Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)

Local and Global: Temporal Question Answering via Information Fusion

Yonghao Liu[1] , **Di Liang**[2] , **Mengyu Li**[1] , **Fausto Giunchiglia**[3] , **Ximing Li**[1] , **Sirui Wang**[2] ,
Wei Wu[2] , **Lan Huang**[1] , **Xiaoyue Feng**[1*] and **Renchu Guan**[1*]
[1]The Key Laboratory for Symbol Computation and Knowledge Engineering of the
Ministry of Education, College of Computer Science and Technology, Jilin University
[2]Center for Natural Language Processing, Meituan Inc
[3]University of Trento
{yonghao20, mengyul21}@mails.jlu.edu.cn,
liximing86@gmail.com, fausto.giunchiglia@unitn.it
{huanglan, fengxy, guanrenchu}@jlu.edu.cn

Abstract

Many models that leverage knowledge graphs (KGs) have recently demonstrated remarkable success in question answering (QA) tasks. In the real world, many facts contained in KGs are time-constrained thus temporal KGQA has received increasing attention. Despite the fruitful efforts of previous models in temporal KGQA, they still have several limitations. (I) They neither emphasize the graph structural information between entities in KGs nor explicitly utilize a multi-hop relation path through graph neural networks to enhance answer prediction. (II) They adopt pre-trained language models (LMs) to obtain question representations, focusing merely on the global information related to the question while not highlighting the local information of the entities in KGs. To address these limitations, we introduce a novel model that simultaneously explores both **L**ocal information and **G**lobal information for the task of temporal KG**QA** (LGQA). Specifically, we first introduce an auxiliary task in the temporal KG embedding procedure to make timestamp embeddings time-order aware. Then, we design information fusion layers that effectively incorporate local and global information to deepen question understanding. We conduct extensive experiments on two benchmarks, and LGQA significantly outperforms previous state-of-the-art models, especially in difficult questions. Moreover, LGQA can generate interpretable and trustworthy predictions.

1 Introduction

QA aims to answer questions expressed in natural language via specific answers and has a wide range of application scenarios. Recently, many studies have been devoted to the use of KGs containing facts in the form of (*subject*, *relation*, *object*) as external knowledge sources to improve the performance of QA [Lukovnikov *et al.*, 2017; Zhang *et al.*, 2018; Liang *et al.*, 2019a; Liang *et al.*, 2019b; Huang *et al.*, 2019]. Notably, some facts are associated with temporal properties (*i.e.*, timestamps or time intervals), which are typically represented in the form of quadruples (*subject*, *relation*, *objective*, *time*), for example, (*Cristiano Ronaldo*, *member of*, *Manchester United FC*, *[2003, 2009]*). Studies on QA of KGs consisting of time-dependent facts have received increasing attention from both academia and industry. This line of work follows a dominant learning paradigm, in which questions are fed into large-scale pre-trained LMs to obtain the corresponding question representations, and then the representations are combined with entity embeddings obtained using KG embedding algorithms to infer the correct answers [Saxena *et al.*, 2021; Mavromatis *et al.*, 2021].

Despite the relative success of previous models in the task of temporal KGQA, existing efforts can be greatly compromised in practice, primarily due to their limitations in the following respects: (I) Existing approaches almost do not emphasize the graph structural information among entities in KGs and fail to model multi-hop relational paths explicitly, which are beneficial for reasoning, as demonstrated in previous research [Ren *et al.*, 2020]. In essence, these models are retrieval-based approaches that perform well in simple question reasoning. For example, the question "*Which team was Cristiano Ronaldo part of in 2006?*" can be answered with a single fact (*Cristiano Ronaldo*, *member of*, *Manchester United FC*, *[2003, 2009]*) from a KG. However, such models struggle when answering the given questions requires multiple facts or multi-hop reasoning (*i.e.*, complex question reasoning). Hence, incorporating the structural information of KGs can facilitate complex question reasoning, which remains unexplored in temporal KGQA tasks.

(II) The global information (*i.e.*, sentence-level semantic information) related to the question and the local information (*i.e.*, entity-level information) of the entities involved are essential to answer the question. However, in previous methods, question understanding is typically performed by pre-trained LMs that implicitly encode the corpus only. In other words, they consider only the global information and ignore the rich semantic information (*i.e.*, local information) of the entities involved. For example, for the question "*With whom did Cris-*

*Corresponding Author

tiano Ronaldo play on the FC in 2006?", the local information (*i.e.*, *Manchester United FC*) is not contained explicitly but exists in the sub-graph containing the entity *Cristiano*'s æ-hop neighbors extracted from KGs. Therefore, it is beneficial to infer the answer if the entity-level semantic information of the extracted sub-graphs is captured. Moreover, these proposed methods lack a certain transparency about their predictions, since they do not model the reasoning paths well and the whole process is invisible. As a result, interpreting the reasoning process is challenging.

To address the aforementioned limitations, in this work, we propose a novel model, LGQA, for temporal KGQA. Our goal is to develop a reasoning model that can effectively infer answer entities for the given questions. Concretely, we first employ temporal KG embedding algorithms based on given temporal KGs to obtain the embeddings of entities, relations, and timestamps. Notably, to build the timestamp embeddings with prior knowledge of the temporal order, we employ an auxiliary task for each pair of timestamp embeddings, which is crucial for further improvements in the model performance. Then, to address limitation I, we explicitly leverage the structural information among entities of KGs via the graph neural networks (GNNs). Moreover, to directly model relational paths, we perform multi-hop message aggregation that allows each node to access its æ-hop neighbors within a single propagation layer, which is significantly superior to one-hop propagation. Next, to solve limitation II, we extract the æ-hop subgraph of the entities from KGs and then perform the above multi-hop message passing to obtain the entities' local information. At the same time, we feed the question into LMs to obtain its global information. Finally, we combine the local and global information into a sophisticated information fusion layer, followed by a model prediction layer. In modeling relational paths, we introduce an attention mechanism to score the reasoning path. In this way, our model can be interpreted according to this score when reasoning.

Overall, our contributions in this work are as follows:

- We propose a novel model named LGQA, which can effectively understand a question and infer the correct answer. To the best of our knowledge, we are the first to apply GNN layers with a multi-hop message passing paradigm for temporal KGQA.
- We leverage the structural information of KGs and combine global and local information for the given questions. Additionally, our model can provide trustworthy predictions based on the attention weights of the relevant reasoning paths.
- We perform extensive experiments on two widely used benchmarks, and the empirical results demonstrate the significant superiority of our model compared to other competitive baselines.

2 Related Work

Temporal KGQA. Generally, KG embedding algorithms [Bordes *et al.*, 2013; Trouillon *et al.*, 2017] are employed to initialize entity and relation embeddings to help answer a question in the task of KGQA [Saxena *et al.*, 2020]. For temporal KGQA, we typically adopt temporal KG embedding approaches, such as TComplEx [Lacroix *et al.*, 2020], for initializing and also obtain the timestamp embeddings. Recently, many researchers have focused on temporal KGQA and have proposed corresponding methods for this task. Among these models, there are three representative ones: CronKGQA [Saxena *et al.*, 2021], TSQA [Shang *et al.*, 2022], and TempoQR [Mavromatis *et al.*, 2021]. CronKGQA utilizes recent advances in temporal KG embeddings and feeds the given questions to pre-trained LMs for answer prediction. TSQA is equipped with a time estimation module that allows unwritten timestamps to be inferred from questions, and presents a contrastive learning module that improves sensitivity to time relation words. TempoQR designs three modules to deepen the question understanding with context, entity, and time-aware information.

Graph Neural Networks. GNNs have attracted much attention due to their ability to model structured data and have been developed for various applications in practice [Liu *et al.*, 2021a; Liu *et al.*, 2021b; Liu *et al.*, 2022]. Among these models, graph convolutional network (GCN) [Kipf and Welling, 2017] is a pioneering work that designs a local spectral graph convolutional layer for learning node embeddings. GraphSAGE [Hamilton *et al.*, 2017] generates node embeddings by learning an aggregator function that samples and aggregates features from the nodes' local neighborhoods. Graph Attention Network (GAT) [Veličković *et al.*, 2018] assigns different weights to different neighbors of a node to learn its representations by introducing self-attention mechanisms. Recently, several models [Feng *et al.*, 2020; Yasunaga *et al.*, 2021] have been designed to shift the power of GNNs to general QA tasks. However, these models use vanilla GNNs that adopt a one-hop neighbor aggregation mechanism, which may limit their expressiveness. Additionally, these models cannot be directly applied to our focused scenarios, *i.e.*, temporal KGQA.

3 Definition

Temporal KGQA aims to find suitable answers from KGs $G = (V, E, R, T)$ for given free-text questions. The answer is either an entity from entity set V or a timestamp from timestamp set T. Here, R and E represent the union sets of relations and edges. Each edge represents a valid fact in the form of quadruples (s, r, o, t), where $s, o \in V$ are the subject and objective entities, $r \in R$ is the relation, and $t \in T$ is the timestamp, respectively.

Following previous models [Saxena *et al.*, 2021], we formalize temporal KGQA as a link prediction problem. The underlying idea is to regard the question as a virtual relation to infer the answer. For example, for the question q "*What award did Cristiano Ronaldo receive in 2008?*", we can answer it with the single fact (*Cristiano Ronaldo, award received, Ball d'Or*). In fact, we can infer the relation "award received" from the question's content, *i.e.*, virtual relation. Thus, we can solve it by the link prediction manner, which can be transformed into (*Cristiano Ronaldo, q, ?, 2008*).

Temporal KG embeddings aim to learn low-dimensional embeddings based on the facts contained in the KG. Concretely, we embed $s, o \in V$, $r \in R$, and $t \in T$ based on the predefined score function $\phi(\cdot)$ to obtain the corresponding embeddings $e_s, e_o, e_r, e_t \in \mathbb{R}^{2D}$. Typically, the valid fact (s, r, o, t) is scored much higher than invalid facts (s', r', o', t'), *i.e.*, $\phi(s, r, o, t) \gg \phi(s', r', o', t')$.

4 Method

In this section, we introduce our proposed model, LGQA, for temporal KGQA, which includes three key modules: *time-sensitive KG embedding*, *information fusion* and *answer prediction*. To better describe the method, we present the overall framework in Fig. 1. Next, we will elaborate on each module.

4.1 Time-Sensitive KG Embedding

We start by obtaining the embeddings of the entity, relation, and timestamp in the temporal KG using a time-sensitive KG algorithm. TComplEx, a prevalent method, can produce high-quality temporal KG embeddings. Specifically, it is defined in the complex space and its score function is as follows:

$$\phi(e_s, e_r, e_o, e_t) = \mathbf{Re}(\langle e_s, e_r \odot e_t, \bar{e}_o \rangle) \tag{1}$$

where $\mathbf{Re}$ denotes the real part in the complex space and $\langle\cdot\rangle$ represents the multi-linear product operation. Additionally, e_s, e_r, e_o, e_t are complex-valued embeddings and $\bar{e}_o$ is the complex conjugate of e_o.

Due to the learning procedure of TComplEx, it is proficient at inferring missing facts in temporal KGs, such as $(s, r, ?, t)$ and $(s, r, o, ?)$, which is suitable for our scenarios. Therefore, in this work, we combine it with temporal order information to generate pre-trained temporal KG embeddings.

However, the vanilla TComplEx algorithm does not explicitly consider the sequential ordering information of timestamps, which is detrimental to reasoning based on temporal signals. For example, for the question "*Who was awarded the Ballon d'Or after Lionel Messi?*", the relevant facts are (*Lionel Messi, award received, Ballon d'Or, [2009, 2009]*) and (*Cristiano Ronaldo, award received, Ballon d'Or, [2013, 2013]*). In the embedding space, it is helpful to be aware that 2013 is later than 2009 when answering this question. Inspired by the usage of position embeddings [Vaswani *et al.*, 2017; Jia *et al.*, 2021], we inject temporal order information into timestamp embeddings via an auxiliary task while training temporal KGs. Specifically, we define the position embedding of the k-th timestamp $\mathbf{t}_k$ as follows:

$$\mathbf{t}_k(c) = \begin{cases} \sin(k/10000^{2i/2d}), \text{if } c = 2i \\ \cos(k/10000^{2i/2d}), \text{if } c = 2i+1 \end{cases} \tag{2}$$

where $2d$ is the dimension of timestamps and c denotes the even or odd position in the $2d$-dimensional vector. We can obtain the position embedding $\mathbf{t}_k \in \mathbb{R}^{2d}$ via Eq. 2. This position encoding method has the properties of uniqueness (*i.e.*, different timestamps have different position embeddings) and sequential ordering (*i.e.*, it can reflect the relative positions among timestamps). Next, we adopt linear regression to obtain the probability of timestamp m being ahead of timestamp n for the given pair (m, n). A binary cross-entropy objective function is employed in this auxiliary task. The concrete formulas are as follows:

$$\begin{aligned} \rho(m,n) &= \sigma(\mathbf{W}_{ts}^{\top}((e_m + \mathbf{t}_m) - (e_n + \mathbf{t}_n))) \\ \mathcal{L}_{ts}(m,n) &= -\alpha(m,n)\log(\rho(m,n)) \\ &\quad - (1-\alpha(m,n))\log(1-\rho(m,n)) \end{aligned} \tag{3}$$

where $\sigma(\cdot)$ and $\mathbf{W}_{ts}$ are the sigmoid function and learnable parameters. e_* and $\mathbf{t}_*$ are the trainable timestamp embeddings and the corresponding position embeddings. $\alpha(m, n)$=1 if $m < n$ and 0 otherwise, and $\rho(m, n)$ is the predicted probability of the time order. The subsequent timestamp embeddings e_t are obtained with the corresponding position embeddings added (*i.e.*, $e_{t,i} = e_{t,i} + \mathbf{t}_i$). And the final loss function of this module is combined with the loss function of the auxiliary task, *i.e.*, $\mathcal{L}_{ts}$. We can obtain the desired trained embeddings of temporal KGs by performing joint training.

4.2 Information Fusion

This module aims to generate enhanced question representations by incorporating the local information of temporal KGs and the global information of pre-trained LMs.

(I) Local Information. Given the temporal KG $G = (V, E, R, T)$, we initialize the node and edge features by the pre-trained time-sensitive KG encoder. Specifically, the value of a node is the corresponding entity embedding. The value of an edge is the concatenation of the relation and timestamp embedding, *i.e.*, $e_r || e_t$. The idea is to propagate both relations and timestamps via graph structures, which is specific to temporal KGQA tasks.

Next, we obtain annotated entities $\{\text{ent}_1, \text{ent}_2, \cdots, \text{ent}_w\}$, which are pre-annotated by hand-crafted templates, from each question q. For each entity ent_i, we then extract its æ-hop sub-graph G_i. The final relevant æ-hop sub-graph G_q for the question can be obtained by combining each entity's sub-graph, *i.e.*, $G_q = \cup_{i=1}^{w} G_i$. Note that we restrict the answer selection to G_q via the latent sub-graph extraction procedure, which can greatly reduce the search space and effectively facilitate the training process.

To directly leverage the structural information among entities of temporal KGs, we apply GNNs to the extracted sub-graph. Typically, the classic message passing paradigm of GNNs can be formulated as:

$$\begin{aligned} a_v^{\ell} &= \textbf{AGGREGATE}(\{h_u^{\ell-1} : u \in \mathcal{N}_v\}) \\ h_v^{\ell} &= \textbf{COMBINE}(h_v^{\ell-1}, a_v^{\ell}) \end{aligned} \tag{4}$$

where $\mathcal{N}_v$ is the set of node v's neighbors. a_v^{ℓ} is the aggregated message at layer ℓ, and h_v^{ℓ} is node v's embeddings at layer ℓ obtained by combining $h_v^{\ell-1}$ and a_v^{ℓ}. However, in the above framework, the nodes in the graph can only access their one-hop neighbors through a single graph layer. In other words, suppose two nodes are not directly connected, they can only interact with each other by stacking a sufficient number of layers, which severely limits the capability of GNNs to explore the relationships between disjoint nodes.

To address this problem, we adopt a multi-hop message passing mechanism that works on all possible paths between

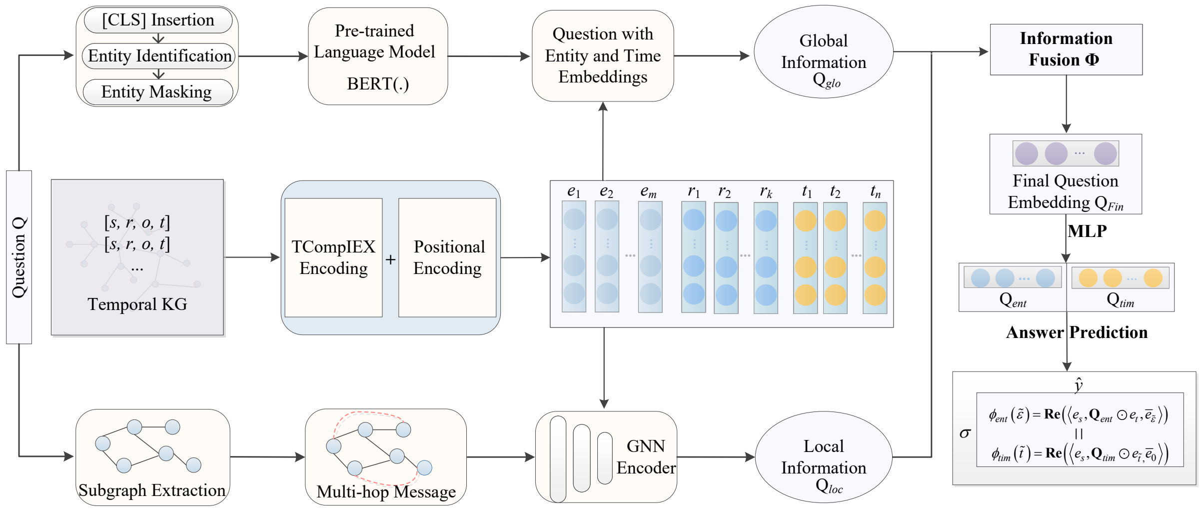

Figure 1: The overall architecture of our model (Best viewed in color).

two nodes. The first step is to compute the normalized attention using Eq. 5.

$$\mathscr{A}_{irj}^{\ell} = \begin{cases} \delta(\mathbf{W}_{ad}^{\ell}(h_i^{\ell}||h_j^{\ell}||h_r||h_t)), (v_i, r, v_j, t) \in G \\ -\infty, \text{otherwise} \end{cases} \tag{5}$$
$$\mathbf{A}^{\ell} = \text{softmax}(\mathscr{A}^{\ell})$$

where $\mathbf{W}_{ad}^{\ell}$ is the learnable weight shared by the ℓ-th layer. h_i^{ℓ} is the embedding of node i, initialized by $h_i^0 = e_i$. h_r and h_t are the embeddings of relation r and timestamp t, respectively. δ denotes the ReLU activation function. $\mathscr{A}^{\ell}$ and $\mathbf{A}^{\ell}$ represent the attention matrix obtained by applying the edges appearing in G and the normalized attention matrix derived by performing a row-wise softmax function, respectively. $||$ denotes the concatenation operation. In addition, since paths with different importance are assigned corresponding weights using Eq. 5, we can derive the reasoning path based on these weights, which is further discussed in Section 6.

To enable the aggregation of multi-hop messages to a target node within a single propagation layer, we employ a mechanism defined as follows:

$$\mathbf{D} = \sum\nolimits_{\tau=0}^{\aleph} \xi_{\tau} \mathbf{A}^{(\tau)} \tag{6}$$

where ξ_{τ} are trainable coefficients. $\mathbf{A}^{(\tau)}$ is the powers of $\mathbf{A}$, which considers relational paths with length limits up to τ from neighboring nodes to the target node. In other words, the target node's context (*i.e.*, intermediate neighbors) and its local graph structure are involved in attention calculation. This procedure successfully creates attentional interactions between a node and its disjoint neighbors beyond one-hop. In practice, we can achieve impressive performance when empirically setting the diffusion distance $\aleph \in [2, 4]$ since many graphs have small-world properties with lower diameters.

Subsequently, the transition matrix $\mathbf{D}$ is leveraged to update the nodes' embeddings to obtain $\mathbf{H}^{\ell+1}$ in Eq. 7.

$$\mathbf{H}^{\ell+1} = \mathbf{D}\mathbf{H}^{\ell} \tag{7}$$

Finally, we perform an average pooling operation on the nodes of the extracted sub-graph to acquire the question's local information $\mathbf{Q}_{loc}$, formulated as Eq. 8.

$$\mathbf{Q}_{loc} = \frac{1}{|V_q|} \sum\nolimits_{i \in V_q} h_i^L \tag{8}$$

where V_q is the node set of the sub-graph and h_i^L is the node embeddings at the L-th layer.

(II) Global Information. To obtain the global information of the question, we feed the question to the pre-trained LMs, such as BERT, since such models implicitly encode world knowledge. Concretely, we first insert the [CLS] token into question q. Then, we identify all the entities in q and mask them with the [MASK] token. For example, for q "*Who is the president of USA after Obama?*", we identify the entities "president of USA" and "Obama" and transform q into "[CLS] Who is the [MASK] after [MASK]". Finally, the tokenized question is fed into BERT, and it can be expressed as:

$$\bar{\mathbf{Q}} = \mathbf{W}_q \text{BERT}(q) \tag{9}$$

where $\mathbf{W}_q$ is the projection matrix. In addition, $\bar{\mathbf{Q}} = [\bar{\mathbf{Q}}_{\text{[CLS]}}, \bar{\mathbf{Q}}_1, \cdots, \bar{\mathbf{Q}}_o]$ is an embedding matrix. We adopt the [CLS] token embedding, $\bar{\mathbf{Q}}_{\text{[CLS]}}$, as the representation of the entity-independent question q. For the masked entities, we use pre-trained temporal KG entity embeddings. In other words, if the question contains two annotated entities, the global information is $\mathbf{Q}_{glo} = \bar{\mathbf{Q}}_{\text{[CLS]}} + e_1 + e_2$. To further enhance question representation and to make full use of the available data, we retrieve the relevant facts of the annotated entities in the question from the temporal KG, so that we can obtain the question-specific time scope. If we retrieve multiple timestamps of relevant facts, we sort them and keep only the start time and end time. For example, for question q, we can retrieve the fact (*Barack Obama, held position, president of USA, [2008, 2016]*) and obtain two time embeddings t_1 and t_2 that correspond to the temporal KG embedding for start time 2008 and end time 2016, respectively. Hence, the global information can be rewritten as $\mathbf{Q}_{glo} = \bar{\mathbf{Q}}_{\text{[CLS]}} + e_1 + e_2 + t_1 + t_2$.

To better integrate the question's local and global information, we employ a sophisticated knowledge fusion layer, $\Phi(\cdot)$, that contains several Transformer encoder layers. After performing the Transformer-based information fusion layer, we obtain the final question representation, *i.e.*, $\mathbf{Q}_{fin} = \Phi(\mathbf{Q}_{loc}||\mathbf{Q}_{glo})$.

4.3 Answer Prediction

We use two-layer MLPs to transform $\mathbf{Q}_{fin}$ into $\mathbf{Q}_{ent}$ and $\mathbf{Q}_{tim}$, which correspond to entity and timestamp prediction, respectively, and are defined in Eq. 10.

$$\begin{aligned} \mathbf{Q}_{ent} &= \mathbf{MLP}(\mathbf{Q}_{fin}) \\ \mathbf{Q}_{tim} &= \mathbf{MLP}(\mathbf{Q}_{fin}) \end{aligned} \tag{10}$$

Next, we define an entity score function $\phi_{ent}(\cdot)$ and a timestamp score function $\phi_{tim}(\cdot)$ to obtain the scores of candidate entities and timestamps, as shown in Eq. 11.

$$\begin{aligned} \phi_{ent}(\tilde{\varepsilon}) &= \mathbf{Re}(\langle e_s, \mathbf{Q}_{ent} \odot e_t, \bar{e}_{\tilde{\varepsilon}} \rangle) \\ \phi_{tim}(\tilde{t}) &= \mathbf{Re}(\langle e_s, \mathbf{Q}_{tim} \odot e_{\tilde{t}}, \bar{e}_o \rangle) \end{aligned} \tag{11}$$

where $\tilde{\varepsilon} \in E_q$ and $\tilde{t} \in T_q$, in which $E_q \subseteq E$ and $T_q \subseteq T$ are specified by the sub-graph G_q with respect to the given question q.

Finally, we concatenate the obtained scores for the entities and timestamps and perform the softmax function over them to obtain the answer probability. The objective function is the cross-entropy loss, as shown in Eq. 12.

$$\begin{aligned} \hat{y}_i &= \text{softmax}(\phi_{ent}(\cdot)||\phi_{tim}(\cdot)) \\ \mathcal{L}_{predict} &= -\sum\nolimits_i y_i \log(\hat{y}_i) \end{aligned} \tag{12}$$

where y_i is the true answer to the question.

5 Experiment

Datasets. We employ two temporal KGQA benchmarks, *i.e.*, CRONQUESTIONS [Saxena *et al.*, 2021] and TimeQuestions [Jia *et al.*, 2021]. **CRONQUESTIONS** is the largest known dataset, which has 410K unique question-answer pairs, where each question contains annotated entities and timestamps. Moreover, this dataset can be divided into entity and time questions based on the type of answers. It can also be divided into simple reasoning (*i.e.*, Simple Entity and Simple Time) and complex reasoning (*i.e.*, Before/After, First/Last and Time Join) based on the questions' difficulty. **TimeQuestions** is another challenging dataset, which has 16k manually tagged temporal questions and is divided into four categories (*i.e.*, Explicit, Implicit, Temporal, and Ordinal) according to the type of time reasoning. We present the statistical information of the datasets in Tables. 1 and 2.

Baselines. We select three types of baselines for comparison on CRONQUESTIONS: (I) pre-trained LMs, including BERT [Devlin *et al.*, 2019], RoBERTa [Liu *et al.*, 2019] and KnowBERT [Peters *et al.*, 2019]; (II) general KG embedding-based models, including EaE [Févry *et al.*, 2020] and EmbedKGQA [Saxena *et al.*, 2020]; and (III) temporal KG embedding-based models, including CronKGQA [Saxena *et al.*, 2021], TMA [Liu *et al.*, 2023], TSQA [Shang *et al.*, 2022], TempoQA [Mavromatis *et al.*, 2021], and CTRN [Jiao *et al.*, 2022]. For another dataset, TimeQuestions, we use temporal KG embedding-based models for comparison.

Model Implementations. We set the weighted coefficient in the KG encoder stage as $\lambda = 0.5$. In the second stage, we extract a 3-hop sub-graph of the question, *i.e.*, æ=3. Moreover, the hop is set to $\aleph = 3$. We perform 2-layer GNNs to obtain the updated node embeddings, *i.e.*, $L = 2$. Furthermore, we use 3-layer Transformers with 4 heads per layer in the knowledge fusion layer $\Phi(\cdot)$. We train our model for 20 epochs with Adam methods, and the validation performance determines its final parameters. We conduct all experiments ten times and take the average values as the final results. We leverage two popular evaluation metrics, *i.e.*, Hits@1 and Hits@10, following previous studies.

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

Table 1: Statistical information of CRONQUESTION.

Category	Train	Dev	Test
Explicit	2,724	1,302	1,311
Implicit	651	291	292
Temporal	2,657	1,073	1,067
Ordinal	938	570	567
Total	6,970	3,236	3,237

Table 2: Statistics information of TimeQuestions.

6 Result

Model Performance. We present the results of our proposed LGQA and baselines on CRONQUESTIONS in terms of Hits@1 and Hits@10 in Table 3 and on TimeQuestions for Hits@1 in Table 4. LGQA achieves the best performance in all experimental settings, indicating its superiority on the temporal KGQA task. Remarkably, LGQA significantly outperforms the second-best model on both datasets. It achieves 7.6% and 4.9% absolute improvements on Hits@1 with respect to complex reasoning and time questions on CRONQUESTION, respectively. It also performs far better than other models for various types of questions in the TimeQuestions dataset. For example, it achieves absolute improvements of 6.3% and 6.0% on Hits@1 for questions involving 'Explicit' and 'Implicit' types. While in the 'Temporal' type of questions, our model gains an absolute improvement of 9.3% compared to the second best performing model. We attribute this to the use of the multi-hop propagation of knowledge fusion and the time-sensitive KG embedding.

Model	Hits@1					Hits@10				
	Overall	Question Type		Answer Type		Overall	Question Type		Answer Type	
		Complex	Simple	Entity	Time		Complex	Simple	Entity	Time
BERT	0.243	0.239	0.249	0.277	0.179	0.620	0.598	0.649	0.628	0.604
RoBERTa	0.225	0.217	0.237	0.251	0.177	0.585	0.542	0.644	0.583	0.591
KnowBERT	0.226	0.220	0.238	0.252	0.177	0.586	0.539	0.646	0.582	0.592
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
TMA	0.784	0.632	0.987	0.792	0.743	0.943	0.904	0.995	0.947	0.936
TSQA	0.831	0.713	0.987	0.829	0.836	0.980	0.968	0.997	0.981	0.978
TempoQR	0.918	0.864	0.990	0.926	0.903	0.978	0.967	0.993	0.980	0.974
CTRN	0.920	0.869	0.990	0.921	0.917	0.980	0.970	0.993	0.982	0.976
LGQA	**0.969**	**0.945**	**0.992**	**0.962**	**0.966**	**0.991**	**0.985**	**0.998**	**0.991**	**0.988**

Table 3: Performance of different models on CRONQUESTIONS.

Model	Overall	Explicit	Implicit	Temporal	Ordinal
CronKGQA	0.393	0.388	0.380	0.436	0.332
TMA	0.436	0.442	0.419	0.476	0.352
TempoQR	0.459	0.503	0.442	0.458	0.367
CTRN	0.465	0.469	0.446	0.512	0.382
LGQA	**0.529**	**0.532**	**0.506**	**0.605**	**0.402**

Table 4: Hits@1 for different models on TimeQuestions.

Category	Complex Question			Simple Question		All
	Before/ After	First/ Last	Time Join	Simple Entity	Simple Time	
EmbedKGQA	0.199	0.324	0.223	0.421	0.087	0.288
T-EaE-add	0.256	0.285	0.175	0.296	0.321	0.278
T-EaE-replace	0.256	0.288	0.168	0.318	0.346	0.288
CronKGQA	0.288	0.371	0.511	0.988	0.985	0.647
TMA	0.581	0.627	0.675	0.988	0.987	0.784
TSQA	0.504	0.721	0.799	0.988	0.987	0.831
TempoQR	0.714	0.853	0.978	0.988	0.987	0.918
CTRN	0.747	0.880	0.897	0.991	0.987	0.920
LGQA	**0.902**	**0.936**	**0.991**	**0.991**	**0.995**	**0.969**

Table 5: Hits@1 for different question types on CRONQUESTIONS.

We find that pre-trained LMs (*e.g.*, BERT and RoBERTa) achieve unsatisfactory performance in this scenario, lagging far behind the general and temporal KG embedding-based models on CRONQUESTIONS. A plausible reason is that these models do not introduce KG into this task, which is detrimental to question understanding. Despite the relative success of general KG embedding-based models (*e.g.*, EaE and EmbedKGQA) in common QA tasks, they still perform worse than temporal KG embedding-based models (*e.g.*, TSQA, TempoQR and CTRN) in our focused scenario. A possible reason is that they do not explicitly leverage temporal KG and neglect temporal information, which is crucial for the temporal KGQA task.

We present the Hits@1 results of our model and other competitive baselines on different question types in Table 5. LGQA is significantly superior to other models, especially for complex questions. Our model gains 15.5%, 5.6%, and 9.4% absolute improvement over “Before/After”, “First/Last” and “Time Join”, respectively, due to the consideration of the timestamp order and multi-hop structural information of the temporal KG. Additionally, our model has comparable performance on simple questions.

Model	Hits@1				
	Overall	Question Type		Answer Type	
		Complex	Simple	Entity	Time
LGQA	**0.969**	**0.945**	**0.992**	**0.962**	**0.966**
w/o time order	0.932	0.889	0.972	0.934	0.900
w/o multi-hop	0.926	0.878	0.970	0.932	0.899
w/o local	0.916	0.872	0.965	0.928	0.891
w/o global	0.716	0.643	0.652	0.625	0.596

Table 6: Results of ablation studies on CRONQUESTIONS.

Ablation Study. We conduct extensive ablation experiments on the crucial components by designing some model variants on the CRONQUESTION dataset. (I) *w/o time order*: We exclude the auxiliary task of encoding temporal order information and use the vanilla TComplEx method. (II) *w/o multi-hop*: We use the one-hop attention computed from the direct neighbors without multi-hop attention, similar to GAT. (III) *w/o local*: We remove the module for extracting local information. (IV) *w/o global*: We remove the module considering global information. The experimental results are presented in Table 6. We can obtain the following insights: First, after eliminating the global information module, the model's performance drops drastically, which is in line with our expectations. This result indicates that this module can provide helpful contextual information for accurately understanding the question. Second, since the local information can bring additional valuable information from KGs, eliminating it can negatively affect the model. Moreover, the performance declines when we perform one-hop message passing instead of multi-hop, empirically demonstrating that multi-hop message passing is more expressive. Finally, complex questions require the temporal order information to be captured, thus removing this information inevitably harms the model.

Hyperparameter Sensitivity. We empirically explore the effects of different hyperparameters by observing the performance of LGQA on the CRONQUESTIONS dataset. We

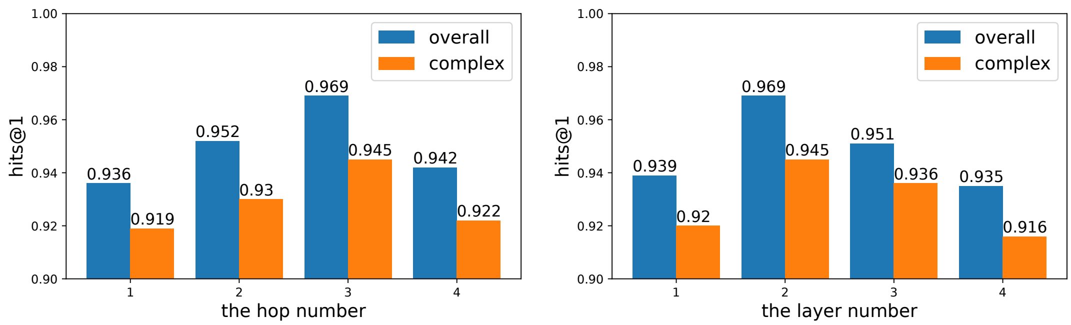

Figure 2: Performance changes for different hyperparameters on CRONQUESTIONs.

study the effect of the number of hops æ in the extracted sub-graphs and the number of layers L in GNNs. The hits@1 results in terms of all questions and complex questions are presented in Fig. 2. We find that the model can achieve the best performance when extracting the 3-hop sub-graph. A possible reason for this is that smaller sub-graphs may exclude correct answers, while larger sub-graphs increase the search space for candidate answers but may bring exponential noise from KGs. Moreover, as illustrated on the right side of Fig. 2, the model's performance shows a trend of increasing and then decreasing as the number of GNN layers increases.

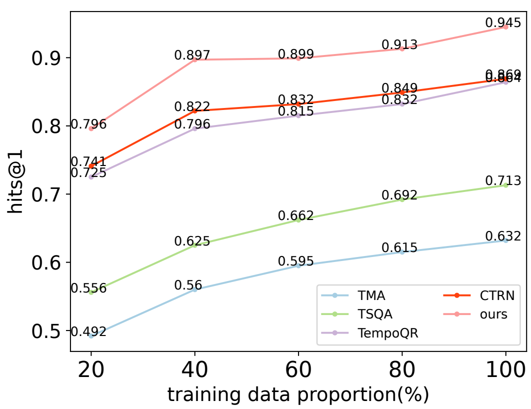

Figure 3: Performance *w.r.t.* different training data.

Impact of Training Data Size. We select several competitive models for comparison on complex questions of the CRONQUESTION dataset regarding hits@1 with different training data sizes. The experimental results are presented in Fig. 3. We find that our model consistently outperforms other baselines in all cases. Taking the 20% training data as an example, our model's hits@1 absolute improvement reaches 18.6% compared to the second-best-performing model. This demonstrates that, first, our proposed model exhibits superior expressive power in complex question reasoning. Second, it does not rely on large amounts of training data.

Model Interpretability. To interpret our model's reasoning process, we investigate the relational path attention weights induced by the attention layer of GNNs described in Eq. 5. Specifically, we trace high attention weights from entity

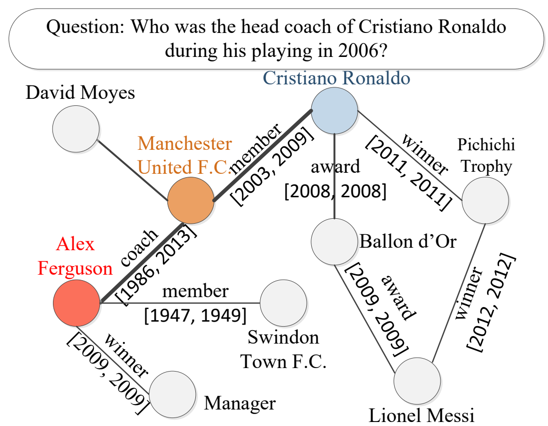

Figure 4: Visualization of a case study of the interpretability of our model. For brevity, we only show the key entities.

nodes to the candidate answer nodes on the retrieved sub-graph G_q by leveraging Best First Search (BFS). Fig. 4 illustrates one example. In this example, we note that the reasoning path contains "Cristiano Ronaldo" in the question and "Alex Ferguson" and "Manchester United F.C." in KGs. LGQA can make accurate predictions, *i.e.*, "Alex Ferguson", given the question. Notably, LGQA promotes rational reasoning by introducing "Manchester United F.C.", which is not mentioned in the question, revealing the importance of background knowledge. It provides an interpretable reasonable path "Cristiano Ronaldo→Manchester United F.C.→Alex Ferguson".

7 Conclusion

In this work, we propose a novel model, LGQA, to perform temporal KGQA tasks. Three specific modules are introduced to significantly improve the model's performance. Specifically, the time-sensitive KG embedding module is employed to add temporal ordering information. Moreover, the information fusion module with multi-hop message passing during the extraction of the æ-hop sub-graphs combines the local information with global information to understand questions. Finally, we obtain the answer based on the answer prediction module. Extensive experiments on two widely used datasets imply that LGQA achieves satisfying performance.

Acknowledgments

Our work is supported by the National Key Research and Development Program of China No.2021YFF1201200, the National Natural Science Foundation of China No.62172187, No.61972175, and No. 61972174.

Contribution Statement

Note that the first three authors, Yonghao Liu, Di Liang and Mengyu Li, are equal contributions. Y.H.L., D.L., and M.Y.L. designed and developed the method and analysed the data. S.R.W. and W.W. contributed to the implementation and data analysis. Y.H.L., M.Y.L., and X.Y.F. drafted the paper. F.G., X.M.L., and L.H. revised the paper. X.Y.F., and R.C.G. supervised the project and contributed to the conception of the project. All authors read and approved the final manuscript.

References

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NeurIPS*, 2013.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[Feng *et al.*, 2020] Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *EMNLP*, 2020.

[Févry *et al.*, 2020] Thibault Févry, Livio Baldini Soares, Nicholas Fitzgerald, Eunsol Choi, and Tom Kwiatkowski. Entities as experts: Sparse memory access with entity supervision. In *EMNLP*, 2020.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

[Huang *et al.*, 2019] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. Knowledge graph embedding based question answering. In *WSDM*, 2019.

[Jia *et al.*, 2021] Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. Complex temporal question answering on knowledge graphs. In *CIKM*, 2021.

[Jiao *et al.*, 2022] Songlin Jiao, Zhenfang Zhu, Wenqing Wu, Zicheng Zuo, Jiangtao Qi, Wenling Wang, Guangyuan Zhang, and Peiyu Liu. An improving reasoning network for complex question answering over temporal knowledge graphs. *Applied Intelligence*, pages 1–14, 2022.

[Kipf and Welling, 2017] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

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

[Liang *et al.*, 2019a] Di Liang, Fubao Zhang, Qi Zhang, and Xuan-Jing Huang. Asynchronous deep interaction network for natural language inference. In *EMNLP-IJCNLP*, 2019.

[Liang *et al.*, 2019b] Di Liang, Fubao Zhang, Weidong Zhang, Qi Zhang, Jinlan Fu, Minlong Peng, Tao Gui, and Xuanjing Huang. Adaptive multi-attention network incorporating answer information for duplicate question detection. In *SIGIR*, 2019.

[Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[Liu *et al.*, 2021a] Yonghao Liu, Renchu Guan, Xiaoyue Feng, and Ximing Li. Vpalg: Paper-publication prediction with graph neural networks. In *CIKM*, 2021.

[Liu *et al.*, 2021b] Yonghao Liu, Renchu Guan, Fausto Giunchiglia, Yanchun Liang, and Xiaoyue Feng. Deep attention diffusion graph neural networks for text classification. In *EMNLP*, 2021.

[Liu *et al.*, 2022] Yonghao Liu, Mengyu Li, Ximing Li, Fausto Giunchiglia, Xiaoyue Feng, and Renchu Guan. Few-shot node classification on attributed networks with graph meta-learning. In *SIGIR*, 2022.

[Liu *et al.*, 2023] Yonghao Liu, Di Liang, Fang Fang, Sirui Wang, Wei Wu, and Rui Jiang. Time-aware multiway adaptive fusion network for temporal kgqa. In *ICASSP*, 2023.

[Lukovnikov *et al.*, 2017] Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. Neural network-based question answering over knowledge graphs on word and character level. In *The Web Conference*, 2017.

[Mavromatis *et al.*, 2021] Costas Mavromatis, Prasanna Lakkur Subramanyam, Vassilis N Ioannidis, Soji Adeshina, Phillip R Howard, Tetiana Grinberg, Nagib Hakim, and George Karypis. Tempoqr: Temporal question reasoning over knowledge graphs. In *AAAI*, 2021.

[Peters *et al.*, 2019] Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. Knowledge enhanced contextual word representations. In *EMNLP-IJCNLP*, 2019.

[Ren *et al.*, 2020] Hongyu Ren, Weihua Hu, and Jure Leskovec. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. In *ICLR*, 2020.

[Saxena *et al.*, 2020] Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *ACL*, 2020.

[Saxena *et al.*, 2021] Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. Question answering over temporal knowledge graphs. In *ACL-IJCNLP*, 2021.

[Shang *et al.*, 2022] Chao Shang, Guangtao Wang, Peng Qi, and Jing Huang. Improving time sensitivity for question answering over temporal knowledge graphs. In *ACL*, 2022.

[Trouillon *et al.*, 2017] Théo Trouillon, Christopher R Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, and Guillaume Bouchard. Knowledge graph completion via complex tensor factorization. *JMLR*, 2017.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[Yasunaga *et al.*, 2021] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *NAACL*, 2021.

[Zhang *et al.*, 2018] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. Variational reasoning for question answering with knowledge graph. In *AAAI*, 2018.