attaches a textual description as an annotation;
add_data_pro: Invokes the attribute assignment method of an individual to write standardized time values;
add_obj_pro: Establishes object property links between individuals.

3.3 Deductive Symbolic Temporal Reasoning

After constructing the event-temporal ontology O, We employ a reasoner over O to infer implicit temporal relations among events. We adopt Allen's interval algebra. Following prior work (Allen and Hayes, 1989; Grüninger and Li, 2017), we adopt Allen's interval algebra (Figure 7 in Appendix A) and extend it to include time points, supporting unified reasoning over point–point, point–interval, and interval–interval relations.

Based on this extended framework, we design a set of SWRL rules (Horrocks et al., 2004) covering all primitive temporal relation types:

$$R = f_{SWRL}(T, S) \tag{8}$$

Where R is the set of constructed SWRL rules, T denotes the extended Allen-style temporal relations, and S represents the event-temporal ontology schema; f_{SWRL} represents the process of generating the rules. For example, the *before* relation from Allen's interval algebra is encoded as:

```
Event(?i), hasEnd(?i, ?t1),
Event(?j), hasStart(?j, ?t2),
lessThan(?t1, ?t2),
->before(?i, ?j)
```

Listing 1: SWRL rule for the `before` temporal relation

We apply these rules using the Pellet reasoner (Figure 8 in Appendix B) (Sirin et al., 2007) to infer implicit temporal relationships among events:

$$o_i' = Pellet(R, o_i) \tag{9}$$

where $O' = \{o_i'\}_{i=1}^{N}$ denotes the set of enriched ontologies.

3.4 Logic-Constrained Retrieval-Augmented Generation

LLMs suffer from attention dilution when processing long-context inputs (Shi et al., 2023; Liu et al., 2024), making it difficult to focus on the precise temporal dependencies needed for reasoning.

In contrast, humans answering temporal questions focus on the events mentioned in the query and their temporal context (e.g., what happens *before* or *after* a given event)—a strategy that often directly leads to the correct answer. So we emulate this with a logic-constrained retrieval mechanism to retain only the most relevant information.

Specifically, we propose three constraint-driven strategies:

- **Entity-Constrained Retrieval**: Retains only temporal relations involving entities explicitly mentioned in the question;
- **Temporal-Anchor-Constrained Retrieval**: Retains only relations linked to temporal keywords (e.g., *before*, *after*) in the question;
- **Dual-Relevance Consensus**: Takes the intersection of the above, ensuring both topical and temporal relevance.

For each strategy, we design parameterized SPARQL (Harris and Seaborne, 2013) query templates to retrieve qualifying temporal facts, forming a compact prompt $P = \{p_i\}_{i=1}^{N}$ from the enriched ontology:

$$p_i = f_{query}(o_i') \tag{10}$$

where f_{query} denotes the logic-constrained relation selection process.

Experimental results show that Dual-Relevance Consensus achieves the best performance in improving reasoning accuracy.

Finally, the retrieved temporal facts are combined with the original question to construct a compact and logic-constrained prompt, which is then fed into the LLM to generate the final answer $A = \{a_i\}_{i=1}^{N}$:

$$a_i = LLM(q_i \circ c_i \circ p_i) \tag{11}$$

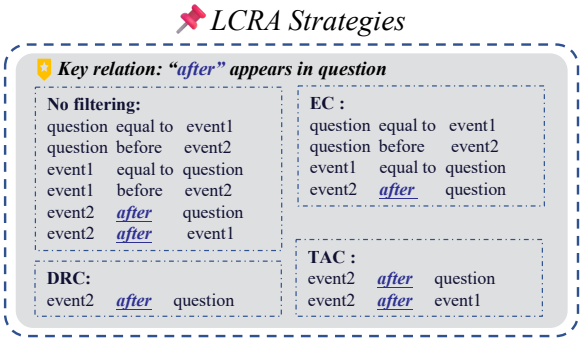

Figure 4: Illustration of the slogic-constrained retrieval process. Retrieved temporal relations are filtered based on question entities and temporal anchors to construct a concise and relevant context.

4 Experiment

In this section, we evaluate our approach by answering the following questions:

- **RQ1:** How does ODL-TempLLM perform in temporal reasoning and understanding compared to current SOTA methods? (§ 4.2)
- **RQ2:** What is the impact of each key component in ODL-TempLLM on the correctness of temporal reasoning? (§ 4.3)
- **RQ3:** Does explicit deductive symbolic temporal reasoning outperform LLMs' implicit, unconstrained reasoning in accuracy and reliability? (§ 4.4.1)
- **RQ4:** How do different retrieval-augmented strategies affect the accuracy of temporal reasoning, and what is the trade-off between context relevance and information completeness? (§ 4.4.2)
- **RQ5:** Under what conditions and in which types of temporal reasoning scenarios does ODL-TempLLM still fail, and what are the underlying causes of these failures? (§ 4.4.3)

4.1 Experimental Setup

Datasets: We demonstrate ODL-TempLLM is a general framework by applying it to three datasets: CoTempQA, MentaQA and TOT.

CoTempQA (Su et al., 2024a) evaluates LLMs' reasoning about *co-temporal* relations (*Equal*, *Overlap*, *During*, and *Mix*) in real-world simultaneous or overlapping events.

MentaQA (Wei et al., 2023) is the first QA benchmark assessing temporal reasoning across

Model	CotempQA								MentaQA		TOT	
	Equal		During		Overlap		Mix		All Factors		Semantic	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
Temporal Reasoning Models												
BigBird	11.6	3.9	16	6.3	12.8	5.3	13.6	6.3	35.81	24.22	8.75	8.28
FiD	43.89	26.14	24.31	11.14	21.6	8.57	28.23	14.56	36.12	26.32	5.66	4.75
Large Language Models												
GPT-3.5-Turbo	66.3	59.4	42.9	31.5	48.5	40.1	46.1	0.7	27.99	24.45	27.79	26.18
GPT-3.5-Turbo(FS+Cot)	75.9	72.48	47.17	37.65	46.78	37.69	66.19	58.06	71.28	57.45	38.49	36.71
GPT-3.5-Turbo(ReAct)	37.94	28.89	44.17	38.79	40.57	34.15	52.97	47.24	46.57	33.83	25.12	22.67
GPT-3.5-Turbo(Temp-CoT)	79.84	71.42	72.46	67.3	68.53	61.29	79.2	78.24	71.67	68.53	68.16	62.74
GPT-3.5-Turbo(ODL-TempLLM)	81.96	73.76	71.46	65.4	70.35	64.31	84.17	79.57	75.72	61.95	70.32	65.42
GPT-4	94.3	91.1	55.8	44.3	55.3	63.5	66.5	23.4	76.93	64.35	58.75	53.75
GPT-4(FS+Cot)	90.52	87.34	60.89	49.2	66.7	64.23	72.84	48.12	69.8	59.04	71.42	65.33
GPT-4(ReAct)	77.65	76.32	46.3	36.81	45.92	57.11	55.29	19.46	63.97	53.4	57.36	54.1
GPT-4(Temp-CoT)	96.55	91.32	71.3	66.12	65.91	63.93	83.21	69.46	69.47	63.4	77.21	74.3
GPT-4(ODL-TempLLM)	**98.67**	**93.12**	73.3	67.47	79.31	72.97	81.92	51.86	79.44	67.33	82.42	78.23
GPT-5.1	90.05	82.33	71.43	69.6	71.43	69.6	85.7	74.44	74.2	59.8	61.75	57.75
GPT-5.1(FS+Cot)	89.2	85.6	78.43	72.6	77.64	72.1	89.42	81.3	81.4	72.3	73.42	66.23
GPT-5.1(ReAct)	75.6	74.2	69.55	66.6	71.21	65.38	82.1	79.4	66.12	63.4	58.1	57.72
GPT-5.1(Temp-CoT)	93.41	92.63	86.13	81.6	86.22	*85.64*	90.31	88.1	79.6	*75.24*	**89.92**	**84.61**
GPT-5.1(ODL-TempLLM)	94.75	*91.97*	**86.72**	**84.28**	89.59	84.33	**93.58**	**91.11**	**88.36**	**81.06**	81.6	79.3
Llama2	59.36	53.21	39.95	28.87	40.33	30.16	52.4	39.61	71.57	58.35	17.5	16.01
Llama2(FS+Cot)	74.27	65.82	37.99	24.61	36.61	23.27	46.15	28.95	72.11	59.85	28.6	25.43
Llama2(ReAct)	36.12	34.49	26.14	18.38	21.12	14.75	27.08	28.64	51.03	42.33	12.3	13.1
Llama2(Temp-CoT)	56.52	54.76	61.44	53.37	49.96	41.33	67.65	63.42	69.43	59.26	44.3	43.2
Llama2(ODL-TempLLM)	62.32	51.83	64.33	55.07	50.69	40.58	71.51	59.68	74.18	60.85	48.16	44.35
Llama3	92.39	83.25	62.62	55.87	65.92	58.65	80.29	69.27	75.03	64.26	53.21	52.68
Llama3(FS+Cot)	93.68	82.33	61.95	52.35	67.64	57.27	74.43	54.7	75.52	64.76	68.75	62.18
Llama3(ReAct)	64.23	60.18	28.76	19.42	32.77	29.51	27.11	25.31	49.23	41.76	44.87	39.42
Llama3(Temp-CoT)	91.33	87.78	73.16	68.29	72.58	61.89	84.31	85.72	76.23	63.25	79.61	72.31
Llama-3(ODL-TempLLM)	*94.85*	88.07	77.36	70.25	77.26	68.6	86.87	74.24	78.48	66.40	81.13	76.57
Llama3.1	91.89	81.88	78.53	73.26	81.23	72.41	85.1	67.6	73.95	60.16	58.21	56.68
Llama3.1(FS+Cot)	91.12	85.4	80.17	76.45	83.6	79.42	88.71	75.44	80.6	71.3	70.75	69.14
Llama3.1(ReAct)	78.36	71.24	79.18	75.55	75.26	73.44	79.32	66.13	70.41	55.33	45.87	39.7
Llama3.1(Temp-CoT)	91.12	82.9	81.72	78.9	*86.93*	85.9	*91.46*	*89.1*	75.14	64.38	87.42	83.24
Llama-3.1(ODL-TempLLM)	94	84.42	*85.43*	*80.26*	**91.45**	**91.36**	92.12	90.74	*81.21*	75.36	*85.18*	*82.57*
Human	97.0	98.3	91.1	93.5	88.0	96.2	82.0	87.0	-	-	-	-

Table 1: Mainresults on CoTempQA, MentaQA and TOT across different models and configuration.We report token-level F1 scores and exact match(EM).**Bold**: best; underlined: second best; *italic underlined*: third best.

Scope, *Order*, and *Counterfactual*, revealing model biases in time-sensitive contexts.

TOT (Fatemi et al., 2024) is a fully synthetic and controllable dataset that generates questions for event ordering and time span calculations, eliminating pretraining contamination to test genuine temporal reasoning.

We primarily compare our framework against leading LLMs, including *Llama2* (Touvron et al., 2023), *Llama3* (Dubey et al., 2024), *Llama3.1* (Dubey et al., 2024), *GPT-3.5-turbo* (Ouyang et al., 2022), *GPT-4* (Achiam et al., 2023) and *GPT-5.1*s. *Llama2* and *Llama3* are run locally with 4-bit quantization for the 70B variants due to limited computational resources, while *Llama3.1* is executed without quantization.

To assess temporal reasoning capabilities, we evaluate the following four configurations on all datasets, using identical base LLMs for fair comparison: Standard zero-shot prompting; *Few-shot* (Brown et al., 2020) + *Chain-of-Thought* (Wei et al., 2022) (FS+CoT); *ReAct* (Yao et al., 2022); Temp-CoT (Chen et al., 2023); ODL-TempLLM (our method).For comprehensive comparison, we also include *FiD* (Izacard and Grave, 2021) and *BigBird* (Zaheer et al., 2020) as additional baselines.

All reported results are averaged over five independent runs to account for variability in model outputs.

4.2 Main Results: Temporal Reasoning Performance

(1) **Performance Gains:** ODL-TempLLM consistently delivers substantial improvements over all methods. As shown in Table 1, it rehabilitates GPT-3.5-Turbo with absolute gains of F1/EM: ↑38.07/↑78.87 on CotempQA-Mix. It boosts

Llama2 by F1/EM: ↑30.66/↑28.34 on TOT, significantly narrowing the human-machine gap.
(2) **Comparison with SOTA:** ODL-TempLLM systematically outperforms FS+CoT, ReAct and Temp-CoT. While ReAct exhibits instability—e.g., on CotempQA-Equal, where Llama3's performance drops by ↓28.16 F1 / ↓23.07 EM—ODL-TempLLM maintains superior robustness, consistently securing the best or second-best results across nearly all settings.

> ***Findings:****ODL-TempLLM mitigates the temporal reasoning bottleneck in LLMs by converting implicit cues into explicit relational guidance, outperforming current SOTA frameworks in both accuracy and robustness.*

4.3 Ablation Study

We ablate key components of ODL-TempLLM on the datasets. Since the symbolic reasoner depends on the time-event ontology, ablating ODL-TempLLM naturally leads to two variants:
(1) **LLM-Ontology**: the reasoner is removed, and the LLM directly extracts temporal relations from the ontology;
(2) **LLM-Context**: the ontology is removed, and the LLM operates on raw input text.

Without an ontology, SPARQL-based retrieval augmentation is infeasible; we simulate retrieval using regular expression matching (Appendix C). On TOT, disabling retrieval augmentation results in invalid outputs, as generated temporal relations exceed the model's context window.

Model	CoTempQA		MentaQA		TOT	
	F1	EM	F1	EM	F1	EM
ODL-TempLLM	84.17	79.57	75.72	61.95	70.32	65.42
w/o LCRAG	69.88	62.87	52.12	39.45	-	-
LLM-Ontology	71.62	65.01	70.17	56.85	46.2	44.8
w/o LCRAG	59.6	51.33	69.36	56.05	34.2	33.8
LLM-Context	67.84	60.56	68.76	55.45	38.7	38.3
w/o LCRAG	64.08	56.12	69.22	55.95	35.23	34.84

Table 2: Ablation study on CoTempQA, MentaQA and TOT with GPT-3.5-Turbo, "LCRAG" denotes "Logic-Constrained Retrieval-Augmented Generation", and "w/o" denotes "without".

As shown in Table 2, disabling both the reasoning engine and LCRAG yields the worst performance, with absolute drops of F1/EM: -24.57/-28.24 on CoTempQA and -36.12/-31.62 on TOT. In contrast, removing all ODL-TempLLM components results in a more moderate degradation (e.g., F1/EM: -6.50/-6.00 on MentaQA) compared to the sharper -23.60/-22.50 loss when only LCRAG is removed. This suggests that unfiltered structured outputs are more harmful than incomplete structure. This is further analyzed in (§ 4.4.2). While removing only the reasoning engine yields the smallest relative drop among all ablations, its impact remains critical; specifically, on the TOT dataset, the performance plummets by over 20%.

> ***Findings:*** *All three components are crucial; externalizing temporal reasoning is essential and disabling LCRAG causes the largest drop due to excessive invalid relations.*

4.4 In-Depth Analysis and Limitations

4.4.1 Performance on Temporal Relation Extraction

As shown in Figure 5, ODL-TempLLM achieves significantly higher accuracy than the LLM-Context and LLM-Ontology, with improvements ranging from 25% to 30%.

Here, accuracy is micro-averaged over all event pairs, each labeled with a unique relation from Allen's Interval Algebra (see Appendix D for annotation protocol and evaluation details).

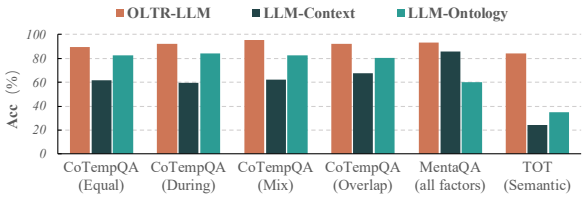

Figure 5: Accuracy of temporal relation extraction methods using GPT-3.5-Turbo as the base model, evaluated on 100 randomly sampled instances per dataset with expert-annotated gold labels.

This gap is particularly pronounced on the TOT dataset,where ODL-TempLLM reaches over 80% accuracy while the other methods drop below 40%. Since TOT inputs are extremely long (averaging 10,000 characters), standard LLMs often miss key facts in such lengthy texts, leading to inaccurate temporal reasoning. This demonstrates that explicit ontology-guided reasoning effectively mitigates the limitations of LLMs.

> ***Findings:****Explicit ontology learning outperforms implicit, data-driven approaches in temporal knowledge acquisition.*

Constraint-Driven Strategies	CotempQA F1	CotempQA EM	ANTR	MentaQA F1	MentaQA EM	ANTR	TOT F1	TOT EM	ANTR
No Filtering	69.88	62.87	373.78	52.12	39.45	87.19	-	-	22410.4
TA	74.94	69.8	26.12	59.95	47.35	12.75	55.45	50.22	311.05
EC	82.08	77.08	16.6	65.1	51.55	7.00	63.46	49.28	51.36
DRC	84.17	79.57	2.24	68.97	55.35	1.74	70.32	65.42	5.32

Table 3: Performance of different retrieval-augmented strategies with GPT-3.5-Turbo. We evaluate the effectiveness of retrieval-augmented strategies by counting the reduction in the average number of temporal relations(ANTR) per example before and after augmentation.

4.4.2 Impact of Constraint-Driven Strategies

To analyze the impact of logic-constrained retrieval-augmented generation, we evaluate three strategies. Without augmentation, the average number of retained temporal relations obtained by reasoner(§ 3.3) is over 373 in CoTempQA, 87 in MentaQA, and 22K in TOT (The TOT dataset is the largest, so it has the most relationships), resulting in the lowest accuracy. This indicates that excessive, irrelevant temporal relations introduce significant noise and impair model reasoning.

Most notably, DRC(§ 3.4) retains less than 2% of the original relations on the datasets, yet achieves the highest performance. This sharp reduction in input complexity, combined with superior accuracy, strongly suggests that excessive temporal relations act as noise, overwhelming the model's reasoning capacity. This aligns with human intuition in answering temporal questions: humans typically focus on events central to the query, and temporal keywords in the question often directly point to the answer.

Model	average number of temporal relations		
	CotempQA	MenatQA	TOT
ODL-TempLLM	373.78	87.19	22410.4
LLM-Context	127.35	15.15	200.17
LLM-Ontology	267.01	21.45	505.44

Table 4: Average number of temporal relationships extracted by different methods with GPT-3.5-Turbo.

We observe a clear negative correlation between the average number of retained temporal relations (ANTR) and reasoning performance. This explains why removing the semantic filter leads to a significant performance drop. As shown in the table,The number of temporal relationships extracted by LLM-Context is the smallest. Therefore, when all components are removed, the model reverts to extracting temporal relations directly from the input question, allowing it to perform implicit filtering based on query content—hence, its performance is not the worst.

Findings:*Excessive temporal relations introduce noise; DRC's superior performance shows that effective filtering—retaining only high-quality, relevant relations—is key to accurate reasoning.*

4.4.3 Error Analysis

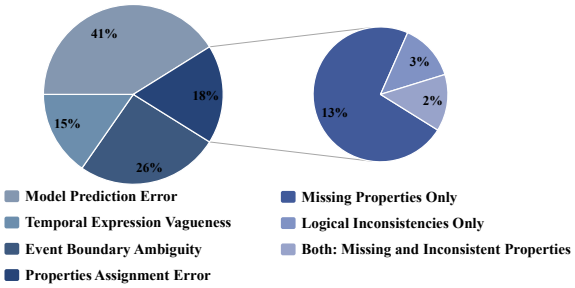

Figure 6: Error type distribution in ODL-TempLLM predictions.

As shown in Figure 6, the dominant error source for ODL-TempLLM is Model Prediction Error (41%), occurring even when inputs and relations are correct—suggesting the model fails to fully leverage valid temporal knowledge, possibly due to over-aggressive filtering.

The next largest sources are Event Boundary Ambiguity (26%) and Temporal Expression Vagueness (18%), reflecting core challenges in segmenting multi-event sentences and grounding vague time phrases.

The remaining 15% are Properties Assignment Errors: mostly missing attributes (13%), with minor cases of logical inconsistencies (2%) or both (3%), indicating insufficient modeling of temporal ontologies. Among these, missing properties dominate, reflecting insufficient capacity to capture temporal attributes during modeling. This further highlights weaknesses in the learning of the event-time ontology.

Findings: *(1) Ambiguous events and vague temporal expressions hinder knowledge construction; (2) Overly strict semantic filtering can discard critical relations; (3) Symbolic reasoning helps, but robust parsing and faithful representation remain open challenges.*

5 Conclusion

We introduce ODL-TempLLM, a neuro-symbolic framework that overcomes LLMs' limitations in temporal reasoning by replacing implicit, context-based inference with explicit modeling and deductive symbolic reasoning. It works in three stages: (1)learning an event-temporal ontology from input text via LLM function calling to explicitly model time; (2) performing deductive symbolic inference to derive reliable implicit temporal relations; and (3) retrieving only semantic-aligned temporal relations via logical constraints. Experiments across multiple benchmarks and diverse LLMs show that ODL-TempLLM consistently outperforms state-of-the-art methods, highlighting the effectiveness of explicit temporal reasoning.

Limitations

While ODL-TempLLM demonstrates superior performance, it has several limitations that warrant further investigation:

Efficiency Bottleneck. The current framework relies on a traditional symbolic reasoning engine for logic verification. Since these engines were not originally optimized for integration with large-scale neural pipelines, they introduce significant latency compared to pure prompting methods. Future work could explore the development of more light-weight, parallelized reasoning modules or the distillation of symbolic logic into the LLM itself to improve inference speed.

Lack of Localized Reasoning. Our approach currently performs a global reasoning process across the entire extracted temporal graph for every query. However, many questions can be resolved using only localized temporal clusters rather than the full context. This "one-size-fits-all" global reasoning strategy may lead to redundant computations and suboptimal efficiency, especially for simpler queries. Implementing a dynamic sub-graph selection mechanism remains a key area for future optimization.

Acknowledgments

This project is funded by Zhiyuan Laboratory(NO.ZYL2024003)

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

James F Allen and Patrick J Hayes. 1989. Moments and points in an interval-based temporal logic. *Computational Intelligence*, 5(3):225–238.

Adrián Bazaga, Rexhina Blloshmi, Bill Byrne, and Adrià de Gispert. 2025. Learning to reason over time: Timeline self-reflection for improved temporal reasoning in language models. *arXiv preprint arXiv:2504.05258*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ziyang Chen, Jinzhi Liao, and Xiang Zhao. 2023. Multi-granularity temporal question answering over knowledge graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11378–11392.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. 2024. Timebench: A comprehensive evaluation of temporal reasoning abilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1204–1228.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Bahare Fatemi, Mehran Kazemi, Anton Tsitsulin, Karishma Malkan, Jinyeong Yim, John Palowitch, Sungyong Seo, Jonathan Halcrow, and Bryan Perozzi. 2024. Test of time: A benchmark for evaluating llms on temporal reasoning. *arXiv preprint arXiv:2406.09170*.

Michael Grüninger and Zhuojun Li. 2017. The time ontology of allen's interval algebra. In *24th International Symposium on Temporal Representation and Reasoning (TIME 2017)*, pages 16–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019. Joint event and temporal relation extraction with shared representations and structured prediction. *arXiv preprint arXiv:1909.05360*.

Steven Harris and Andy Seaborne. 2013. SPARQL 1.1 query language. W3C recommendation, W3C. https://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, and 1 others. 2004. Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21(79):1–31.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pages 874–880.

Raghav Jain, Daivik Sojitra, Arkadeep Acharya, Sriparna Saha, Adam Jatowt, and Sandipan Dandapat. 2023. Do language models have a common sense regarding time? revisiting temporal commonsense reasoning in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6750–6774.

Mayuko Kimura, Lis Kanashiro Pereira, and Ichiro Kobayashi. 2021. Towards a language model for temporal commonsense reasoning. In *Proceedings of the Student Research Workshop Associated with RANLP 2021*, pages 78–84.

Mayuko Kimura, Lis Kanashiro Pereira, and Ichiro Kobayashi. 2022. Toward building a language model for understanding temporal commonsense. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 17–24.

Chen Lin, Dmitriy Dligach, Timothy A Miller, Steven Bethard, and Guergana K Savova. 2016. Multilayered temporal modeling for the clinical domain. *Journal of the American Medical Informatics Association*, 23(2):387–395.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Feng Pan and Jerry R Hobbs. 2006. Time ontology in owl. *W3C working draft, W3C*, 1(1):1.

Guy D Rosin, Ido Guy, and Kira Radinsky. 2022. Time masking for temporal language models. In *Proceedings of the fifteenth ACM international conference on Web search and data mining*, pages 833–841.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. 2007. Pellet: A practical owl-dl reasoner. *Journal of Web Semantics*, 5(2):51–53.

Zhaochen Su, Juntao Li, Jun Zhang, Tong Zhu, Xiaoye Qu, Pan Zhou, Yan Bowen, Yu Cheng, and Min Zhang. 2024a. Living in the moment: Can large language models grasp co-temporal reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13014–13033.

Zhaochen Su, Jun Zhang, Tong Zhu, Xiaoye Qu, Juntao Li, Min Zhang, and Yu Cheng. 2024b. Timo: Towards better temporal reasoning for language models. *arXiv preprint arXiv:2406.14192*.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835.

Weizhi Tang and Vaishak Belle. 2024. Ltlbench: Towards benchmarks for evaluating temporal logic reasoning in large language models. *arXiv preprint arXiv:2407.05434*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jonas Wallat, Abdelrahman Abdallah, Adam Jatowt, and Avishek Anand. 2025. A study into investigating temporal robustness of llms. *arXiv preprint arXiv:2503.17073*.

Jiexin Wang, Adam Jatowt, Masatoshi Yoshikawa, and Yi Cai. 2023. Bitimebert: Extending pre-trained language representations with bi-temporal information. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 812–821.

Yuqing Wang and Yun Zhao. 2024. Tram: Benchmarking temporal reasoning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6389–6415.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yifan Wei, Yisong Su, Huanhuan Ma, Xiaoyan Yu, Fangyu Lei, Yuanzhe Zhang, Jun Zhao, and Kang Liu. 2023. Menatqa: A new dataset for testing the temporal comprehension and reasoning abilities of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1434–1447.

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. 2024. Large language models can learn temporal reasoning. *arXiv preprint arXiv:2401.06853*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

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

Chenhan Yuan, Qianqian Xie, Jimin Huang, and Sophia Ananiadou. 2024. Back to the future: Towards explainable temporal reasoning with large language models. In *Proceedings of the ACM Web Conference 2024*, pages 1963–1974.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and 1 others. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Xinliang Frederick Zhang, Nicholas Beauchamp, and Lu Wang. 2024. Narrative-of-thought: Improving temporal reasoning of large language models via recounted narratives. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16507–16530.

A Allen's Interval Algebra

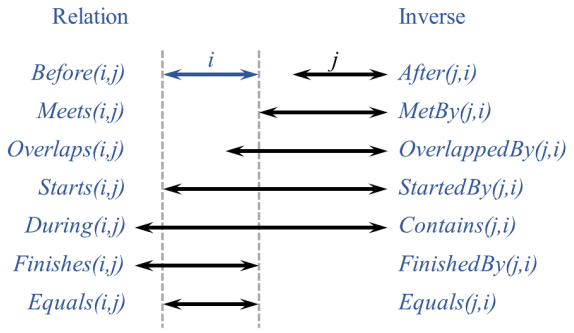

Figure 7: Allen's Interval Algebra: Temporal Relations between Two Events i and j

Allen's Interval Algebra (Allen, 1983) is a formalism for reasoning about temporal relations between time intervals. It defines 13 mutually exclusive and exhaustive binary relations that capture all possible ways two intervals can be ordered in time. These relations are: before, after, meets, met by, overlaps, overlapped by, starts, started by, during, contains, finishes, finished by, and equals. The algebra provides a foundation for qualitative temporal reasoning and has been widely used in natural language processing, planning, and knowledge representation.

B Pellet's reasoning pipeline

We use the Pellet reasoner (Sirin et al., 2007) to infer implicit knowledge from our OWL ontology. As shown in Figure 8, Pellet first parses the input ontology and repairs minor syntax issues to ensure OWL DL compliance. It then applies a Tableaux-based reasoning algorithm to check logical consistency, classify concept hierarchies, and answer queries over individuals (ABox). This process enables the derivation of implicit temporal relationships between events that are not explicitly stated but follow logically from the ontology axioms.

C Simulating Retrieval via Regular Expression Matching

Since our framework lacks a formal ontology, SPARQL-based semantic retrieval is infeasible. Instead, we simulate retrieval augmentation by leveraging the *structured representation* of extracted temporal relations.

After temporal relation extraction, each relation

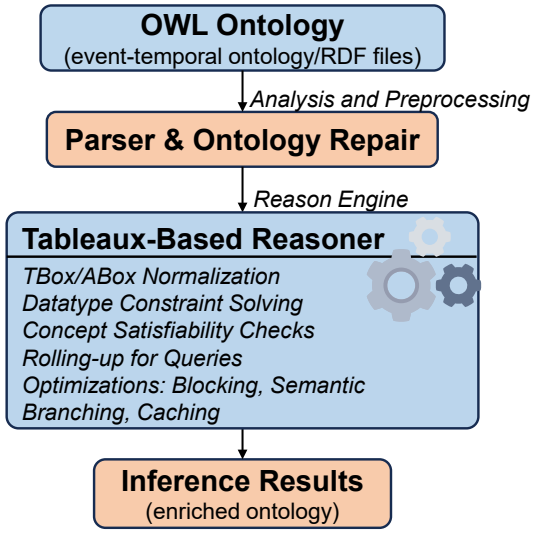

Figure 8: Overview of Pellet's reasoning pipeline.

is normalized into one of the following three canonical forms:

1. `Event_x [REL] Event_y`
2. `Question [REL] Event_y`
3. `Event_x [REL] Question`

where `[REL]` denotes a temporal relation label from Allen's Interval Algebra, and `Event_x`, `Event_y` are surface strings of detected events. The term `Question` refers to the temporal query phrase (e.g., "When did X happen?").

Based on the structured format of the extracted temporal relations, regular expression matching rules are designed according to a predefined retrieval strategy to simulate SPARQL-based retrieval in the absence of an ontology.

D Annotation Protocol for Temporal Relations

In our evaluation, the ground-truth temporal relations are derived from expert annotation based on Allen's Interval Algebra. For any two events, there exists a unique and well-defined temporal relation among the 13 base relations. Given a data with N annotated events, this yields N^2 ordered event pairs, each assigned exactly one temporal label.

To construct the evaluation set, we randomly sampled 100 examples from each of the three datasets (CoTempQA, MentaQA, and TOT). There domain experts independently labeled all event pairs in these samples following the Allen framework. Disagreements were resolved through discussion to ensure consistency. These expert-annotated

labels serve as the gold standard for computing accuracy in our experiments.

Instructions provided to annotators:

- **Task Goal:** For each document, examine every ordered pair of pre-identified events (e_i, e_j) and assign exactly one temporal relation from Allen's Interval Algebra. Use only information explicitly stated or strongly implied in the text.
- **Label Set:** Choose from the 13 base relations: *before*, *after*, *meets*, *met-by*, *overlaps*, *overlapped-by*, *starts*, *started-by*, *during*, *contains*, *finishes*, *finished-by*, *equals*. If the relation is ambiguous or cannot be determined with high confidence, select *vague*.
- **Procedure:** For each of the 100 sampled examples per dataset:

  1. List all N annotated events (as provided by the original dataset).
  2. Consider all N^2 ordered pairs (e_i, e_j), including self-pairs and reverse pairs.
  3. Assign one label per pair based on contextual evidence.

  Judgments were recorded directly in a spreadsheet; no specialized annotation interface was used.
- **Risk Notice:** All texts are from public QA datasets and contain no personally identifiable, sensitive, or harmful content. The task involves only reading short passages and poses no ethical or psychological risk.
- **Compensation:** Annotation was conducted voluntarily by the authors as part of the research effort. No external annotators were involved, and no payment was made.

E Prompt

The prompt formats are showcased in Figure 9

Concept Extraction and Entity Construction Prompt

The ontology contains the following concept classes:
#Concept Classes:
1. Event: Any event will belong to this concept.
2. TimeInstant (subClassOf TemporalEntity): A temporal entity with zero extent or duration.
3. ProperInterval (subClassOf TimeInterval): A temporal entity with non-zero extent or duration, i.e. for which the value of the beginning and end are different.

#MoreDetails:
You can use the api_call to represent the relation.
For example:
"He played the score first in 2017-01-09." belongs to Event, output the api_call:
"add_individual('Event_1', 'Event', 'He played the score first in 2017-01-09')".
And "2017-01-09" belongs to TimeInstant, output the api_call: "add_individual('TimeInstant_1', 'TimeInstant', '2017-01-09')"
And "from 2017-09-01 to 2019-10-09" belongs to ProperInterval,
output the api_call: add_individual('ProperInterval_1', 'ProperInterval', 'from 2017-09-01 to 2019-10-09')"
However, "2017-09-01" and "2019-10-09" belong to TimeInstant, you should also output the api_calls:
"add_individual('TimeInstant_2', 'TimeInstant', '2017-09-01')",
"add_individual('TimeInstant_3', 'TimeInstant', '2019-10-09')".
Please do not forget add TimeInstant individuals for TimeInterval.

#Context:{}

#Reqiurement:
Please strictly output the response as the following json format without any other comments:
{"explanation": "xxxx"; "api_calls": []}

Object Property Linking Prompt

Then, you should link them with Object Properties for the above individuals.
And this ontology contains the following Object Properties:
1. eventTime. (domain: Event, range: TemporalEntity).
2. hasTime. (domain: No constraint, range: TimeInstant or TimeInterval).
3. hasBeginning. (domain: TimeInterval, range: TimeInstant).
4. hasEnd. (domain: TimeInterval, range: TimeInstant).

#More Details:
There is a api_call for this task: "add_obj_pro('xxx', 'xxx', 'xxx')"
For example, you can use "add_obj_pro('Event_1', 'eventTime', '2019-01')" to represents the event1's time information.
And you can use "add_obj_pro('ProperInterval_1', 'hasBeginning', 'TimeInstant_1')"
and "add_obj_pro('ProperInterval_1', 'hasEnd', 'TimeInstant_2')" to represent the beginning and ent time.

#Requirement Please strictly output the response as the following json format without any other comments:
{"explanation": "xxxx"; "api_calls": []}

Data Property Assignment Prompt

Then, you should set the data properties for the Individuals.
And this ontology contains the following Data Properties:
1. inXSDDateTimeStamp (domain: TimeInstant)

#More Details:
There is a api_call for this task:
"add_data_pro('xxx', 'xxx', 'xxxx-xx-xxTxx:xx:xx')"
For example, you can use "add_data_pro('TimeInstant_1', 'inXSDDateTimeStamp', '1962-10-01T00:00:00')" for the TimeInstant individual "1962-10-01".

#Requirement: Please strictly output the response as the following json format without any other comments:
{"explanation": "xxxx"; "api_calls": []}

Figure 9: event-temporal ontology learning instructions and input formats.