MusTQ: A Temporal Knowledge Graph Question Answering Dataset for Multi-Step Temporal Reasoning

Tingyi Zhang[♠], **Jiaan Wang**[♠], **Zhixu Li**[♢*], **Jianfeng Qu**[♠*], **An Liu**[♠]
Zhigang Chen[♡] **and Hongping Zhi**[♣]

[♠]School of Computer Science and Technology, Soochow University, Suzhou, China
[♢]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai, China
[♡]Jilin Kexun Information Technology Co., Ltd.
[♣]iFLYTEK Research, Suzhou, China

{zhangtingyi712,jawang.nlp}@gmail.com zhixuli@fudan.edu.cn
{jfqu,anliu}@suda.edu.cn {zgchen,hpzhi}@iflytek.com

Abstract

Question answering over temporal knowledge graphs (TKGQA) is an emerging topic, which has attracted increasing interest since it considers the dynamic knowledge in the world. Several datasets along with model developments are proposed in the TKGQA research field. However, existing studies generally focus on fact-centered reasoning, with limited attention to temporal reasoning. To tackle the intricate and comprehensive nature of temporal reasoning, we propose a new TKGQA dataset, MusTQ, which contains 666K multi-step temporal reasoning questions as well as a TKG. The multi-step temporal reasoning is established based on six basic temporal reasoning types derived from a well-established measure theory. Using MusTQ, we evaluate previous TKGQA methods and find that they typically fall short in multi-step temporal reasoning. Furthermore, we propose a TKGQA model, MusTKGQA, which enhances multi-step reasoning ability with entity-time attention mechanism and optimized temporal knowledge graph representation. Extensive experiments on MusTQ show that our model achieves state-of-the-art multi-step temporal reasoning performance.[1]

1 Introduction

Given natural questions, question answering over temporal knowledge graphs (TKGQA) aims to provide the corresponding answers using a temporal knowledge graph (TKG) as the knowledge base. Since many real-world facts will change over time and formulating questions with temporal constraint is a straightforward method to ensure factual validity, TKGQA has gradually attracted increasing research attention (Saxena et al., 2021; Ding et al., 2023; Chen et al., 2022; Long et al., 2022; Xiao et al., 2022; Shang et al., 2022; Mavromatis et al., 2022; Jiao et al., 2023; Ding et al., 2022) .

Compared with the traditional knowledge graph question answering (KGQA) that adopts static knowledge graphs as the knowledge base, TKGQA should not only consider the knowledge formed in triples but also emphasize its temporal information. Therefore, reasoning on temporal information could naturally become a vital perspective for questions that involve temporal constraints. Existing TKGQA datasets either query the information in TKG (Saxena et al., 2021; Chen et al., 2023) or make prediction in the future timestamps (Ding et al., 2023). In this paper, we mainly focus on the former type of TKGQA datasets. Though great success has been achieved in previous work, existing TKGQA datasets typically focus on fact-centered reasoning questions, with less attention on temporal reasoning. Specifically, they only involve single-step temporal reasoning, ignoring the systematical exploration of temporal reasoning. Among them, CronQuestions (Saxena et al., 2021) constructs its questions with single-step temporal reasoning related to explicit time or entity. Although MultiTQ (Chen et al., 2023) contains two-step temporal reasoning questions, its questions only involve fact-centered reasoning and lack systematic explorations on temporal reasoning.

In this paper, we propose MusTQ (**Mu**ti-**S**tep **T**emporal Reasoning **Q**uestions), a new TKGQA dataset, which aims to enhance the temporal reasoning ability in TKGQA. In order to systematically study temporal reasoning, we start with the measure of time. According to the well-established theory for the scale of measurement (Stevens, 1946), the measure of time determines the basic operations that can be applied to the temporal data. The timestamp information in TKG involves the following three operations: determination of i) equality; ii) greater or less; iii) equality of intervals or differences. The first operation maps the facts to their

* Corresponding authors.
[1] https://github.com/theTyZ/MusTQ

Findings of the Association for Computational Linguistics ACL 2024, pages 11688–11699
August 11-16, 2024

Dataset	KG	Temporal facts	Temporal reasoning			# Questions
			Comparison	Numeric	Multi-step	
TempQuestions	FreeBase	✗	✓	✗	✗	1271
TimeQuestions	WikiData	✗	✓	✗	✗	16k
CronQuestions	WikiData	✓	✓	✗	✗	410k
MultiTQ	ICEWS	✓	✓	✗	✓	500k
MusTQ	WikiData	✓	✓	✓	✓	666k

Table 1: The comparison of different temporal questions datasets over KG in information query scenario.

corresponding timestamps, while the second operation sorts facts according to their chronology. The final operation enables timestamp reasoning with the addition and subtraction operations. Under the guidance of this theory, we classify temporal reasoning into temporal comparison reasoning and temporal numeric reasoning. The former uses operations i) and ii) to reason timestamps based on the comparison, while the latter uses operation iii) to reason timestamps based on the interval. Further, we divide these two categories into six detailed reasoning types and construct the diverse multi-step temporal reasoning by their combination. Finally, MusTQ contains ~666K temporal questions and an underlying TKG (with 125k entities and 326k facts) as its knowledge base.

Based on our MusTQ, we build and evaluate various baseline systems, including the pre-trained language models and the representative KGQA and TKGQA models. The experimental results demonstrate that all existing methods show their limited capabilities in multi-step temporal reasoning. Furthermore, we propose a new TKGQA model, MusTKGQA, which improves the multi-step temporal reasoning ability via carefully designed TKG representation learning and entity-time attention mechanism. In detail, to enhance the temporal reasoning, we optimize the TKG representation in time and fact perspectives with sequence alignment and fact alignment respectively. The sequence alignment enhances the reasoning between timestamps, while the fact alignment improves the awareness for time boundary of interval facts. Then, we propose an entity-time attention mechanism to infer the temporal information of the question steply and finally reason the answer. Extensive experiments on MusTQ show that our model achieves state-of-the-art multi-step temporal reasoning performance. The results on the previous CronQuestions dataset also show the strong ability of our model. Our main contributions are concluded as follows:

- We construct the first TKGQA dataset (*i.e.*, MusTQ) for multi-step temporal reasoning. The involved single-step temporal reasoning types are founded systematically upon the operations for the measure of time. Our dataset contains about 666K questions and includes temporal comparison reasoning and temporal numeric reasoning as its basic stepwise reasoning categories.
- We propose a model MusTKGQA that conducts multi-step temporal reasoning through the entity-time attention mechanism based on TKG representation. To enhance temporal reasoning ability, the TKG representation is optimized by sequence alignment and fact alignment.
- Experiments on MusTQ show that our model achieves state-of-the-art performance on multi-step temporal reasoning.

2 Related Work

Temporal Questions over KG. Several datasets are proposed for temporal question answering. TempQuestions (Jia et al., 2018) and TimeQuestions (Jia et al., 2021) are two datasets that collect temporal questions from KGQA datasets. However, they adopt traditional static knowledge graphs as their knowledge base (Bollacker et al., 2008; Vrandečić and Krötzsch, 2014). CronQuestions (Saxena et al., 2021) is the first temporal question dataset over TKG (Lacroix et al., 2020), whose questions are classified into simple and complex questions based on the number of facts leveraged to answer the questions. Additionally, MultiTQ (Chen et al., 2023) focuses on questions in multiple temporal granularities and contains two-step comparison temporal reasoning questions with ICEWS (García-Durán et al., 2018). However, its questions are fact-centered and lack temporal reasoning variety. The questions in the above datasets require models to query information within the provided KGs. ForecastTKGQuestions (Ding et al., 2023) is another TKGQA dataset that aims to predict facts in the future timestamps beyond the given TKG. We mainly focus on datasets that query information within TKG. Different from existing datasets that generally focus on fact-centered reasoning, our MusTQ contains multi-step temporal reasoning questions constructed based on the systematical exploration of temporal reasoning. Table 1 shows the comparison between our MusTQ and previous datasets.

TKGQA Models. Question answering on TKG generally leverages the Temporal Knowledge Graph Embedding (TKGE) for answer reasoning, which aims to learn the low-dimensional representation of entities, relations, and timestamps. CronKGQA (Saxena et al., 2021) leverages TKGE to make answer reasoning with the semantics of the questions. Based on CronKGQA, various TKGQA models are proposed to emphasize the time information of the question. Among them, TempoQR (Mavromatis et al., 2022) introduces time scope information to make time-aware question representation for answer reasoning. TSQA (Shang et al., 2022) infers the timestamps through the time estimation module. SubGTR (Chen et al., 2022) adopts logical reasoning on the subgraph of the question. Besides, LGQA (Liu et al., 2023) emphasizes the graph structural information in QA. Moreover, MultiQA (Chen et al., 2023) considers the multi-granularity temporal questions and ForecastTKGQA (Ding et al., 2023) is for TKGQA in the forecast scenario. However, none of these methods consider multi-step temporal reasoning in various temporal reasoning categories. To tackle this real-world challenge, we propose MusTKGQA.

3 MusTQ Construction

MusTQ consists of multi-step temporal questions and a TKG as its knowledge base. We first introduce the TKG in MusTQ (§ 3.1). Then we describe the systematic temporal reasoning categories, and the single-step reasoning types in each category (§ 3.2). Next, we present the process of developing multi-step temporal reasoning questions (§ 3.3). Finally, we show the data statistics of MusTQ (§ 3.4).

3.1 Temporal Knowledge Graph

A TKG $\mathcal{G}$ consists of temporal facts in the form of (s, r, o, τ), where $s, o \in \mathcal{E}$ and $r \in \mathcal{R}$. $\mathcal{E}$, $\mathcal{R}$ denote the set of entities and relations in $\mathcal{G}$, respectively. $\tau \in \mathcal{T}$ represents the valid time of the fact (s, r, o) in the set of timestamps $\mathcal{T}$. We adopt the Wikidata subset (Lacroix et al., 2020) as the base of our TKG because its time annotations are in the form of timestamps τ_p: *(s, r, o, τ_p, τ_p)* and time intervals (τ_s, τ_e): *(s, r, o, τ_s, τ_e)*, which are denoted as timestamp facts and interval facts, respectively. Following Saxena et al. (2021), we balance the number of facts in different relations and supplement necessary temporal information for the occurrence of its entities via the following heuristic rules: (1) The dataset adopts the time granularity down to the year; (2) We eliminate duplicate facts and maintain a refined TKG by narrowing the time span to the period the majority of facts occurrences are concentrated. Finally, the TKG includes 125K entities, 202 relations, and 326K facts with the time span from 1 AD to 2027 AD (*i.e.*, 2027 timestamps).

3.2 Temporal Reasoning Category

To propose multi-step temporal questions, we first systematically construct multiple basic temporal reasoning types based on all basic operations for time. According to a well-established measure theory (Stevens, 1946), the timestamp information contains three fundamental empirical operations, *i.e.*, determination of i) equality; ii) greater or less; iii) equality of intervals or differences. These operations are used to perform the following two temporal reasoning categories:

Temporal Comparison Reasoning. Based on the operations i) and ii), this category compares the temporal information of different facts. In detail, the operation i) enables the measuring of equivalence for categorized variables (*e.g.*, timestamp). Each fact in TKG could be mapped to a specific timestamp based on the operation i). Consequently, we establish the fine-grained reasoning types (1) *Explicit Time* and (2) *Fact* which need equivalence reasoning by considering the timestamp or occurrence of fact, separately. The operation ii) corresponds to '$>$/$<$' in mathematics. Under its guidance, variables could compare with each other to establish their rank-ordering. Thus, we introduce the fine-grained types: (3) *Before/After* reasons the time span based on the occurrence of the fact. (4) *Ordinal* infers the target timestamp based on facts occurring at the ordinal positions.

Temporal Numeric Reasoning. This category processes the numeric reasoning between timestamps with the operation iii). The operation makes it possible to measure the differences between timestamps, and it corresponds '$+$/$-$' in mathematics. We develop the following reasoning types to make inferences based on time intervals: (5) *Time Dif-*

Category	Operation	Type	Example Templates
Temporal Comparison Reasoning	Equality	Explicit Time	in {time}
		Fact	when {time}
	Greater or less	Before/After	before {time}
		Ordinal	When did {head} receive the {ordinal} award?
Temporal Numeric Reasoning	Equality of intervals or differences	Time Difference	Did {headA} win {tail} {diff} years before {headB}?
		Count	How many awards has {head} won overall?

Table 2: The reasoning categories and fundamental operations of timestamp. Each reasoning type corresponds to an example time constraint template or basic template. "{head}" and "{tail}" denote head and tail entities, respectively. "{diff}" is a digit that denotes the explicit time difference and "{ordinal}" indicates the ordinal number.

Process	Template
Base	Did {head} secure {tailA} {diff} years earlier than winning {tailB}?
+ Time Cons.	Following {timeA}, did {head} secure {tailA} {diff} years earlier than winning {tailB}?
+ Time Fill.	Following {tailA} **was given to** {headA}, did {head} secure {tailA} {diff} years earlier than winning {tailB}?
+ Entity	Following Order of Georgi Dimitrov was given to Gherman Titov, did Valentina Stepanovna Grizodubova secure Order of Lenin five years earlier than winning Order of the October Revolution?

Table 3: The construction process from the base template to the final multi-step temporal reasoning question. Cons.: Constraints; Fill.: Filling.

ference and (6) *Count*, where the temporal interval is denoted by explicit time gaps and the cardinal number of factual occurrences, respectively.

Among the above six reasoning types, types (1-3) only form time constraints, while types (4-6) lead to the questions themselves.

3.3 Temporal Question Construction

We leverage the manual templates w.r.t six reasoning types to construct multi-step temporal reasoning questions in four steps (c.f., Table 3): First, based on reasoning types (4-6), we employ human experts to construct base templates using four high-frequency relations in TKG. The human experts construct 63 unique base templates (for the construction details, please refer to Appendix A). The base template examples are shown in Table 2.

Second, we extend the base templates with essential time constraints to incorporate multi-step temporal reasoning. We adopt the reasoning types (1-3) to introduce time constraints into base templates. This process collects 691 templates with time constraints. To enhance the semantics diversity, human experts are further employed to create several paraphrasing templates to rewrite these

Type	Example Templates
2 steps	Following {timeA}, did {head} secure {tailA} {diff} years earlier than winning {tailB}?
3 steps	How many award nominations has {head} received in total in the time span from {timeA} to {timeB}?
4 steps	Who took on the {ordinal} job as {tail} in {time} during the phase between {timeA} and {timeB}?

Table 4: Example templates with time constraints.

templates. As a result, we obtain 12,754 templates with time constraints. Table 4 gives several example templates with different numbers of reasoning steps. Third, we fill time slots (*e.g.*, {time}) in each template with the time information described as the timestamp or fact time.

Finally, multi-step questions are completed by completing the time-filled templates with entity aliases from the TKG (entity filling). To ensure reality and validity, we only use the entities within a 3-hop distance of the entity in the basic template during entity filling. Appendix B lists several examples of question construction. Ultimately, we construct 666K unique question-answer pairs. The boundaries of entities and timestamps within questions are also provided in MusTQ.

3.4 Data Statistics

To ensure temporal reasoning without guessing from seen questions, we make sure there is no overlap of the multi-step reasoning (*i.e.*, the combination of entities, timestamps and reasoning types) between the training set and the test set. The data statistics of the dataset splitting are shown in Table 5. MusTQ includes questions with a range of two to four temporal reasoning steps. Most questions require temporal reasoning within three steps. There are four types of answers in MusTQ. In ad-

	Train	Valid	Test
Entity Answer	119,451	10,246	10,259
Time Answer	124,501	10,638	10,651
Boolean Answer	212,474	11,266	11,296
Numeric Answer	124,345	10,600	10,638
Two steps	297,126	20,771	20,798
Three steps	261,766	20,100	20,168
Four steps	21,879	1,879	1,878
Total	580,771	42,750	42,844

Table 5: The number of questions of MusTQ w.r.t different types of answers and different numbers of temporal reasoning steps.

dition to the entity and time answers, which are commonly involved in existing datasets (Saxena et al., 2021; Chen et al., 2023), MusTQ contains boolean answers to verify the mentioned time differences in question. The questions with numeric answers are also included to evaluate the ability to understand the time interval by reasoning the fact times within the interval.

4 MusTKGQA

To answer the temporal reasoning questions, we propose MusTKGQA which first enhances the temporal knowledge graph embedding (TKGE) with sequence alignment and fact alignment (§ 4.1). Then, our model leverages the enhanced TKGE to perform the multi-step temporal reasoning through an entity-time attention mechanism (§ 4.2).

4.1 Enhanced TKGE

TKGE represents the TKG by learning the embeddings of entities, relations and timestamps, which is widely adopted in TKGQA. TNTComplEx (Lacroix et al., 2020) is a state-of-the-art TKGE model that learns representation in the *complex vector space* through entity prediction $(s, r, ?, t)$ and time prediction $(s, r, o, ?)$ tasks which correspond to the information query scenario of our MusTQ. Besides, it considers nontime-sensitive information together with dynamic facts, which could help to query related entities when there is no explicit time information (please refer to Appendix C for more details). However, TNTComplEx is weak for multi-step temporal reasoning due to the following two drawbacks: (1) It neglects the explicit modeling of **chronological order** between timestamps, resulting in limited ability in numeric reasoning. (2) It is weak in recognizing the validity period of the fact. In the time prediction task, given an interval fact that happens at t_s and ends at t_e, each timestamp within its valid period (t_s, t_e) is sampled equally as the golden labels. Therefore, the **time boundaries** (t_s and t_e) of the interval fact are not fully modeled. Given these, we introduce sequence alignment and fact alignment in TKGE to enhance the modeling of chronological order and time boundary, respectively.

Sequence Alignment. We perform sequence alignment to sort timestamps in chronological order with fixed-length intervals. Specifically, we introduce three relations in TKG to enhance the representation of timestamps: next year ($next_y$), prior year ($prior_y$), and this year ($this_y$). For example, facts (2005, $next_y$, 2006) and (2006, $prior_y$, 2005) describe the chronological order between the years 2005 and 2006. Besides, with fact (2006, $this_y$, 2006), the awareness of the same year is enhanced.

Fact Alignment. To model the time boundary information of interval facts, we introduce additional time prediction samples. For each interval fact (s, r, o, t_s, t_e), two timestamp facts can be constructed: (s, r_s, o, t_s), (s, r_e, o, t_e), where r_s and r_e are two relations denote the start time and end time. In this way, the boundary information of facts could be explicitly considered in TKGE.

Equipped with these optimizations, the trained TKGE model ($\Theta(\cdot)$) is further used to generate the low-dimensional representation of elements in $\mathcal{G}$.

4.2 Temporal Question Reasoning

After enhancing the TKGE, for a given TKGQA question q, we reason its answer based on both q and the enhanced TKGE. In detail, we first fuse the TKGE information with q to obtain the TKGE-enhanced question representation. Second, based on the fused information, we leverage an entity-time attention mechanism to get attention distributions over the mentioned entities in question. Third, we perform temporal reasoning for each entity. Finally, we reason the answer for the question q.

TKGE-enhanced Question Representation. At the start, we integrate question q with the entity and timestamp information in $\mathcal{G}$. q is first tokenized into a token sequence $q = \{w_1, w_2, ..., w_{|q|}\}$, where w_i indicates the i-th token. Then, we obtain the contextual embeddings for each question token via a frozen BERT (Devlin et al., 2019):

$$h^q_{cls}, h_{w_1}, ..., h_{w_{|q|}} \leftarrow W_c \, \mathrm{BERT}(\{w_1, ..., w_{|q|}\}) \quad (1)$$

where W_c is trainable parameters, $h^q_{cls} \in \mathbb{R}^d$ denotes the hidden state of special token [CLS], which gathers the whole information of q. $h_{w_i} \in \mathbb{R}^d$ denotes the hidden state of w_i. Next, the TKGE-enhanced question representation is calculated as:

$$\mathrm{h}^q_{cls}, \mathrm{h}_{w_1}, ..., \mathrm{h}_{w_{|q|}} \leftarrow \mathrm{g}(h^q_{cls}, h'_{w_1}, ..., h'_{w_{|q|}}) \quad (2)$$

$$h'_{w_i} = \begin{cases} \Theta(w_i) & w_i \in \mathcal{T} \cup \mathcal{E} \\ h_{w_i} & \text{otherwise} \end{cases} \quad (3)$$

where g denotes the transformer encoder (Vaswani et al., 2017). In this manner, h^q_{cls} and h^q_{cls} denote the overall representation of question q before and after TKG information fusion, respectively.

Entity-Time Attention Mechanism. The proposed attention mechanism is used to obtain the attention distributions over entities in question during different reasoning steps. To introduce the time scope information for multiple entities mentioned in q, we leverage two-step temporal reasoning (which provides the related time scope information) before the answer reasoning step.

For the initial step of temporal and answer reasoning, we construct queries q_t and q_a guided by the question representation:

$$q_t, q_a \leftarrow W_t[h^q_{cls}; \mathrm{h}^q_{cls}], W_a[h^q_{cls}; \mathrm{h}^q_{cls}] \quad (4)$$

Subsequently, since the temporal constraints reasoning steps depend on each other, q_t serves as guidance for the next step query $q'_t = W'_t[q_t; \mathrm{h}^q_{cls}]$, where W_t, W_a and $W'_t \in \mathbb{R}^{d \times 2d}$ are trainable parameters. Assuming there are k entities mentioned in q, denotes as $E_q = \{e_1, ..., e_{|E_q|}\}$ ($e_i \in \mathcal{E}$), we next use these three queries to calculate the attention distributions over them:

$$\mathrm{Attn}_a = \sum_{i=1}^{|E_q|} \beta^a_i \Theta(e_i) \mid \beta^a_i = \mathrm{softmax}(q_a \Theta(e_i)), \quad (5)$$

where Attn_a denotes the attention distribution calculated by query q_a. Similarily, Attn_t, Attn'_t, β^t_i and $\beta^{t'}_i$ are obtained based on queries q_t and q'_t.

Temporal Reasoning. For each entity $e_i \in E_q$, we calculate its contextual representation based on other entities and the mentioned time τ in q:

$$\mathbf{e}^t_i = \sum_{j=1, j \neq i}^{|E_q|} \beta^t_j \Theta(e_j) + \gamma^t_i \Theta(e_i) \quad (6)$$

$$\mathbf{e}^{t'}_i = \sum_{j=1, j \neq i}^{|E_q|} \beta^{t'}_j \Theta(e_j) + \gamma^{t'}_i \Theta(e_i) \quad (7)$$

where $\mathbf{e}^t_i$ and $\mathbf{e}^{t'}_i$ denote the contextual representation of e_i based on queries q_t and q'_t, respectively. γ^t_i and $\gamma^{t'}_i$ are attention weights of e_i, which gather the time information τ:

$$\gamma^t_i = \beta^t_i \,\mathrm{Sigmoid}(\mathrm{W}_a q_a) \mathrm{W}_t q_t + \Theta(\tau) \quad (8)$$

$$\gamma^{t'}_i = \beta^{t'}_i \,\mathrm{Sigmoid}(\mathrm{W}_a q_a) \mathrm{W}_{t'} q'_t + \Theta(\tau) \quad (9)$$

where W_a, W_t and $\mathrm{W}_{t'}$ are trainable parameters. Following Mavromatis et al. (2022), we next conduct temporal reasoning by inferring the time information ν_i for entity e_i:

$$\Re(\nu^t_i) = \Re(\Theta(e_i)) \odot \Re(\omega^t_i) - \Im(\Theta(e_i)) \odot \Im(\omega^t_i) \quad (10)$$

$$\Im(\nu^t_i) = \Re(\Theta(e_i)) \odot \Im(\omega^t_i) + \Im(\Theta(e_i)) \odot \Re(\omega^t_i) \quad (11)$$

where ν^t_i denotes the time information after reasoning based on query q_t and $\omega^t_i = (W^t_q q_t) \odot \overline{\mathbf{e}}^t_i$. $\odot$ denotes the element-wise product. $\overline{\mathbf{e}}^t_i$ is the conjugate vector of $\mathbf{e}^t_i$. $\Re$ and $\Im$ represent the real part and imaginary part of the complex embedding. Similarly, based on query q'_t, $\nu^{t'}_i$ is calculated with $\omega^{t'}_i$, where $\omega^{t'}_i = (W^{t'}_q q'_t) \odot \overline{\mathbf{e}}^{t'}_i$. W^t_q and $W^{t'}_q$ are trainable parameters.

Answer Reasoning. We combine the reasoned temporal information $\nu^{t'}_i / \nu^t_i$ when token w_j in q is the entity e_i through a transformer g_q:

$$m^q_{cls}, m_{w_1}, ..., m_{w_{|q|}} \leftarrow g_q(\mathrm{h}^q_{cls}, \mathrm{h}'_{w_1}, ..., \mathrm{h}'_{w_{|q|}}) \quad (12)$$

$$\mathrm{h}'_{w_j} = \begin{cases} \mathrm{h}_{w_j} + \nu^{t'}_i + \nu^t_i & w_j = e_i \in E_q \\ \mathrm{h}_{w_j} & \text{otherwise} \end{cases} \quad (13)$$

where m^q_{cls} is utilized for final question representation that incorporates information from both TKG $\mathcal{G}$ and the reasoned temporal information.

In the answer reasoning step, we use the queried attention information Attn_a for reasoning. Specifically, various answer types are considered based on m^q_{cls}. We make the answer reasoning into four classification tasks:

(1) *Entity Prediction*: we score each entity $\epsilon \in \mathcal{E}$ with time τ mentioned in q:

$$s_e = \Re(< \mathrm{Attn}_a, W_e m^q_{cls}, \overline{\Theta(\epsilon)}, \Theta(\tau) >) \quad (14)$$

where $< ... >$ denotes inner product. W_e is the trainable parameters. $\overline{\Theta(\epsilon)}$ is the conjugate vector of $\Theta(\epsilon)$. $s_e \in \mathbb{R}^{|\mathcal{E}|}$ represents the prediction score over all entities in $\mathcal{G}$.

(2) *Time Prediction*: we leverage max function to score each timestamp $\tau \in \mathcal{T}$ with time reasoning queried attention information Attn_t, Attn'_t:

$$\begin{aligned} s_t = \max(&\Re(< \mathrm{Attn}_a, W_\tau m^q_{cls}, \overline{\mathrm{Attn}_t}, \Theta(\tau) >), \\ &\Re(< \mathrm{Attn}_a, W_\tau m^q_{cls}, \overline{\mathrm{Attn}'_t}, \Theta(\tau) >)) \end{aligned} \quad (15)$$

where $s_t \in \mathbb{R}^{|\mathcal{T}|}$ denotes the prediction score over all timestamps. W_τ is the trainable parameters.
(3) *Boolean Prediction:* we get the maximum score with time reasoning queried attention information and the time τ mentioned in q. Then, the boolean answer is obtained through projection f_b:

$$\begin{aligned} s_b = f_b(\max(\Re(<\text{Attn}_a, W_b m^q_{cls}, \overline{\text{Attn}_t}, \Theta(\tau)>), \\ \Re(<\text{Attn}_a, W_b m^q_{cls}, \overline{\text{Attn}'_t}, \Theta(\tau)>))) \end{aligned} \tag{16}$$

where $s_b \in \mathbb{R}^2$ is the boolean prediction score. W_b is the trainable parameters.
(4) *Numeric Prediction*: we use the entity and time score distribution to predict numeric answer through projection f_c: $s_c = f_c([s_e; s_t])$.

The model employs the softmax function to calculate the prediction probability and adopts cross-entropy loss during training.

5 Experiments

5.1 Experimental Setups

We evaluate the performance of previous baselines and our MusTKGQA on MusTQ.

Baselines. We select pre-trained language models (PLMs), traditional KGQA, and strong TKGQA models as baselines.

(1) *Pre-trained language models*: We include two representative PLMs **BERT** (Devlin et al., 2019) and **RoBERTa** (Liu et al., 2019) as baselines. To adopt these PLMs in TKGQA, following Saxena et al. (2021), we use PLMs to encode the questions, and then add the prediction head on the top of the question representation followed by a softmax function to make the answer prediction. We also combine the TKG information and introduce two variants for each PLM. a) **RoBERTa**$_{TC}$ and **BERT**$_{TC}$: We obtain the TKG-enhanced question representation by concatenating question representation obtained by PLMs with the mentioned entity and timestamp embeddings in TKG. Then we acquire the fused information with projection. We calculate the prediction score of all entities and timestamps through dot-product against their TKG embeddings. For the boolean answer and numeric answer prediction, we only use projection to predict answers. Here, we use TComplEx (Lacroix et al., 2020) to obtain the TKGE. b) **RoBERTa**$_{AL}$ and **BERT**$_{AL}$ further employ our proposed sequence alignment and fact alignment.

(2) *KGQA model*: **EmbedKGQA** (Saxena et al., 2020) is a widely used KGQA model that leverages KG embeddings to perform multi-hop KGQA. However, this model cannot model the temporal information, thus we leverage the random time embeddings when adapting this model in TKGQA.

(3) *TKGQA model*: **CronKGQA** (Saxena et al., 2021) and **TempoQR** (Mavromatis et al., 2022) are two strong TKGQA models that specialize in single-step temporal reasoning. The TempoQR has two versions, *i.e.*, soft-supervision and hard-supervision. Here, we use the soft-supervision version in our baseline since it can efficient reasoning with questions including multiple entities.

Metrics. For questions with entity and time answers, we report the Hits@n to show the proportion that the golden answer is included in the top-n of the candidate list. Following Saxena et al. (2021), we report Hits@1 and Hits@10. For questions with boolean and numeric answers, we use accuracy as the evaluation metric, which is equal to Hits@1.

Implementation Details. The details of training hyper-parameters are given in Appendix D.

5.2 Results & Analyses

Table 6 shows the result of baselines and our model on MusTQ w.r.t different answer types. "Overall" indicates the overall performance on all questions. For EmbedKGQA, CronKGQA and TempoQR, their overall performances are not reported due to they cannot handle questions with boolean and numeric answers.

The performance of baselines. For PLM baselines, compared to the performance of original PLMs with their TKG-enhanced variants, the TKG information greatly benefits the performances of TKGQA. Additionally, equipped with our optimized TKGE, PLMs significantly improve their multi-step temporal reasoning ability. For the KGQA baseline, EmbedKGQA shows its limited ability in questions with time answers due to its unawareness of time. When answering the entity-answer questions, it could leverage the structured information between entities and achieve better results than original PLMs.

With temporal information derived from TKG, TKGQA baselines can infer temporal information and perform temporal reasoning. Both CronKGQA and TempoQR outperform EmbedKGQA by a large margin. TempoQR shows its superiority in dealing with entity-answer and time-answer questions. For example, TempoQR achieves 0.371 Hits@1, while the counterparts of EmbedKGQA and CronKGQA are 0.155 and 0.278, respectively. However, the per-

Model	Overall	Entity		Time		Boolean	Numeric
	Hits@1	Hits@1	Hits@10	Hits@1	Hits@10	Hits@1	Hits@1
RoBERTa	0.303	0.143	0.465	0.041	0.342	0.714	0.282
BERT	0.303	0.117	0.431	0.045	0.342	0.724	0.294
EmbedKGQA	-	0.155	0.532	0.017	0.088	-	-
CronKGQA	-	0.278	0.623	0.065	0.455	-	-
TempoQR	-	0.371	0.733	0.083	0.538	-	-
RoBERTa$_{TC}$	0.383	0.416	0.765	0.087	0.533	0.732	0.277
BERT$_{TC}$	0.386	0.406	0.773	0.079	0.527	0.744	0.294
RoBERTa$_{AL}$	0.395	0.448	0.799	0.089	0.542	0.737	0.286
BERT$_{AL}$	0.399	0.443	0.806	0.091	0.546	0.745	0.297
MusTKGQA	**0.547**	**0.591**	**0.873**	**0.297**	**0.799**	**0.907**	**0.373**

Table 6: Experimental results on MusTQ. The results in **bold** and underline denote the best and second results, separately. "-" denotes the corresponding models cannot deal with boolean-answer and numeric-answer questions.

Model	Hits@1		
	2 steps	3 steps	4 steps
RoBERTa	0.332	0.293	0.097
BERT	0.337	0.288	0.090
RoBERTa$_{TC}$	0.351	0.421	0.333
BERT$_{TC}$	0.360	0.418	0.322
RoBERTa$_{AL}$	0.357	0.435	0.375
BERT$_{AL}$	0.365	0.435	0.384
MusTKGQA	0.579	0.518	0.506

Table 7: Experimental results (Hits@1) on questions with different temporal reasoning steps on MusTQ.

formance of TempoQR is still worse than the PLM variants. This is because TempoQR is initially for single-step temporal reasoning and models limited entity information, while the PLM variants consider all mentioned entities in question.

MusTKGQA VS. All. MusTKGQA outperforms all baselines in all answer types. It outperforms the best baseline models (*i.e.*, BERT$_{AL}$) by 0.148 Hits@1. Compared with BERT$_{AL}$, MusTKGQA increases 0.148, 0.206, 0.162, 0.076 Hits@1 for entity-, time-, boolean-, and numeric-answer questions, respectively, demonstrating its effectiveness in multi-step temporal reasoning.

Fine-grained Multi-Step Reasoning Comparison. We further analyze the ability of PLMs and MusTKGQA to tackle questions with different reasoning steps. As shown in Table 7, MusTKGQA outperforms the baseline models for questions w.r.t different reasoning steps. For 4-step reasoning questions, MusTKGQA gets great improvements of 0.122 Hits@1 over the best baseline BERT$_{AL}$.

Fine-grained Single-Step Reasoning Comparison. CronQuestions (Saxena et al., 2021) is a widely-used TKGQA dataset with single-step reasoning questions. We also measure the single-step

Model	Hits@1		
	Overall	Entity	Time
RoBERTa	0.070	0.082	0.048
BERT	0.071	0.077	0.060
EmbedKGQA	0.288	0.411	0.057
CronKGQA	0.647	0.699	0.549
TempoQR	0.799	0.876	0.653
MusTKGQA	0.875	0.871	0.881

Table 8: Experimental results on CronKGQA (Hits@1).

Model	Hits@1			
	Overall	2 steps	3 steps	4 steps
MusTKGQA	0.547	0.579	0.518	0.506
-Seq	0.540	0.571	0.513	0.493
-Fact	0.543	0.574	0.518	0.466
-Seq&Fact	0.540	0.576	0.510	0.468
MusTKGQA$_{TC}$	0.541	0.571	0.514	0.489

Table 9: Hits@1 results of different MusTKGQA variants on MusTQ w.r.t different reasoning steps.

temporal reasoning performance of MusTKGQA as well as baselines on CronQuestions. As shown in Table 8, MusTKGQA outperforms all baselines, indicating its strong single-step reasoning ability.

5.3 Ablation Study

We introduce the following variants by moving the modules in MusTKGQA: (1) *-Seq* and (2) *-Fact* remove the sequence alignment and fact alignment in the original MusTKGQA. (3) *-Seq&Fact* removes both alignments. (4) *MusTKGQA$_{TC}$* replaces the TKGE module from TNTComplEx to TComplEx. Table 9 shows that two alignment tasks greatly improve the effects of MusTKGQA, and TNTComplEx could enhance the information fusion in optimization, verifying the rationality of our model.

6 Conclusion

We propose the first multi-step temporal reasoning TKGQA dataset, MusTQ, which constructs multi-step reasoning under the guidance of a well-established measure theory. MusTQ is centered on temporal reasoning and is challenging for existing TKGQA models. Further, we establish the MusTKGQA model with entity-time attention mechanism and optimized TKG representation to tackle temporal reasoning in multi-steps. Experimental results on MusTQ show that our model achieves state-of-the-art multi-step temporal reasoning performance. The results on the previous CronQuestions dataset also show the strong ability of our model.

Limitations

While we show the multi-step temporal reasoning in the MusTQ dataset, there are some limitations worth considering in future work: (1) Our MusTQ is constructed through manual templates. Though paraphrasing is used in the construction process, the dataset might be limited in linguistic diversity. (2) Following (Saxena et al., 2021), our dataset also adopts the time granularity down to the year. Thus, the dataset cannot capture the fine-grained time information with year level.

Ethical Considerations

In this section, we consider the potential ethical issues of our work. In this paper, we propose the MusTQ dataset with multi-step temporal reasoning questions and a TKG. The questions are collected via manually-constructed templates. During template collection, the salary for each human annotator is determined by the average time of annotation and local labor compensation standards. For the TKG, we use the subset of WikiData (Lacroix et al., 2020), and the corresponding license is the CC0 License, which is granted to copy, distribute and modify the contents.

Acknowledgements

We thank the anonymous reviewers for their valuable comments. This work is supported by the National Natural Science Foundation of China (No.62072323, U21A20488, No.62102276), Shanghai Science and Technology Innovation Action Plan (No.22511104700), China Postdoctoral Science Foundation (Grant No. 2023M732563), Zhejiang Lab Open ResearchProject (No.K2022NB0AB04), the Natural Science Foundation of Jiangsu Province (Grant No. BK20210705) and Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250. ACM.

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, Toronto, Canada. Association for Computational Linguistics.

Ziyang Chen, Xiang Zhao, Jinzhi Liao, Xinyi Li, and Evangelos Kanoulas. 2022. Temporal knowledge graph question answering via subgraph reasoning. *Knowledge-Based Systems*, page 109134.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Zifeng Ding, Zongyue Li, Ruoxia Qi, Jingpei Wu, Bailan He, Yunpu Ma, Zhao Meng, Shuo Chen, Ruotong Liao, Zhen Han, et al. 2023. Forecasttkgquestions: A benchmark for temporal question answering and forecasting over temporal knowledge graphs. In *International Semantic Web Conference*, pages 541–560. Springer.

Alberto García-Durán, Sebastijan Dumancic, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4816–4821. Association for Computational Linguistics.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018. Tempquestions: A benchmark for temporal question answering. In *Companion Proceedings of the The Web Conference 2018*, pages 1057–1062. ACM.

Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. 2021. Complex temporal question answering on knowledge graphs. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 792–802. ACM.

Songlin Jiao, Zhenfang Zhu, Wenqing Wu, Zicheng Zuo, Jiangtao Qi, Wenling Wang, Guangyuan Zhang, and Peiyu Liu. 2023. An improving reasoning network for complex question answering over temporal knowledge graphs. *Applied Intelligence*, 53(7):8195–8208.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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. *arXiv preprint arXiv:1907.11692*.

Yonghao Liu, Di Liang, Mengyu Li, Fausto Giunchiglia, Ximing Li, Sirui Wang, Wei Wu, Lan Huang, Xiaoyue Feng, and Renchu Guan. 2023. Local and global: Temporal question answering via information fusion. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 5141–5149. ijcai.org.

Shaonan Long, Jinzhi Liao, Shiyu Yang, Xiang Zhao, and Xuemin Lin. 2022. Complex question answering over temporal knowledge graphs. In *International Conference on Web Information Systems Engineering*, pages 65–80. Springer.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

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 and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6663–6676, Online. Association for Computational Linguistics.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507, Online. 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, Dublin, Ireland. Association for Computational Linguistics.

Stanley Smith Stevens. 1946. On the theory of scales of measurement. *Science*, 103(2684):677–680.

Théo Trouillon, Christopher R. Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, and Guillaume Bouchard. 2017. Knowledge graph completion via complex tensor factorization. *J. Mach. Learn. Res.*, 18:130:1–130:38.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Yao Xiao, Guangyou Zhou, and Jin Liu. 2022. Modeling temporal-sensitive information for complex question answering over knowledge graphs. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 418–430. Springer.

A Base Template Construction

We invite five graduate students specializing in Knowledge Graphs to serve as human experts for dataset construction. Before template construction, we first provide them with an illustration of all three single-step temporal reasoning types adopted in the base template. Then we show them some example templates in existing datasets. As the examination, we ask all human experts to write three base templates for each reasoning type that query the facts with the 'win' relation. Based on the analysis of templates written in the examination, we put forward the following requirements for template construction:

1) The template must be a question and its statement should be fluent.

2) Each template should adopt one given single-step temporal reasoning type and query one relation.

3) The entities and the type-specific reasoning information in the template should be annotated as corresponding slots. In detail, the head entity and tail entity should be annotated as {head} and {tail} separately. The ordinal number is represented as {ordinal}. The time difference is denoted as {diff} and the count number is shown as {count}.

Next, we ask the human experts to make templates for four frequently occurring relations *Position held*, *Member of sports team*, *Nominated for*, and *Win* based on the proposed template construction requirements. To ensure the template is satisfying, we ask all experts to vote. The template was considered desirable if more than half of the votes passed. Finally, we obtain 63 unique base templates.

B Example Questions In MusTQ

We give some examples to show the multi-step temporal reasoning questions of MusTQ in Table 10.

C TComplex & TNTComplex

Temporal Knowledge Graph Embedding is a method for TKG representation. For each fact (s, r, o, τ) in TKG, the entities s, o, relation r and timestamp τ are embedded as low-dimensional vectors. Lacroix et al. (2020) utilizes tensor decomposition and proposed the TComplEx and TNTComplEx based on ComplEx (Trouillon et al., 2017). ComplEx is a KG embedding methods. For each fact (s, r, o) in static KG, all entities and relation are embeded into Complex vector space $\mathbb{C}$. In detail, ComplEx represents head entity as $\boldsymbol{e}_s \in \mathbb{C}^{d\times 1}$, tail entity as $\boldsymbol{e}_o \in \mathbb{C}^{d\times 1}$ and relation as $\boldsymbol{v}_r \in \mathbb{C}^{d\times 1}$ by applying the score function $\phi(s, r, o)$.

$$\begin{aligned}\phi(s, r, o) &= \Re(< \boldsymbol{e}_s, \boldsymbol{v}_r, \overline{\boldsymbol{e}}_o >) \\ &= \Re(\sum_{i=1}^{d} \boldsymbol{e}_{s_i} \boldsymbol{v}_{r_i} \overline{\boldsymbol{e}}_{o_i})\end{aligned}$$

where $\overline{\boldsymbol{e}}$ represents the complex conjugate of e, $\Re$ denotes to adopt the real part of the complex number in the score function.

In TComplEx, the timestamp of temporal facts is also denoted as complex vector $\boldsymbol{t}_\tau \in \mathbb{C}^{d\times 1}$ through the score function $\phi(s, r, o, \tau)$.

$$\begin{aligned}\phi(s, r, o, \tau) &= \Re(< \boldsymbol{e}_s, \boldsymbol{v}_r, \overline{\boldsymbol{e}}_o, \boldsymbol{t}_\tau >) \\ &= \Re(< \boldsymbol{e}_s, \boldsymbol{v}_r \odot \boldsymbol{t}_\tau, \overline{\boldsymbol{e}}_o >)\end{aligned}$$

$\odot$ denotes the element-wise product. Since some facts may not be affected by time, TNTComplEx introduces the non-temporal part together with TComplEx. The score function for TNTComplEx denotes as $\Re(< \boldsymbol{e}_s, \boldsymbol{v}_r, \overline{\boldsymbol{e}}_o, \boldsymbol{t}_\tau > + < \boldsymbol{e}_s, \boldsymbol{u}_r, \overline{\boldsymbol{e}}_o, \mathbf{1} >)$, where $\boldsymbol{u}_r$ represents the temporal agnostic relation representation.

D Implementation Details

All experiments are implemented with PyTorch (Paszke et al., 2019) on a server equipped with an Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz and a GeForce GTX 1080 Ti GPU. We adopt the dimension for TKG representation d as 512. In the experiments, we adopt the original BERT-base, RoBERTa-base, and DistillBERT-base (Sanh et al., 2019) for text semantics encoding. Same as exiting TKGQA models, we only retrieve the information embedded by the TKGE model and the LM models without further updating in MusTQ. The transformer models for both TKGE fusion and final question representation have 3 layers and 8 attention heads. We choose the Adam (Kingma and Ba, 2015) optimizer in the training process and the learning rate is 2e-5. In all, MusTQ has 120M trainable parameters and the average training time is 20 GPU hours.

The training process of the enhanced TKGE is generally the same as the original TKGE method (i.e., TNTComplEx) training process. The five-tuple and the triples would transform into the quadruples in training. Specifically, the five-tuple

Type	Sampled Template	Example Questions
2 steps	Following {timeA}, did {head} secure {tailA} {diff} years earlier than winning {tailB}?	Following Order of Georgi Dimitrov was given to Gherman Titov, did Valentina Stepanovna Grizodubova secure Order of Lenin five years earlier than winning Order of the October Revolution?
3 steps	How many award nominations has {head} received in total in the time span from {timeA} to {timeB}?	How many award nominations has Arkin Alan received in total in the time span from Donald Madden was proposed as Tony Award for Best Actor in a Play to Haing S. Ngor got Academy Award for Best Supporting Actor?
4 steps	Who took on the {ordinal} job as {tail} in {time} during the phase between {timeA} and {timeB}?	Who took on the third job as Teachta Dala in 1981 during the phase between Mark Clinton became Minister for Agriculture, Food and the Marine and Alan James Dukes became Minister for Finance?

Table 10: Examples of multi-step temporal reasoning questions on MusTQ.

knowledge could be trained by the following strategy: In the original TKGE method, for a five-tuple knowledge $(s, r, o, \tau_s, \tau_e)$ the model would sample one timestamp between (τ_s, τ_e) to construct the golden quadruple (s, r, o, τ_{sam}). For the triples we construct in alignment, we leverage the dummy time embedding to fill the absent timestamp information. In answer reasoning, we would leverage the dummy time embedding to represent the embedding of time τ mentioned in question q if none is present.

We adopt the training strategy that implements all four classification tasks simultaneously to fully leverage data in the training process since there is an overlap in the types of temporal reasoning involved in the questions for each answer type. We keep the boolean prediction and numeric prediction tasks. For entity and time prediction tasks, following previous TKGQA models(Saxena et al., 2021; Mavromatis et al., 2022), we merge them by concatenating their prediction scores and calculating the answer probability over the combined scores.

Additionally, in order to obtain TKG-enhanced question representation of PLM variants, we concatenate each entity mentioned in question with the question representation obtained by PLMs and the mentioned timestamp, during which we leverage the TKG embeddings to represent the mentioned entity and timestamp. Then we fuse the information through projection and add the fusion information of each entity mentioned in question to acquire the final TKG-enhanced question representation.