Time-aware ReAct Agent for Temporal Knowledge Graph Question Answering

Qianyi Hu[1,2,3], **Xinhui Tu**[1,2,3,*], **Cong Guo**[1,2,3], **Shunping Zhang**[1,2,3]
[1]Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning,
[2]National Language Resources Monitoring and Research Center for Network Media,
[3]School of Computer, Central China Normal University, Wuhan, China
{huqianyi,guo_c,shunpingzhang}@mails.ccnu.edu.cn
tuxinhui@ccnu.edu.cn

Abstract

Temporal knowledge graph question answering (TKGQA) addresses time-sensitive queries using knowledge bases. Although large language models (LLMs) and LLM-based agents such as ReAct have shown potential for TKGQA, they often lack sufficient temporal constraints in the retrieval process. To tackle this challenge, we propose *TempAgent*, a novel autonomous agent framework built on LLMs that enhances their ability to conduct temporal reasoning and comprehension. By integrating temporal constraints into information retrieval, TempAgent effectively discards irrelevant material and concentrates on extracting pertinent temporal and factual information. We evaluate our framework on the MultiTQ dataset, a real-world multi-granularity TKGQA benchmark, using a fully automated setup. Our experimental results reveal the remarkable effectiveness of our approach: TempAgent achieves a 41.3% improvement over the baseline model and a 32.2% gain compared to the Abstract Reasoning Induction (ARI) method. Moreover, our method attains an accuracy of 70.2% on the @hit1 metric, underscoring its substantial advantage in addressing time-aware TKGQA tasks.

1 Introduction

Temporal knowledge graph question answering (TKGQA) is an emerging research area that focuses on time-sensitive queries through the use of structured knowledge graphs. In practice, however, TKGQA poses significant challenges due to the fluid and evolving nature of factual information over time (Gottschalk and Demidova, 2018). For instance, the entity serving as Prime Minister of the United Kingdom changed from Boris Johnson in 2021 to Rishi Sunak in 2023. Given the rapidly changing landscape of real-world events, temporal knowledge graphs (TKGs) have garnered increasing attention as a means to capture evolving information. Meanwhile, recent developments in large language models (LLMs), particularly those instantiated through LLM-based agents such as ReAct (Yao et al., 2022), have exhibited remarkable proficiency in language understanding, generation, interaction, and reasoning (Wei et al., 2022), thereby facilitating improved QA performance through integrated reasoning and tool usage (Zhuang et al., 2024).

Nevertheless, existing solutions often fail to incorporate temporal constraints effectively when interacting with knowledge bases. Although LLMs demonstrate strong performance in general natural language processing tasks, they frequently struggle with questions requiring robust temporal reasoning (Huang and Chang, 2023; Liang et al., 2023). For example, directly applying ReAct—a state-of-the-art LLM-based agent for general QA tasks (Yao et al., 2022; Ding et al., 2024)—to TKGQA reveals a critical limitation: the absence of temporal constraints in retrieval. This shortcoming obstructs effective filtering of facts within temporal knowledge bases and complicates the process of locating time-relevant information.

To overcome these limitations, we introduce *TempAgent*, an LLM-based autonomous agent framework specifically designed for complex, multi-granularity TKGQA tasks. TempAgent enables filtering of irrelevant information based on the varying granularities of temporal constraints embedded within questions. It thus aims to enhance both flexibility and adaptability in a zero-shot setting for tackling complex temporal reasoning queries. Figure 1 illustrates the overall flowchart of TempAgent. Concretely, our approach comprises three central components: *(i)* a vector database containing embedded representations of each knowledge base entry, *(ii)* specialized tools to conduct temporally constrained searches within this database and isolate time-relevant content, *(iii)* an LLM-based agent capable of iterative, multi-

* Corresponding author.

Findings of the Association for Computational Linguistics: NAACL 2025, pages 6028–6039

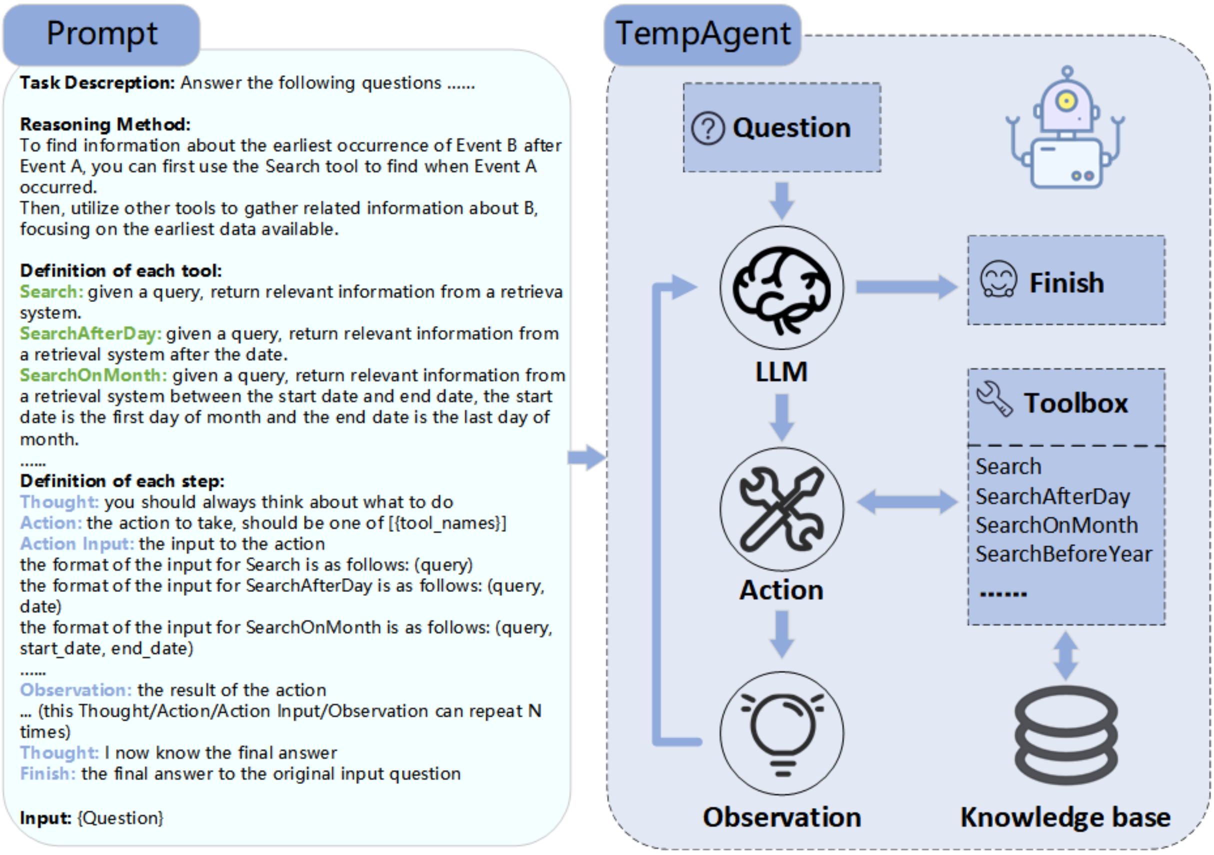

Figure 1: The execution flow of TempAgent

step interactions with the vector database using these custom tools.

We evaluate TempAgent on two datasets: MultiTQ (Chen et al., 2023), which features complex temporal constraints, and CronQuestions (Saxena et al., 2021), a standard TKGQA benchmark. All experiments are conducted in a zero-shot setting using two proprietary LLMs and one open-source LLM. Our results show that TempAgent consistently surpasses naive RAG (Chen et al., 2024a) and the ReAct RAG agent (Yao et al., 2022) across all tested models, achieving a 32.2% improvement over the Abstract Reasoning Induction (ARI) (Chen et al., 2024c) baseline. Notably, under GPT-4, TempAgent achieves a 70.2% accuracy on the Hits@1 metric, underscoring its efficacy in addressing time-sensitive queries. This performance highlights TempAgent's scalability, as it leverages autonomous reasoning within the model. Overall, our findings underscore TempAgent's potential to effectively handle TKGQA tasks and adapt to more sophisticated LLMs. In summary, the primary contributions of our work are as follows:

- **Multi-Granularity Time-Filtering.** We conduct a systematic analysis of temporal constraints in TKGQA, culminating in the development of a multi-granularity time-filtering tool to retrieve time-relevant knowledge more precisely. We further design tailored prompts based on the type and number of temporal constraints to enhance the model's temporal reasoning.

- **Innovative TKGQA Framework.** We propose an innovative framework that harnesses the capabilities of LLM-based agents to address the challenges posed by TKGQA. This framework operates in a zero-shot manner, thereby establishing a foundation for further advances in the field.

- **Comprehensive Evaluation.** Through extensive experimentation on the MultiTQ dataset (Chen et al., 2023) and CronQuestions (Saxena et al., 2021), we demonstrate that TempAgent's temporal reasoning over large-scale knowledge bases significantly outperforms the baseline model. On MultiTQ, our approach yields an improvement of up to 41.3% in Hits@1.

2 Related Work

2.1 TKGQA

Temporal Knowledge Graph Question Answering (TKGQA) involves answering natural language queries using information derived from Temporal Knowledge Graphs (TKGs) (Jia et al., 2018a). Existing TKGQA approaches typically fall into two main categories: *embedding-based methods* and *semantic parsing techniques* (Li et al., 2024b).

Embedding-based Methods. These methods rely on TKG embeddings to represent entities, relations, and temporal information, and then employ scoring functions to identify appropriate answers (Saxena et al., 2021; Shang et al., 2022; Mavromatis et al., 2022; Chen et al., 2023). (Lacroix et al., 2020) introduced TComplEx as a foundational model for TKG embeddings; subsequent work has extended its functionality in various ways. For instance, (Shang et al., 2022) incorporated a time-aware knowledge graph encoder, while CRONKGQA (Saxena et al., 2021) leveraged transformer-based pre-trained TKG models. TempoQR (Mavromatis et al., 2022) introduced specialized modules for contextual, entity, and temporal reasoning. More recently, MultiQA (Chen et al., 2023) proposed a multi-granularity time aggregation mechanism to address more complex temporal queries.

Semantic Parsing Techniques. Although less prevalent, semantic parsing methods concentrate on capturing and structuring temporal constraints within queries. TEQUILA (Jia et al., 2018b) decomposes questions into sub-questions, while SF-TQA (Ding et al., 2022) represents time constraints using predefined structured formats. More recently, SERQA (Du et al., 2024) integrates syntactic information with Masked Self-Attention. Building on the powerful semantic capabilities of Large Language Models (LLMs), (Chen et al., 2024c) introduced the ARI framework to enhance LLMs' temporal reasoning, and Prog-TQA (Chen et al., 2024b) exploits in-context learning for generating and executing program drafts. Our proposed method diverges from these approaches by introducing a time-aware agent that dynamically filters knowledge based on multi-granularity temporal constraints and leverages LLMs' robust reasoning to tackle more intricate temporal queries.

2.2 LLM-Based Agents

Recent advancements in LLMs demonstrate notable reasoning abilities (Sun et al., 2024a), sparking heightened research interest in LLM-based agents that can autonomously address complex QA tasks (Zhao et al., 2024; Wang et al., 2024; Abbasiantaeb et al., 2024). A pioneering work in this domain, ReAct (Yao et al., 2022), introduces a prompting strategy that transforms models such as ChatGPT into interactive language agents. These agents can collaborate with external resources, incorporate feedback, and generate step-by-step reasoning. Building on ReAct, Reflexion (Shinn et al., 2024) proposes a mechanism that uses linguistic feedback instead of model weight updates. Through an episodic memory buffer, Reflexion stores verbal reflections on prior task outcomes, facilitating improved decision-making in subsequent interactions.

Concurrently, efforts to equip open-source LLMs with agent-like capabilities through fine-tuning have gained traction (Lin et al., 2024a,b). However, these approaches often rely heavily on LLM-generated data for training. Such reliance can introduce invalid or suboptimal trajectories into the training corpus, making it challenging to maintain consistent performance. To mitigate these issues, we focus on more reliable, closed-source LLM APIs, such as ChatGPT and GPT-4, investigating strategies to solve complex tasks in a manner less susceptible to the pitfalls of fine-tuning.

3 Approach

3.1 Problem Formulation

TKGQA. A temporal knowledge graph typically consists of a set of fact tuples that capture relationships between entities over time, represented as $\mathcal{G} = \{\langle e, r, e', t\rangle \mid e, e' \in \mathcal{E}, r \in \mathcal{R}, t \in T\}$, where $\mathcal{E}$, $\mathcal{R}$, and T denote the sets of entities, relationships, and timestamps, respectively. Each tuple $\langle e, r, e', t\rangle$ encapsulates factual knowledge, indicating that a relationship r exists between the head entity e and tail entity e' at timestamp t. Given a total of N questions and O answers, we denote the sets of questions and answers as $Q = \{q_1, ..., q_n, ..., q_N\}$ and $A = \{a_1, ..., a_o, ..., a_O\}$ respectively. The objective of TKGQA is to accurately answer questions by leveraging relevant tuples in $\mathcal{G}$.

Tool	Definition
Search(q)	Returns relevant knowledge.
SearchOnDay(q, t)	Returns relevant knowledge on a given date.
SearchAfterDay(q, t)	Returns relevant knowledge strictly after a given date.
SearchBeforeDay(q, t)	Returns relevant knowledge strictly before a given date.
SearchOnMonth(q, t_{start}, t_{end})	Returns relevant knowledge between t_{start} and t_{end}.
SearchAfterMonth(q, t)	Returns relevant knowledge after the last day of a specified month.
SearchBeforeMonth(q, t)	Returns relevant knowledge before the first day of a specified month.
SearchOnYear(q, t_{start}, t_{end})	Returns relevant knowledge between t_{start} and t_{end}.
SearchAfterYear(q, t)	Returns relevant knowledge after the last day of a specified year.
SearchBeforeYear(q, t)	Returns relevant knowledge before the first day of a specified year.

Table 1: Descriptions of the retrieval tools.

3.2 Overview

Current approaches to TKGQA often fail to efficiently incorporate temporal constraints when searching knowledge bases, and contemporary LLMs face inherent challenges in processing time-sensitive tasks. To address these issues, we introduce *TempAgent*, a novel framework that augments multi-granularity QA over TKGs by integrating sophisticated temporal reasoning. First, we embed the TKG, treating it as our temporally structured knowledge base. Next, we enhance the LLM's capabilities by constructing a dedicated "toolbox" that can manipulate TKG data or intermediate outputs.

In tackling complex temporal reasoning tasks, we adopt an agent-based paradigm. Specifically, we design distinct prompts for different question types, allowing our method to adopt a divide-and-conquer approach. Our agent provides targeted reasoning for each category of question, ensuring that each query employs a customized reasoning process. The complete process is illustrated in Figure 1.

3.3 Toolbox for retrieval

To facilitate robust retrieval from our external knowledge base, we develop a suite of specialized retrieval tools. Broadly, these tools can be grouped into two categories: those that do not impose temporal constraints and those that filter data based on specific time conditions. In the latter category, we analyze a range of question types in detail, classifying time-related keywords into three principal categories: "before," "after," and "on." Given that real-world time naturally has multiple granularities, our chosen dataset similarly exhibits this property. Consequently, each principal category is subdivided by time granularity—namely "year," "month," and "day"—yielding nine time-aware filters. We also include a single tool without time filtering. Table 1 lists all ten tools in our retrieval toolbox.

Tool Inputs and Processing. Each tool receives as input a $query$ alongside one or two timestamps. The $query$ is dynamically generated by the model based on the requirements of the specific question-answering step. For queries that target a particular year or month, we interpret that year or month as a time span from its first day to its last day. Hence, tools such as SearchOnMonth and SearchOnYear require two timestamps, t_{start} and t_{end}.

When a tool is invoked, it first computes the embedding v_{query}. Next, it applies a filtering step on $\mathcal{G}$ using time constraints determined by the tool and timestamps. For example, invoking [SearchAfterDay("Richard Boucher visit", "2008-04-02")] excludes all tuples from $\mathcal{G}$ dated prior to April 2, 2008. Formally, we define

$$\mathcal{G}_{\text{filter}} = \begin{cases} \text{filter}(\mathcal{G}, t_{\text{start}}, t_{\text{end}}) \\ \text{for on-year/month queries,} \\ \text{filter}(\mathcal{G}, t_i) \\ \text{for single time-bound queries,} \end{cases} \tag{1}$$

where $\mathcal{G}_{\text{filter}}$ is the time-filtered subset of $\mathcal{G}$.

Subsequently, we calculate the semantic similarity between v_{query} and each tuple in $\mathcal{G}_{\text{filter}}$. For tools that do not incorporate time filtering, the similarity is computed against the entire $\mathcal{G}$. We then select the top k tuples, converting them back to textual form:

$$RT = \text{top}_k\Big(S\big(v_{query}, \mathcal{G}\big)\Big), \tag{2}$$

$$RT = \text{top}_k\Big(S\big(v_{query}, \mathcal{G}_{\text{filter}}\big)\Big), \tag{3}$$

where $S(\cdot)$ represents the semantic similarity function, and $RT = \{rt_1, \ldots, rt_k\}$ denotes the set of k most relevant tuples returned to the LLM-based agent.

Property	Sample Question
By Question Type	
Equal	*Who paid a visit to Iraq in December 2008?*
Before/After	*Which country signed formal agreements with Iraq after 2014-12?*
First/Last	*In which year did the head of government of Finland last visit China?*
Equal Multi	*Who threatened Iraq last, before Najm-al-Din Karim did?*
Before Last	*Before Ethiopia, with whom did Swaziland last formally sign an agreement?*
After First	*Who visited China first, after the International Federation of Red Cross and Red Crescent Societies?*
By Time Granularity	
Year	*Who announced the intention to negotiate with Swaziland in 2012?*
Month	*Which country signed formal agreements with Glaxosmithkline in August 2009?*
Day	*When did the Australian professor praise Iraq?*
By Answer Type	
Entity	*Which country was condemned by Thailand after Kuwait on 21 June 2011?*
Time	*When did the presidential family of the United States make an appeal to China?*

Table 2: Examples of Various Question Types.

3.4 Agent for reasoning

Having established our retrieval toolbox, we now focus on the core reasoning mechanism of our TKGQA system. Motivated by ReAct (Yao et al., 2022), we conceptualize the LLM as an agent that interacts with its environment to handle TKGQA tasks. As shown in Figure 1, our carefully crafted prompt P consists of four key components: a comprehensive task description, tailored reasoning methods for diverse question types, definitions of each tool available in the retrieval toolbox and detailed step-by-step instructions.

In this setup, the agent's planning trajectory is framed as a sequence of *thought–action–observation* triplets $(\mathcal{T}, \mathcal{A}, \mathcal{O})$. We denote by $\mathcal{T}$ the internal thoughts or deliberations of the LLM-based agent, $\mathcal{A}$ the set of available actions (i.e., invoking a specific tool or *Finish* the process), and $\mathcal{O}$ the observations or feedback derived from each action. Let q be a given question, and let the final set of answers be $FA = \{fa_1, \ldots, fa_L\}$. Each step i proceeds as follows:

$$RT_i = Tool(action\_input, \mathcal{G}) \tag{4}$$

$$o_i = RT_i \tag{5}$$

Here, $Tool$ represents the invocation of a tool from the toolbox. $action\_input$ denotes the input required for the tool call, which is generated by thought t_i and utilized by action a_i. We have specified the format of $action\_input$ in P to ensure stable tool invocation. $\mathcal{G}$ represents the external knowledge graph database. The observation $o_i \in \mathcal{O}$ captures the feedback and information received from the environment in response to these actions. If the action is Finish $f_i \in \mathcal{F}$, it indicates the agent finishes the task with a final answer fa. This can be represented as:

$$fa = LLM(P, t_i, \{o_1, ..., o_{i-1}\}), \tag{6}$$

where fa is the final answer generated by the LLM, P is the prompt that provides instructions, t_i is the current thought at step i, and $o_1, ..., o_{i-1}$ is the set of all previous observations up to step $i-1$.

At each iteration, the prompt P is expanded to include the newly formed $\langle t_i, a_i, o_i \rangle$. This iterative scheme enables the agent to track and reference the complete history of reasoning steps, tool calls, and observations, thus supporting an adaptive decision-making process. Equipped with continuous feedback and intermediate outputs, TempAgent can refine its plan and actions accordingly, leading to a more coherent, accurate, and efficient solution trajectory.

3.5 Answer Evaluation

Due to the time-consuming nature of human evaluation and the limitations of code-based assessments, LLMs are increasingly adopted to evaluate model outputs, often achieving results that align closely with human judgments given appropriate instructions. In our experiments, we randomly sampled 100 questions and evaluated the answers generated by the model using `gpt-3.5-turbo`. Manual verification of these LLM-based evaluations revealed a high agreement rate of 93.33%.

Our evaluation prompt guides the LLM through a systematic procedure: it first assesses whether the

Model	Hits@1				
	Overall	Question Type		Answer Type	
		Single	Multiple	Entity	Time
MultiQA	0.289	0.361	0.112	0.329	0.203
ARI	0.380	0.680	0.210	0.394	0.344
Naive RAG(LLama3)	0.323	0.401	0.124	0.180	0.627
Naive RAG(GPT3.5)	0.320	0.404	0.112	0.201	0.576
Naive RAG(GPT4)	0.379	0.469	0.155	0.242	0.672
ReAct RAG Agent(LLama3)	0.332	0.406	0.149	0.248	0.514
ReAct RAG Agent(GPT3.5)	0.330	0.411	0.130	0.202	0.610
ReAct RAG Agent(GPT4)	0.398	0.506	0.130	0.243	0.735
TempAgent(LLama3)	0.543	0.697	0.162	0.483	0.672
TempAgent(GPT3.5)	0.539	0.684	0.168	0.478	0.661
TempAgent(GPT4)	**0.702**	**0.857**	**0.316**	**0.624**	**0.870**

Table 3: Overall results of baselines and our framwork on the **MULTITQ**.

Model	Hits@1				
	Overall	Question Type		Answer Type	
		Simple	Complex	Entity	Time
ARI	0.707	0.860	0.570	0.660	0.800
Naive RAG(LLama3)	0.533	0.621	0.200	0.537	0.526
Naive RAG(GPT3.5)	0.492	0.589	0.200	0.524	0.474
Naive RAG(GPT4)	0.633	0.726	0.280	0.610	0.684
ReAct RAG Agent(LLama3)	0.725	0.789	0.480	0.683	0.816
ReAct RAG Agent(GPT3.5)	0.692	0.758	0.440	0.671	0.737
ReAct RAG Agent(GPT4)	0.809	0.863	0.600	0.768	0.895
TempAgent(LLama3)	0.800	0.853	0.600	0.768	0.868
TempAgent(GPT3.5)	0.767	0.821	0.560	0.732	0.842
TempAgent(GPT4)	**0.842**	**0.895**	**0.640**	**0.805**	**0.921**

Table 4: Overall results of baselines and our framework on the **CronQuestions** dataset. This dataset consists of single time-granularity data and is divided into "simple" and "complex" categories based on difficulty.

predicted answer matches the ground-truth answer via a preliminary logical analysis and then provides a concise decision (i.e., *correct* or *incorrect*). This conservative grading practice tends to classify ambiguous or partially correct answers as *incorrect*, thus leading to a slightly lower score than a typical human evaluation might yield. Nevertheless, the resulting metric offers a more rigorous assessment of model performance.

4 Experiment

4.1 Settings

Dataset: We conducted experiments on MultiTQ(Chen et al., 2023), a multi-granularity TKGQA dataset. In addition to its diverse temporal granularities, the dataset has a sufficiently large scale, providing abundant relevant facts per query. The original MultiTQ dataset contains over 50,000 question–answer pairs; for our experiments, we randomly sampled 1% of these pairs, resulting in 560 questions as our test set.

To further evaluate the generalization of our approach, we also tested on CronQuestions (Saxena et al., 2021). In MultiTQ, temporal information is represented as time points, and questions can include either a single temporal constraint or multiple constraints. We label those with a single constraint as *Single questions*, and those with multiple time constraints as *Multiple questions*. Consequently, this dataset thoroughly tests a model's capacity to handle varied temporal demands while coordinating across multiple granularities. Table 2 provides illustrative examples of different question types from both categories.

Baselines: We compare *TempAgent* against four sets of baselines:

MultiQA (Chen et al., 2023): This is the primary baseline of the MultiTQ dataset (Chen et al., 2023), relying on an embedding-based strategy.

ARI (Chen et al., 2024c): The Abstract Reasoning Induction (ARI) framework designed to enhance LLMs' temporal knowledge and reasoning capacities.

Naive RAG (Chen et al., 2024a): One of the earliest retrieval-augmented generation approaches post-ChatGPT. It follows a "retrieve-then-read" paradigm (Ma et al., 2023), incorporating indexing, retrieval, and generation steps. It has served as a foundation for later, more advanced retrieval-based

Model	Equal			Before/After			Equal Multi		
	Day	Month	Year	Day	Month	Year	Day	Month	Year
MultiQA	0.446	0.397	0.330	0.614	0.574	0.667	0.154	0.333	0.276
Naive RAG(LLama3-70b)	0.261	0.510	0.536	0.195	0.300	0.556	0.167	0.167	0.444
Naive RAG(GPT-3.5)	0.261	0.388	0.714	0.232	0.550	0.444	0.083	0.000	0.278
Naive RAG(GPT-4)	0.307	0.531	0.571	0.317	0.450	0.444	0.083	0.000	0.278
ReAct RAG Agent(LLama3-70b)	0.318	0.510	0.571	0.293	0.350	0.556	0.250	0.167	0.389
ReAct RAG Agent(GPT-3.5)	0.250	0.388	0.643	0.268	0.400	0.556	0.583	0.000	0.389
ReAct RAG Agent(GPT-4)	0.273	0.612	0.714	0.402	0.550	0.444	0.250	0.333	0.222
TempAgent(LLama3-70b)	0.875	0.918	0.875	0.488	0.800	0.667	0.167	0.000	0.444
TempAgent(GPT-3.5)	0.875	0.898	0.821	0.500	0.750	0.667	0.167	0.167	0.444
TempAgent(GPT-4)	**0.989**	**0.939**	**0.893**	**0.720**	**0.900**	**0.889**	**0.750**	**0.833**	**0.611**

Table 5: Experiment results of multi-granular time.

methods.

ReAct (Yao et al., 2022): A straightforward yet powerful approach enabling collaborative reasoning and decision-making in LLMs by maintaining interpretable step-by-step traces. It achieves strong performance in multi-hop QA and interactive decision tasks.

Implementation Details. Following prior work (Li et al., 2024a; Sun et al., 2024b; Jiang et al., 2023; Liu et al., 2024), we employ Hits@1 to measure the proportion of queries whose top-ranked candidate answer matches the ground truth. Because LLMs do not produce output probabilities, we rely on an exact match approach, effectively mirroring the Hits@1 metric used in earlier studies.

We evaluate our framework on two proprietary models and one open-source model: gpt-3.5-turbo[1], gpt-4-turbo-2024-04-09[2] and llama-3-70b-instruct[3].

4.2 Experimental results

Tables 3, 4, and 5 present the experimental results on both the MultiTQ and CronQuestions datasets, with the highest scores in **bold**. In Table 3, our approach outperforms Naive RAG and ReAct RAG Agent under all tested models, validating the advantages of TempAgent's design. Notably, **TempAgent** surpasses the baseline methods by a substantial margin of 41.3% when integrated with GPT-4. Although LLaMA3-70b delivers performance slightly above GPT-3.5, it remains below GPT-4.

This outcome underscores the versatility of TempAgent: by seamlessly guiding the model's chain-of-thought and incorporating specialized tools, it exploits the inherent strengths of the underlying LLM to achieve robust performance across different scenarios. For single-answer questions, TempAgent registers nearly a 50% improvement over MultiQA, and on multiple-answer questions, it demonstrates roughly triple the effectiveness of MultiQA. As a time-aware agent, TempAgent excels in time-specific queries, reaching an accuracy of 87%.

Additionally, we performed a comprehensive examination of various question types, as illustrated in Figure 2. Among them, *equal* questions are handled most effectively, achieving nearly 100% accuracy. The *First/Last*, *Before/After*, and *Equal multi* question types show broadly comparable performance, although *First/Last* questions exhibit a marginally better success rate. Conversely, questions with multiple constraints, specifically *Before last* and *After first*, pose significant challenges, scoring far below *Equal multi* queries. These results imply that TempAgent can interpret both implicit and explicit temporal information yet struggles with the heightened multi-hop reasoning demanded by *Before last* and *After first*.

To evaluate the agent's capacity for multi-granularity temporal reasoning, we conducted experiments involving queries that span multiple time granularities. Table 5 reports these results. Our model demonstrates robust performance across different temporal granularities yet underperforms on *Equal multi* questions at the yearly granularity. Two key factors contribute to this shortfall: (1) the additional complexity arising from multiple temporal constraints and (2) the daily resolution of time in the knowledge base, which complicates the extraction of coarse-grained yearly information. Interestingly, TempAgent attains notably high Hits@1 on *Equal multi* questions at both day and month gran-

[1]https://platform.openai.com/docs/models/gpt-3-5-turbo
[2]https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4
[3]https://ollama.com/library/llama3:70b-instruct

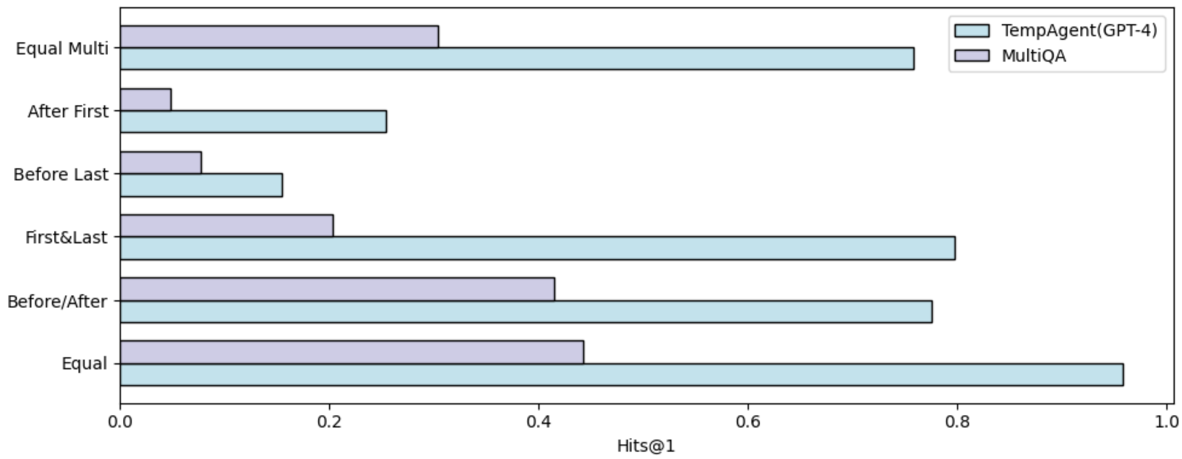

Figure 2: Performance (Hits@1) of TempAgent and MultiQA on the MultiTQ dataset under different question types.

Model	**Hits@1**				
	Overall	Question Type		Answer Type	
		Single	Multiple	Entity	Time
w/o toolbox	0.398	0.516	0.106	0.243	0.734
w/o multiprompt	0.355	0.396	0.087	0.337	0.395
TempAgent	0.539	0.684	0.168	0.478	0.661
k=3	0.479	0.627	0.112	0.381	0.689
k=5	0.509	0.667	0.118	0.415	0.712
k=10	0.539	0.684	0.168	0.478	0.661
k=15	0.529	0.681	0.149	0.449	0.644

Table 6: Overall results of our ablation experiments with different values of k. All experiments are conducted using *gpt-3.5-turbo*.

ularities, showing a substantial improvement over the baseline. This underscores the model's strong temporal reasoning in these particular contexts.

4.3 Ablation Study

To further clarify the contributions of each TempAgent component, we carried out an ablation study (see Table 6). Specifically, we tested different values of k (3, 5, 10, 15) to identify the optimal number of retrieved tuples. Our findings suggest that performance improves as k increases from 3 to 10, before dipping at $k = 15$, likely due to excessive noise and resultant hallucinations. Consequently, $k = 10$ is selected for all other experiments.

We also evaluated the necessity of two key components: Toolbox. Removing the time-filtering toolbox (*w/o toolbox*) led to a marked drop in performance, indicating that time-constrained retrieval mitigates irrelevant noise and reduces hallucinations. Multi-Prompting. Eliminating the specialized prompts for distinct question types (*w/o multiprompt*) also degraded results, confirming that targeted question-specific prompts reinforce temporal reasoning.

5 Conclusion

We introduced *TempAgent*, an LLM-based framework dedicated to multi-granularity TKGQA. By integrating time-aware retrieval with robust reasoning strategies, TempAgent addresses a key shortcoming in existing ReAct-style approaches: the inability to effectively filter knowledge bases by multi-granularity temporal constraints. Such an omission often leads to excessive or irrelevant information, amplifying the risk of LLM hallucinations. Our method bridges this gap through a novel retrieval mechanism that diligently enforces temporal filtering, thereby substantially reducing hallucination rates and bolstering efficiency.

6 Limitation

Although large language models exhibit remarkable capabilities, they are not impervious to inaccuracies or biases. Manual verification remains important for improving overall evaluation fidelity. Our experiments show that retrieving the top 10 chunks offers the best balance between coverage and noise. Nonetheless, potential information gaps remain, and we are exploring more sophisticated retrieval methods to optimize the volume and quality of retrieved data without exacerbating hallucinations. While iterative retrieval partly mitigates information gaps, it is not a comprehensive solution. Additionally, the agent's performance is predominantly influenced by the underlying LLM's capacity; smaller models may struggle to attain comparable outcomes.

7 Acknowledgement

This work was supported by the National Natural Science Foundation of China (62472192). Authors are grateful to the anonymous reviewers for helpful comments.

References

Zahra Abbasiantaeb, Yifei Yuan, Evangelos Kanoulas, and Mohammad Aliannejadi. 2024. Let the llms talk: Simulating human-to-human conversational qa via zero-shot llm-to-llm interactions. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, page 8–17, New York, NY, USA. Association for Computing Machinery.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024a. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Zhuo Chen, Zhao Zhang, Zixuan Li, Fei Wang, Yutao Zeng, Xiaolong Jin, and Yongjun Xu. 2024b. Self-improvement programming for temporal knowledge graph question answering. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14579–14594.

Ziyang Chen, Dongfang Li, Xiang Zhao, Baotian Hu, and Min Zhang. 2024c. Temporal knowledge question answering via abstract reasoning induction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4872–4889, Bangkok, Thailand. Association for Computational Linguistics.

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.

Wentao Ding, Hao Chen, Huayu Li, and Yuzhong Qu. 2022. Semantic framework based query generation for temporal question answering over knowledge graphs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1867–1877.

Yujuan Ding, Wenqi Fan, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meets llms: Towards retrieval-augmented large language models. *CoRR*, abs/2405.06211.

Chenyang Du, Xiaoge Li, and Zhongyang Li. 2024. Semantic-enhanced reasoning question answering over temporal knowledge graphs. *Journal of Intelligent Information Systems*, pages 1–23.

Simon Gottschalk and Elena Demidova. 2018. Eventkg: A multilingual event-centric temporal knowledge graph. In *The Semantic Web*, pages 272–287, Cham. Springer International Publishing.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *61st Annual Meeting of the Association for Computational Linguistics, ACL 2023*, pages 1049–1065. Association for Computational Linguistics (ACL).

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018a. Tempquestions: A benchmark for temporal question answering. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 1057–1062, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018b. Tequila: Temporal question answering over knowledge bases. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1807–1810.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. In *International Conference on Learning Representations*.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2024a. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Zhidong Li, Licai Wang, Qibin Luo, and Silong Qiao. 2024b. Large language model based on full-text retrieval for temporal knowledge q&a approach. In *2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, pages 441–446. IEEE.

Ke Liang, Yue Liu, Sihang Zhou, Wenxuan Tu, Yi Wen, Xihong Yang, Xiangjun Dong, and Xinwang Liu.

2023. Knowledge graph contrastive learning based on relation-symmetrical structure. *IEEE Transactions on Knowledge and Data Engineering*.

Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. 2024a. Data-efficient fine-tuning for llm-based recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 365–374.

Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. 2024b. Data-efficient fine-tuning for llm-based recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 365–374, New York, NY, USA. Association for Computing Machinery.

Jingping Liu, Xianyang Tian, Hanwen Tong, Chenhao Xie, Tong Ruan, Lin Cong, Baohua Wu, and Haofen Wang. 2024. Enhancing chinese abbreviation prediction with llm generation and contrastive evaluation. *Information Processing & Management*, 61(4):103768.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315.

Costas Mavromatis, Prasanna Lakkur Subramanyam, Vassilis N Ioannidis, Adesoji Adeshina, Phillip R Howard, Tetiana Grinberg, Nagib Hakim, and George Karypis. 2022. Tempoqr: Temporal question reasoning over knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 5825–5833.

Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. 2021. Question answering over temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.

Chao Shang, Guangtao Wang, Peng Qi, and Jing Huang. 2022. Improving time sensitivity for question answering over temporal knowledge graphs. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8017–8026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Hongda Sun, Weikai Xu, Wei Liu, Jian Luan, Bin Wang, Shuo Shang, Ji-Rong Wen, and Rui Yan. 2024a. Determlr: Augmenting llm-based logical reasoning from indeterminacy to determinacy. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9828–9862.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024b. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):1–26.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

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*.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. 2024. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36.

A Appendix

A.1 Prompt for answer evaluation

Table 7 presents our carefully designed prompt for answer evaluation. The LLM first offers a preliminary judgment on whether the generated answer matches the reference, followed by a structured chain-of-thought that validates the initial assessment. Concluding the process, the LLM issues a definitive verdict (*correct* or *incorrect*). Notably, this stringent evaluation paradigm can result in conservative grading—ambiguous or partially correct answers often receive an *incorrect* label. As a consequence, the final accuracy evaluated by the LLM tends to be slightly lower than that from human annotators, though it also tends to be more rigorous.

A.2 Case study

To illustrate how TempAgent's runtime differs from prior approaches, we performed a case study comparing Naive RAG (Chen et al., 2024a), ReAct (Yao et al., 2022), and our method. Figure 3(a) shows that Naive RAG relies solely on keyword searches; it fails to locate the necessary knowledge

a) Naive RAG

Question: Which country hosted the first visit of Richard Boucher after Iraq?
> Entering new AgentExecutor chain...
Thought: To find out which country hosted Richard Boucher after his visit to Iraq, I need to search for information about his travel itinerary or diplomatic visits around the time he was involved in Iraq.
Action: Search
Action Input: "Richard Boucher first visit after Iraq"
Observation:['Richard_Boucher\tMake_a_visit\tIraq\t2008-04-02', 'Richard_Boucher\tMake_statement\tIraq\t2005-04-08', 'Iraq\tHost_a_visit\tRichard_Boucher\t2008-04-02', 'Iraq\tHost_a_visit\tStephen_Hadley\t2006-10-30', 'Iraq\tHost_a_visit\tStephen_Hadley\t2006-11-13','Iraq\tHost_a_visit\tStephen_Hadley\t2006-11-29', 'Iraq\tHost_a_visit\tStephen_Hadley\t2008-07-18','Iraq\tHost_a_visit\tRobert_L._Ehrlich\t2010-02-28', 'Iraq\tHost_a_visit\tChuck_Hagel\t2014-12-13', 'Iraq\tHost_a_visit\tStephen_Hadley\t2006-11-01']
Final Answer: I am unable to determine which country hosted Richard Boucher first after his visit to Iraq in 2008 based on the available information.

b) React RAG

Question: Which country hosted the first visit of Richard Boucher after Iraq?
> Entering new AgentExecutor chain...
Thought: To find out which country hosted Richard Boucher after his visit to Iraq, I need to search for information about his travel itinerary or diplomatic visits around the time he was involved in Iraq.
Action: Search
Action Input: "Richard Boucher first visit after Iraq"
Observation:['Richard_Boucher\tMake_a_visit\tIraq\t2008-04-02', 'Richard_Boucher\tMake_statement\tIraq\t2005-04-08', 'Iraq\tHost_a_visit\tRichard_Boucher\t2008-04-02', 'Iraq\tHost_a_visit\tStephen_Hadley\t2006-10-30', 'Iraq\tHost_a_visit\tStephen_Hadley\t2006-11-13', 'Iraq\tHost_a_visit\tStephen_Hadley\t2006-11-29', 'Iraq\tHost_a_visit\tStephen_Hadley\t2008-07-18', 'Iraq\tHost_a_visit\tRobert_L._Ehrlich\t2010-02-28', 'Iraq\tHost_a_visit\tChuck_Hagel\t2014-12-13', 'Iraq\tHost_a_visit\tStephen_Hadley\t2006-11-01']
Thought: The search results show multiple entries related to Richard Boucher's visit to Iraq, but none of them specify the first country he visited after Iraq. I need to refine my search to find more specific information about his subsequent travel after leaving Iraq.
Action: Search
Action Input: "Richard Boucher next country visited after Iraq April 2008"
Observation:['Iraq\tHost_a_visit\tRichard_Boucher\t2008-04-02', 'Richard_Boucher\tMake_statement\tIraq\t2005-04-08', 'Richard_Boucher\tMake_a_visit\tIraq\t2008-04-02', 'Iraq\tHost_a_visit\tStephen_Hadley\t2008-07-18', 'Iraq\tHost_a_visit\tStephen_Hadley\t2006-10-30', 'Iraq\tHost_a_visit\tStephen_Hadley\t2006-11-13', 'Iraq\tHost_a_visit\tStephen_Hadley\t2006-11-29', 'Iraq\tHost_a_visit\tRobert_L._Ehrlich\t2010-02-28', 'Iraq\tHost_a_visit\tStephen_Hadley\t2006-11-01', 'Iraq\tHost_a_visit\tChancellor_of_the_Exchequer\t2006-11-18']
Thought: Despite multiple search attempts, I have not been able to find specific information about the first country Richard Boucher visited after Iraq in April 2008. The search results repeatedly show his visits to Iraq but do not provide details about his subsequent travel destinations.
Final Answer: I am unable to determine which country hosted Richard Boucher first after his visit to Iraq in 2008 based on the available information.

c) TempAgent

Question: Which country hosted the first visit of Richard Boucher after Iraq?
> Entering new AgentExecutor chain...
Thought: To find the country that hosted the first visit of Richard Boucher after his visit to Iraq, I need to first determine the date of his visit to Iraq.
Action: Search
Action Input: ('Richard Boucher visit Iraq')
Observation:[{'text': 'Richard_Boucher Make_a_visit Iraq', 'date': '2008-04-02'}, {'text': 'Richard_Boucher Make_statement Iraq', 'date': '2005-04-08'}, {'text': 'Iraq Host_a_visit Richard_Boucher', 'date': '2008-04-02'}, {'text': 'Richard_Boucher Make_statement Sudan', 'date': '2005-02-25'}, {'text': 'Richard_Boucher Express_intent_to_meet_or_negotiate Sudan', 'date': '2005-01-06'}, {'text': 'Richard_Boucher Express_intent_to_meet_or_negotiate Sudan', 'date': '2005-01-07'}, {'text': 'Richard_Boucher Make_a_visit Iran', 'date': '2007-06-14'}, {'text': 'Richard_Boucher Consult Foreign_Affairs_(France)', 'date': '2007-03-02'}, {'text': 'Iraq Host_a_visit Stephen_Hadley', 'date': '2006-11-03'}, {'text': 'Iraq Host_a_visit Stephen_Hadley', 'date': '2006-10-30'}
]Richard Boucher visited Iraq on 2008-04-02. Now, I need to find the first visit he made to another country after this date.
Action: Search_after_Day
Action Input: ('Richard Boucher visit', '2008-04-02')
Observation:[{'text': 'Richard_Boucher Make_a_visit France', 'date': '2008-10-24'}, {'text': 'France Host_a_visit Richard_Boucher', 'date': '2008-10-24'}, {'text': "Richard_Boucher Meet_at_a_'third'_location Foued_Mebazaa", 'date': '2011-09-23'}, {'text': 'Richard_Boucher Express_intent_to_meet_or_negotiate Prachanda', 'date': '2008-12-03'}, {'text': 'Richard_Boucher Consult Gurbanguly_Berdymukhammedov', 'date': '2008-05-30'}, {'text': 'Lawyer/Attorney_(United_States) Make_statement Richard_Boucher', 'date': '2010-05-27'}, {'text': 'Lawyer/Attorney_(United_States) Make_statement Richard_Boucher', 'date': '2009-08-05'}, {'text': 'Richard_Boucher Engage_in_negotiation Rehman_Malik', 'date': '2008-10-18'}, {'text': 'Richard_Boucher Express_intent_to_cooperate Rehman_Malik', 'date': '2008-10-18'}, {'text': 'Richard_Boucher Consult Head_of_Government_(Sri_Lanka)', 'date': '2008-09-24'}]
Thought: The first visit of Richard Boucher after his visit to Iraq on 2008-04-02 was to France on 2008-10-24.
Final Answer: France hosted the first visit of Richard Boucher after Iraq.

Figure 3: Comparison of four methods on TKGQA: (a) Naive RAG (Chen et al., 2024a; Ma et al., 2023), (b) ReAct (Yao et al., 2022), and (c) TempAgent. Incorrect finishes are shown in red; correct ones are in green. Orange highlights indicate retrieved content that does not conform to the question's time constraints, leading to potential hallucinations.

Prompt of Answer Evaluation
You are assessing a submitted answer to a question relative to the true answer based on the provided criteria: QUESTION: {question} ANSWER: {answer} TRUE ANSWER: {correct_answer} **Criteria:** **Relevance:** Is the submission referring to a real quote from the text? **Conciseness:** Is the answer concise and to the point? **Correct:** Is the answer correct?
Does the submission meet the criterion? First, write out in a step-by-step manner your reasoning about the criterion to be sure that your conclusion is correct. Avoid simply stating the correct answers at the outset. Then print the **CORRECT** or **INCORRECT** (without quotes or punctuation) on its own line corresponding to the correct answer. Reasoning:

Table 7: This prompt describes the entire process of model evaluation, where {question} represents the question, {correct_answer} represents the correct answer, and {answer} represents the answer generated by the model.

and produces an incorrect final answer. ReAct (Figure 3(b)) attempts a second retrieval step by updating the query with partial knowledge gleaned from the first search, but also fails to identify the crucial information. By contrast, TempAgent (Figure 3(c)) deconstructs the query into two subtasks: (1) identify the date Richard Boucher visited Iraq, and (2) isolate the first country he visited thereafter, applying temporal filtering appropriately. This case underscores TempAgent's ability to resolve queries that stymie both Naive RAG and ReAct, revealing how multi-step temporal reasoning and time-aware retrieval yield a more effective TKGQA workflow.