Two-stage Generative Question Answering on Temporal Knowledge Graph Using Large Language Models

Yifu Gao[1], **Linbo Qiao**[1*], **Zhigang Kan**[1], **Zhihua Wen**[1], **Yongquan He**[3], **Dongsheng Li**[1,2*]
[1] National Key Laboratory of Parallel and Distributed Computing,
National University of Defense Technology, Changsha, China
[2] Xiangjiang Laboratory, Changsha, China
[3] Meituan, Beijing, China
{gaoyifu, qiao.linbo, kanzhigang13, zhwen, dsli}@nudt.edu.cn
heyongquan@meituan.com

Abstract

Temporal knowledge graph question answering (TKGQA) poses a significant challenge task, due to the temporal constraints hidden in questions and the answers sought from dynamic structured knowledge. Although large language models (LLMs) have made considerable progress in their reasoning ability over structured data, their application to the TKGQA task is a relatively unexplored area. This paper first proposes a novel **gen**erative **t**emporal **k**nowledge **g**raph **q**uestion **a**nswering framework, GenTKGQA, which guides LLMs to answer temporal questions through two phases: Subgraph Retrieval and Answer Generation. First, we exploit LLM's intrinsic knowledge to mine temporal constraints and structural links in the questions without extra training, thus narrowing down the subgraph search space in both temporal and structural dimensions. Next, we design virtual knowledge indicators to fuse the graph neural network signals of the subgraph and the text representations of the LLM in a non-shallow way, which helps the open-source LLM deeply understand the temporal order and structural dependencies among the retrieved facts through instruction tuning. Experimental results on two widely used datasets demonstrate the superiority of our model.

1 Introduction

Real-world knowledge is frequently updated rather than static (Erxleben et al., 2014; Boschee et al., 2015), e.g., (*Obama*, *hold_position*, *President*) is merely valid only for a certain period [*2009*, *2016*]. Hence, the temporal knowledge graph (TKG) is proposed as a database for storing dynamic structured facts associated with timestamps, denoted as (*subject*, *relation*, *object*, *timestamp*). Temporal knowledge graph question answering (TKGQA) aims to answer a natural question with explicit or implicit temporal constraints based on the TKG, e.g., "*Who held the position of president (in 2017) or (after Obama)?*". Due to the temporal constraints hidden in questions and the answers sought from dynamic structured knowledge, TKGQA is one of the most challenging QA tasks.

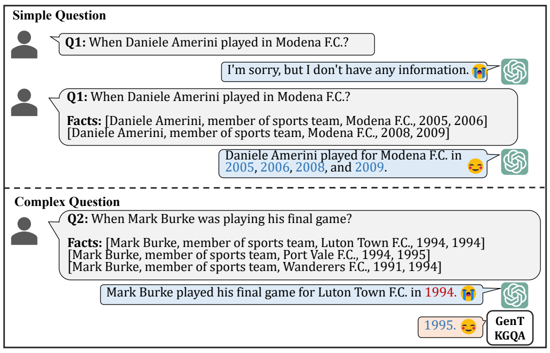

Simple Question
Q1: When Daniele Amerini played in Modena F.C.?
I'm sorry, but I don't have any information.
Q1: When Daniele Amerini played in Modena F.C.?
Facts: [Daniele Amerini, member of sports team, Modena F.C., 2005, 2006]
[Daniele Amerini, member of sports team, Modena F.C., 2008, 2009]
Daniele Amerini played for Modena F.C. in 2005, 2006, 2008, and 2009.
Complex Question
Q2: When Mark Burke was playing his final game?
Facts: [Mark Burke, member of sports team, Luton Town F.C., 1994, 1994]
[Mark Burke, member of sports team, Port Vale F.C., 1994, 1995]
[Mark Burke, member of sports team, Wanderers F.C., 1991, 1994]
Mark Burke played his final game for Luton Town F.C. in 1994.
1995.
GenT KGQA

Figure 1: Examples of the responses of LLM and GenTKGQA to the simple and complex temporal questions.

Recently, large language models (LLMs) have shown strong competitiveness in various fields (Fei et al., 2023; Ye et al., 2023). Some researchers explore the reasoning capability of LLMs for structured knowledge based on KGQA tasks (Baek et al., 2023; Kim et al., 2023), and some works examine the temporal reasoning capabilities of LLMs through time-sensitive QA tasks (Chen et al., 2021; Tan et al., 2023). Intuitively, LLMs have the ability to deal with temporal structured knowledge. Based on the above findings, we attempt to utilize LLMs for the TKGQA task and summarize the following two challenges: 1) ***Question-relevant Subgraph Retrieval.*** A common practice to enhance the LLM's domain-specific reasoning capability is to input query-relevant information as additional knowledge into the LLMs (Hu et al., 2023). As shown in Figure 1, ChatGPT [1] cannot answer temporal questions directly, but it can

[*] Corresponding Author

[1] https://openai.com/blog/chatgpt

Findings of the Association for Computational Linguistics: ACL 2024, pages 6719–6734
August 11-16, 2024 ©2024 Association for Computational Linguistics

answer simple questions when the relevant facts are provided. However, finding facts relevant to the problem is a struggle due to the large search space with both structural and temporal dimensions. For example, for the complex temporal question "*Who held the position of president after Obama?*", the structural link *hold position* between the entities and the temporal constraints *after 2016* in the problem are unknown, so directly finding all relevant facts about the given entities *Obama* and *President* is bound to introduce too much noise information. How to accurately retrieve relevant facts from a two-dimensional space is the first challenge. 2) ***Complex-type Question Reasoning.*** Recent works about LLM-based KG reasoning mostly input structured knowledge in the natural text form into the task prompt and reason about the answers in a training-free manner (Yang et al., 2024). However, these approaches fuse subgraph information with the LLM in a shallow way, which limits the inference performance on the complex question type. As illustrated in Figure 1, ChatGPT cannot understand the chronological order of the relevant facts and answers incorrectly on the complex question type "final". How subgraph information can be integrated into LLM representations in a non-superficial way to simulate structured reasoning remains an open question.

Hence, we propose GenTKGQA, a novel generative temporal knowledge graph question answering framework consisting of two phases, subgraph retrieval and answer generation, which is used to address the above two challenges, respectively. At the first phase, we find that the structural and temporal scope of the subgraph is determined by the relation links and the temporal constraints in the question, respectively. Therefore, we use a divide-and-conquer strategy to reduce the subgraph search space by decomposing the complex subgraph retrieval problem into two subtasks, namely, relation ranking and time mining. Then, we utilize the LLM's internal knowledge to mine structural connections between entities and time constraints in the problem without extra training. We only need to input few-shot examples into the prompt to accomplish subgraph retrieval of the entire data. At the second phase, we fine-tune the open-source LLM with instruction tuning to incorporate structural and temporal information of the subgraph in a non-shallow way. Recent works illustrate that fusing graph neural network (GNN) representations and language text representations can enhance the ability of LMs to perceive graph structure (Zhang et al., 2022b). Thus, we design three novel virtual knowledge indicators to bridge the links between pre-trained GNN signals of the temporal subgraph and text representations of the LLM, which guides the LLMs in deeply understanding the graph structure and improves their reasoning ability for complex temporal questions. Overall, our contribution can be summarized in the following four points:

1) We present a novel two-stage generative framework for the TKGQA task, which explores LLM's temporal reasoning capabilities in the context of dynamic structured knowledge.

2) We motivate the LLM's intrinsic knowledge to mine the temporal constraints and structural connections in the questions without extra training, which reduces the subgraph search space from both structural and temporal dimensions.

3) We design virtual knowledge indicators to fuse the GNN signals and text representations in a non-shallow way, which helps the open-source LLMs improve their reasoning on the complex question type through instruction tuning.

4) Experiment results on two widely used datasets show that GenTKGQA as a generative model performs consistently better than embedding-based methods on the Hits@1 metric.

2 Related Work

2.1 TKGQA Methods

Temporal knowledge graph question answering (TKGQA) task aims to answer complex questions in the natural language format using entities and timestamps from the given TKG (Jia et al., 2018, 2021; Chen et al., 2023b). Existing mainstream methods employ TKG embeddings to represent the entities, relations and timestamps, and use the scoring function to select the entity or time with the highest relevance as the answer (Saxena et al., 2021). However, single embedding methods have difficulty handling complex reasoning problems with implicit time constraints. Therefore, recent methods try to incorporate other modules to improve the model performance on complex problems. Specifically, TSQA (Shang et al., 2022) proposes a contrastive approach to enhance the model's time sensitivity. TempoQR (Mavromatis et al., 2022) designs three modules, namely context, entity and time-aware information, to enhance the incorporation of the TKG into questions. Besides, some approaches propose to solve the TKGQA task with problem-relevant subgraph reasoning (Chen et al., 2022; Liu et al., 2023b; Sharma et al., 2023). Despite the effectiveness of these approaches, few studies have explored how LLMs can address the TKGQA task.

2.2 LMs for Temporal Question Answering

Language models (LMs) have exhibited strong performance on the question answering task (Raffel et al., 2020). In recent years, some researchers have explored the temporal reasoning capabilities of LMs and propose several typical time-sensitive QA datasets. They focus on temporal question answering either within a closed-book setting to assess models' internal memorization of temporal facts (Liska et al., 2022; Dhingra et al., 2022), or within an open-book setting to evaluate models' temporal understanding and reasoning capability over unstructured texts (Zhang and Choi, 2021; Chen et al., 2021; Tan et al., 2023). In the context of the latter setting, some works propose to use the graph structure extracted from text to assist the model in determining the temporal order between events (Mathur et al., 2022; Su et al., 2023; Yang et al., 2023; Xiong et al., 2024), which is similar but fundamentally different from our work. These approaches aim to answer temporal questions based on the known natural text context. In contrast, our model focuses on structured temporal knowledge as auxiliary information that needs to be retrieved by the model.

2.3 LMs for KG Question Answering

How to combine LMs and KG for question answering has become a hot issue. Some works attempt to enhance question representation and relation matching with PLMs in the multi-hop KGQA task (Saxena et al., 2020; Zhang et al., 2022a; Jiang et al., 2023b), but there is no interaction between the LM and KG representations. Other works try to use one modality to ground the other, i.e., using the encoded representation of a linked KG to augment the text representation (Lin et al., 2019; Yang et al., 2019), or using the text representation of the PLM to enhance the graph reasoning model (Feng et al., 2020a). The most recent approaches enable deep integration of the two modalities by jointly updating the GNN and LM representations (Yasunaga et al., 2021; Zhang et al., 2022b).

However, the emergence of large language models (LLMs) has changed how LMs handle the KGQA task, which is divided into two main approaches: training-free and fine-tuning (Yang et al., 2024). Recent works attempt to append query-relevant facts as the input prompt for LLMs and make inferences without extra training (Baek et al., 2023; Wu et al., 2023; Jiang et al., 2023a; Kim et al., 2023; Li et al., 2024). Fine-tuning the full parameters of the LLM can be cost-prohibitive. Hence, KPE (Zhao et al., 2023) enables knowledge integration by freezing PLM parameters and introducing trainable parameter adapters. ChatKBQA (Luo et al., 2023) employs the LoRA (Hu et al., 2022) technique to fine-tune open-source LLMs, achieving the logical query form generation. Besides, KoPA (Zhang et al., 2023) incorporates the KG embeddings into the LLMs with a prefix adapter, aiming to achieve structrual-aware reasoning in the LLMs. Applying LLMs to the temporal KGQA task remains an unexplored area.

3 Preliminaries

TKGQA. A temporal knowledge graph (TKG) $\mathcal{G} := (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$ is a multi-relational, directed graph with timestamped edges between entities, where $\mathcal{E}, \mathcal{R}$ and $\mathcal{T}$ represent the sets of entities, relations and timestamps, respectively. Each fact in the $\mathcal{G}$ can be represented as a quadruple $(s, r, o, t) \in \mathcal{F}$, corresponding to entity $s/o \in \mathcal{E}$, relation type $r \in \mathcal{R}$ and timestamp $t \in \mathcal{T}$. Given a natural language question q, TKGQA aims to extract entities s/o or timestamps t that correctly answer the question q.

ICL and IT. Applying LLM to the TKGQA task, the goal is to generate the answer $\mathcal{A}$ based on the input text sequence $\mathcal{S}$ and the LLM $\mathcal{M}$. $\mathcal{S}$ consists of several parts: the instruction prompt $\mathcal{I}$, the task-specific input prompt $\mathcal{Q}$, and the auxiliary demonstration prompt $\mathcal{D}$. In-context Learning (ICL) method is an efficient approach to employ LLMs to solve downstream tasks without extra training, the input sequence of ICL can be denoted as $\mathcal{S} = \mathcal{I} : \mathcal{D} : \mathcal{Q}$, where : means to concatenate the different prompts. Meanwhile instruction tuning (IT) aims to fine-tune LLMs to follow human instructions and accomplish the distinct tasks in the instruction prompt, the input sequence of IT can be denoted as $\mathcal{S} = \mathcal{I} : \mathcal{Q} : \mathcal{A}$.

4 Method

We apply the LLMs processing TKGQA task in a two-phase process, i.e., an ICL-based subgraph retrieval phase and an IT-based answer generation phase. At the first stage, we utilize internal knowledge of LLM for unlabeled subgraph retrieval. At the second stage, we incorporate external knowledge for structure-aware temporal inference.

4.1 Subgraph Retrieval

We split the complex subgraph retrieval problem into a relation ranking subtask in the structural dimension and a time mining subtask in the temporal dimension. With such a strategy, we only need to provide a small number of samples to complete the subgraph retrieval.

4.1.1 Relation Ranking

We aim to determine the structural scope of the subgraph $\mathcal{G}_{sub,i}$ by retrieving the corresponding relations from the candidate relation set $\mathcal{R}_i$ for each question q_i. Recent work has shown that LLMs can better handle the information extraction task as reranking agents (Sun et al., 2023). Therefore, we feed the question q_i and the candidate set $\mathcal{R}_i$ to the LLM to obtain the top k relations $\mathcal{R}_{i,k}$ relevant to the question. Relations in $\mathcal{R}_i$ are linearized, i.e., [employer, member_of_sports_team, ...,]), and the retrievd relations can bridge the entities identified within the questions. The specific relation ranking prompt is shown in Appendix C.

4.1.2 Time Mining

We find that natural questions contain temporal constraints, either explicit or implicit, and we can easily determine the range of relevant facts in the temporal dimension by using explicit temporal constraints such as "in 2008", "at the year of 2012", etc. How to capture implicit temporal constraints is the key to improving the efficiency of searching for relevant facts. For example, for the complex temporal question "*Who held the position of president after Obama?*", the implicit temporal restriction is known to be (*after 2016*) based on the temporal validity (*2009*, *2016*) of the fact (*Obama*, *hold_position*, *President*). We design specific prompt templates based on different answer types as well as consider the temporal validity of the facts in the questions (given entities and relations matched in the Section 4.1.1) to get the temporal constraints. The details of the templates are shown in Appendix C.

Through the above process, we narrow down the search space of subgraphs and use relevant facts under structural and temporal constraints as additional knowledge to assist the LLM inference.

4.2 Answer Generation

In this section, we will discuss how to incorporate the knowledge retrieved in the previous section 4.1 into the LLM. Previous fundamental approaches to incorporate KG structural information focus on adding the knowledge to the input prompt in the text form, i.e., (subject, relation, object). However, incorporating query-relevant facts into LLMs in text form is not a good choice. Because such shallow interactions do not enable the model to understand the structural dependencies and temporal order between facts, leading to weak temporal reasoning in the complex problem type.

Recent works show that language models can enhance their ability to perceive graph structures by incorporating knowledge representations expressed by graph neural networks (GNN) (Zhang et al., 2022b). Inspired by this, we first extract the structural and temporal information of entities and relations with pre-trained temporal GNN embeddings. Then, we bridge the links between GNN and text representations through pre-designed virtual knowledge indicators. At last, we fine-tune the open-source LLM to deeply understand the temporal order and structural dependencies of the retrieved query-relevant facts.

4.2.1 Temporal GNN

Given the retrieved temporal subgraph $\mathcal{G}_{sub,z}$ of question q_z, we first initialise entity, relation and time representations in $\mathcal{G}_{sub,z}$ using the TKG embedding method (Lacroix et al., 2020). Then, to fully explore the structural information among entities and relations of the temporal subgraph, we propose a temporal graph neural network (T-GNN), which is a variant of graph attention networks (Velickovic et al., 2018). The important distinction between them is that T-GNN captures the correlation scores of neighbouring nodes by incorporating temporal embeddings. Therefore, T-GNN computes message $\mathbf{m}_{ij}$ between entities e_i and e_j as follows:

$$\mathbf{m}_{ij} = \mathbf{W}_m(\mathbf{e}_i^{(l-1)} + \mathbf{r}_{ij} + \mathbf{t}_{ij}), \qquad (1)$$

where $\mathbf{e}_i^{(l-1)}$ is the entity representation of e_i at the l-1 layer, $\mathbf{r}_{ij}$ and $\mathbf{t}_{ij}$ are the embeddings of the relation and timestamp connecting e_i and e_j. $\mathbf{W}_m$ is a linear transformation. Next, the node representation $\mathbf{e}_j^{(l)}$ is calculated via message passing between

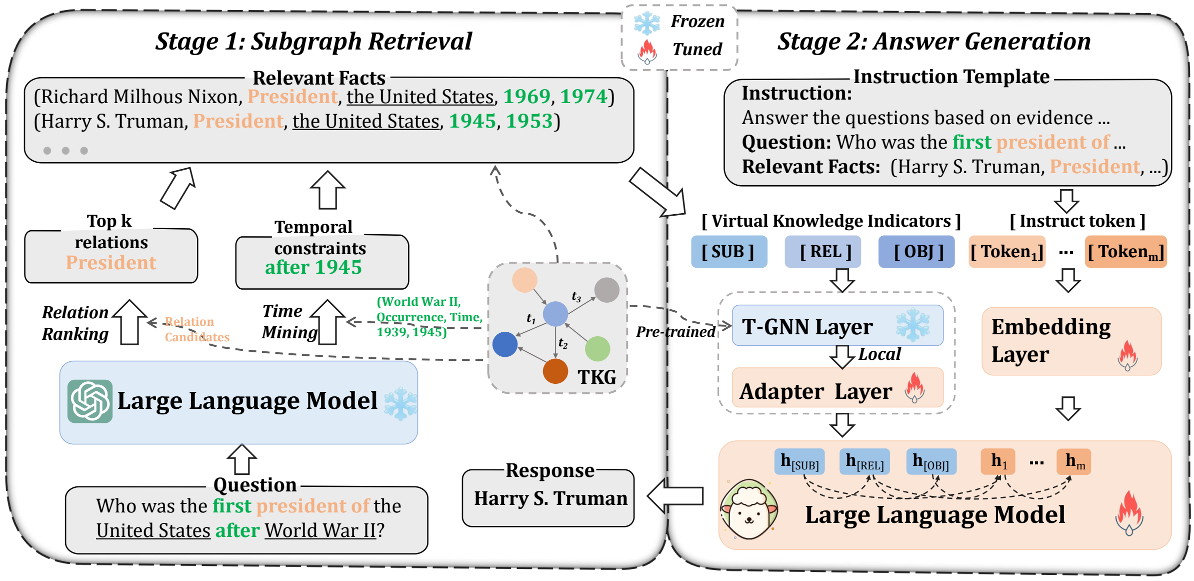


Figure 2: The overall architecture of our proposed GenTKGQA can be divided into two stages, subgraph retrieval and answer generation. Given a temporal question, we mark the entities provided, the implied time constraints and the links between the entities with underline, blue font and orange font, respectively.

neighbors on the $\mathcal{G}_{sub,z}$:

$$\mathbf{e}_j^{(l)} = \sum_{i \in \mathcal{N}_j} \alpha_{ij} \mathbf{m}_{ij}, \tag{2}$$

here $\mathcal{N}_j$ represents the neighbor entities of the arbitrary node e_j, and α_{ij} denotes the attention values with $\mathbf{e}_i^{(l-1)}$ as query and $\mathbf{m}_{ij}$ as key:

$$\alpha_{ij} = \frac{\exp\left(u_{ij}\right)}{\sum_{w \in \mathcal{N}_j} \exp\left(u_{wj}\right)}, \tag{3}$$

$$u_{ij} = f_n((\mathbf{W}_q \mathbf{e}_i^{(l-1)})^\top (\mathbf{W}_k \mathbf{m}_{ij})), \tag{4}$$

$\mathbf{W}_q, \mathbf{W}_k$ are linear transformations, f_n is the RELU activation function. Through the above process, we obtain the entity representation $\mathbf{e}_j$ with subgraph structural and temporal information. Following embedding-based TKGQA methods, we use the link prediction task to pre-train the graph neural network representations. Specifically, for each fact (s, r, o, t) in the TKG, we generate a query $(s, r, [mask], t)$ or $([mask], r, o, t)$ by masking the object or subject entity. Then, we obtain the mask embedding $\mathbf{e}_{[mask]}^{(l)}$ through Eq.(2), and feed it into the multi-layer perceptron (MLP) decoder to maximize the probability of the missing entity o and s through the cross-entropy loss function:

$$p\left(\mathbf{e}\right) = \text{Softmax}(\mathbf{e}_{[mask]}^{(l)} \mathbf{w} + \mathbf{b}), \tag{5}$$

$$\mathcal{L} = -\sum_{(s,r,o,t) \in \mathcal{G}} \log p(o_t) + \log p(s_t). \tag{6}$$

4.2.2 Virtual Knowledge Indicators

We design three knowledge indicators to link graph signals and input prompt text, namely [SUB], [REL] and [OBJ], correspond to the virtual tokens of the head entities, relations and tail entities in the subgraph, respectively. We then try to incorporate structural and temporal information from the subgraph into the indicator representations. Specifically, we use the Local operator to get the structure representations $\mathbf{h}^s$ of entities and relationships in the subgraph, respectively:

$$\mathbf{h}_{[\text{SUB}]}^s = \text{Local}(\mathbf{e}_{[\text{SUB}]}), \tag{7}$$

here $\mathbf{e}_{[\text{SUB}]}$ represents pre-trained T-GNN embeddings of all subject entities in the subgraph, Local indicates the max or mean pooling operator. Besides, we leverage the time embeddings to enhance the indicator representations with temporal information.

$$\mathbf{h}_{[\text{SUB}]}^{st} = \mathbf{h}_{[\text{SUB}]}^s + \mathbf{t}_{min} + \mathbf{t}_{max}, \tag{8}$$

where $\mathbf{t}_{min}$ and $\mathbf{t}_{max}$ denote the embeddings for the minimum and maximum values of time in the subgraph, respectively. The intuition follows BERT that use position embeddings for tokens (Devlin et al., 2019). Here, time embeddings can be seen as entity positions in the time dimension. The relation and object indicators are the same as subject. At last, we employ a simple linear layer $\mathbf{W}_p$ to project them into the textual representation space of the

LLM. The final input prompt sequence $\mathcal{S} = \mathcal{V} : \mathcal{I} : \mathcal{Q} : \mathcal{A}$, $\mathcal{V}$ represent virtual indicator tokens. Details of the instruction template can be found in the Appendix C. The optimization objective of the LLM $\mathcal{M}$ can be formulated as:

$$\mathcal{A} = \arg\max_{\mathcal{A}} P_{\mathcal{M}} \left(\mathcal{A} \mid \mathcal{V}, \mathcal{I}, \mathcal{Q}, \mathcal{A}\right). \qquad (9)$$

5 Experiments

We design experiments to answer the following questions:

Q1. How does GenTKGQA perform on the TKG question answering task? (Section 5.2)

Q2. How do the two stages contribute to the model performance respectively? (Section 5.3)

Q3. How does GenTKGQA perform under changes in hyper-parameters? (Section 5.4)

Q4. How does GenTKGQA outperform ChatGPT in answering complex temporal questions? (Section 5.5)

5.1 Datasets, Metrics and Baselines

CronQuestions (Saxena et al., 2021) is a temporal QA dataset, which contains 410K unique question-answer pairs, including annotated entities and timestamps, with 350k for training and 30k for validation and testing. The dataset can be categorized into simple reasoning (Simple Entity and Simple Time) and complex reasoning (Before/After, First/Last, and Time Join) based on temporal constraints. The TimeQuestions dataset (Sharma et al., 2023) has 13.5k manually edited questions and is divided into three parts: training, validation, and testing, containing 7k, 3.2k, and 3.2k questions, respectively. The questions are categorized into four types: Explicit, Implicit, Temporal and Ordinal. Following previous studies, we use two popular evaluation metrics, Hits@1 and Hits@10. More information about datasets and metrics can be found in Appendix A.

We compare four types of baselines: 1) KG embedding-based models including EaE (Feng et al., 2020b) and EmbedKGQA (Saxena et al., 2020); 2) TKG embedding-based models including CronKGQA (Saxena et al., 2021), EntityQR (Mavromatis et al., 2022), TMA (Liu et al., 2023a), TSQA (Shang et al., 2022) and TempoQR (Mavromatis et al., 2022); and 3) Language models, including BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and ChatGPT. For another TimeQuestions dataset, we use CronKGQA, TempoQR and TwiRGCN (Sharma et al., 2023) for comparison. The implementation details of the baselines are described in Appendix B.

5.2 Main Results

Table 1 reports the performance of all methods on the CronQuestions dataset for various question types. We can observe that GenTKGQA consistently outperforms the baselines in terms of "Overall" performance, and achieves significant improvements of 11.3% in the "Complex" question type and 9.6% in the "Time" answer type on the Hits@1 metric over the second best method. Especially, our model achieves nearly 100% for the "Simple" question type. The possible reason is that simple questions usually involve single facts, GenTKGQA can easily retrieve the relevant facts containing the answer and infer the correct answer through instruction tuning technique.

Furthermore, compared to KG embedding methods, temporal KG embedding methods show significant results on various metrics, thanks to the fact that the temporal information of the TKG is taken into account in the question representation. This is also why KG embedding methods are particularly ineffective for the "Time" answer type. However, most TKG embedding methods treat the QA task as a link prediction, which works for the "Simple" question type containing a single fact compared to the "Complex" question type.

We find that PLMs (BERT, RoBERTa) and LLMs (ChatGPT) have the lowest performance on the TKGQA task without TKG information. This suggests that language models (LM), whether encoded or generated, with a large or small number of parameters, have difficulty answering temporal questions without any relevant context. *w/ tkg* indicates that LMs use entity/time embeddings or relevant facts from the TKG. Obviously, LMs *w/ tkg* have significantly better performance, which suggests that the language models have some degree of temporal reasoning capability when relevant TKG information is provided, validating the importance of the subgraph retrieval phase. It is worth noting that ChatGPT *w/ tkg* still performs weakly in reasoning about complex problem types when providing the facts retrieved in the first stage by GenTKGQA, while our model achieves the best results. This demonstrates the effectiveness of interacting GNN and LM representations in dealing with complex temporal problems. The above findings demonstrate the adequacy of our two motiva-

Model	Hits@1					Hits@10				
	Overall	Question Type		Answer Type		Overall	Question Type		Answer Type	
		Complex	Simple	Entity	Time		Complex	Simple	Entity	Time
EmbedKGQA	0.288	0.286	0.290	0.411	0.057	0.672	0.632	0.725	0.850	0.341
EaE	0.288	0.257	0.329	0.318	0.231	0.678	0.623	0.753	0.668	0.698
CronKGQA	0.647	0.392	0.987	0.699	0.549	0.884	0.802	0.990	0.898	0.857
EntityQR	0.745	0.562	0.990	0.831	0.585	0.944	0.906	0.993	0.962	0.910
TMA	0.784	0.632	0.987	0.792	0.743	0.943	0.904	0.995	0.947	0.936
TSQA	0.831	0.713	0.987	0.829	0.836	0.980	0.968	0.997	**0.981**	0.978
TempoQR	0.918	0.864	0.990	0.926	0.903	0.978	0.967	0.993	0.980	0.974
BERT *w/o tkg*	0.071	0.086	0.052	0.077	0.06	0.213	0.205	0.225	0.192	0.253
RoBERTa *w/o tkg*	0.07	0.086	0.05	0.082	0.048	0.202	0.192	0.215	0.186	0.231
ChatGPT *w/o tkg*	0.151	0.144	0.160	0.134	0.182	0.308	0.308	0.307	0.257	0.402
BERT *w/ tkg*	0.243	0.239	0.249	0.277	0.179	0.620	0.598	0.649	0.628	0.604
RoBERTa *w/ tkg*	0.225	0.217	0.237	0.251	0.177	0.585	0.542	0.644	0.583	0.591
ChatGPT *w/ tkg*	0.754	0.579	0.987	0.689	0.873	0.852	0.746	0.992	0.808	0.933
GenTKGQA	**0.978**	**0.962**	**0.999**	**0.967**	**0.990**	**0.983**	**0.971**	**0.999**	0.974	**0.994**

Table 1: Performance comparison of different models on CronQuestions. The best and second best results are marked in **bold** and underlined, respectively. *w/o tkg* indicates that LMs answer the questions directly without using TKG information, and *w/ tkg* indicates that LMs answer the questions with TKG background knowledge.

Model	Overall	Explicit	Implicit	Temporal	Ordinal
CronKGQA	0.462	0.466	0.445	0.511	0.369
TempoQR	0.416	0.465	0.360	0.400	0.349
TwiRGCN(average)	**0.605**	**0.602**	0.586	0.641	0.518
TwiRGCN(interval)	0.603	0.599	0.603	**0.646**	0.494
GenTKGQA	0.584	0.596	**0.611**	0.563	**0.578**

Table 2: Hits@1 for different models on TimeQuestions.

Model	Hits@1				
	Overall	Question Type		Answer Type	
		Complex	Simple	Entity	Time
GenTKGQA	0.978	0.962	0.999	0.967	0.990
w/o SR	0.119	0.140	0.090	0.127	0.103
w/o SR inference	0.475	0.381	0.601	0.294	0.812
w/ SR random	0.766	0.613	0.970	0.661	0.961
w/o T-GNN	0.935	0.914	0.965	0.920	0.963
w/o VKI	0.843	0.824	0.870	0.831	0.867

Table 3: Ablation study results on CronQuestions.

tions for solving complex temporal problems with LLMs. A possible reason for the poor improvement of our model's Hits@10 metric for the "Entity" type is that the LMs provide irrelevant responses when asked to generate multiple answers.

As shown in Table 2, GenTKGQA still achieves surprising results on the TimeQuestions dataset, especially improving on the "Implicit" and "Ordinal" question types by 1.3% and 11.6%, respectively. Both question types have implied time constraints, similar to the "Complex" question type on the CronQuestions dataset. Overall, compared to the baseline methods, GenTKGQA achieves significant results on the complex question type with different datasets, validating the motivation of our work and the effectiveness of the proposed model. Last but not least, GenTKGQA, as a generative QA model, achieves better results than most traditional extractive QA methods.

5.3 Ablation Study

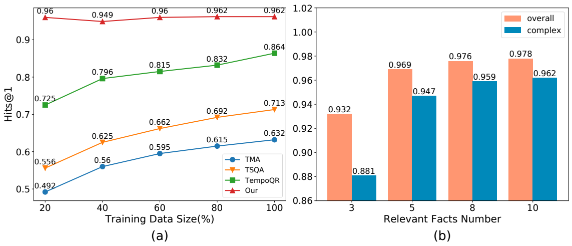
As shown in Table 3, to verify each module's importance, we conduct ablation experiments on the CronQuestions dataset.

w/o SR means that we directly perform problem inference without using relevant subgraph information in the model training and inference phases, while *w/o SR inference* means that we do not provide subgraphs only at the inference. We can observe a sharp decrease in model effectiveness due to the lack of use of subgraph information, which is consistent with the other LMs (Section 5.2). This result shows that current LMs are weak in dealing directly with temporal reasoning problems, validating the importance of the subgraph retrieval module. The *w/o SR inference* result indicates that GenTKGQA remembers part of the structured knowledge during the training phase and improves the temporal inference performance without providing subgraph information. *w/ SR random* denotes the random selection of relevant facts involving entities in the question, and the drop in results proves the validity of our first-stage approach.

w/o T-GNN indicates that we directly use the temporal embeddings (Lacroix et al., 2020) to represent entities and relations of subgraphs. The slight decrease in the results indicates that the T-GNN is able to perceive the structural information

Type	Question/ Retrieved Graph	Response	
		ChatGPT *w/ tkg*	GenTKGQA
Simple Entity	**dean in 1997 was the person?** [Xavier Darcos, position held, dean, 1995, 1998] [Zinaida Belykh, position held, dean, 1988, 1998] [José Miguel Pérez García, position held, dean, 1990, 1998] [Jiří Zlatuška, position held, dean, 1994, 1998] [Katarzyna Olbrycht, position held, dean, 1981, 1998]	[Katarzyna Olbrycht, José Miguel Pérez García, Jiří Zlatuška, Xavier Darcos, Zinaida Belykh, Andrei Fursenko, Anatoly Torkunov, Alexander Konovalov, Anatoly Vichnyakov, Anatoly Vishnevsky]	[Xavier Darcos, Zinaida Belykh, José Miguel Pérez García, Jiří Zlatuška, Katarzyna Olbrycht, Catalina Enseñat Enseñat, Catalina Enseñat Enseñat, Marcel Berger, Miguel Beltrán Lloris, Miklós Réthelyi]
Simple Time	**When Daniele Amerini played in Modena F.C.?** [Daniele Amerini, member of sports team, Modena F.C., 2005, 2006] [Daniele Amerini, member of sports team, Modena F.C., 2008, 2009]	[2005, 2006, 2008, 2009]	[2005, 2006, 2008, 2009]
Before/ After	**Who held the position of Sociétaire of the Comédie-Française before Catherine Hiegel?** [François Jules, position held, Comédie-Française, 1850, 1894] [Jean Martinelli, position held, Comédie-Française, 1930, 1950] [Yvonne Gaudeau, position held, Comédie-Française, 1950, 1986] [Lise Delamare, position held, Comédie-Française, 1951, 1967]	[Lise Delamare]	[Yvonne Gaudeau]
First/ Last	**When Mark Burke was playing his final game?** [Mark Burke, member of sports team, Luton Town F.C., 1994, 1994] [Mark Burke, member of sports team, Port Vale F.C., 1994, 1995] [Mark Burke, member of sports team, Wanderers F.C., 1991, 1994] [Mark Burke, member of sports team, Darlington F.C., 1990, 1990]	[1994]	[1995]
Time Join	**Who was Iowa Governor in Greater German Reich during the World War II?** [Bourke B. H., position held, Governor of Iowa, 1943, 1945] [George A. W., position held, Governor of Iowa, 1939, 1943] [Robert D. B., position held, Governor of Iowa, 1945, 1949]	[George A. W.]	[George A. W., Bourke B. H.]

Table 4: Comparison of responses to five different question types between our GenTKGQA and ChatGPT *w/ tkg*. Marked in blue is the correct answer.



(a)

(b)

Figure 3: Parameter sensitivity on our GenTKGQA.

of the TKG. *w/o VKI* means that we try to remove the virtual knowledge indicators from the input prompt. Model performance degradation shows that indicators can bridge the gap between distinct representations.

5.4 Sensitivity Analysis

Impact of training data size. We explore the impact of different training data sizes to reason about complex temporal questions. As shown in Figure 3(a), by comparing the Hits@1 metric of several methods for the complex question type, our method consistently outperforms others as the training data expands. In particular, at 20% of the training data, our model outperforms the second best model by 32%, demonstrating that our model has strong inference ability in the case of few-shot samples due to its intrinsic knowledge.

Impact of the number of relevant facts. We report the performance changes on the CronQuestions dataset by varying the number of retrieved facts n in Figure 3(b). It can be seen that the model performs poorly with a small number of relevant facts (n=3), and there is a slight increase in performance at n=10. Fewer facts do not provide sufficient context knowledge, while more facts may introduce noise. Taking this into consideration, we set the hyper-parameter n to 10.

5.5 Qualitative Results

We provide specific examples for each question type to compare the answer results of ChatGPT and ChatGPT *w/ tkg*. Table 4 includes the graphs retrieved by our method, along with the answer results for five different question types.

When providing relevant facts retrieved by GenTKGQA as background knowledge, ChatGPT performs competitively in the simple question type, correctly answering questions with entity or time as the answer. However, it has difficulty answering complex types of questions. For example, in the "Before/After" and "First/Last" question types, ChatGPT struggles to understand the temporal order of the relevant facts and gives incorrect answers. Besides, in the case of the "Time Join" question type, some of the correct answers are missing from the generated responses because ChatGPT does not fully understand the implicit time in the question. On the contrary, GenTKGQA performs well in both simple and complex question types due to the fact that we use a deep manner to incorporate subgraph information into the LLM. However, similar to other LLMs, GenTKGQA randomly generates some irrelevant answers when generating multiple answers, e.g., in the "Simple Entity" question type.

6 Conclusion

We propose a novel generative framework, GenTKGQA, which guides the LLM in a two-stage manner to handle temporal question answering on TKG. Specifically, at the subgraph retrieval phase, we exploit the LLM's intrinsic knowledge to mine the temporal constraints and structural links in the temporal questions, which reduces the search space of the subgraphs in both temporal and structural dimensions. We employ the in-context learning approach to complete subgraph retrieval for the entire dataset with a small number of samples. In order to improve the inference performance of the LLM on complex question types, at the answer generation phase, we present the instruction tuning technique to make the open-source LLM truly understand the temporal order and structural dependencies among retrieved facts. Most significantly, we design novel virtual knowledge indicators to establish a bridge between subgraph neural information and text representations. Experimental results show that our framework can effectively utilize the LLM to solve the complex question type of TKGQA task and validate the adequacy of our motivation.

Limitations

Although the complex temporal question on the CronQuestions dataset involves multiple facts, the inter-entity connection in each fact is single-hop, so the hyper-parameter k of our model is set to 1 to achieve the best results. In fact, the vast majority of current TKGQA datasets involve facts that are single-hop. So, in the future, we will explore more datasets to solve inference for multi-hop complex temporal problems over TKG. In addition, we use the in-context learning approach to prompt the ChatGPT baseline to answer the questions, saving the labor cost of checking whether the answers are correct. However, the design of different templates may result in incomplete consistency with the manual results, but this does not affect the conclusions of this paper. Because the results provided by other works similarly show the poor performance of ChatGPT's temporal question answering (Tan et al., 2023; Chen et al., 2023a).

Ethics Statement

This work presents a novel two-stage framework for the temporal knowledge graph question answering task using large language models. Our experiments use the publicly available datasets and language models from open sources. The dataset is developed to be used for the TKG-based temporal QA task. The language models are used to generate answers to temporal questions with entities or timestamps, which does not involve toxic content. This paper uses the above dataset and models with their initial intention. We believe that this work is consistent with ACL's ethics policy and presents no potential risk.

Acknowledgements

This work is sponsored in part by the National Natural Science Foundation of China under Grant No. 62025208, and the Xiangjiang Laboratory Fund under Grant No. 22XJ01012.

References

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations*, pages 78–106.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. Icews coded event data. *Harvard Dataverse*, 12.

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. In *NeurIPS Datasets and Benchmarks*.

Ziyang Chen, Dongfang Li, Xiang Zhao, Baotian Hu, and Min Zhang. 2023a. Temporal knowledge question answering via abstract reasoning induction. *CoRR*, abs/2311.09149.

Ziyang Chen, Jinzhi Liao, and Xiang Zhao. 2023b. Multi-granularity temporal question answering over knowledge graphs. In *ACL*, pages 11378–11392.

Ziyang Chen, Xiang Zhao, Jinzhi Liao, Xinyi Li, and Evangelos Kanoulas. 2022. Temporal knowledge graph question answering via subgraph reasoning. *Knowledge-Based Systems*, 251:109–134.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandecic. 2014. Introducing wikidata to the linked data web. In *ISWC*, pages 50–65.

Hao Fei, Bobo Li, Qian Liu, Lidong Bing, Fei Li, and Tat-Seng Chua. 2023. Reasoning implicit sentiment with chain-of-thought prompting. In *ACL*, pages 1171–1182.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020a. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *EMNLP*, pages 1295–1309.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020b. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *EMNLP*, pages 1295–1309.

Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. 2023. Chatdb: Augmenting llms with databases as their symbolic memory. *CoRR*, abs/2306.03901.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018. Tempquestions: A benchmark for temporal question answering. In *WWW*, pages 1057–1062.

Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. 2021. Complex temporal question answering on knowledge graphs. In *CIKM*, pages 792–802.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. 2023a. Structgpt: A general framework for large language model to reason over structured data. In *EMNLP*, pages 9237–9251.

Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. 2023b. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *ICLR*.

Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. 2023. KG-GPT: A general framework for reasoning on knowledge graphs using large language models. In *EMNLP (Findings)*, pages 9410–9421.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. In *ICLR*.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2024. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. In *ICLR*.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *EMNLP-IJCNLP*, pages 2829–2839.

Adam Liska, Tomás Kociský, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien de Masson d'Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, Ellen Gilsenan-McMahon, Sophia Austin, Phil Blunsom, and Angeliki Lazaridou. 2022. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *ICML*, pages 13604–13622.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yonghao Liu, Di Liang, Fang Fang, Sirui Wang, Wei Wu, and Rui Jiang. 2023a. Time-aware multiway adaptive fusion network for temporal knowledge graph question answering. In *ICASSP*, pages 1–5.

Yonghao Liu, Di Liang, Mengyu Li, Fausto Giunchiglia, Ximing Li, Sirui Wang, Wei Wu, Lan Huang, Xiaoyue Feng, and Renchu Guan. 2023b. Local and global: Temporal question answering via information fusion. In *IJCAI*, pages 5141–5149.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Haoran Luo, Haihong E, Zichen Tang, Shiyao Peng, Yikai Guo, Wentai Zhang, Chenghao Ma, Guanting Dong, Meina Song, and Wei Lin. 2023. Chatkbqa: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models. *CoRR*, abs/2310.08975.

Puneet Mathur, Vlad I. Morariu, Verena Kaynig-Fittkau, Jiuxiang Gu, Franck Dernoncourt, Quan Hung Tran, Ani Nenkova, Dinesh Manocha, and Rajiv Jain. 2022. Doctime: A document-level temporal dependency graph parser. In *NAACL*, pages 993–1009.

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 *AAAI*, pages 5825–5833.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67.

Apoorv Saxena, Soumen Chakrabarti, and Partha P. Talukdar. 2021. Question answering over temporal knowledge graphs. In *ACL-IJCNLP*, pages 6663–6676.

Apoorv Saxena, Aditay Tripathi, and Partha P. Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *ACL*, pages 4498–4507.

Chao Shang, Guangtao Wang, Peng Qi, and Jing Huang. 2022. Improving time sensitivity for question answering over temporal knowledge graphs. In *ACL*, pages 8017–8026.

Aditya Sharma, Apoorv Saxena, Chitrank Gupta, Seyed Mehran Kazemi, Partha P. Talukdar, and Soumen Chakrabarti. 2023. Twirgcn: Temporally weighted graph convolution for question answering over temporal knowledge graphs. In *EACL*, pages 2041–2052.

Xin Su, Phillip Howard, Nagib Hakim, and Steven Bethard. 2023. Fusing temporal graphs into transformers for time-sensitive question answering. In *EMNLP (Findings)*, pages 948–966.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agent. In *EMNLP*, pages 14918–14937.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. In *ACL*, pages 14820–14835.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio', and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.

Yike Wu, Nan Hu, Sheng Bi, Guilin Qi, Jie Ren, Anhuan Xie, and Wei Song. 2023. Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering. *CoRR*, abs/2309.11206.

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. 2024. Large language models can learn temporal reasoning. *CoRR*, abs/2401.06853.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *ACL*, pages 2346–2357.

Linyao Yang, Hongyang Chen, Zhao Li, Xiao Ding, and Xindong Wu. 2024. Give us the facts: Enhancing large language models with knowledge graphs for fact-aware language modeling. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–20.

Sen Yang, Xin Li, Lidong Bing, and Wai Lam. 2023. Once upon a time in graph: Relative-time pretraining for complex temporal reasoning. In *EMNLP*, pages 11879–11895.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: reasoning with language models and knowledge graphs for question answering. In *NAACL-HLT*, pages 535–546.

Hai Ye, Qizhe Xie, and Hwee Tou Ng. 2023. Multi-source test-time adaptation as dueling bandits for extractive question answering. In *ACL*, pages 9647–9660.

Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022a. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *ACL*, pages 5773–5784.

Michael J. Q. Zhang and Eunsol Choi. 2021. Situatedqa: Incorporating extra-linguistic contexts into QA. In *EMNLP*, pages 7371–7387.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D. Manning, and Jure Leskovec. 2022b. Greaselm: Graph reasoning enhanced language models for question answering. In *ICLR*.

Yichi Zhang, Zhuo Chen, Wen Zhang, and Huajun Chen. 2023. Making large language models perform better in knowledge graph completion. *CoRR*, abs/2310.06671.

Ziwang Zhao, Linmei Hu, Hanyu Zhao, Yingxia Shao, and Yequan Wang. 2023. Knowledgeable parameter efficient tuning network for commonsense question answering. In *ACL*, pages 9051–9063.

Category	**Train**	**Dev**	**Test**
Simple Entity	90,651	7,745	7,812
Simple Time	61,471	5,197	5,046
Before/After	23,869	1,982	2,151
First/Last	118,556	11,198	11,159
Time Join	55,453	3,878	3,832
Simple Reasoning	152,122	12,942	12,858
Complex Reasoning	197,878	17,058	17,142
Entity Answer	225,672	19,362	19,524
Time Answer	124,328	10,638	10,476
Total	350,000	30,000	30,000

Table 5: Dataset Statistics of CronQuestions.

Category	**Train**	**Dev**	**Test**
Explicit	2,725	1,302	1,312
Implicit	660	296	297
Temporal	2,810	1,177	1,163
Ordinal	976	587	593
Total	7,171	3,362	3,365

Table 6: Dataset Statistics of TimeQuestions[2].

A Dataset Statistics and Metrics

We use the CronQuestions and TimeQuestions dataset in our experiments. TimeQuestions was first proposed by Jia et al. (2021), but it provides only static knowledge graphs with temporal attributes, not strictly temporal knowledge graphs. Later, Sharma et al. (2023) expanded this dataset by preprocessing all the contained facts into the temporal knowledge graph format of (subject entity, relationship, object entity, [start time, end time]) and restricting all times to years. Dataset statistics are described in Table 5 and 6, respectively.

Following previous studies, we leverage two popular evaluation metrics, Hits@1 and Hits@10. Specifically, Hits@K $= \frac{1}{|\text{Test}|}\sum_{q\in \text{Test}} \text{ind}(\text{rank}(q) \leq \text{K})$, where $\text{rank}(q)$ denotes the ranking of the answer to question q obtained by the model in the candidate list. ind is 1 if the inequality holds and is 0 otherwise, $\text{K} = 1, 10$.

B Baselines and Implementation Details

We use the OpenAI-API[3] (gpt-3.5-turbo-0613[4]) for all ChatGPT-related experiments, including subsequent ChatGPT baselines.

In the subgraph retrieval phase, we use ChatGPT to mine temporal constraints and structural links between entities and add 5 samples to the in-context learning prompt templates, which are presented as Table 7 and 8. We set k=1 for the top-k relations. In the answer generation phase, following (Lacroix et al., 2020), we select the dimension of entity/relation/time embeddings to 512. For T-GNN, the layer l is set to 1, the linear transformations $\mathbf{W}_q$, $\mathbf{W}_k$ and $\mathbf{W}_m$ are 512×512, and the $\mathbf{m}$ and $\mathbf{b}$ of the MLP layer are $512\times|\mathcal{E}|$. We use the open-source Llama 2-7B (Touvron et al., 2023) for instruction tuning and select up to n=10 relevant facts as additional knowledge. The linear projection layer $\mathbf{W}_p$ is 512×4096. We fine-tune Llama 2-7B using LoRA (Hu et al., 2022) with rank 64. The number of epochs is set to 4 and the learning rate is 3e-4. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a fixed batch size of 8. We conduct all the experiments with NVIDIA A100 GPUs, and the results of each experiment are averaged over three runs. We will release the source code upon acceptance.

We compare our model with the following baselines:

EmbedKGQA (Saxena et al., 2020): Timestamps are ignored during pre-training and random time embeddings are used during the QA task.

EaE (Feng et al., 2020b): In the experiment, we follow use TKG embeddings to enhance the question representation, and then predict the answer probabilities via dot-product.

CronKGQA (Saxena et al., 2021): CronKGQA is the TKGQA embedding-based method that first uses a LM model to get question embeddings and then utilize a TKG embedding-based scoring function for answer prediction.

EntityQR and TempoQR (Mavromatis et al., 2022): Based on EaE, EntityQR utilizes a TKG embedding-based scoring function for answer prediction. TempoQR utilizes a TKG embedding-based scoring function for answer prediction and fuse additional temporal information.

[2] The actual number of questions in the training, validation and test set is 6,970, 3,236 and 3,237, respectively. The total number exceeds the number of questions as some questions belong to multiple categories.

[3] `https://platform.openai.com/docs/api-reference`

[4] `https://platform.openai.com/docs/models/gpt-3-5-turbo`

TMA (Liu et al., 2023a): TMA improves QA performance through enhanced fact retrieval and adaptive fusion network.

TSQA (Shang et al., 2022): TSQA presents a contrastive learning module that improves sensitivity to time relation words.

TwiRGCN (Sharma et al., 2023): TwiRGCN is a method for processing TKGQA tasks with the relational graph convolutional network (RGCN).

BERT and RoBERTa (Devlin et al., 2019; Liu et al., 2019): For ***w/o tkg***, following CronKGQA (Saxena et al., 2021), we add a prediction head on top of the [CLS] token of the final layer, and then do a softmax over it to predict the answer probabilities. For ***w/ tkg***, following TempoQR (Mavromatis et al., 2022), we generate their LM-based question embedding and concatenate it with the annotated entity and time embeddings, followed by a learnable projection. The resulted embedding is scored against all entities and timestamps via dot-product.

ChatGPT: To ensure that the output format meets the expected requirements, we use the in-context learning approach to motivate ChatGPT to answer the questions and provide 5 examples in the prompt template. The specific templates are presented in Table 10 of Appendix C. ***w/o tkg*** and ***w/ tkg*** differ in whether or not question-relevant facts are provided in the input prompts, which are retrieved in the first stage by our proposed GenTKGQA framework.

C Prompt Template

The prompts for relation ranking and time mining can be found in Table 7 and Table 8, respectively. The template used for instruction tuning is shown in Table 9. The ChatGPT baseline prompt is presented in Table 10.

Relation Ranking Prompt
l will give you a list of words. Find the $\{k\}$ words from the list that are most semantically related to the given sentence. If there are no semantically related words, pick out any $\{k\}$ words. Examples) Sentence A: When was the first time Martin Taylor played for The Hatters? Words List: [‘member of sports team’, ‘position held’, ‘award received’, ‘spouse’, ‘employer’] Top $\{k\}$ Answers: [‘member of sports team’] . . . Sentence E: Which was awarded to Daniel Walther in 1980? Words List: [‘member of sports team’, ‘position held’, ‘award received’, ‘spouse’, ‘employer’] Top $\{k\}$ Answers: [‘award received’] Now let’s find the top $\{k\}$ words. Sentence: $\{sentence\}$ Words List: $\{relation\_list\}$ Top $\{k\}$ Answer:

Table 7: Relation Ranking Prompt. This prompt is used to extract structural links between entities in the question.

Time Mining Prompt
I will give you a natural language question with a temporal constraint. Answer the temporal constraint involved in the question based on the knowledge context and the question type. Answer only in "before", "after", "between and" format. Examples) Question A: Who held Governor of Connecticut position after Lowell P. Weicker? Knowledge Context: [‘Lowell P. Weicker’, ‘position held’, ‘Governor of Connecticut’, ‘1991’, ‘1995’] Question Type: after Response: after 1995 . . . Question E: Who’s the player who played in AC Reggiana with Daniele Magliocchetti? Knowledge Context: [‘Daniele Magliocchetti’, ‘member of sports team’, ‘A.C. Reggiana’, ‘2012’, ‘2014’] Question Type: time_join Response: between 2012 and 2014 Next, let’s answer the time constraints involved in the following question. Question: $\{question\}$ Knowledge Context: $\{context\}$ Question Type: $\{type\}$ Response:

Table 8: Time Mining Prompt. This prompt is used to find the time constraints involved in the complex question.

Instruction Tuning Template

Below is an instruction that describes a task, paired with an input that provides further context.
Write a response that appropriately completes the request.

Instruction:
Answer the questions based on evidence.
Each evidence is in the form of [head, relation, tail, start_time, end_time]
and it means ‘head relation is tail between start_time and end_time’.
You must list the 10 most relevant answers.

Input:
Question: $\{question\}$
Evidence set: $\{evidence\_set\}$

Response:$\{answer\}$

Table 9: This is the template for instruction tuning.

ChatGPT *w/ tkg*

Answer the questions based on evidence.
Each evidence is in the form of [head, relation, tail, start_time, end_time]
and it means ‘head relation is tail between start_time and end_time’.
You must list the 10 most relevant answers separated by ‘\t’.

Examples)

Question A: Who was the Member of the House of Representatives in 1990?
Evidence set: [[‘Simon Crean’, ‘position held’, ‘Member of the House of Representatives’, ‘1990’, ‘2013’],. . .]
Answer: Simon Crean\tJohn Dawkins\t. . .

. . .

Question E: With whom did Steve Haslam play on the Sheffield Wednesday F.C.?
Evidence set: [[‘Ola Tidman’, ‘member of sports team’, ‘Sheffield Wednesday F.C.’, ‘2003’, ‘2005’], . . .]
Answer: Ola Tidman\tChris Marsden\t. . .

Now let’s answer the Question based on the Evidence set.
Please do not say there is no evdience, you must list the 10 most relevant answers separated by ‘\t’.
Question: $\{question\}$
Evidence set: $\{evidence\_set\}$
Answer:

ChatGPT *w/o tkg*
Answer the questions directly.
You must answer the 10 most relevant answers separated by ‘\t’.

Examples)

Question A: Who was the Member of the House of Representatives in 1990?
Answer: Simon Crean\tJohn Dawkins\t. . .

. . .

Question E: With whom did Steve Haslam play on the Sheffield Wednesday F.C.?
Answer: Ola Tidman\tChris Marsden\t. . .

Now let’s answer the Question, you must answer the 10 most relevant answers separated by ‘\t’.
Question: $\{question\}$
Answer:

Table 10: ChatGPT Baseline Prompt.