

Improving Time Sensitivity for Question Answering over Temporal Knowledge Graphs

Chao Shang¹, Guangtao Wang¹, Peng Qi¹, Jing Huang^{2*}

¹JD AI Research

²Alexa AI, Amazon

{chao.shang3, guangtao.wang, peng.qi}@jd.com

jhuangz@amazon.com

Abstract

Question answering over temporal knowledge graphs (KGs) efficiently uses facts contained in a temporal KG, which records entity relations and when they occur in time, to answer natural language questions (e.g., “Who was the president of the US before Obama?”). These questions often involve three time-related challenges that previous work fail to adequately address: 1) questions often do not specify exact timestamps of interest (e.g., “Obama” instead of 2000); 2) subtle lexical differences in time relations (e.g., “before” vs “after”); 3) off-the-shelf temporal KG embeddings that previous work builds on ignore the temporal order of timestamps, which is crucial for answering temporal-order related questions. In this paper, we propose a time-sensitive question answering (TSQA) framework to tackle these problems. TSQA features a timestamp estimation module to infer the unwritten timestamp from the question. We also employ a time-sensitive KG encoder to inject ordering information into the temporal KG embeddings that TSQA is based on. With the help of techniques to reduce the search space for potential answers, TSQA significantly outperforms the previous state of the art on a new benchmark for question answering over temporal KGs, especially achieving a 32% (absolute) error reduction on complex questions that require multiple steps of reasoning over facts in the temporal KG.

1 Introduction

Temporal knowledge graphs (KGs) record the relations between entities and the timestamp or time period when such relation hold, e.g., in the form of a quadruple: (Franklin D. Roosevelt, position held, President of USA, [1933, 1945]). This makes them a perfect source of knowledge to answer questions that involve knowledge of when certain events occurred as well as how they are related temporally (see Figure 1 for an example). Unlike question

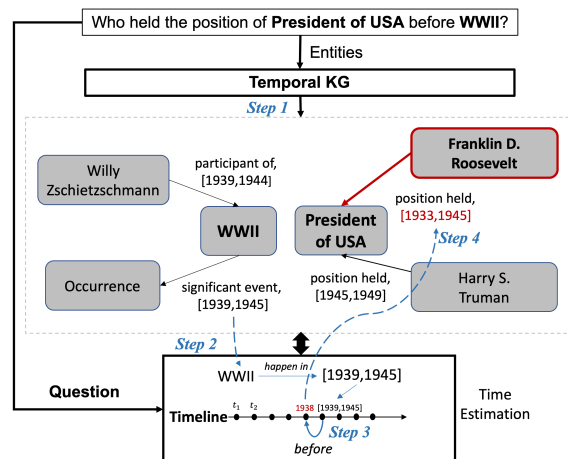


Figure 1: An example of complex temporal question on a temporal KG.

answering (QA) over non-temporal KGs that is mainly concerned with relational inference, a core challenge in temporal KGQA is correctly identifying the time of reference mentioned explicitly or implicitly in the question, and locating relevant facts by jointly reasoning over relations and timestamps.

Inspired by work on relational KGQA (Huang et al., 2019; Saxena et al., 2020), where knowledge graph embeddings (Dasgupta et al., 2018; García-Durán et al., 2018; Goel et al., 2020; Wu et al., 2020; Lacroix et al., 2020) learned independently of question answering are used as input to KGQA models, previous work (Saxena et al., 2021) employs temporal KG embeddings to attack the problem of temporal KGQA. Despite its relative success on simple temporal questions that directly queries facts in the KG with one out of the four facts left as the answer (e.g., “When was Franklin D. Roosevelt the President of USA?” or “What position did Franklin D. Roosevelt hold between 1933 and 1945?”), this approach still struggles to handle questions that require multiple steps of relational-temporal reasoning (e.g., the example in Figure

* Work done at JD AI Research.

1).

We identify three main challenges that hinder further progress on temporal KGQA. Firstly, complex temporal questions often require inferring the correct point of reference in time, which is not considered by previous work. For instance, to correctly answer the question in Figure 1, it is crucial that we first identify that World War II took place between 1939 and 1945, and look for entities with the desired relation with President of USA in the time interval specified by these times. Secondly, unlike entity relations, which are usually expressed in natural language with a handful of content words that correspond well with their recorded relations in KGs (e.g., “What position did ... hold ...” vs the “position held” relation), temporal relations often involve just one or two prepositions (e.g., “before” or “during”) and are expressed only implicitly in temporal KGs (e.g., nowhere is it clearly stated that 1931 is earlier than, or before, 1934, by a gap of 3 years). As a result, a small lexical change can drastically alter the temporal relation expressed by the question, and therefore the answer set. Thirdly, previous work on temporal KGQA build on temporal KG embeddings, where each timestamp is assigned a randomly initialized vector representation that is jointly optimized with entity and relation representations to reconstruct quadruples in the KG from embeddings. While sound as a standalone method for encoding knowledge in temporal KGs, this approach does not guarantee that the learned timestamp representations can recover implicit temporal relations like temporal orders or distance, which are crucial for temporal KGQA.

In this paper, we propose a time-sensitive question answering framework (TSQA) to address these challenges. We first equip the temporal KGQA model with a time estimation module that infers the unstated timestamps from questions as the first step of reasoning, and feed the result into relational inference as a reference timestamp. Even without explicit training data for this module, the explicit factorization of the problem yields significant improvement over previous work on complex questions that require reasoning over multiple temporal quadruples. To improve the sensitivity of our question encoder to time relation words, we also propose auxiliary contrastive losses that contrast the answer prediction and time estimation for questions that differ only by the time relation word (e.g., “before” vs “after”). By leveraging the mutual ex-

clusiveness of answers and the prior knowledge regarding potential time estimates from different time relation words, we observe further improvements in model performance on complex questions. Next, to learn temporal KG embeddings with prior knowledge of temporal order and distance built in, we introduce an auxiliary loss of time-order classification between each pair of timestamp embeddings. As a result, the knowledge in the temporal KG can be distilled into the entity, relation, and timestamp embeddings where the timestamp embeddings can naturally recover order and distance information between the underlying timestamps, thus improving the performance of temporal KGQA where such information is crucial. Finally, we enhance TSQA with KG-based approaches to narrow the search space to speed up model training and inference, as well as reduce the number of false positives in model prediction. As a result, TSQA outperforms the previous state of the art on the CRONQUESTIONS benchmark (Saxena et al., 2021) by a large margin.

To summarize, our contributions in this paper are: a) we propose a time-sensitive question answering framework (TSQA) that performs time estimation for complex temporal answers; b) we present contrastive losses that improve model sensitivity to time relation words in the question; c) we propose a time-sensitive temporal KG embedding approach that benefits temporal KGQA; d) with the help of KG-based pruning technique, our TSQA model outperforms the previous state of the art by a large margin.

2 Related Work

Temporal Knowledge Graph Embedding. Knowledge graph embedding learning (Bordes et al., 2013; Yang et al., 2014; Trouillon et al., 2016; Dettmers et al., 2018; Shang et al., 2019; Sun et al., 2019; Tang et al., 2019; Ji et al., 2021) has been an active research area with applications directly in knowledge base completion and relation extractions. Recently, there are several works that extended the static KG embedding models to temporal KGs. Jiang et al. (2016) first attempt to extend TransE (Bordes et al., 2013) by adding a timestamp embedding into the score function. Later, Hyte (Dasgupta et al., 2018) projects each timestamp with a corresponding hyperplane and utilizes the TransE score in each space. García-Durán et al. (2018) extend TransE and

DistMult by utilizing recurrent neural networks to learn time-aware representations of relation types. TCompLex (Lacroix et al., 2020) extends the ComplEx with time based on the canonical decomposition of tensors of order 4.

Temporal QA on Knowledge Graph. Temporal QA have mostly been studied in the context of reading comprehension. ForecastQA (Jin et al., 2021) formulates the forecasting problem as a multiple-choice question answering task, where both the articles and questions include the timestamps. The recent released TORQUE (Ning et al., 2020) is a dataset that explores the temporal ordering relations between events described in a passage of text.

Another direction is the temporal question answering over knowledge bases (KB) (Jia et al., 2018b,a), which retrieves time information from the KB. TempQuestions (Jia et al., 2018a) is a KGQA dataset specifically aimed at temporal QA. Based on this dataset, Jia et al. (2018b) design a method that decomposes and rewrites each question into nontemporal sub-question and temporal sub-question. Here the KG used in TempQuestions is based on a subset of FreeBase which is not a temporal KG. Later Jia et al. (2021) proposes a first end-to-end system (EXAQT) for answering complex temporal questions, which takes advantage of the question-relevant compact subgraphs within the KG, and relational graph convolutional networks (Schlichtkrull et al., 2018) for predicting the answers. All previous datasets only include a limited number of temporal questions. Recently, a much larger temporal KGQA dataset CRONQUESTIONS (Saxena et al., 2021) is released, which includes both the temporal questions and the temporal KG with time annotation for all edges. Based on this dataset, the CronKGQA model (Saxena et al., 2021) is presented that exploits recent advances in Temporal KG embeddings and achieves performance superior to all baselines.

3 Method

In this section, we first give the problem definition of temporal question answering over temporal knowledge graph. Then, we introduce the framework to solve this problem, which integrates time sensitivity into KG embedding and answer inference. Finally, we describe the key modules of our proposed system in details.

3.1 Problem Definition and Framework

QA on Temporal KG aims at finding out the answer from a given temporal KG $G = (\mathcal{V}, \mathcal{E}, \mathcal{R}, T)$ for a given free-text temporal question Q containing implicit temporal expression, and the answer is either an entity of entity set \mathcal{V} or a timestamp of timestamp set T . Here, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges, and \mathcal{R} is the set of relations. Edge from a quadruple $(s, r, [t_s, t_e], o)$ indicates the relation $r \in \mathcal{R}$ holds between subject entity s and object entity o during time interval $[t_s, t_e]$ ($t_s < t_e$ and $t_{e/s} \in T$).

Framework. Our framework resorts to KG embeddings along with pretrained language models to perform temporal KGQA. Figure 2 shows the architecture which consists of two modules: 1) time-aware TKG encoder; 2) time-sensitive question answer.

The time-aware TKG encoder extends the existing TKG embedding method by adding an auxiliary time-order learning task to consider the quadruple orders. And the time sensitive QA module first performs neighboring graph extraction to reduce the search space for question answer, then performs joint training for answer/time prediction and time-sensitive contrastive learning to enhance the model ability in capturing temporal signals in free-text question. Next, we will introduce these two modules in details.

3.2 Time-aware KG Encoder

We first briefly review a time-aware KG embedding method based on TCompLex (Lacroix et al., 2020) since it has been used in (Saxena et al., 2021) for TKGQA and shows competitive performance. Next, we show that how to perform TCompLex on temporal KG, then analyze its weakness in TKGQA especially for complex question and further overcome such weakness by introducing an auxiliary time-order learning task in TKG embedding.

TCompLex for TKG. TCompLex is an extension of ComplEx considering time information, which not only encodes the entity and relation to complex vectors, but also maps each timestamp to a complex vector. To perform TCompLex over temporal KG in our problem definition, we first reformulate each quadruple to a set of new quadruples by

$$(s, r, [t_s, t_e], o) = \{(s, r, t, o) | t_s \leq t \leq t_e\} \quad (1)$$

Let $e_s, e_r, e_t, e_o \in \mathbb{C}^d$ be the complex-value embeddings of s, r, t, o , respectively. Then, TCom-

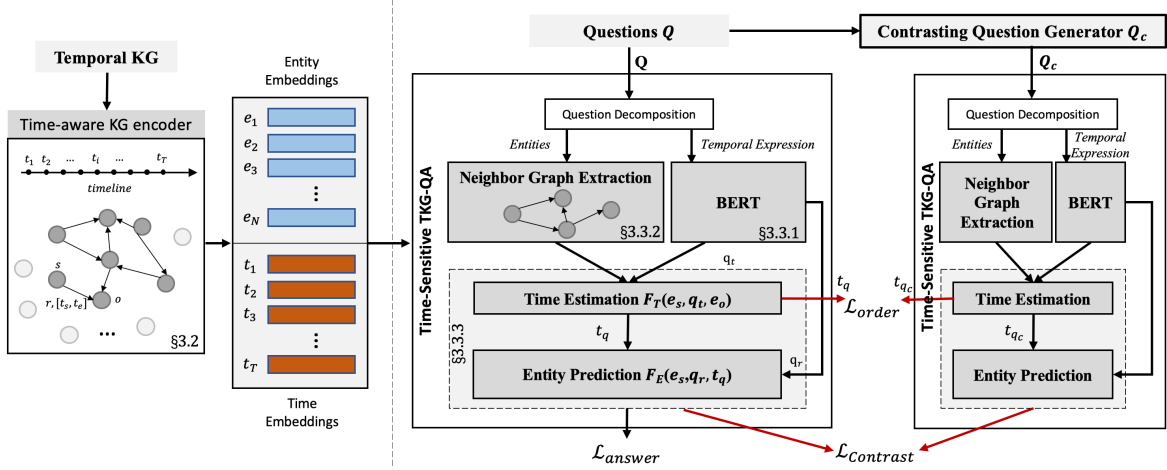


Figure 2: The architecture of our TSQA model (**Left**: Time-aware TKG encoder; **Right**: Time-Sensitive TKG-QA).

pLEx scores each quadruple (s, r, t, o) by

$$\mathcal{S}(s, r, o, t) = \text{Re}(\langle e_s, e_r, e_o, e_t \rangle) \quad (2)$$

where $\text{Re}(\cdot)$ denotes the real part of a complex vector, and $\langle \cdot \rangle$ denotes the multi-linear product.

Finally, we use a loss function similar to the negative sampling loss for effectively TCompLEx training.

$$L_{TC} = -\log(\phi(\gamma - \mathcal{S}(s, r, o, t))) - \frac{1}{K} \sum_{i=1}^K (\log(\phi(\mathcal{S}(s'_i, r, o'_i, t'_i) - \gamma))), \quad (3)$$

where γ is a fixed margin, ϕ is the sigmoid function, (s'_i, r, o'_i, t'_i) is the i -th negative quadruple.

According to the loss function in equation 3, we observe that TCompLEx only cares about whether the quadruple is true or false and ignores the orders of different quadruples occur. However, the time orders are critical to find the correct answer in knowledge graphs. For example, to answer the ‘Who is the President of USA before William J. Clinton?’, we need not only the two facts (President of USA, Position Held, Ronald Reagan, [1981, 1989]) and (President of USA, Position Held, William J. Clinton, [1993, 2001]), but also the time order of these facts. To overcome such a limit of TCompLEx in TKGQA, we introduce an auxiliary time-order learning task over time-embeddings.

Time-order learning in TKG. To keep the time order in embedding spaces, we first sort the timestamps in T by an ascending order and get $(t_1, t_2, \dots, t_{|T|})$ and $t_i < t_j$ if $1 \leq i < j \leq |T|$.

Let $\mathbf{t}_i = [\text{Re}(e_{t_i}), \text{Im}(e_{t_i})] \in \mathbb{R}^{2d}$ be the concatenation the real and imaginary components of embedding e_{t_i} of timestamp t_i . Inspired by position embedding in (Vaswani et al., 2017), we first initialize the timestamp embedding \mathbf{t}_i as follows.

$$\begin{aligned} \mathbf{t}_i[2k] &= \sin\left(\frac{i}{10000^{2k/2d}}\right) \\ \mathbf{t}_i[2k+1] &= \cos\left(\frac{i}{10000^{2k/2d}}\right) \end{aligned} \quad (4)$$

where $0 \leq k \leq d-1$.

Afterwards, for any pair of timestamps (t_i, t_j) , we calculate the probability of time order as:

$$p_t(i, j) = \text{sigmoid}((\mathbf{t}_1 - \mathbf{t}_2)^T \mathbf{W}_t), \quad (5)$$

where $\mathbf{W}_t \in \mathbb{R}^{2d}$ represents a parameter vector.

Based on the time-order probabilities, we introduce a binary cross-entropy loss as a time-order constraint over timestamp embeddings as follow:

$$\begin{aligned} L_{TO} &= -\delta(i, j) \log(p_t(i, j)) \\ &\quad - (1 - \delta(i, j)) \log(1 - p_t(i, j)), \end{aligned} \quad (6)$$

where $\delta(i, j) = 1$ if $t_i < t_j$ else $\delta(i, j) = 0$.

Joint-training. A weighted sum of T-CompLEx training loss and time-order constraint is considered as the final objective function for the joint training for TKG embedding.

3.3 Time-Sensitive TKG-QA

In this section, we introduce our time-sensitive question answering module from the following aspects in details: 1) question decomposition which

divides the questions as entities and relations described in free-text; 2) entity neighboring sub-graph extraction which reduces the search space of candidate timestamps and answer entities; and 3) time-sensitive question answer which explores the time information implied in both KG and questions to help the model find the answer.

3.3.1 Question Decomposition and Encoder

For each question Q , we first identify all the entities $\{\text{Ent}_1, \text{Ent}_2, \dots, \text{Ent}_k\}$ in Q which also appear in KG G , i.e., $\text{Ent}_i \in E$ ($1 \leq i \leq k$). Then, by replacing the entities in question Q with special token $[subject]$ and $[object]$ in order, we obtain an entity-independent temporal relation description in free-text named **temporal expression** \hat{Q} .

Taking the question ‘‘When did **Obama** hold the position of **President of USA**?’’ as an example, by replacing the identified entities ‘‘President of USA’’ and ‘‘Obama’’, we get its temporal expression as ‘‘When did *subject* hold the position of *object*?’’.

Next $[\text{CLS}] + \hat{Q}$ are fed into BERT that outputs $[\text{CLS}]$ token embedding as $e_q \in \mathbb{R}^{d_{bert}}$, where d_{bert} is the output dimension of BERT, and two kinds of question representations as follows.

$$\mathbf{q}_r = \mathbf{W}_q^r(\delta(\mathbf{W}e_q)) \quad (7)$$

$$\mathbf{q}_t = \mathbf{W}_q^t(\delta(\mathbf{W}e_q)), \quad (8)$$

where $\mathbf{q}_r, \mathbf{q}_t \in \mathbb{R}^{2d}$ represents the embedding of relation and time implied in question, respectively. $\mathbf{W} \in \mathbb{R}^{d_{bert} \times 2d}$, $\mathbf{W}_q^r, \mathbf{W}_q^t \in \mathbb{R}^{2d \times 2d}$ are the parameter matrix, and δ represents the activation function. Finally, to facilitate the calculation with KG embeddings, we reformulate $\mathbf{q}_r, \mathbf{q}_t$ in complex space as:

$$\mathbf{q}_r = \mathbf{q}_r[0 : d] + \sqrt{-1} \cdot \mathbf{q}_r[d : 2d] \quad (9)$$

$$\mathbf{q}_t = \mathbf{q}_t[0 : d] + \sqrt{-1} \cdot \mathbf{q}_t[d : 2d] \quad (10)$$

3.3.2 Entity Neighbor Graph Extraction

Let $\{\text{Ent}_1, \text{Ent}_2, \dots, \text{Ent}_k\}$ be the k entities extracted from question Q , we first extract the m -hop neighboring sub-graph G_i for each entity Ent_i . Then, by combining these k sub-graphs, we obtain the search graph G_q for question Q : $G_q = \cup_{i=1}^k G_i$. Suppose that E_q and T_q are the sets of entities and timestamps appearing in G_q , respectively, they constitute the search space of time and entity prediction in our TKG-QA method. In training stage, we set the hop number m as the minimum value which results in correct answer entity appearing in

G_q . In testing stage, we set m as the largest hop number used in training stage. In practice, the size of graph G_q is usually much smaller than that of whole graph G . For example, in CronKGQA, the average value of $|G_q|/|G|$ is about 3%.

Entity Neighboring graph extraction aims at reducing the search space of candidate timestamps and answer entities. This results in not only more efficient training procedure, but also performance improvement of question answer because a larger number of candidates usually means a much more difficult learning problem.

3.3.3 Time-Sensitive Question Answering

For temporal question answer over KG, the interaction of time and answer entity prediction is very important since the time range brings a strong constraint on the search space of answers. However, the existing method (Saxena et al., 2021) usually performs such two predictions independently which results in poor performance especially for complex questions which need to consider multiple facts to get the answer. To overcome this limitation, we directly feed the intermediate time representation t_q learned from time estimation to answer prediction to enhance the interaction of these two tasks.

Time Estimation. Based on the embeddings e_s and e_o of subject entity s and object entity o from KG and the time embedding \mathbf{q}_t from a question, we design the time estimation function F_T for learning the time embedding t_q as follows:¹

$$\begin{aligned} t_q &= F_T(e_s, \mathbf{q}_t, e_o) \\ &= \mathbf{W}_q^t([\text{Re}(\langle e_s, \mathbf{q}_t, e_o \rangle), \text{Im}(\langle e_s, \mathbf{q}_t, e_o \rangle)]), \end{aligned} \quad (11)$$

where $\mathbf{W}_q^t \in \mathbb{R}^{2d \times 2d}$ represents the parameter matrix. $[\cdot]$ is the concatenation function, $\text{Re}(\cdot)$ denotes the real part of a complex vector and $\text{Im}(\cdot)$ is the imaginary part.

After getting the time embedding w.r.t. question t_q , for timestamp prediction, the following score function to estimate the score for each timestamp $t \in T_q$ as follow:

$$\mathcal{S}_t = \text{Re}(\langle t_q, t \rangle) \quad (12)$$

Entity Prediction. In enhance the interaction between time prediction and answer prediction, we

¹A simple temporal question might contain the timestamp (e.g. 2001). In this case, we set t_q as the linear combination of this learned time embedding and the timestamp embedding from KG.

update the embedding of entity w.r.t. question by considering time embedding t_q by an entity function F_E as follow:

$$e_q = F_E(e_s, \mathbf{q}_r, t_q) = \langle e_s, \mathbf{q}_r, t_q \rangle \quad (13)$$

Finally, we score the entity $e \in E_q$ by:

$$S_e = Re(\langle e_q, e \rangle) \quad (14)$$

The answer entity of the question is either timestamp or entity. Let S_a be the answer score and thus $S_a = S_t$ or S_e when the answer is timestamp or entity. Suppose C represents the number of candidate answers (i.e., $C = |E_q| + |T_q|$), then we can define the probability of i -th candidate answer being true as:

$$P_{a,i} = \frac{\exp(S_{a,i})}{\sum_{j=1}^C \exp(S_{a,j})}. \quad (15)$$

Finally, we train the answer prediction model by minimizing the cross-entropy loss as follow:

$$L_{answer} = - \sum_i^C \mathbf{y}_i \log(P_{a,i}), \quad (16)$$

where $\mathbf{y}_i = 1$ if the i -th candidate is the true answer, otherwise $\mathbf{y}_i = 0$.

3.3.4 Temporal Contrastive Learning

The temporal question answer system should be sensitive to the temporal relation implied in the question. For example, the answer of ‘‘What does happen before a given event?’’ is quite different from that of ‘‘What does happen after a given event?’’. Existing works on TKG-QA usually resort to pre-trained language models for question understanding. But these models are not sensitive to the difference of temporal expressions in free-text (Ning et al., 2020; Dhingra et al., 2021; Shang et al., 2021; Han et al., 2021), and thus prone to wrong predictions.

To make the system sensitive to the temporal relation implied in question, we resort to a contrastive learning method: we construct a contrastive question to the original question, then add auxiliary contrastive learning tasks to distinguish the latent temporal representation and prediction results coming from the pair of contrastive questions.

Contrastive Question Generation. To generate the contrastive question \bar{Q} for the given question Q , we first extract all the temporal words based on large number of questions in temporal question answer dataset, and then build a contrastive word pair

dictionary by finding the antonyms. The dictionary consists of $\mathcal{D}_{contr} = \{(\text{first}, \text{last}), (\text{before}, \text{after}), (\text{before}, \text{during}), (\text{during}, \text{after}), (\text{before}, \text{when}), (\text{when}, \text{after})\}$. Based on such dictionary, we replace the temporal word in given question Q by its antonym to generate its contrastive question \bar{Q} .

Contrastive time order learning. For the contrastive question pair Q and \bar{Q} , we follow the same encoder in Eq. 11 to get the corresponding time-aware embeddings t_q and t_{q_c} , respectively. Meanwhile, according to the contrastive temporal word pair dictionary, suppose that we pickup the pair $(\text{word}_1, \text{word}_2) \in \mathcal{D}_{contr}$ for contrastive question construction, we can construct a question order label y_o : $y_o = 0$ if \bar{Q} is achieved by replacing word_1 as word_2 , else $y_o = 1$.

Afterward, we distinguish the temporal orders implied by word_1 and word_2 by predicting of the order label y_o based on t_q and t_{q_c} as follow:

$$p_o = \text{sigmoid}((t_q - t_{q_c})^T \mathbf{W}_o) \quad (17)$$

$$\mathcal{L}_{order} = -y_o \log(p_o) - (1 - y_o) \log(1 - p_o), \quad (18)$$

where $\mathbf{W}_o \in \mathbb{R}^{2d}$ represents the parameter vector to be learned.

Answer-guided Contrastive Learning. Let $\mathbf{S} = [s_1, \dots, s_C]$, $\bar{\mathbf{S}} = [\bar{s}_1, \dots, \bar{s}_C]$ be the answer scores w.r.t. questions Q and its contrastive question \bar{Q} , respectively, where $C = |E_q| + |T_q|$. By stacking these two scores together, we get $\mathbf{S}_q = [\mathbf{S}; \bar{\mathbf{S}}] \in \mathbb{R}^{2 \times C}$. Then, we can apply softmax over \mathbf{S}_q along the last dimension and get the probability scores $\mathbf{P}_q = \text{softmax}(\mathbf{S}_q) \in \mathbb{R}^{2 \times C}$ and $\text{sum}(\mathbf{P}_q[:, i]) = 1$ for $i = 1, \dots, C$.

Due to the fact that the answers of question Q are definitely not for question \bar{Q} , we construct an answer-guided learning labels as $\mathbf{y}_a = [y_1, \dots, y_C]$, where $y_i = 1$ if and only if the i -th candidate is true answer for Q , otherwise $\mathbf{y}_i = 0$. Then, we get an answer-guided contrastive loss as follow:

$$\mathcal{L}_{contrast} = -\frac{1}{C} \sum_{j=0}^C y_i \log(\mathbf{P}_q[0, i]) \quad (19)$$

Joint Training. We combine the answer prediction loss and contrastive losses as the final objective function for joint training:

$$Loss = \mathcal{L}_{answer} + \lambda_o \cdot \mathcal{L}_{order} + \lambda_c \cdot \mathcal{L}_{contrast}, \quad (20)$$

where $\lambda_o > 0$, $\lambda_c > 0$ are the weight factors to make tradeoffs between different losses.

4 Experiments

In this section, we conduct experiments to assess the effectiveness of our proposed method TSQA for TKG-QA. Our experimental results show that our approach obtains significant improvements over the baseline models.

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

Table 1: CRONQUESTIONS dataset statistics as well as the numbers of questions across different types of reasoning required and answer types.

4.1 Experimental Setup

Data. CRONQUESTIONS² is the largest known Temporal KGQA dataset consisting of two parts: a KG with temporal annotations, and a set of free-text questions requiring temporal reasoning. This Temporal KG has 125k entities and 328k facts (quadruples), while a set of 410k questions is given. The facts have the time spans in the edge. These time spans or timestamps were discretized to years.

This dataset consists of questions that can be categorized into two groups based on their answer type: entity questions where the answer is an entity in the KG, and time questions where the answer is a timestamp. The authors also categorize these questions into “simple reasoning” (including *simple entity* and *simple time* subtypes) and “complex reasoning” (including *before/after*, *first/last* and *time join* subtypes). Table 1 provides the number of questions across different categories. Complex questions require complex temporal reasoning which takes advantage of multiple facts and temporal order of these facts.

Evaluation Metrics include Hits@1 and Hits@10, which is the standard evaluation metrics on CRONQUESTIONS (Saxena et al., 2021).

Hyper-parameter setting. We train the TSQA models by setting the hyper-parameters as: learning rate = $\{1e^{-4}, 2e^{-5}, 1e^{-5}\}$, $\lambda_o = \{0.5, 1.0, 2.0, 3.0, 5.0\}$ and $\lambda_c = \{0.5, 1.0, 2.0, 3.0, 5.0\}$, and pick up

²<https://github.com/apoorvumang/CronKGQA>

the best hyper-parameters on dev set by the overall Hits@1 metrics. Our models are implemented by PyTorch and trained using NVIDIA Tesla V100 GPUs.

Baselines. We select several recent SOTA TKG-QA models as our baselines as follow:

- EmbedKGQA (Saxena et al., 2020) is the first method to use KG embeddings for the multi-hop KGQA task. It uses ComplEx (Trouillon et al., 2016) embeddings and can only deal with non-temporal KGs and single entity questions.
- T-EaE-add/replacement (Saxena et al., 2021) are two modifications of KG enhanced language model EaE (Février et al., 2020), which integrates entity knowledge into a transformer-based language model and has been used for TKG-QA (Saxena et al., 2020). T-EaE-add has all grounded entities and time spans marked in the question, and T-EaE-replace replaces the BERT embeddings with the entity/time embeddings instead of adding them with token embeddings.
- CronKGQA (Saxena et al., 2021) extends EmbedKGQA to the temporal QA task, and takes advantage of the temporal KG embeddings to answering temporal questions. This is the current SOTA model on CRONQUESTIONS.

4.2 Main Results

Table 2 compares different TKG-QA methods in terms of Hits@1 and Hits@10. From this table, we observe that: 1) our proposed TSQA has achieved state-of-the-art performance in terms of all types of questions on both Hits@1 and Hits@10. 2) The performance improvement over the SOTA model is significant. TSQA outperforms the SOTA results by more than 82% Hits@1 relative improvement (32% absolute error reduction) on complex questions and 21% Hits@10 relative improvement on simple questions. These results proved the excellent performance of our proposed TSQA on question answering on the temporal knowledge graph, especially for complex temporal reasoning.

We also compare our method with baselines in terms of Hits@1 on different subtype questions in Table 3. From this table, we observe that: on complex questions, our proposed TSQA model outperforms all baseline models significantly. The relative improvement is up to 75%, 94%, 56%, for “before/after”, “first/last” and “Time Joint”, respectively. The first two kinds of questions are more

Model	Hits@1					Hits@10				
	Overall	Question Type		Answer Type		Overall	Question Type		Answer Type	
		Complex	Simple	Entity	Time		Complex	Simple	Entity	Time
EmbedKGQA	0.288	0.286	0.290	0.411	0.057	0.672	0.632	0.725	0.850	0.341
T-EaE-add	0.278	0.257	0.306	0.313	0.213	0.663	0.614	0.729	0.662	0.665
T-EaE-replace	0.288	0.257	0.329	0.318	0.231	0.678	0.623	0.753	0.668	0.698
CronKGQA	0.647	0.392	0.987	0.699	0.549	0.884	0.802	0.992	0.898	0.857
TSQA	0.831	0.713	0.987	0.829	0.836	0.980	0.968	0.997	0.981	0.978

Table 2: Comparison of different TKG-QA models on CRONQUESTIONS dataset.

Question type	Before After	First Last	Time Join	Simple Entity	Simple Time
EmbedKGQA	0.199	0.324	0.223	0.421	0.087
T-EaE-add	0.256	0.285	0.175	0.296	0.321
T-EaE-replace	0.256	0.288	0.168	0.318	0.346
CronKGQA	0.288	0.371	0.511	0.988	0.985
TSQA	0.504	0.721	0.799	0.988	0.987

Table 3: Comparison of different models w.r.t. question type in terms of Hits@1.

challenging as they require a better understanding of the temporal expressions in question. Our method is better in capturing such time-sensitivity change in temporal words and thus results in great improvement. Moreover, for the simple questions, our method still keeps competitive performance compared to the SOTA model.

4.3 Ablation Study

To understand the contributions of the proposed modules in our method, we perform an ablation study by sequentially removing the following components from our proposed TSQA: temporal Contrastive learning (TC), time-aware TKG embeddings (TKE), entity neighboring graph extractor (NG), and time estimation for question answer (TE) in Table 4. It is noted that removing TKE means that we replace TKE with T-CompLEx as KG encoder, and removing NG means that we perform QA over the whole knowledge graph.

By comparing the two adjacent rows of this table, we can infer the contributions of TC, TKE, NG and TE, respectively: 1) all these modules improve the overall performance in terms of Hits@1, especially for complex questions; 2) by comparing the last two adjacent rows, the proposed time estimation brings significant Hits@1 improvement (14.5%), since this module supplies the latent time embedding which not only enhances the interaction of timestamp estimation and answer estimation but also supplies a good anchor for finding the answer entity, which is very crucial for answering complex questions; 3) entity neighboring graph extraction gets 7.8% Hits@1 improvement over

complex questions by comparing rows “TC-TKE” and “TC-TKE-NG”, since it significantly narrows down the search space of the candidate answers; 4) by comparing the first three rows, time-aware TKG embedding (TKE) and temporal contrastive learning (TC) further boost the Hits@1 over complex questions. This is because the complex questions usually require the model to capture time ordering information implied in temporal words of the question. And these two modules enhance temporal order learning by adding explicit time-order constraints.

Model	Hits@1				
	Overall	Question Type		Answer Type	
		Complex	Simple	Entity	Time
TSQA	0.831	0.713	0.987	0.829	0.836
-TC	0.821	0.696	0.984	0.820	0.822
-TC-TKE	0.816	0.688	0.985	0.816	0.818
-TC-TKE-NG	0.757	0.583	0.986	0.797	0.687
-TC-TKE-NG-TE	0.661	0.412	0.989	0.719	0.556

Table 4: Results of the ablation study. “-” means to remove a module.

5 Conclusion

In this paper, we propose a time-sensitive question answering framework (TSQA) over temporal knowledge graphs (KGs). To facilitate the reasoning over temporal and relational facts over multiple facts, we propose a time estimation component to infer the unstated timestamp in the question. To further improve the model’s sensitivity to time relation words in the question and facilitate temporal reasoning, we enhance the model with a temporal KG encoder that produces KG embeddings that can recover the implicit temporal order and distance between different timestamps, and with contrastive losses that compare temporally exclusive questions. With the help of answer search space pruning from entity neighboring sub-graphs, our TSQA model significantly improves the performance on complex temporal questions that require reasoning over multiple pieces of facts, and outperforms the previous state of the art by a large margin.

Acknowledgements

This work is supported by the National Key Research and Development Program of China under Grant No. 2020AAA0108600.

References

- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.
- Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2001–2011.
- Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-second AAAI conference on artificial intelligence*.
- Bhuvan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. 2021. Time-aware language models as temporal knowledge bases. *arXiv preprint arXiv:2106.15110*.
- Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. *arXiv preprint arXiv:2004.07202*.
- Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. *arXiv preprint arXiv:1809.03202*.
- Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Rujun Han, Xiang Ren, and Nanyun Peng. 2021. Econet: Effective continual pretraining of language models for event temporal reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5367–5380.
- Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 105–113.
- Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*.
- Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jan-nik Strötgen, and Gerhard Weikum. 2018a. Tempquestions: A benchmark for temporal question answering. In *Companion Proceedings of the The Web Conference 2018*, pages 1057–1062.
- Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jan-nik Strötgen, and Gerhard Weikum. 2018b. Tequila: Temporal question answering over knowledge bases. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1807–1810.
- 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.
- Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Baobao Chang, Sujian Li, and Zhifang Sui. 2016. Towards time-aware knowledge graph completion. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1715–1724.
- Woojeong Jin, Rahul Khanna, Suji Kim, Dong-Ho Lee, Fred Morstatter, Aram Galstyan, and Xiang Ren. 2021. Forecastqa: A question answering challenge for event forecasting with temporal text data. 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 4636–4650.
- Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. *arXiv preprint arXiv:2004.04926*.
- Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. Torque: A reading comprehension dataset of temporal ordering questions. *arXiv preprint arXiv:2005.00242*.
- Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. 2021. Question answering over temporal knowledge graphs. *arXiv preprint arXiv:2106.01515*.
- 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.
- Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.
- Chao Shang, Peng Qi, Guangtao Wang, Jing Huang, Youzheng Wu, and Bowen Zhou. 2021. Open temporal relation extraction for question answering. In

3rd Conference on Automated Knowledge Base Construction.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

Yun Tang, Jing Huang, Guangtao Wang, Xiaodong He, and Bowen Zhou. 2019. Orthogonal relation transforms with graph context modeling for knowledge graph embedding. *arXiv preprint arXiv:1911.04910*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jiapeng Wu, Meng Cao, Jackie Chi Kit Cheung, and William L Hamilton. 2020. Temp: temporal message passing for temporal knowledge graph completion. *arXiv preprint arXiv:2010.03526*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.