Multi-granularity Temporal Question Answering over Knowledge Graphs

Ziyang Chen[1], **Jinzhi Liao**[2], **Xiang Zhao**[1,*]
[1] Laboratory for Big Data and Decision, National University of Defense Technology, China
[2] National Defense University, China
{chenziyangnudt, liaojinzhi12, xiangzhao}@nudt.edu.cn

Abstract

Recently, question answering over temporal knowledge graphs (i.e., TKGQA) has been introduced and investigated, in quest of reasoning about dynamic factual knowledge. To foster research on TKGQA, a few datasets have been curated (e.g., CronQuestions and Complex-CronQuestions), and various models have been proposed based on these datasets. Nevertheless, existing efforts overlook the fact that real-life applications of TKGQA also tend to be complex in temporal granularity, i.e., the questions may concern mixed temporal granularities (e.g., both day and month). To overcome the limitation, in this paper, we motivate the notion of multi-granularity temporal question answering over knowledge graphs and present a large-scale dataset for multi-granularity TKGQA, namely MultiTQ. To the best of our knowledge, MultiTQ is among the first of its kind, and compared with existing datasets on TKGQA, MultiTQ features at least two desirable aspects—ample relevant facts and multiple temporal granularities. It is expected to better reflect real-world challenges, and serve as a test bed for TKGQA models. In addition, we propose a competing baseline MultiQA over MultiTQ, which is experimentally demonstrated to be effective in dealing with TKGQA. The data and code are released at https://github.com/czy1999/MultiTQ.

1 Introduction

In real-life applications factual knowledge is apt to evolve over time (Nonaka et al., 2000; Roddick and Spiliopoulou, 2002; Hoffart et al., 2011; Gottschalk and Demidova, 2018); for instance, The host city of the Winter Olympic Games in 2018 was South Korea, while in 2022 it was Beijing.In this connection, there is a current trend to investigate knowledge graphs (KGs) involving *time*, and these KGs are coined as *temporal* knowledge graphs (TKGs). In a TKG, fact triplets are equipped with temporal information (e.g., timestamps), and a *temporal fact* can be stated in the form like "(*Beijing, held, Winter Olympic Games, 2022*)".

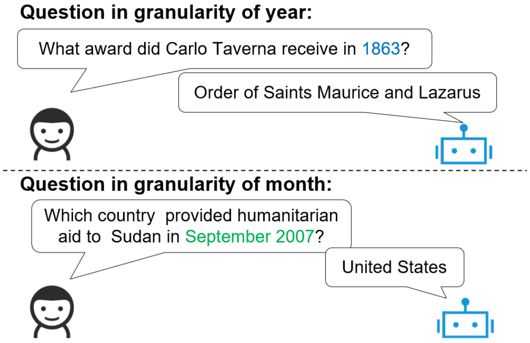
Question in granularity of year:
What award did Carlo Taverna receive in 1863?
Order of Saints Maurice and Lazarus
Question in granularity of month:
Which country provided humanitarian aid to Sudan in September 2007?
United States

Figure 1: Examples of temporal question answering.

To exploit the value of TKGs, recent research effort has been devoted to process natural language questions over TKG, i.e., question answering over TKG (TKGQA in short) (Saxena et al., 2021). Given a question and a background TKG, it retrieves from the TKG an answer to the question. To foster research on TKGQA, several datasets have been introduced, among which CronQuestions (Saxena et al., 2021) is by far the largest. We explain the task with a sample question in CronQuestions.

Example 1 *In the upper part of Figure 1, the agent is supplied with the question* "What award did Carlo Taverna receive in 1863?" *as well as a TKG. By considering the semantic relevance of the facts, the agent locates the candidate fact "(*Carlo Taverna, receive, Order of Saints Maurice and Lazarus, *1863)", and hence,* Order of Saints Maurice and Lazarus *is returned as the answer.*

Specifically, CronQuestions comprises 410k *temporal* questions, each of them has a temporal constraint, e.g., "*in 1863*" in the example above. Albeit large scale, the questions in CronQuestions tend to be "pseudo-temporal" (Chen et al., 2022). By looking into the construction of Cron-

[*] Corresponding author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 11378–11392
July 9-14, 2023 ©2023 Association for Computational Linguistics

QUESTIONS, we find that most of the questions are related to, respectively, only one fact, which can be well located without enforcing the temporal constraint in the question; for example, *Carlo Taverna* only received one award, which was *Order of Saints Maurice and Lazarus*. In this case, the temporal constraint does not further restrict the candidate facts to answering the question, and the question is essentially *atemporal* in the context of the given KG. Moreover, in CRONQUESTIONS, questions and the TKG are designed to be both described in the temporal granularity of *year*. This simplification, however, is less practical, since questions and knowledge in the real world are not limited to the time frame of years. For instance, as shown in the lower part of Figure 1, the agent is likely to be given a question in the granularity of month, which is common in the real world. In short, these two important aspects are not well attended by existing TKGQA datasets, which thus may be insufficient in evaluating TKGQA models.

In this research, we are motivated to address the shortcomings by presenting a new dataset for TKGQA, namely MULTITQ. MULTITQ is a large-scale dataset featuring *ample relevant facts* and *multiple temporal granularities* (comparison of statistics in Table 1). To avoid the pseudo-temporal issue, we intentionally generate temporal questions that are relevant to more than one fact triplet, such that the temporal constraint is always necessary to correctly locate the answer. This characteristic is of importance to evaluating TKGQA models, since temporal reasoning is a unique challenge arising out of the task. Further, MULTITQ features multiple temporal granularities, which is largely overlooked by existing datasets. We resort to a template-based question generation method, which automatically constructs question templates (and hence questions) of multiple temporal granularities. In this way, MULTITQ is expected to serve as a test bed for evaluating TKGQA models, especially in reasoning with temporal constraints and coordinating between temporal granularities. In addition, to provide a competing baseline on MULTITQ, we propose a transformer-based model for multi-granularity TKGQA, namely MultiQA.

In summary, our contribution is three-fold:

- To the best of our knowledge, we are among the first to elicit the notion and motivate the challenges of multi-granularity TKGQA.
- We present a multi-granularity TKGQA dataset MULTITQ. Besides multiple temporal granularities, the dataset is also prominent in its large scale with ample relevant facts regarding each questions therein.
- We propose MultiQA, a strong baseline to handle multi-granularity TKGQA, the performance of which is demonstrated by comprehensive experiments on MULTITQ.

2 Related Work

2.1 Datasets for TKGQA

TEMPQUESTIONS (Jia et al., 2018a) is one of the first publicly available TKGQA datasets consisting of 1,271 questions. SYGMA (Neelam et al., 2021) introduced a subset of TEMPQUESTIONS that can be answered over `Wikidata` called TEMPQA-WD. Previous collections on temporal questions contain only about a thousand questions and are not suitable for building neural models. TIMEQUESTIONS (Jia et al., 2021) searches through eight datasets of question answering over conventional KGs for time-related questions and contains 16k questions. CRONQUESTIONS (Saxena et al., 2021) is another TKGQA dataset that uses its KG drawn from `Wikidata`, which comprises a total of 410k questions. While it alleviates problem of incomplete learning of large models due to small amount of data, CRONQUESTIONS contains a large number of pseudo-temporal questions (Chen et al., 2022). This reduces the applicability of CRONQUESTIONS for evaluating the temporal reasoning capability of TKGQA models.

Since these datasets focus on single-time granularity, consistent with the KGs, they do not reflect the real-world challenges of multi-granularity temporal question answering. It motivates us to close the gap by presenting a novel dataset for TKGQA.

2.2 TKGQA Models

There are two streams of approaches to tackle TKGQA. The first decomposes the original question into several non-temporal questions and time constraints. Then models designed for question answering over conventional KGs are applied to answer these questions, and time constraints finally compare and select the most proper answer, e.g., TEQUILA (Jia et al., 2018b). However, this approach needs handcrafted decomposition rules and cannot cope with complex questions (Jia et al., 2021).

The methods in the second stream try to acquire TKG embedding to calculate the semantic similarities for the answer determination. CronKGQA (Saxena et al., 2021) provides a learnable reasoning process for TKGQA, which does not rely on handcrafted rules. Although CronKGQA performs well in answering simple questions, it fails to solve complex questions requiring inference of certain time constraints. TempoQR (Mavromatis et al., 2021) introduces time scope information for each question and employs EaE method (Févry et al., 2020) to enhance the semantic information of the question representation.

However, limited by the single granularity of available datasets, none of these methods have considered the multi-granularity problem, making them lacking in real-world applications. In this paper, we address the challenges by proposing multi-granularity temporal QA methods, MultiQA.

2.3 Analysis for Temporal Questions

It is noted that temporality also gains attention in community question answering (CQA) and multi-modal question answering (MQA).

Models (Duan et al., 2018; Wu et al., 2017; Zhang et al., 2020) and datasets (Pal et al., 2012; Figueroa, 2010; Figueroa et al., 2016, 2019) for temporal community questions are emerging in recent years. There are two viewpoints for temporality across CQA sites: 1) a measure of the usefulness of the answers (Pal et al., 2012), and 2) the recurrent attention given to questions during different time-frames (Figueroa et al., 2016). Based on these two viewpoints, a new set of time-frame specific categories are proposed (Figueroa et al., 2019).

In the field of multimodal question answering, a series of temporal question answering datasets integrating audio and video have been proposed (Lei et al., 2020; Fayek and Johnson, 2020; Jang et al., 2017). Techniques such as spatio-temporal attention (Jang et al., 2017), motion-appearance memory (Gao et al., 2018), spatio-temporal grounded audio-visual network (Li et al., 2022) and spatio-temporal graph models (Cherian et al., 2022) have been proposed and demonstrated their effectiveness on different VideoQA and AudioQA datasets.

3 The MultiTQ Dataset

MultiTQ is a new complex temporal question answering dataset with multi-granularity temporal information. Compared to existing datasets, our dataset features in a few advantages, including large scale, ample relations and multiple temporal granularity, which hence better reflects real-world scenarios, as shown in Table 1.

3.1 Analysis of KG

Most TKGQA datasets use `Wikidata` as the KG. However, `Wikidata` suffers from relation-sparsity problem. Specifically, for each entity in the KG, the number of relation types involved is fairly homogeneous. We define semantic complexity.

Definition 1 *For a TKG $\mathcal{K} := (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$, where $\mathcal{E}, \mathcal{R}, \mathcal{T}$ denote entities, relations, and timestamps respectively. KG semantic complexity $SC_{\mathcal{K}}$ is defined as the average of the number of relation types involved in each entity:*

$$SC_{\mathcal{K}} = \frac{1}{|\mathcal{E}|} \sum_{e_i \in \mathcal{E}} N_{e_i}^{Rtype}, \tag{1}$$

where $N_{e_i}^{Rtype}$ is the number of relation types involved in e_i.

A larger $SC_{\mathcal{K}}$ indicates a richer KG $\mathcal{K}$ in terms of relation semantic information. For example, CronQuestions uses a subset of `Wikidata` as the KG, but the SC value of this KG is only 1.32, i.e., an average of 1.32 types of relation per entity, which is more descriptive, with over 80% of entities having only one relation type and 99% of entities having no more than two relation types. Thus, even though it contains a rich number of entities and relations, the KG of CronQuestions is fairly sparse at semantic level.

Unlike previous datasets, we take `ICEWS05-15` (García-Durán et al., 2018), a subset from the Integrated Crisis Early Warning System (`ICEWS`) database, as the KG for MultiTQ. `ICEWS` captures and processes millions of pieces of data from digital news media, social media and other sources, with a wealth of dynamic semantic information that provides an adequate KG for temporal question answering. As shown in Table 2, `ICEWS05-15` is rich in semantic information with SC value 7.05. The richness of relation types makes it more in line with real-life scenarios.

3.2 Question Construction

Following CronQuestions, we filter through `ICEWS05-15` to find 22 most frequent relations to build templates and generate questions.

Dataset	KG	SC Value	Multiple Temporal	No Pseudo-temporal	Multi-Granularity	#Questions
TEMPQUESTIONS	FreeBase	/	✗	✗	✗	1,271
TEMPQA-WD	FreeBase,Wikidata	/	✗	✗	✗	839
TIMEQUESTIONS	WikiData	/	✗	✗	✗	160k
CRONQUESTIONS	WikiData	1.32	✓	✗	✗	410k
Complex-CRONQUESTIONS	WikiData	1.32	✓	✓	✗	45k
MULTITQ	**ICEWS**	**7.05**	✓	✓	✓	**500k**

Table 1: Comparison of TKGQA datasets. SC value denotes semantic complexity of a KG.

	Wikidata Subset	ICEWS05-15
Entities	125,726	10,488
Relations	203	251
Timestamps	1,643	4,017
Fact triplets	328,635	479,329
SC value	1.32	7.05
Time Span	0 - 9620[1]	2005 - 2015

Table 2: Statistics for various KGs.

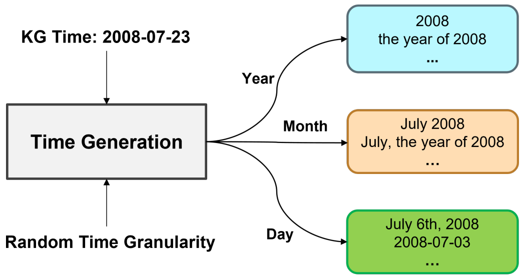

Figure 2: Multi-granular time generation.

Category	Representative expanded templates
Equal	Who visited to {tail} in {time}?
Before/After	Before {tail2}, who visited {tail}?
First/Last	Who first visited {tail}?
Equal Multi	Who visited {tail} on the same year of {tail2}?
Before Last	Who visited {tail} last before {tail2} did?
After First	After {time}, Who visited {tail} first?

Table 3: Representative expanded templates for core template *'Who first visited {tail}'*.

		Train	Dev	Test
Single	Equal	135,890	18,983	17,311
	Before/After	75,340	11,655	11,073
	First/Last	72,252	11,097	10,480
Multiple	Equal Multi	16,893	3,213	3,207
	After First	43,305	6,499	6,266
	Before Last	43,107	6,532	6,247
Total		386,787	587,979	54,584

Table 4: Statistics of question categories in MULTITQ.

Firstly, 246 unique core templates are constructed by five experts in social computing based on the 22 most frequently occurring relations. Taking the relation 'make a visit' as an example, human experts have constructed several core templates based on their expert knowledge, e.g., "Who first visited {tail}". Next, the core template will be expanded by the question category (cf. Section 3.2.2). Time constraints and multi-granularity temporal information are added to the core template, as shown in Table 3, enriching and diversifying the semantics of templates. Finally, we ended up with 7,334 templates. Each of these templates has a corresponding procedure that could be executed over the TKG to extract all possible answers for that template. These templates were then filled using entity aliases from ICEWS to generate 500k unique question-answer pairs.

3.2.1 Multi-Granularity Temporal Questions

Time is naturally multi-granular, a property that previous models have ignored. The motivation for proposing Multi-Granularity Temporal Questions is to drive the attention of temporal questions reasoning on multi-granular time.

ICEWS provides time information at a day granularity, which allows us to generate higher granularity information, such as year and month granularity. Questions in MULTITQ contain three temporal granularities, i.e., day, month and year. In order to generate multi-granular time information, we have designed a time generation module that can randomly generate different formats and types of year-month-day granularity time expressions from the daily granularity time according to syntactic criteria, as shown in Figure 2, effectively increasing the variety and complexity of the question texts.

3.2.2 Question Categorization

To make the problem more challenging, we propose the concept of multiple temporal reasoning questions, where there are multiple temporal constraint words in one question and the QA model needs multiple complex reasoning to obtain final. We categorize questions into "Single questions" and "Multiple questions". Please refer Table 5 for examples of these questions.

[1] This abnormality is brought by some science fiction-type knowledge, and some erroneous time information.

Single questions. These questions contain a single temporal constraint, where the answer can be either an entity or a time instance. For example, the question “Who visited the United States in 2008?” requires a single temporal constraint to answer the question, namely *Equal*. In our dataset, single questions are further categorized into three types: *Equal*, *Before/After* and *First/Last*.

Multiple questions. These questions contain multiple temporal constraints to answer and can be more varied. For example “Which country first visited United States in 2015?” This requires reasoning over multiple temporal constraints, including $Equal$ and $First$. In our dataset, Multiple questions are further categorized into three types: *Equal multi*, *Before last* and *After first*.

3.2.3 Question Filtering and Splitting

We follow CronQuestions and ensure that there is no entity overlap between train questions and test questions. This policy ensures that models are doing temporal reasoning rather than guessing from entities seen during training. Specifically, we split the ICEWS05-15 into train/dev/test folds without entity overlap, and then perform question generation protocol on each divided TKG.

Automatic question generation via templates may lead to some questions with low quality, including pseudo-temporal questions and nonsensical questions. To compensate for these shortcomings, we follow Chen et al. (2022) and eliminate all pseudo-temporal questions, making the dataset more challenging. Furthermore, due to the factual sparsity of the KG, automatic generation through templates may result in questions such as "Who visited the United States in 2005?" where there may be hundreds of answers. To avoid this, we eliminate questions with more than 20 answers to ensure that the questions in the dataset are of practical importance.

Finally, we get train/dev/test folds with a ratio of roughly 8:1:1, and 500k questions in total. Dataset statistics are shown in Table 4. We believe that providing entity and time annotations directly would significantly affect the performance of the model, reducing reasoning on simple questions to a KG query task. Therefore, we do not provide corresponding entity and time annotations in our dataset. Summarizing, each of our examples contains a natural language temporal question and a set of ‘gold’ answers (entity or time).

Property	Sample Question
By question type	
Equal	*Which country provided humanitarian aid to Sudan in 2007?*
Before/After	*Who commended the Military of Mali before the Armed Rebel of Mali did?*
First/Last	*When did the Militant of Taliban first commend the Government of Pakistan?*
Equal Multi	*In 2012, who last did Barack Obama appeal for?*
Before Last	*Who was threatened by Benjamin Netanyahu last before Middle East?*
After First	*Who first wanted to negotiate with Evo Morales after the Citizen of Brazil did?*
By time granularity	
Year	*Who first made Abu Sayyaf suffer from conventional military forces In 2015?*
Month	*In Dec, 2008, who would wish to negotiate with the Senate of Romania?*
Day	*In Jul 21st, 2011, who criticized the Media of Ecuador?*
By answer type	
Entity	*Which country visited Japan in 2013?*
Time	*When did China express intent to meet with the Government of Pakistan?*

Table 5: Representative examples from MultiTQ.

Statistic	Train	Dev.	Test
#tokens per question	13.50	11.28	11.41
#tokens per answer	2.15	2.02	1.96
#answers per question	1.88	2.36	2.43
#entities per question	1.61	1.65	1.64
#distinct words	14,714	5,712	5,843
#distinct timestamps	4,159	3,787	3,763

Table 6: Core statistics of each split in MultiTQ

3.3 Statistics of MultiTQ

We summarize the number of questions in MultiTQ across different types in Table 4, and the core statistics of each split in Table 6. In Table 5, we present sample questions from MultiTQ as per question type, time granularity and answer type.

Overall, the resulting MultiTQ dataset contains 500k questions from 22 relations (More statistic are listed in Appendix A.3). In Figure 3, we show how questions in our benchmark are distributed by length (in words), and contrast this with CronQuestions and TempQuestions. Questions in our benchmark are between 4 and 35 words long, and the average question length is 13.01 words. The figure shows that a good proportion of questions in MultiTQ are relatively verbose, implying increased parsing difficulty for QA systems.

4 The MultiQA Model

We overview the model architecture in Figure 4.

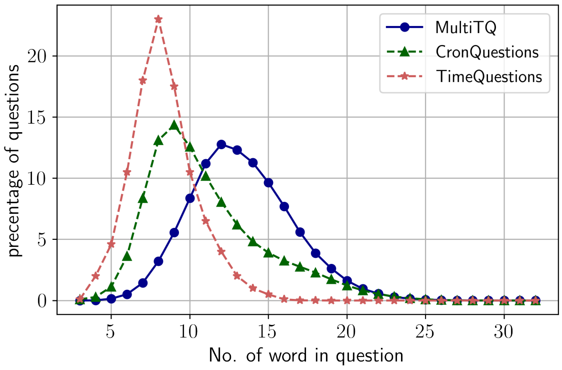

Figure 3: Length distribution of datasets.

4.1 Question Pre-processing

To obtain the entity and time information in the question, we use named entity recognition (NER) and time extraction tools to enable the model to more accurately access the information in the KG by aligning it with the entities and times in the KG (See Appendix A.2).

We obtain the semantic information of questions by a pre-trained language model. Specifically, the natural language form of the question $\mathbf{q}_{\text{text}}$ is transformed into a semantic matrix $\mathbf{Q}_R$ by the pre-trained RoBERTa (Liu et al., 2019).

$$\mathbf{Q}_R = \mathbf{W}_R \, \text{RoBERTa}\left(\mathbf{q}_{\text{text}}\right), \tag{2}$$

where $\mathbf{Q}_R = [\mathbf{q}_{\text{CLS}}, \mathbf{q}_{R_1}, ..., \mathbf{q}_{R_N}]$ is a $D \times L$ embedding matrix. L is the number of tokens and D is the dimensions of the TKG embeddings. $\mathbf{W}_R$ is a $D \times D_{roberta}$ projection matrix where $D_{roberta}$ is the dimension of the RoBERTa embeddings. The finial question representation $\mathbf{q} = \mathbf{q}_{\text{CLS}}$.

4.2 Multi-Granularity Time Aggregation

As the facts provided by the KG are all at day granularity, e.g., 2008-03-19, the TKG embedding thus trained contains only semantic information at day granularity (See Appendix A.1). However, the question contains reasoning about year and month granularity, and no semantic information can be obtained directly from the pre-trained TKG embeddings. To solve this problem and obtain time embeddings at a coarser granularity, we propose a multi-granularity time aggregation module. Taking the example of month granularity time aggregation, we want to aggregate all related day information to get that of month granularity in the question.

Specifically, For the month granularity time m in the question, we first extract all contained day timestamps $d_1, d_2, ..., d_N$ and their TKG embeddings $\mathbf{t}_{d_1}, \mathbf{t}_{d_2}, ..., \mathbf{t}_{d_N}$, which are rich in temporal information. N is the number of related days. To obtain the time representation at month granularity, we construct the temporal semantic matrix $\mathbf{T}_d$,

$$\mathbf{T}_d = [\mathbf{t}_{d_1}, \mathbf{t}_{d_2}, ..., \mathbf{t}_{d_N}], \tag{3}$$

where $\mathbf{T}_d \in \mathbb{R}^{N \times D}$ is a matrix containing all day embeddings for month m.

Time as an ordering sequence has an inherent similarity to the positions of words in the text, so we enrich its sequential property by employing a sinusoidal position encoding method (Vaswani et al., 2017; Jia et al., 2021). Here, the k^{th} position in $\mathbf{T}_d$ will be encoded as:

$$PE(k,j) = \begin{cases} \sin\left(k/10000^{\frac{2i}{D}}\right), & \text{if } j = 2i \\ \cos\left(k/10000^{\frac{2i}{D}}\right), & \text{if } j = 2i+1 \end{cases} \tag{4}$$

where j is the (even/odd) position in the D-dimensional vector. Further, we get $\mathbf{T}'_d$ by adding positional embedding to $\mathbf{T}_d$. Adding positional embedding ensures sequential ordering among the timestamps, which is vital for reasoning signals like before and after in temporal questions.

Next, we propose an information fusion layer to fuse the information into a single time representation $\mathbf{t}_m$. Following Févry et al. (2020), we use an information fusion layer that consists of a dedicated learnable encoder $Transformer(\cdot)$ which consists of 2 Transformer encoding layers (Vaswani et al., 2017). This encoder allows the time tokens to attend each other, which fuses all days' embeddings into a single month embedding. The final token embedding matrix $\mathbf{T}_m$ is calculated as

$$\mathbf{T}_m = \text{Transformer}(\mathbf{T}'_d), \tag{5}$$

where $\mathbf{T}_m = [\mathbf{t}_{\text{CLS}}, \mathbf{t}_{m_1}, ..., \mathbf{t}_{m_N}]$, and the finial question representation $\mathbf{t}_m = \mathbf{t}_{\text{CLS}}$.

Repeating the aggregation, we obtain a time representation of year granularity $\mathbf{t}_\mathbf{y}$. The final time representation is $\mathbf{t}_\tau$ for the question at τ.

4.3 Answer Scoring Module

Finally, we get the scores of the candidate answers, consisting of all entities and timestamps,

$$\begin{aligned} \max\left(\phi\left(\mathbf{e}_s, \mathbf{W}_e\mathbf{q}, \mathbf{e}_\epsilon, \mathbf{t}_\tau\right)\right., \\ \left.\phi\left(\mathbf{e}_o, \mathbf{W}_e\mathbf{q}, \mathbf{e}_\epsilon, \mathbf{t}_\tau\right)\right) \\ \bigoplus\phi\left(\mathbf{e}_s, \mathbf{W}_t\mathbf{q}, \mathbf{e}_o, \mathbf{t}_\tau\right). \end{aligned} \tag{6}$$

where s, o and τ are the annotated subject, object and timestamp, respectively. ϵ represents candidate answers (all entities in the TKG). $\mathbf{W}_e$ and $\mathbf{W}_t$ are $D \times D$ learnable matrix specific for entity predictions and time predictions respectively. ϕ denotes

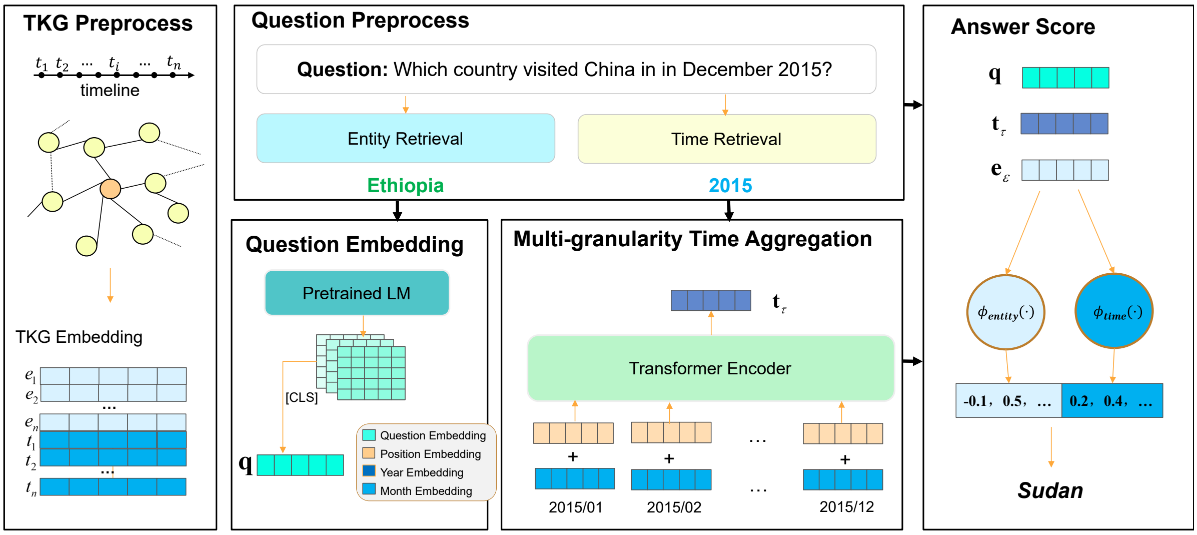

Figure 4: Model architecture of MultiQA. The question text is parsed to obtain the corresponding entity and time. TKG embedding is utilized to obtain the semantic information of KG, and the temporal semantic embeddings of different granularities are obtained by the multi-granularity aggregation module. Finally, the candidate answers are calculated by the scoring function.

the score function in TComplEx (Lacroix et al., 2020). We treat the annotated subject and object interchangeably, and $\max(\cdot)$ function ensures that we ignore the scores when s or o is a dummy entity.

During training, softmax is used to calculate probabilities over this combined score vector, and cross-entropy loss is employed.

5 Experiments

We experimentally evaluates MultiQA against five baselines. In the interest of space, experiment settings are in Appendix A.6.

5.1 Baseline Methods

- **Pre-trained LMs**: To evaluate BERT (Devlin et al., 2019), DistillBERT (Sanh et al., 2019) and ALBERT (Lan et al., 2020), we generate their LM-based question embedding and concatenate it with the entity and time embeddings, followed by a learnable projection. The resulted embedding is scored against all entities and timestamps via dot-product.

- **EmbedKGQA** (Saxena et al., 2020) is designed with static KGs. To deal with multiple temporal granularities, timestamps are ignored during pre-training and random time embeddings are used.

- **CronKGQA** (Saxena et al., 2021) is designed for single temporal granularity. To deal with multiple granularities, time embeddings at the year/month granularity are drawn at random from corresponding day embeddings.

5.2 Overall Results

Table 7 shows the results of our method compared to other baselines on MULTITQ. First, by comparing EmbedKGQA to pre-trained LMs (BERT, DistillBERT, ALBERT), we see that introducing KG representations with score function significantly improves the model's reasoning ability, even without providing any temporal information. We hypothesize that this is because KG embeddings specific to the TKG helps the model to focus on those entities.

Since EmbedKGQA has non-temporal embeddings, its performance on questions where the answer is a time is very low. By comparing CronKGQA to EmbedKGQA, we see that introducing a pre-trained time representation it refers significantly helps in answering temporal questions. In this case, the absolute improvement for all questions is 7% and 15% at Hits@1 and Hits@10, respectively. Further, we see the benefit of multi-granular time aggregation to the question, which effectively improves the inference on multi-granularity temporal questions (cf. Section 5.3). The absolute improvement of MultiQA over CronKGQA is 1% at Hits@1.

With the results of the paired t-test, we find that the MultiQA outperforms the best baseline significantly in most tasks, which demonstrates that multi-granular time aggregation is an effec-

Model	Hits@1					Hits@10				
	Overall	Question Type		Answer Type		Overall	Question Type		Answer Type	
		Multiple	Single	Entity	Time		Multiple	Single	Entity	Time
BERT	0.083	0.061	0.092	0.101	0.040	0.441	0.392	0.461	0.531	0.222
DistillBERT	0.083	0.074	0.087	0.102	0.037	0.482	0.426	0.505	0.591	0.216
ALBERT	0.108	0.086	0.116	0.139	0.032	0.484	0.415	0.512	0.589	0.228
EmbedKGQA	0.206	0.134	0.235	0.290	0.001	0.459	0.439	0.467	0.648	0.001
CronKGQA	0.279	0.134	0.337	0.328	0.156	0.608	0.453	0.671	0.696	0.392
MultiQA	**0.293****	**0.159****	**0.347***	**0.349****	**0.157**	**0.635****	**0.519****	**0.682***	**0.733****	**0.396**

Table 7: Overall results of baselines and our methods on the MULTITQ dataset. $^{*}(p \leq 0.05)$ and $^{**}(p \leq 0.005)$ indicate paired t-test of MultiQA versus the best baseline.

Model	Equal			Before/After			Equal Multi		
	Day	Month	Year	Day	Month	Year	Day	Month	Year
BERT	0.049	0.103	0.136	0.150	0.164	0.175	0.064	0.102	0.090
DistillBERT	0.041	0.087	0.113	0.160	0.150	0.186	0.096	0.127	0.089
ALBERT	0.069	0.082	0.132	0.221	0.277	0.308	0.103	0.144	0.144
EmbedKGQA	0.200	0.336	0.218	**0.392**	0.518	0.511	0.145	0.321	0.263
CronKGQA	0.425	0.389	0.331	0.375	0.474	0.450	0.295	**0.333**	0.251
MultiQA	**0.445****	**0.393***	**0.350****	0.379	**0.548****	**0.525****	**0.308***	0.321	**0.283****

Table 8: Experiment results of multi-granular time on Hits@1. $^{*}(p \leq 0.05)$ and $^{**}(p \leq 0.005)$ indicate paired t-test of MultiQA versus the best baseline.

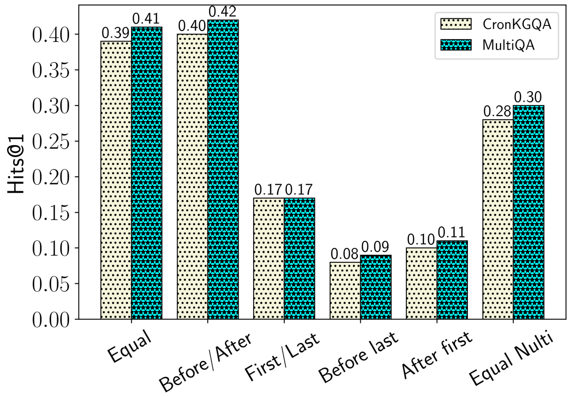

Figure 5: Model performance (Hits@1) against question types for MultiQA and CronKGQA.

tive solution to improve the overall performance of multi-granularity TKGQA. As shown in Figure 5, MultiQA achieves a better performance than CronKGQA in most categories of questions.

5.3 Results on Multi-Granular Time

To verify the effectiveness of the model on multi-granularity temporal reasoning, we experiment on multi-granularity temporal questions.

First, by comparing CronKGQA to EmbedKGQA, paradoxically, CronKGQA, while outperforming EmbedKGQA in overall results, is rather less effective at multi-granularity temporal reasoning. We argue that CronKGQA's introduction of a single granularity time representation improves inference at the corresponding time granularity, but misleads inference at the other granularities, causing the results to fall instead on multi-granularity TKGQA. This also highlights previous models' lack of inference capability for multi-granularity temporal question answering.

In addition, due to the multi-granularity aggregation module, MultiQA improves significantly at month and year granularity. Specifically, it outperforms by 7% at month and year granularity on before/after types, respectively. Similar pattern is also observed on the other types.

5.4 Effect of Training Dataset Size

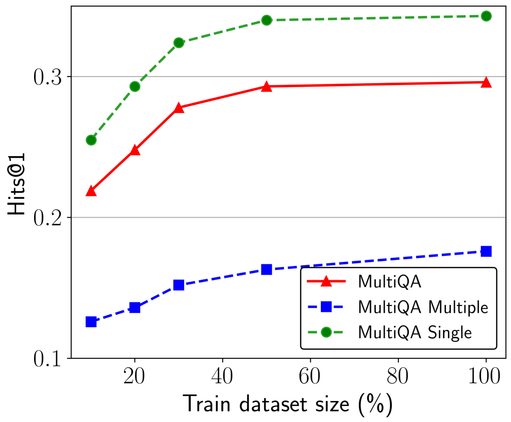

Figure 6: Model performance (Hits@1) against training dataset size (percentage) for MultiQA.

Although the dataset is constructed from templates and is semantically narrowed, the large dataset is still effective in improving model effectiveness. Figure 6 shows the effect of training dataset size on model performance. As we can see, for MultiQA, increasing the training dataset size from 10% to 100% steadily increases its performance for both single and Multiple reasoning type questions. We hypothesize that this is because the large number of entities and facts in the KG and the large number of model trainable parameters. These results affirm the hypothesis that having a large, even if synthetic, the dataset is useful for training temporal reasoning models (Saxena et al., 2021).

6 Conclusion and Limitation

In this paper, we introduce the concept of multi-granularity temporal question answering and construct a benchmark dataset MultiTQ, which features ample relevant facts and multiple temporal granularities. We also propose a multi-granularity temporal question Answering model MultiQA, serving as a strong baseline for follow-up research.

Limitation. The main drawback of our data creation protocol is that the question/answer pairs were generated automatically, leading the question distribution to be artificial from a semantic perspective. In addition, the KG adopted in the research focuses on a single event domain, and extending the dataset to multiple domains is planned as future work.

Acknowledgement

The authors would like to thank the anonymous reviewers for their insightful and constructive comments, which greatly contributed to improving the quality of the paper. This work was partially supported by National Key R&D Program of China No. 2020AAA0108800, NSFC under grants Nos. 62272469 and 71971212, and The Science and Technology Innovation Program of Hunan Province under grant No. CX20220067.

References

Ziyang Chen, Xiang Zhao, Jinzhi Liao, Xinyi Li, and Evangelos Kanoulas. 2022. Temporal knowledge graph question answering via subgraph reasoning. *Knowl. Based Syst.*, 251:109134.

Anoop Cherian, Chiori Hori, Tim K. Marks, and Jonathan Le Roux. 2022. (2.5+1)D spatio-temporal scene graphs for video question answering. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 444–453. AAAI Press.

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.

Xinyu Duan, Siliang Tang, Shengyu Zhang, Yin Zhang, Zhou Zhao, Jianru Xue, Yueting Zhuang, and Fei Wu. 2018. Temporality-enhanced knowledgememory network for factoid question answering. *Frontiers Inf. Technol. Electron. Eng.*, 19(1):104–115.

Haytham M. Fayek and Justin Johnson. 2020. Temporal reasoning via audio question answering. *IEEE ACM Trans. Audio Speech Lang. Process.*, 28:2283–2294.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4937–4951. Association for Computational Linguistics.

Alejandro Figueroa. 2010. Surface language models for discovering temporally anchored definitions on the web - producing chronologies as answers to definition questions. In *WEBIST 2010, Proceedings of the 6th International Conference on Web Information Systems and Technologies, Volume 1, Valencia, Spain, April 7-10, 2010*, pages 269–275. INSTICC Press.

Alejandro Figueroa, Carlos Gómez-Pantoja, and Ignacio Herrera. 2016. Search clicks analysis for discovering temporally anchored questions in community question answering. *Expert Syst. Appl.*, 50:89–99.

Alejandro Figueroa, Carlos Gómez-Pantoja, and Günter Neumann. 2019. Integrating heterogeneous sources for predicting question temporal anchors across yahoo! answers. *Inf. Fusion*, 50:112–125.

Jiyang Gao, Runzhou Ge, Kan Chen, and Ram Nevatia. 2018. Motion-appearance co-memory networks for video question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6576–6585. Computer Vision Foundation / IEEE Computer Society.

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.

Simon Gottschalk and Elena Demidova. 2018. Eventkg: A multilingual event-centric temporal knowledge graph. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 272–287. Springer.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, and Gerhard Weikum. 2011. YAGO2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011 (Companion Volume)*, pages 229–232. ACM.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2017. TGIF-QA: toward spatio-temporal reasoning in visual question answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1359–1367. IEEE Computer Society.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018a. Tempquestions: A benchmark for temporal question answering. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, pages 1057–1062. ACM.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018b. TEQUILA: temporal question answering over knowledge bases. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 1807–1810. ACM.

Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. 2021. Complex temporal question answering on knowledge graphs. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 792–802. ACM.

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.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jie Lei, Licheng Yu, Tamara L. Berg, and Mohit Bansal. 2020. TVQA+: spatio-temporal grounding for video question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8211–8225. Association for Computational Linguistics.

Guangyao Li, Yake Wei, Yapeng Tian, Chenliang Xu, Ji-Rong Wen, and Di Hu. 2022. Learning to answer questions in dynamic audio-visual scenarios. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 19086–19096. IEEE.

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.

Costas Mavromatis, Prasanna Lakkur Subramanyam, Vassilis N. Ioannidis, Soji Adeshina, Phillip R. Howard, Tetiana Grinberg, Nagib Hakim, and George Karypis. 2021. Tempoqr: Temporal question reasoning over knowledge graphs. *CoRR*, abs/2112.05785.

Sumit Neelam, Udit Sharma, Hima Karanam, Shajith Ikbal, Pavan Kapanipathi, Ibrahim Abdelaziz, Nandana Mihindukulasooriya, Young-Suk Lee, Santosh K. Srivastava, Cezar Pendus, Saswati Dana, Dinesh Garg, Achille Fokoue, G. P. Shrivatsa Bhargav, Dinesh Khandelwal, Srinivas Ravishankar, Sairam Gurajada, Maria Chang, Rosario Uceda-Sosa, Salim Roukos, Alexander G. Gray, Guilherme Lima, Ryan Riegel, Francois P. S. Luus, and L. Venkata Subramaniam. 2021. SYGMA: system for generalizable modular question answering overknowledge bases. *CoRR*, abs/2109.13430.

Ikujiro Nonaka, Ryoko Toyama, and Noboru Konno. 2000. Seci, ba and leadership: a unified model of dynamic knowledge creation. *Long range planning*, 33(1):5–34.

Aditya Pal, James Margatan, and Joseph A. Konstan. 2012. Question temporality: identification and uses. In *CSCW '12 Computer Supported Cooperative Work, Seattle, WA, USA, February 11-15, 2012*, pages 257–260. ACM.

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 Z. Yang, Zachary 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.

John F. Roddick and Myra Spiliopoulou. 2002. A survey of temporal knowledge discovery paradigms and methods. *IEEE Trans. Knowl. Data Eng.*, 14(4):750–767.

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

Apoorv Saxena, Soumen Chakrabarti, and Partha P. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6663–6676. Association for Computational Linguistics.

Apoorv Saxena, Aditay Tripathi, and Partha P. 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, ACL 2020, Online, July 5-10, 2020*, pages 4498–4507. Association for Computational Linguistics.

Stefan Schweter and Alan Akbik. 2020. FLERT: document-level features for named entity recognition. *CoRR*, abs/2011.06993.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org.

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.

Fei Wu, Xinyu Duan, Jun Xiao, Zhou Zhao, Siliang Tang, Yin Zhang, and Yueting Zhuang. 2017. Temporal interaction and causal influence in community-based question answering. *IEEE Trans. Knowl. Data Eng.*, 29(10):2304–2317.

Xuchao Zhang, Wei Cheng, Bo Zong, Yuncong Chen, Jianwu Xu, Ding Li, and Haifeng Chen. 2020. Temporal context-aware representation learning for question routing. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 753–761. ACM.

A Appendix

A.1 TKG Embeddings

A TKG $\mathcal{K} := (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$ is a multi-relational directed graph with time-stamped edges between entities. A fact in $\mathcal{K}$ can be formalized as $(s, r, o, \tau) \in \mathcal{F}$, where $s, o \in \mathcal{E}$ denote the subject and object entities, $r \in \mathcal{R}$ denotes the relation between them, and $\tau \in \mathcal{T}$ is the timestamp associated with that relation. TKG embedding methods learn a K-dimensional vector $\mathbf{e}_\epsilon, \mathbf{v}_r, \mathbf{t}_\tau \in \mathbb{R}^K$ of each $\epsilon \in \mathcal{E}, r \in \mathcal{R}$ and $\tau \in \mathcal{T}$ in $\mathcal{K}$, such that each fact $(s, r, o, \tau) \in \mathcal{F}$ has a higher score than the one $(s', r', o', \tau') \notin \mathcal{F}$ through a scoring function $\phi(\cdot)$, formally $\phi\left(\mathbf{e}_s, \mathbf{v}_r, \mathbf{e}_o, \mathbf{t}_\tau\right) > \phi\left(\mathbf{e}_{s'}, \mathbf{v}_{r'}, \mathbf{e}_{o'}, \mathbf{t}_{\tau'}\right)$. TComplEx (Lacroix et al., 2020) is an extension of ComplEx (Trouillon et al., 2016) considering time information, which encodes each entity, relation and timestamp to complex vector. The score function $\phi(\cdot)$ of TComplEx is defined by

$$\phi\left(\mathbf{e}_s, \mathbf{v}_r, \overline{\mathbf{e}}_o, \mathbf{t}_\tau\right) = \mathrm{Re}\left(\left\langle \mathbf{e}_s, \mathbf{v}_r \odot \mathbf{t}_\tau, \overline{\mathbf{e}}_o \right\rangle\right), \quad (7)$$

where $\mathrm{Re}(\cdot)$ denotes the real part, $(\cdot)$ is the complex conjugate of the embedding vector and $\odot$ is the element-wise product.

We train TComplEx on `ICEWS05-15` with the TKG completion task. We learn the entity and relation representations in the complex space C_d, where d denotes the dimension of the complex vectors.

A.2 Entity and Time Retrieval

Unlike previous QA datasets, our dataset does not contain entities and time annotations, so the only information the QA model can use is the text of the questions and the corresponding KG information. This is also in line with the TKGQA task in a practical application scenario. Due to the lack of entity linking tools for the `ICEWS`, we first used a pre-trained generic NER tool (Schweter and Akbik, 2020) to extract the question text, filter out the entity names in it, and then match it with the entities in the KG through fuzzy matching to find the most similar entity as the entity result for subsequent inference.

$$Q_{\text{entity}} = \mathrm{FuzzyMatch}\left(Q_{\text{ner}}, \mathcal{E}\right), \quad (8)$$

where Q_{ner} is the list of identified entities, and we fuzzy match the identified entities with the entities $\mathcal{E}$ in the KG by calculating the similarity,

$$Sim(e_1, e_2) = 2 \cdot \frac{M_{e_1e_2}}{L_{e_1} + L_{e_2}}, \tag{9}$$

where L_e is the text length of entity e, $M_{e_1e_2}$ is the the maximum length that can be matched between e_1 and e_2. Entity with highest similarity in $\mathcal{E}$ will be added to entity linking set Q_{entity}.

As the expression of time is more fixed, we adopt a rule-based method to extract time information from the question text for subsequent reasoning. Specifically, a series of regular expressions based on common time formats have written to extract the corresponding time information in the question.

A.3 More Statistics of MultiTQ

We summarize the statistics of different time granularities in Table 9 and distribution of relations in Figure 7 in MultiTQ.

	Time granularities		
	Day	Month	Year
Equal	77,738	55,221	39,225
Before/After	65,641	20,443	11,984
Equal Multi	2,364	7,971	12,978
Total	145,743	83,635	64,187

Table 9: Statistics for the various time granularities.

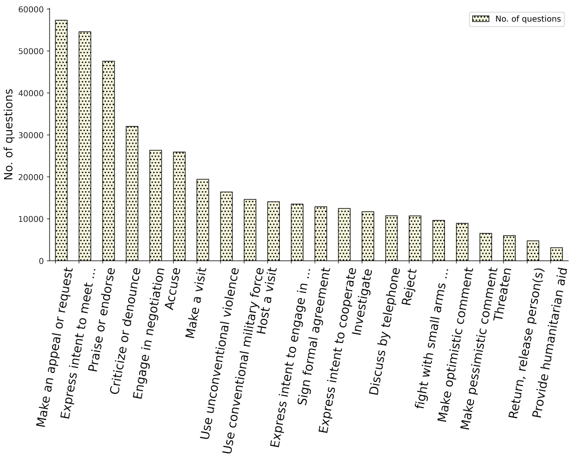

Figure 7: Relation distribution in MultiTQ.

A.4 Results on Single-Granularity Time

We conduct an additional experiment to analyze the behavior of baseline models on single-time granularity datasets. We partition MultiTQ by time granularity, ensuring that there is only single granularity of time in each divided dataset (Day, Month, and Year). At the setting of single-day, since the temporal granularity of KG coincides with that of the dataset, our model degenerates to CronKGQA. From the experiment results at a single granularity, even if the time granularity of the KG is kept consistent with that of the questions, the existing model still struggles to achieve excellent results as that on previous datasets (e.g., CronQuestions), mainly because our proposed dataset has more complex question types and KG with higher semantic complexity, which hence better reflects real-world scenarios.

As observed in Table 10, we can see that the introduction of time information on fine-grained questions can significantly improve the performance of the temporal QA system. Consistent with the observations on the multi-granularity experiments discussed in Section 5.3, MultiQA is able to achieve substantially improved performance at coarse-grained timescales thanks to the multi-granularity time aggregation module. This further validates the efficacy of this module.

	Model	Hits@1
Day	ALBERT	0.091
	EmbedKGQA	0.186
	CronKGQA	0.270
	MultiQA	/
Month	ALBERT	0.083
	EmbedKGQA	0.269
	CronKGQA	0.303
	MultiQA	0.317
Year	ALBERT	0.117
	EmbedKGQA	0.184
	CronKGQA	0.254
	MultiQA	0.266

Table 10: Experiment results of single-granularity time on Hits@1.

A.5 Error Analysis

For error analysis, we randomly sample 100 error instances from the test set and summarized the following three types of typical errors: (1) Retrieving irrelevant entities , meaning the model obtained wrong entities from the KG; Although our entity linking model can achieve a high prediction accuracy, wrong entities still exist in some questions. (2) Wrong reasoning at the semantic level, meaning the model failed to obtain the entities related to the semantics of the question. Limited by the representation of the question and the reasoning ability, even when the time constraint is not taken into account, there are still cases where the reasoning yields irrelevant entities or times. Such a phenomenon is especially common in complex questions. (3) Lacking the ability of reasoning about complex temporal constraints, meaning the model design cannot support complex temporal constraints. The inference ability of MultiQA comes from the complementary inference ability obtained in the pre-training of TKG Embedding, which is limited to simple temporal inference. This prevents our model from achieving efficient reasoning about complex constraints such as First, before, etc.

This demonstrates more efforts are needed to strengthen the model's reasoning capability, especially in semantic reasoning and complex temporal constraints reasoning. Also, using more advanced NEL models would be an effective direction for enhancement.

A.6 Reproducibility

In this section, we report more experimental details to ensure the reproducibility of this paper.

The model is implemented with PyTorch (Paszke et al., 2019). We use TComplEx (Lacroix et al., 2020) as our TKG embeddings, and their dimensions $D = 512$. We use BERT-base, DistillBERT-base and ALBERT-base in our implementation. Both LM's parameters and the TKG embeddings are not updated during the training. We set the number of transformer layers of the encoder $Transformer(\cdot)$ to $l = 2$ with 4 heads per layer. The model's parameters are updated with Adam (Kingma and Ba, 2015) with a learning rate of 0.0002. All the experiments are conducted on a server that has an Intel(R) Core(TM) i9-10900K@3.70GHz CPU and a 24-GB Nvidia RTX 3090 GPU. The operating system is Ubuntu 20.04. More details about the implementation, e.g., dependency libraries, can be found in the `README` file of the software.

In addition, our model has about 195M parameters. And the average training time is 2.5h.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 6 Conclusion and Limitation

☑ A2. Did you discuss any potential risks of your work?
Section 6 Conclusion and Limitation

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1 introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3.1

☑ B1. Did you cite the creators of artifacts you used?
Section 3.1

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
ICEWS dataset has been sanctioned by the U.S. Government for public release.and is allowed for science research[1].ICEWS is open with the CCO License allowing free access[2].
[1] https://www.lockheedmartin.com/en-us/news/features/2016/ICEWs-10000-dataset-download.html
[2] https://dataverse.harvard.edu/dataverse/icews

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
ICEWS is open with the CCO License allowing free access[1].
[1] https://dataverse.harvard.edu/dataverse/icews

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The ICEWS dataset contains information on significant events such as national institutions and does not involve personal private information.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3 and Appendix A.3

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3 and Appendix A.3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix A.6

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix A.6

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix A.6

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.