ForecastTKGQuestions: A Benchmark for Temporal Question Answering and Forecasting over Temporal Knowledge Graphs

Zifeng Ding$^{\star 1,2}$, Zongyue Li$^{\star 1,3}$, Ruoxia Qi$^{\star 1}$, Jingpei Wu[1], Bailan He[1,2],
Yunpu Ma[1,2], Zhao Meng[4], Shuo Chen[1,2], Ruotong Liao[1,3],
Zhen Han(🖂)[1], and Volker Tresp(🖂)[1]

[1] LMU Munich, Geschwister-Scholl-Platz 1, 80539 Munich, Germany
[2] Siemens AG, Otto-Hahn-Ring 6, 81739 Munich, Germany
[3] Munich Center for Machine Learning (MCML), Munich, Germany
[4] ETH Zürich, Rämistrasse 101, 8092 Zürich, Switzerland
`{zifeng.ding, ruoxia.qi, bailan.he, shuo.chen}@campus.lmu.de`,
`{zongyue.li, jingpei.wu}@outlook.com`, `cognitive.yunpu@gmail.com`,
`zhmeng@ethz.ch`, `liao@dbs.ifi.lmu.de`, `hanzhen02111@hotmail.com`,
`Volker.Tresp@lmu.de`

Abstract. Question answering over temporal knowledge graphs (TKGQA) has recently found increasing interest. Previous related works aim to develop QA systems that answer temporal questions based on the facts from a fixed time period, where a temporal knowledge graph (TKG) spanning this period can be fully used for inference. In real-world scenarios, however, it is common that given knowledge until the current instance, we wish the TKGQA systems to answer the questions asking about future. As humans constantly plan the future, building forecasting TKGQA systems is important. In this paper, we propose a novel task: forecasting TKGQA, and propose a coupled large-scale TKGQA benchmark dataset, i.e., FORECASTTKGQUESTIONS. It includes three types of forecasting questions, i.e., entity prediction, yes-unknown, and fact reasoning questions. For every question, a timestamp is annotated and QA models only have access to TKG information prior to it for answer inference. We find that previous TKGQA methods perform poorly on forecasting questions, and they are unable to answer yes-unknown and fact reasoning questions. To this end, we propose FORECASTTKGQA, a TKGQA model that employs a TKG forecasting module for future inference. Experiments show that it performs well in forecasting TKGQA.

1 Introduction

Knowledge graphs (KGs) model factual information by representing every fact with a triple, i.e., (s, r, o), where s, o, r, are the subject entity, the object entity, and the relation between s and o, respectively. To adapt to the ever-evolving

---

* Equal contribution.

knowledge, temporal knowledge graphs (TKGs) are introduced, where they additionally specify the time validity of every fact with a time constraint t (e.g., a timestamp), and represent each fact with a quadruple (s, r, o, t). Recently, TKG reasoning has drawn increasing attention. While a lot of methods focus on temporal knowledge graph completion (TKGC) where they predict missing facts at the observed timestamps, various recent methods pay more attention to forecasting the facts at unobserved future timestamps in TKGs.

Knowledge graph question answering (KGQA) is a task aiming to answer natural language questions using a KG as the knowledge base (KB). KGQA requires QA models to extract answers from KGs, rather than retrieving or summarizing answers from text contexts. [21] first introduces question answering over temporal knowledge graphs (TKGQA). It proposes a non-forecasting TKGQA dataset CronQuestions that takes a TKG as its underlying KB. Temporal reasoning techniques are required to answer these questions. Though [21] manages to combine TKG reasoning with KGQA, it has limitations. Previous KGQA datasets, including CronQuestions, do not include yes-no and multiple-choice questions, while these two question types have been extensively studied in reading comprehension QA, e.g., [13]. Besides, the questions in CronQuestions are in a non-forecasting style, where all questions are based on the TKG facts that happen in a fixed time period, and an extensive TKG that is fully observable in this period can be used to infer the answers, making the answer inference less challenging. For example, the TKG facts from *2003*, including (*Stephen Robert Jordan*, *member of sports team*, *Manchester City*, *2003*), are all observable to answer the question *Which team was Stephen Robert Jordan part of in 2003?*. CronQuestions manages to bridge the gap between TKGC and KGQA, however, no previous work manages to combine TKG forecasting with KGQA, where only past TKG information can be used for answer inference.

In this work, we propose a novel task: forecasting question answering over temporal knowledge graphs (forecasting TKGQA), together with a coupled large-scale dataset, i.e., ForecastTKGQuestions. We generate forecasting questions based on the Integrated Crisis Early Warning System (ICEWS) Dataverse [2], and label every question with a timestamp. To answer a forecasting question, QA models can only access the TKG information prior to the question timestamp. The contribution of our work is three-folded: (1) We propose forecasting TKGQA, a novel task aiming to test the forecasting ability of TKGQA models. To the best of our knowledge, this is the first work binding TKG forecasting with temporal KGQA; (2) We propose a large-scale benchmark TKGQA dataset: ForecastTKGQuestions. It contains three types of questions, i.e., entity prediction questions (EPQs), yes-unknown questions (YUQs), and fact reasoning questions (FRQs), where the last two types of questions have never been considered in previous KGQA datasets[5]; (3) We propose ForecastTKGQA, a model aiming to solve forecasting TKGQA. It employs a TKG forecasting module and a pre-trained language model (LM) for answer inference. Experimental results show that it achieves great performance on forecasting questions.

---

[5] YUQs are based on yes-no questions and FRQs are multiple-choice questions.

2 Preliminaries and Related Work

TKG Reasoning Let $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{T}$ denote a finite set of entities, relations, and timestamps, respectively. A TKG $\mathcal{G}$ is defined as a finite set of TKG facts represented by quadruples, i.e., $\mathcal{G} = \{(s, r, o, t) | s, o \in \mathcal{E}, r \in \mathcal{R}, t \in \mathcal{T}\}$. We define the TKG forecasting task (also known as TKG extrapolation) as follows. Assume we have a query $(s_q, r_q, ?, t_q)$ (or $(?, r_q, o_q, t_q)$) derived from a target quadruple (s_q, r_q, o_q, t_q), and we denote all the ground-truth quadruples as $\mathcal{F}$. TKG forecasting aims to predict the missing entity in the query, given the observed **past** TKG facts $\mathcal{O} = \{(s_i, r_i, o_i, t_i) \in \mathcal{F} | t_i < t_q\}$. Such temporal restriction is not imposed in TKG completion (TKGC, also known as TKG interpolation), where the observed TKG facts from any timestamp, including t_q and the timestamps after t_q, can be used for prediction. In recent years, there have been extensive works done for both TKGC [16, 15, 6] and TKG forecasting [14, 9, 30, 8, 18]. We give a more detailed discussion about the forecasting methods. RE-NET [14] employs an autoregressive architecture and models fact occurrence as a probability distribution conditioned on the temporal sequences of past related TKG information. TANGO [9] employs neural ordinary differential equations to model temporal dependencies among graph information of different timestamps. CyGNet [30] uses the copy-generation mechanism to extract hints from historical facts for forecasting. xERTE [8] constructs a historical fact-based subgraph and selects prediction answers from it. TLogic [18] is the first rule-based TKG forecasting method that learns temporal logical rules in TKGs and achieves superior results.

Question Answering over KGs Several datasets have been proposed for QA over non-temporal KGs, such as SimpleQuestions [1], WebQuestionsSP [28], ComplexWebQuestions [24], MetaQA [29], TempQuestions [11], and TimeQuestions [12]. Among these datasets, only TempQuestions and TimeQuestions involve temporal questions that require temporal reasoning for answer inference, however, their associated KGs are non-temporal. CronQuestions [21] contains questions based on a time-evolving TKG, i.e., Wikidata [27]. It is proposed for non-forecasting TKGQA. Two types of questions, i.e., entity prediction and time prediction questions, are included. To answer CronQuestions, Saxena et al. propose CronKGQA that uses TKGC methods, along with pre-trained LMs, which shows great effectiveness. A line of methods has been proposed on top of CronKGQA (TempoQR [19], TSQA [23], SubGTR [4]), where they better distinguish question time scopes and reason over subgraphs. CronQuestions is proposed based on the idea of TKGC, and it does not support TKG forecasting and contains no forecasting questions. One recent work, i.e., ForecastQA [13], proposes a QA dataset fully consisting of forecasting questions. However, ForecastQA is not related to KGQA. In ForecastQA, answers to its questions are inferred from text contexts, while KGQA/TKGQA requires models to find the answers from the coupled KGs/TKGs without providing any additional text contexts. As a result, the methods designed for ForecastQA have no ability to address TKGQA. To this end, we propose ForecastTKGQuestions,

Table 1: (a) KGQA dataset comparison. Statistics are taken from [21] and [12]. **T%** denotes the portion of temporal questions. (b) ForecastTKGQuestions statistics: number of questions of different types.

(a)

Datasets	**TKG**	**Forecast**	**T%**	**# Questions**
MetaQA	✗	✗	0%	400k
TempQuestions	✗	✗	100%	1271
TimeQuestions	✗	✗	100%	16k
CronQuestions	✓	✗	100%	410k
ForecastTKGQuestions	✓	✓	100%	727k

(b)

	Train	**Valid**	**Test**
1-Hop Entity Prediction	211,564	36,172	33,447
2-Hop Entity Prediction	85,088	12,266	10,765
Yes-Unknown	251,537	42,884	39,695
Fact Reasoning	3,164	514	517
Total	551,353	91,836	84,424

aiming to bridge the gap between TKG forecasting and KGQA. We compare ForecastTKGQuestions with recent KGQA datasets in Table 1a.

Task Formulation: Forecasting TKGQA Forecasting TKGQA aims to test the forecasting ability of TKGQA models. It requires QA models to predict future facts based on past TKG information. We formulate it as follows. Given a TKG $\mathcal{G}$ and a natural language question q generated based on a TKG fact whose valid timestamp is t_q, forecasting TKGQA aims to predict the answer to q. We label every question q with t_q, and constrain QA models to only use the TKG facts $\{(s_i, r_i, o_i, t_i)|t_i < t_q\}$ before t_q for answer inference. We propose three types of forecasting TKGQA questions, i.e., EPQs, YUQs, and FRQs. The answer to a EPQ is an entity $e \in \mathcal{E}$. The answer to a YUQ is either *yes* or *unknown*. We formulate FRQs as multiple choices and thus the answer to an FRQ corresponds to a choice c. As a novel task, forecasting TKGQA requires models to have the ability of both natural language understanding (NLU) and future forecasting. Compared with it, the traditional TKG forecasting task does not require NLU and non-forecasting TKGQA does not consider future forecasting. Thus, previous methods for TKG forecasting[6], e.g., RE-Net [14], and non-forecasting TKGQA, e.g., TempoQR [19], are not suitable for solving forecasting TKGQA.

3 ForecastTKGQuestions

3.1 Temporal Knowledge Base

A subset from ICEWS [2] is taken as the associated temporal KB for our proposed dataset. We construct a TKG ICEWS21 based on the events taken from the official website of the ICEWS weekly event data[7] [2]. ICEWS contains sociopolitical events in English. We take the events from Jan. 1, 2021, to Aug. 31,

---

[6] Relation set is provided in TKG forecasting and these methods explicitly learn relation representations. However, TKG relations are not annotated in forecasting TKGQA questions. Only question texts are provided and these methods have no way to process. Therefore, we do not consider them in experiments on our new task.

[7] https://dataverse.harvard.edu/dataverse/icews

Table 2: ICEWS21 TKG statistics. N_{train}, N_{valid}, N_{test} denote the number of TKG facts in $\mathcal{G}_{\text{train}}$, $\mathcal{G}_{\text{valid}}$, $\mathcal{G}_{\text{test}}$, respectively. $|\mathcal{E}|$, $|\mathcal{R}|$, $|\mathcal{T}|$ denote ICEWS21's number of entities, relations, timestamps, respectively.

Dataset	N_{train}	N_{valid}	N_{test}	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{T}\rvert$
ICEWS21	252,434	43,033	39,836	20,575	253	243

2021, and extract TKG facts in the following way. For every ICEWS event, we generate a TKG fact (s, r, o, t). We take the content of *Event Date* as the timestamp t of the TKG fact. We take the contents of *Source Name* and *Target Name* as the subject entity s and the object entity o of the TKG fact, respectively. We take the content of *Event Text* as the relation type r of the fact. We present the dataset statistics of ICEWS21 in Table 2. We split ICEWS21 into three parts $\mathcal{G}_{\text{train}} = \{(s, r, o, t) \in \mathcal{G} | t \in [t_0, t_1)\}$, $\mathcal{G}_{\text{valid}} = \{(s, r, o, t) \in \mathcal{G} | t \in [t_1, t_2)\}$, $\mathcal{G}_{\text{test}} = \{(s, r, o, t) \in \mathcal{G} | t \in [t_2, t_3]\}$, where t_0, t_1, t_2, t_3 correspond to *2021-01-01*, *2021-07-01*, *2021-08-01* and *2021-08-31*, respectively. We generate training/validation/test questions based on $\mathcal{G}_{\text{train}}/\mathcal{G}_{\text{valid}}/\mathcal{G}_{\text{test}}$. We ensure that there exists no temporal overlap between every two of them, i.e., $\mathcal{G}_{\text{train}} \cap \mathcal{G}_{\text{valid}} = \emptyset$, $\mathcal{G}_{\text{train}} \cap \mathcal{G}_{\text{test}} = \emptyset$ and $\mathcal{G}_{\text{valid}} \cap \mathcal{G}_{\text{test}} = \emptyset$. In this way, we prevent QA models from observing any information from the evaluation sets during training.

3.2 Question Categorization and Generation

We generate natural language questions based on the TKG facts in ICEWS21 and propose our QA dataset ForecastTKGQuestions. Every relation type in ICEWS21 is coupled with a CAMEO code (specified in the *CAMEO Code* column of the ICEWS weekly event data). In the official CAMEO codebook (can be found in ICEWS database), each CAMEO code is explained with examples and detailed descriptions. We use the official CAMEO codebook provided in the ICEWS dataverse for aiding the generation of natural language relation templates. We create relation templates for 250 out of 253 relation types for question generation[8]. For example, we create a relation template *engage in material cooperation with* for the relation type *engage in material cooperation, not specified below.* Questions in ForecastTKGQuestions are categorized into three categories, i.e., EPQs (including 1-hop and 2 hop EPQs), YUQs, and FRQs. We summarize the number of different types of questions in Table 1b. We use the relation templates to create natural language question templates for all types of questions (examples in Table 3) which are used for question generation. All question templates are presented in our supplementary source code and explained in Appendix C.2. Similar to previous KGQA datasets, e.g., CronQuestions, entity linking is considered as a separate problem and is not covered in our work. We assume complete entity and timestamp linking, and annotate the entities and timestamps in our questions. This applies to all three types of questions in our dataset. Distribution of question timestamps is specified in Appendix C.5.

---

[8] The rest three relation types are not ideal for question generation (Appendix C.1).

Table 3: Example question templates of all types. s_q and o_q are the annotated question entities. t_q is the annotated question timestamp. For FRQ, s_c, o_c, t_c are annotated choice entities and timestamp. We only write one choice in FRQ template for brevity. Better understand with details in Section 3.2.

Question Type	**Example Template**
1-Hop EPQ	*Who will* $\{s_q\}$ *engage in material cooperation with on* $\{t_q\}$*?*
2-Hop EPQ	*Who will threaten a country, while* $\{s_q\}$ *criticizes or denounces this country on* $\{t_q\}$*?*
YUQ	*Will* $\{s_q\}$ *make a pessimistic comment about* $\{o_q\}$ *on* $\{t_q\}$*?*
FRQ	*Why will* $\{s_q\}$ *appeal to* $\{o_q\}$ *to meet or negociate on* $\{t_q\}$*?* *A:* $\{s_c\}$ *threatens* $\{o_c\}$ *on* $\{t_c\}$*; B:...*

Entity Prediction Questions We generate two groups of EPQs, i.e., 1-hop and 2-hop EPQs. Each 1-hop EPQ is generated from a single TKG fact, e.g., the natural language question *Who will Sudan host on 2021-08-01?* is based on (*Sudan*, *host*, *Ramtane Lamamra*, *2021-08-01*). Question templates are used during question generation. The underlined parts in the question denote the annotated entities and timestamps for KGQA. We consider all the facts concerning the 250 selected relations and transform them into 1-hop EPQs. Each 2-hop EPQ is generated from two associated TKG facts in ICEWS21 where they contain common entities. An example is presented in Table 4. The answer to a 2-hop EPQ (*Israel*) corresponds to a 2-hop neighbor of its annotated entity (*Iran*) at the question timestamp (*2021-08-02*). We generate 2-hop questions by utilizing AnyBURL [20], a rule-based KG reasoning model. We first split ICEWS21 into snapshots, where each snapshot $\mathcal{G}_{t_i} = \{(s, r, o, t) \in \mathcal{G} | t = t_i\}$ contains all the TKG facts happening at the same timestamp. Then we train AnyBURL on each snapshot for rule extraction. We collect the 2-hop rules with a confidence higher than 0.5 returned by AnyBURL, and manually check if two associated TKG facts in each rule potentially have a logical causation or can be used to interpret positive/negative entity relationships. After excluding the rules not meeting this requirement, we create question templates based on the remaining ones. We search for the groundings in ICEWS21 at every timestamp, where each grounding corresponds to a 2-hop EPQ. See our source code for the complete list of extracted 2-hop rules and see Appendix C.3 for more EPQ generation details.

Yes-Unknown Questions Based on the idea of triple classification in KG reasoning[9], we introduce yes-no questions into KGQA. We then turn yes-no questions into yes-unknown questions because, according to the Open World Assumption (OWA), the facts not observed in a given TKG are not necessarily wrong [7]. We generalize triple classification to quadruple classification[10], and then translate TKG facts into natural language questions. We take answering YUQs as solving quadruple classification. For every TKG fact concerning the

---

[9] For a KG fact (s, r, o), triple classification aims to predict whether this fact is valid or not.

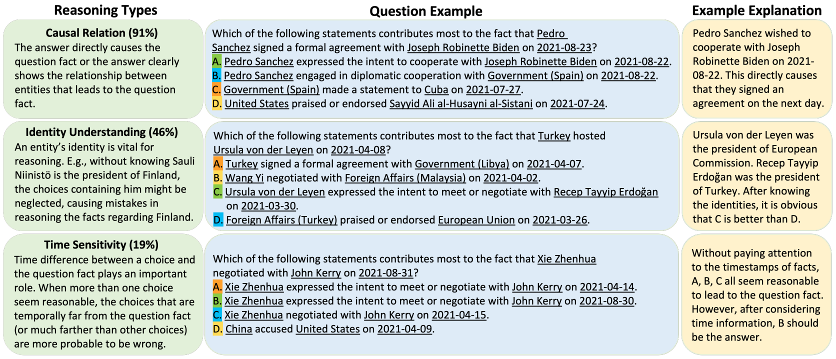
[10] Quadruple classification has never been studied in previous works. We define it as predicting whether a TKG fact (s, r, o, t) is valid or unknown, under OWA.

Table 4: 2-hop EPQ example. To avoid overlong text, we use symbols to represent relations and timestamps in TKG facts and 2-hop rules. r_1 =*accuse*; r_2 =*engage in diplomatic cooperation*; t_1 =*2021-08-02*. m, n are two entities that are 2-hop neighbors of each other at t_1. X is their common 1-hop neighbor at t_1. The extracted rule describes the negative relationship between *Iran* and *Israel*.

Associated TKG Facts	**2-Hop Rule**	**Generated 2-Hop Question**	**Answer**
(*United States*, r_1, *Iran*, t_1) (*United States*, r_2, *Israel*, t_1)	(X, r_1, m) => (X, r_2, n)	*Who will a country engage in diplomatic cooperation with, while this country accuses Iran on 2021-08-02?*	*Israel*

selected 250 relations, we generate either a true or an unknown question based on it. For example, for the fact (*Sudan*, *host*, *Ramtane Lamamra*, *2021-08-01*), a true question is generated as *Will Sudan host Ramtane Lamamra on 2021-08-01?* and we label *yes* as its answer. An unknown question is generated by randomly perturbing one entity or the relation type in this fact, e.g., *Will Germany host Ramtane Lamamra on 2021-08-01?*, and we label *unknown* as its answer. We ensure that the perturbed fact does not exist in the original TKG. We use 25% of total facts in ICEWS21 to generate true questions and the rest are used to generate unknown questions.

Fact Reasoning Questions The motivation for proposing FRQs is to study the difference between humans and machines in finding supporting evidence for reasoning. We formulate FRQs in the form of multiple choices. Each question is coupled with four choices. Given a TKG fact from an FRQ, we ask the QA models to choose which fact in the choices is the most contributive to (the most relevant cause of) the fact mentioned in the question. We provide several examples in Fig. 1. We generate FRQs as follows. We first train a TKG forecasting model xERTE [8] on ICEWS21. Note that to predict a query $(s, r, ?, t)$, xERTE samples its related prior TKG facts and assigns contribution scores to them. It provides explainability by assigning higher scores to the more related prior facts. We perform TKG forecasting and collect the queries where the ground-truth missing entities are ranked as top 1 by xERTE. For each collected query, we find its corresponding TKG fact and pick out four related prior facts found by xERTE. We take the prior facts with the highest, the lowest, and median contribution scores as **Answer**, **Negative**, and **Median**, respectively. Inspired by InferWiki [3], we include a **Hard Negative** fact with the second highest contribution score, making it non-trivial for QA models to make the right decision. We generate each FRQ by turning the corresponding facts into a question and four choices (using templates), and manage to use xERTE to generate a large number of questions. However, since the answers to these questions are solely determined by xERTE, there exist numerous erroneous examples. For example, the **Hard Negative** of lots of them are more suitable than their **Answer** to be the answers. We ask five graduate students (major in computer science) to manually check all these questions and annotate them as reasonable or unreasonable according to their own knowledge or through search engines. If the majority annotate a question as unreasonable, we filter it out. See Appendix C.4 for more details of FRQ generation and annotation, including the annotation instruction and interface.



Reasoning Types	Question Example	Example Explanation
Causal Relation (91%) The answer directly causes the question fact or the answer clearly shows the relationship between entities that leads to the question fact.	Which of the following statements contributes most to the fact that Pedro Sanchez signed a formal agreement with Joseph Robinette Biden on 2021-08-23? A. Pedro Sanchez expressed the intent to cooperate with Joseph Robinette Biden on 2021-08-22. B. Pedro Sanchez engaged in diplomatic cooperation with Government (Spain) on 2021-08-22. C. Government (Spain) made a statement to Cuba on 2021-07-27. D. United States praised or endorsed Sayyid Ali al-Husayni al-Sistani on 2021-07-24.	Pedro Sanchez wished to cooperate with Joseph Robinette Biden on 2021-08-22. This directly causes that they signed an agreement on the next day.
Identity Understanding (46%) An entity's identity is vital for reasoning. E.g., without knowing Sauli Niinistö is the president of Finland, the choices containing him might be neglected, causing mistakes in reasoning the facts regarding Finland.	Which of the following statements contributes most to the fact that Turkey hosted Ursula von der Leyen on 2021-04-08? A. Turkey signed a formal agreement with Government (Libya) on 2021-04-07. B. Wang Yi negotiated with Foreign Affairs (Malaysia) on 2021-04-02. C. Ursula von der Leyen expressed the intent to meet or negotiate with Recep Tayyip Erdoğan on 2021-03-30. D. Foreign Affairs (Turkey) praised or endorsed European Union on 2021-03-26.	Ursula von der Leyen was the president of European Commission. Recep Tayyip Erdoğan was the president of Turkey. After knowing the identities, it is obvious that C is better than D.
Time Sensitivity (19%) Time difference between a choice and the question fact plays an important role. When more than one choice seem reasonable, the choices that are temporally far from the question fact (or much farther than other choices) are more probable to be wrong.	Which of the following statements contributes most to the fact that Xie Zhenhua negotiated with John Kerry on 2021-08-31? A. Xie Zhenhua expressed the intent to meet or negotiate with John Kerry on 2021-04-14. B. Xie Zhenhua expressed the intent to meet or negotiate with John Kerry on 2021-08-30. C. Xie Zhenhua negotiated with John Kerry on 2021-04-15. D. China accused United States on 2021-04-09.	Without paying attention to the timestamps of facts, A, B, C all seem reasonable to lead to the question fact. However, after considering time information, B should be the answer.

Fig. 1: Required reasoning types and proportions (%) in sampled FRQs, as well as FRQ examples. We sample 100 FRQs in each train/valid/test set. For choices, green for **Answer**, blue for **Hard Negative**, orange for **Median** and yellow for **Negative**. Multiple reasoning skills are required to answer each question, so the total proportion sum is not 100%.

To better study the reasoning skills required to answer FRQs, we randomly sample 300 FRQs and manually annotate them with reasoning types. The required reasoning skills and their proportions are shown in Fig. 1.

4 ForecastTKGQA

ForecastTKGQA employs a TKG forecasting model TANGO [9] and a pre-trained LM BERT [5] for solving forecasting questions. We illustrate its model structure in Fig. 2 with three stages. In Stage 1, a TKG forecasting model TANGO [9] is used to generate the time-aware representation for each entity at each timestamp. In Stage 2, a pre-trained LM (e.g., BERT) is used to encode questions (and choices) into question (choice) representations. Finally, in Stage 3, answers are predicted according to the scores computed using the representations from Stage 1 and 2.

4.1 TKG Forecasting Model

We train TANGO on ICEWS21 with the TKG forecasting task. We use ComplEx [26] as its scoring function. We learn the entity and relation representations in the complex space $\mathbb{C}^d$, where d is the dimension of complex vectors. The training set corresponds to all the TKG facts in $\mathcal{G}_{\text{train}}$, and we evaluate the trained model on $\mathcal{G}_{\text{valid}}$ and $\mathcal{G}_{\text{test}}$. After training, we perform a one time inference on

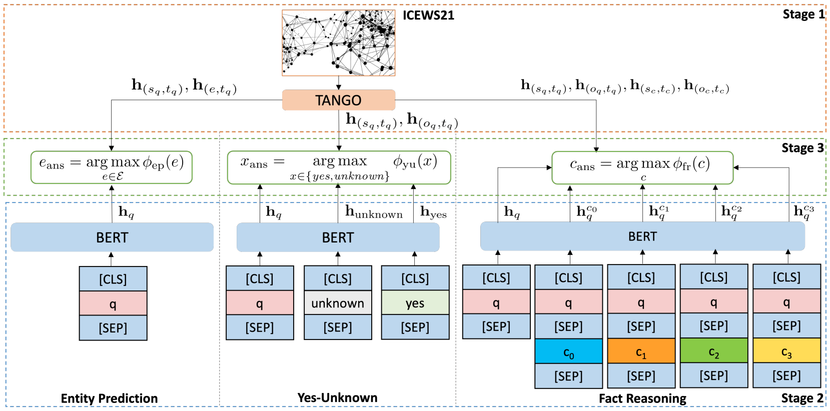


Fig. 2: Model structure of ForecastTKGQA.

$\mathcal{G}_{\text{valid}}$ and $\mathcal{G}_{\text{test}}$. Following the default setting of TANGO, to compute entity and relation representations at every timestamp t, we recurrently input all the TKG facts from $t-4$ to $t-1$, i.e., snapshots from $\mathcal{G}_{t-4}$ to $\mathcal{G}_{t-1}$, into TANGO and take the output representations. Note that it infers representations based on the prior facts, thus not violating our forecasting setting. We compute the entity and relation representations at every timestamp in ICEWS21 and keep them for aiding the QA systems in Stage 1 (Fig. 2). See Appendix B.1 for more details of TANGO training and inference. To leverage the complex representations computed by TANGO with ComplEx, we map the output of BERT to $\mathbb{C}^d$. For each natural language input, we take the output representation of the [CLS] token computed by BERT and project it to a $2d$ real space to form a $2d$ real-valued vector. We take the first and second half of it as the real and imaginary part of a d-dimensional complex vector, respectively. All the representations output by BERT have already been mapped to $\mathbb{C}^d$ without further notice.

4.2 QA Model

Entity Prediction For every EPQ q, we compute an entity score for every entity $e \in \mathcal{E}$. The entity with the highest score is predicted as the answer e_{ans}. To compute the score for e, we first input q into BERT and map its output to $\mathbb{C}^d$ to get the question representation $\mathbf{h}_q$. Inspired by ComplEx, we then define e's entity score as

$$\phi_{\text{ep}}(e) = \text{Re}\left(< \mathbf{h}'_{(s_q,t_q)}, \mathbf{h}_q, \bar{\mathbf{h}}'_{(e,t_q)} > \right). \tag{1}$$

$\mathbf{h}'_{(s_q,t_q)} = f_{\text{ep}}\left(\mathbf{h}_{(s_q,t_q)}\right)$, $\mathbf{h}'_{(e,t_q)} = f_{\text{ep}}\left(\mathbf{h}_{(e,t_q)}\right)$, where f_{ep} denotes a neural network aligning TKG representations to EPQs. $\mathbf{h}_{(s_q,t_q)}$ and $\mathbf{h}_{(e,t_q)}$ denote the TANGO representations of the annotated entity s_q and the entity e at the question timestamp t_q, respectively. Re means taking the real part of a complex vector and $\bar{\mathbf{h}}'_{(e,t_q)}$ means the complex conjugate of $\mathbf{h}'_{(e,t_q)}$.

Yes-Unknown Judgment For a YUQ, we compute a score for each candidate answer $x \in \{\textit{yes}, \textit{unknown}\}$. We first encode each x into a d-dimensional complex representation $\mathbf{h}_x$ with BERT. Inspired by TComplEx [16], we then compute scores as

$$\phi_{\text{yu}}(x) = \text{Re}\left(< \mathbf{h}'_{(s_q,t_q)}, \mathbf{h}_q, \bar{\mathbf{h}}'_{(o_q,t_q)}, \mathbf{h}_x >\right). \tag{2}$$

$\mathbf{h}'_{(s_q,t_q)} = f_{\text{yu}}\left(\mathbf{h}_{(s_q,t_q)}\right)$,$\mathbf{h}'_{(o_q,t_q)} = f_{\text{yu}}\left(\mathbf{h}_{(o_q,t_q)}\right)$, where f_{yu} denotes a neural network aligning TKG representations to YUQs. $\mathbf{h}_{(s_q,t_q)}$ and $\mathbf{h}_{(o_q,t_q)}$ denote the TANGO representations of the annotated subject entity s_q and object entity o_q at t_q, respectively. $\mathbf{h}_q$ is the BERT encoded question representation. We take the candidate answer with the higher score as the predicted answer x_{ans}.

Fact Reasoning We compute a choice score for every choice c in an FRQ by using the following scoring function:

$$\phi_{\text{fr}}(c) = \text{Re}\left(< \mathbf{h}'_{(s_c,t_c)}, \mathbf{h}^c_q, \bar{\mathbf{h}}'_{(o_c,t_c)}, \mathbf{h}'_q >\right), \tag{3}$$

$\mathbf{h}^c_q$ is the output of BERT mapped to $\mathbb{C}^d$ given the concatenation of q and c. $\mathbf{h}'_{(s_c,t_c)} = f_{\text{fr}}\left(\mathbf{h}_{(s_c,t_c)}\right)$ and $\mathbf{h}'_{(o_c,t_c)} = f_{\text{fr}}\left(\mathbf{h}_{(o_c,t_c)}\right)$. f_{fr} is a projection network and $\mathbf{h}_{(s_c,t_c)}$, $\mathbf{h}_{(o_c,t_c)}$ denote the TANGO representations of the entities annotated in c. $\mathbf{h}'_q = f\left(f_{\text{fr}}\left(\mathbf{h}_{(s_q,t_q)}\right) \| \mathbf{h}^c_q \| f_{\text{fr}}\left(\mathbf{h}_{(o_q,t_q)}\right)\right)$, where f serves as a projection and $\|$ denotes concatenation. $\mathbf{h}_{(s_q,t_q)}$ and $\mathbf{h}_{(o_q,t_q)}$ denote the TANGO representations of the entities annotated in the question q. We take the choice with the highest choice score as our predicted answer c_{ans}. We give a more detailed description of Equation 1, 2 and 3 in Appendix A.

Parameter Learning We use cross-entropy loss to train ForecastTKGQA on each type of questions separately. The loss functions of EPQs, FRQs and YUQs are given by $\mathcal{L}_{\text{ep}} = -\sum_{q \in \mathcal{Q}^{\text{ep}}} \log\left(\frac{\phi_{\text{ep}}(e_{\text{ans}})}{\sum_{e \in \mathcal{E}} \phi_{\text{ep}}(e)}\right)$, $\mathcal{L}_{\text{fr}} = -\sum_{q \in \mathcal{Q}^{\text{fr}}} \log\left(\frac{\phi_{\text{fr}}(c_{\text{ans}})}{\sum_c \phi_{\text{fr}}(c)}\right)$ and $\mathcal{L}_{\text{yu}} = -\sum_{q \in \mathcal{Q}^{\text{yu}}} \log\left(\frac{\phi_{\text{yu}}(x_{\text{ans}})}{\sum_{x \in \{\textit{yes},\textit{unknown}\}} \phi_{\text{yu}}(x)}\right)$, respectively. $\mathcal{Q}^{\text{ep}}/\mathcal{Q}^{\text{yu}}/\mathcal{Q}^{\text{fr}}$ denotes all EPQs/YUQs/FRQs and $e_{\text{ans}}/x_{\text{ans}}/c_{\text{ans}}$ is the answer to question q.

5 Experiments

We answer several research questions (RQs) with experiments[11]. **RQ1** (Section 5.2, 5.4): Can a TKG forecasting model better support forecasting TKGQA than a TKGC model? **RQ2** (Section 5.2, 5.4): Does ForecastTKGQA perform well in forecasting TKGQA? **RQ3** (Section 5.3, 5.5): Are the questions in our dataset answerable? **RQ4** (Section 5.7): Is the proposed dataset efficient? **RQ5** (Section 5.6): What are the challenges of forecasting TKGQA?

---

[11] Implementation details and further analysis of ForecastTKGQA in Appendix B.3 and G.

5.1 Experimental Setting

Evaluation Metrics We use mean reciprocal rank (MRR) and Hits@k as the evaluation metrics of the EPQs. For each EPQ, we compute the rank of the ground-truth answer entity among all the TKG entities. Test MRR is then computed as $\frac{1}{|\mathcal{Q}_{\text{test}}^{\text{ep}}|}\sum_{q\in\mathcal{Q}_{\text{test}}^{\text{ep}}}\frac{1}{\text{rank}_q}$, where $\mathcal{Q}_{\text{test}}^{\text{ep}}$ denotes all EPQs in the test set and rank_q is the rank of the ground-truth answer entity of question q. Hits@k is the proportion of the answered questions where the ground-truth answer entity is ranked as top k. For YUQs and FRQs, we employ accuracy for evaluation. Accuracy is the proportion of the correctly answered questions out of all questions.

Baseline Methods We consider two pre-trained LMs, BERT [5] and RoBERTa [17] as baselines. For EPQs and YUQs, we add a prediction head on top of the question representations computed by LMs, and use softmax function to compute answer probabilities. For every FRQ, we input into each LM the concatenation of the question with each choice, and follow the same prediction structure. Besides, we derive two model variants for each LM by introducing TKG representations. We train TComplEx on ICEWS21. For every EPQ and YUQ, we concatenate the question representation with the TComplEx representations of the entities and timestamps annotated in the question, and then perform prediction with a prediction head and softmax. For FRQs, we further include TComplEx representations into choices in the same way. We call this type of variant BERT_int and RoBERTa_int since TComplEx is a TKGC (TKG interpolation) method. Similarly, we also introduce TANGO representations into LMs and derive BERT_ext and RoBERTa_ext, where TANGO serves as a TKG extrapolation backend. Detailed model derivations are presented in Appendix B.2. We also consider one KGQA method EmbedKGQA [22], and two TKGQA methods, i.e., CronKGQA [21] and TempoQR [19] as baselines. We run EmbedKGQA on top of the KG representations trained with ComplEx on ICEWS21, and run TKGQA baselines on top of the TKG representations trained with TComplEx.

5.2 Main Results

We report the experimental results in Table 5. In Table 5a, we show that our entity prediction model outperforms all baseline methods. We observe that EmbedKGQA achieves a better performance than BERT and RoBERTa, showing that employing KG representations helps TKGQA. Besides, LM variants outperform their original LMs, indicating that TKG representations help LMs perform better in TKGQA. Further, BERT_ext shows stronger performance than BERT_int (this also applies to RoBERTa_int and RoBERTa_ext), which proves that TKG forecasting models provide greater help than TKGC models in forecasting TKGQA. CronKGQA and TempoQR employ TComplEx representations as supporting information and perform poorly, implying that employing TKG representations provided by TKGC methods may include noisy information in forecasting TKGQA. ForecastTKGQA injects TANGO representations

Table 5: Experimental results over ForecastTKGQuestions. The best results are marked in bold.

(a) EPQs. Overall results in Appendix D.

	MRR		Hits@1		Hits@10	
Model	1-Hop	2-Hop	1-Hop	2-Hop	1-Hop	2-Hop
RoBERTa	0.166	0.149	0.104	0.085	0.288	0.268
BERT	0.279	0.182	0.192	0.106	0.451	0.342
EmbedKGQA	0.317	0.185	0.228	0.112	0.489	0.333
RoBERTa_int	0.283	0.157	0.190	0.094	0.467	0.290
BERT_int	0.314	0.183	0.223	0.107	0.490	0.344
CronKGQA	0.131	0.090	0.081	0.042	0.231	0.187
TempoQR	0.145	0.107	0.094	0.061	0.243	0.199
RoBERTa_ext	0.306	0.180	0.216	0.108	0.497	0.323
BERT_ext	0.331	0.208	0.239	0.128	0.508	0.369
ForecastTKGQA	**0.339**	**0.216**	**0.248**	**0.129**	**0.517**	**0.386**

(b) YUQs and FRQs.

	Accuracy	
Model	YUQ	FRQ
RoBERTa	0.721	0.645
BERT	0.813	0.634
RoBERTa_int	0.768	0.693
BERT_int	0.829	0.682
RoBERTa_ext	0.798	0.707
BERT_ext	0.837	0.746
ForecastTKGQA	**0.870**	**0.769**
Human Performance (a)	-	0.936
Human Performance (b)	-	0.954

into a scoring module, showing its great effectiveness on EPQs. For YUQs and FRQs, ForecastTKGQA also achieves the best performance. Table 5b shows that it is helpful to include TKG representations for answering YUQs and FRQs and our scoring functions are effective.

5.3 Human vs. Machine on FRQs

To study the difference between humans and models in fact reasoning, we further benchmark human performance on FRQs with a survey (See Appendix E for details). We ask five graduate students to answer 100 questions randomly sampled from the test set. We consider two settings: (a) Humans answer FRQs with their own knowledge and inference ability. **Search engines are not allowed**; (b) Humans can turn to search engines and use the web information published **before the question timestamp** for aiding QA. Table 5 shows that humans achieve much stronger performance than all QA models (even in setting (a)). This calls for a great effort to build better fact reasoning TKGQA models.

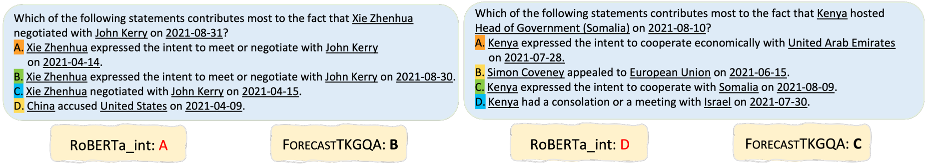
5.4 Performance over FRQs with Different Reasoning Types

Considering the reasoning types listed in Fig. 1, we compare RoBERTa_int with ForecastTKGQA on the 100 sampled test questions that are annotated with reasoning types, to justify performance gain brought by TKG forecasting model on FRQs. Experimental results in Table 6 imply that employing TKG forecasting model helps QA models better deal with any reasoning type on FRQs. We use two cases in Fig. 3 to provide insights into performance gain.

Case 1. Two reasoning skills, i.e., Causal Relation and Time Sensitivity (shown in Fig. 1), are required to correctly answer the question in Case 1. Without considering the timestamps of choices, A, B, C all seem at least somehow reasonable.

Table 6: Performance comparison across FRQs with different reasoning types.

	Accuracy		
Model	Causal Relation	Identity Understanding	Time Sensitivity
RoBERTa_int	0.670	0.529	0.444
ForecastTKGQA	**0.787**	**0.735**	**0.611**



Which of the following statements contributes most to the fact that Xie Zhenhua negotiated with John Kerry on 2021-08-31?
A. Xie Zhenhua expressed the intent to meet or negotiate with John Kerry on 2021-04-14.
B. Xie Zhenhua expressed the intent to meet or negotiate with John Kerry on 2021-08-30.
C. Xie Zhenhua negotiated with John Kerry on 2021-04-15.
D. China accused United States on 2021-04-09.

(a) Case 1.

Which of the following statements contributes most to the fact that Kenya hosted Head of Government (Somalia) on 2021-08-10?
A. Kenya expressed the intent to cooperate economically with United Arab Emirates on 2021-07-28.
B. Simon Coveney appealed to European Union on 2021-06-15.
C. Kenya expressed the intent to cooperate with Somalia on 2021-08-09.
D. Kenya had a consolation or a meeting with Israel on 2021-07-30.

(b) Case 2.

Fig. 3: Case Studies on FRQs. We mark green for **Answer**, blue for **Hard Negative**, orange for **Median** and yellow for **Negative**.

However, after considering choice timestamps, B should be the most contributive reason for the question fact. First, the timestamp of B (*2021-08-30*) is much closer to the question timestamp (*2021-08-31*). Moreover, the fact in choice B directly causes the question fact. RoBERTa_int manages to capture the causation, but fails to correctly deal with time sensitivity, while ForecastTKGQA achieves better reasoning on both reasoning types.

Case 2. Two reasoning skills, i.e., Causal Relation and Identity Understanding (shown in Fig. 1), are required to correctly answer the question in Case 2. *Head of Government (Somalia)* and *Somalia* are two different entities in TKG, however, both entities are about Somalia. By understanding this, we are able to choose the correct answer. ForecastTKGQA manages to understand the identity of *Head of Government (Somalia)*, match it with *Somalia* and find the cause of the question fact. RoBERTa_int makes a mistake because as a model equipped with TComplEx, it has no well-trained timestamp representations of the question and choice timestamps, which would introduce noise in decision making.

5.5 Answerability of ForecastTKGQuestions

To validate the answerability of the questions in ForecastTKGQuestions. We train TComplEx and TANGO over the whole ICEWS21, i.e., $\mathcal{G}_{\text{train}} \cup \mathcal{G}_{\text{valid}} \cup \mathcal{G}_{\text{test}}$, and use them to support QA. Note that this violates the forecasting setting of forecasting TKGQA, and thus we call the TKG models trained in this way as cheating TComplEx (CTComplEx) and cheating TANGO (CTANGO). Answering EPQs with cheating TKG models is same as non-forecasting TKGQA. We couple TempoQR with CTComplEx and see a huge performance increase (Table 7a). Besides, inspired by [10], we develop a new TKGQA model Multi-Hop

Table 7: Answerability study. Models with α means using CTComplEx and β means using CTANGO. ↑ denotes relative improvement (%) from the results in Table 5. Acc means Accuracy.

(a) EPQs.

	MRR				Hits@10			
Model	1-Hop	↑	2-Hop	↑	1-Hop	↑	2-Hop	↑
TempoQR$^{\alpha}$	0.713	391.7	0.233	117.8	0.883	263.4	0.419	110.6
MHS$^{\alpha}$	0.868	-	0.647	-	0.992	-	0.904	-
MHS$^{\beta}$	0.771	-	0.556	-	0.961	-	0.828	-

(b) YUQs and FRQs.

	YUQ		FRQ	
Model	Acc	↑	Acc	↑
BERT_int$^{\alpha}$	0.855	19.6	0.816	14.4
BERT_ext$^{\beta}$	0.873	4.3	0.836	12.1
ForecastTKGQA$^{\beta}$	0.925	6.3	0.821	6.8

Scorer[12] (MHS) for EPQs. Starting from the annotated entity s_q of an EPQ, MHS updates the scores of outer entities for n-hops ($n = 2$ in our experiments) until all s_q's n-hop neighbors on the snapshot $\mathcal{G}_{t_q}$ are visited. Initially, MHS assigns a score of 1 to s_q and 0 to any other unvisited entity. For each unvisited entity e, it then computes e's score as: $\phi_{\text{ep}}(e) = \frac{1}{|\mathcal{N}_e(t_q)|} \sum_{(e',r)\in\mathcal{N}_e(t_q)}(\gamma \cdot \phi_{\text{ep}}(e') + \psi(e', r, e, t_q))$, where $\mathcal{N}_e(t_q) = \{(e', r)|(e', r, e, t_q) \in \mathcal{G}_{t_q}\}$ is e's 1-hop neighborhood on $\mathcal{G}_{t_q}$ and γ is a discount factor. We couple MHS with CTComplEx and CTANGO, and define $\psi(e', r, e, t_q)$ separately. For MHS + CTComplEx, $\psi(e', r, e, t_q) = f_2(f_1(\mathbf{h}_{e'}\|\mathbf{h}_r\|\mathbf{h}_e\|\mathbf{h}_{t_q}\|\mathbf{h}_q))$. f_1 and f_2 are two neural networks. $\mathbf{h}_e, \mathbf{h}_{e'}, \mathbf{h}_r, \mathbf{h}_{t_q}$ are the CTComplEx representations of entities e, e', relation r and timestamp t_q, respectively. For MHS + CTANGO, we take the idea of ForecastTKGQA: $\psi(e', r, e, t_q) = \text{Re}\left(< \mathbf{h}_{(e',t_q)}, \mathbf{h}_r, \bar{\mathbf{h}}_{(e,t_q)}, \mathbf{h}_q >\right)$. $\mathbf{h}_{(e,t_q)}$, $\mathbf{h}_{(e',t_q)}$, $\mathbf{h}_r$ are the CTANGO representations of entities e, e' at t_q, and relation r, respectively. $\mathbf{h}_q$ is BERT encoded question representation. We find that MHS achieves superior performance (even on 2-hop EPQs). This is because MHS not only uses cheating TKG models, but also considers ground-truth multi-hop structural information of TKGs at t_q (which is unavailable in the forecasting setting). For YUQs and FRQs, Table 7b shows that cheating TKG models help improve performance, especially on FRQs. These results imply that given the ground-truth TKG information at question timestamps, our forecasting TKGQA questions are answerable.

5.6 Challenges of Forecasting TKGQA over ForecastTKGQuestions

From the experiments discussed in Section 5.3 and 5.5, we summarize the challenges of forecasting TKGQA: (1) Inferring the ground-truth TKG information $\mathcal{G}_{t_q}$ at the question timestamp t_q accurately; (2) Effectively performing multi-hop reasoning for forecasting TKGQA; (3) Developing TKGQA models for better fact reasoning. In Section 5.5, we have trained cheating TKG models and used them to support QA. We show in Table 7 that QA models substantially improve

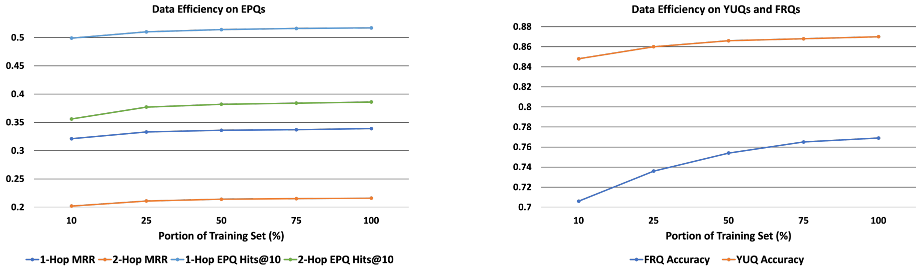
---

[12] See Appendix F for detailed model explanation and model structure illustration.

their performance on forecasting TKGQA with cheating TKG models. This implies that accurately inferring the ground-truth TKG information at t_q is crucial in our task and how to optimally achieve it remains a challenge. We also observe that MHS with cheating TKG models achieves much better results on EPQs (especially on 2-hop). MHS utilizes multi-hop information of the ground-truth TKG at t_q ($\mathcal{G}_{t_q}$) for better QA. In forecasting TKGQA, by only knowing the TKG facts before t_q and not observing $\mathcal{G}_{t_q}$, it is impossible for MHS to directly utilize the ground-truth multi-hop information at t_q. This implies that how to effectively infer and exploit multi-hop information for QA in the forecasting scenario remains a challenge. Moreover, as discussed in Section 5.3, current TKGQA models still trail humans with great margin on FRQs. It is challenging to design novel forecasting TKGQA models for better fact reasoning.

5.7 Study of Data Efficiency

We want to know how the models will be affected with less/more training data. For each type of questions, we modify the size of its training set. We train ForecastTKGQA on the modified training sets and evaluate our model on the original test sets. We randomly sample 10%, 25%, 50%, and 75% of the training examples to form new training sets. Fig. 4 shows that for every type of question, the performance of ForecastTKGQA steadily improves as the size of the training sets increase. This proves that our proposed dataset is efficient and useful for training forecasting TKGQA models.



(a) Data efficiency on EPQs. (b) Data efficiency on YUQs, FRQs.

Fig. 4: Data efficiency analysis.

6 Justification of Task Validity from Two Perpectives

(1) Perspective from Underlying TKG. We take a commonly used temporal KB, i.e., ICEWS, as the KB for constructing underlying TKG ICEWS21. ICEWS-based TKGs contain socio-political facts. It is meaningful to perform forecasting over them because this can help to improve early warning in critical socio-political situations around the globe. [25] has shown with case studies that ICEWS-based TKG datasets have underlying cause-and-effect temporal patterns and TKG forecasting models are built to capture them. This indicates that performing TKG forecasting over ICEWS-based TKGs is also valid. And therefore, developing forecasting TKGQA on top of ICEWS21 is meaningful and valid. **(2) Perspective from the Motivation of Proposing Different Types of Questions.** The motivation of proposing EPQs is to introduce TKG link forecasting (future link prediction) into KGQA, while proposing YUQs is to introduce quadruple classification (stemming from triple classification) and yes-no type questions. We view quadruple classification in the forecasting scenario as deciding if the unseen TKG facts are valid based on previously known TKG facts. To answer EPQs and YUQs, models can be considered as understanding natural language questions first and then performing TKG reasoning tasks. Since TKG reasoning tasks are considered solvable and widely studied in the TKG community, our task over EPQs and YUQs is valid. We propose FRQs aiming to study the difference between humans and machines in fact reasoning. We have summarized the reasoning skills that are required to answer every FRQ in Fig. 1, which also implies the potential direction for QA models to achieve improvement in fact reasoning in the future. We have shown in Section 5.3 that our proposed FRQs are answerable to humans, which directly indicates the validity of our FRQs. Thus, answering FRQs in forecasting TKGQA is also valid and meaningful.

7 Conclusion

In this work, we propose a novel task: forecasting TKGQA. To the best of our knowledge, it is the first work combining TKG forecasting with KGQA. We propose a coupled benchmark dataset ForecastTKGQuestions that contains various types of questions including EPQs, YUQs and FRQs. To solve forecasting TKGQA, we propose ForecastTKGQA, a QA model that leverages a TKG forecasting model with a pre-trained LM. Though experimental results show that our model achieves great performance, there still exists a large room for improvement compared with humans. We hope our work can benefit future research and draw attention to studying the forecasting power of TKGQA methods.

Acknowledgement. This work has been supported by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) as part of the project CoyPu under grant number 01MK21007K.

Supplemental Material Statement: Source code and data are uploaded here[13]. Appendices are published in the arXiv version[14]. We have referred to the corresponding parts in the main body. Please check accordingly.

---

[13] https://github.com/ZifengDing/ForecastTKGQA

[14] https://arxiv.org/abs/2208.06501

References

1. Bordes, A., Usunier, N., Chopra, S., Weston, J.: Large-scale simple question answering with memory networks (2015). https://doi.org/10.48550/ARXIV.1506.02075, https://arxiv.org/abs/1506.02075
2. Boschee, E., Lautenschlager, J., O'Brien, S., Shellman, S., Starz, J., Ward, M.: ICEWS Coded Event Data (2015). https://doi.org/10.7910/DVN/28075, https://doi.org/10.7910/DVN/28075
3. Cao, Y., Ji, X., Lv, X., Li, J., Wen, Y., Zhang, H.: Are missing links predictable? an inferential benchmark for knowledge graph completion. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) 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. pp. 6855–6865. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.acl-long.534, https://doi.org/10.18653/v1/2021.acl-long.534
4. Chen, Z., Zhao, X., Liao, J., Li, X., Kanoulas, E.: Temporal knowledge graph question answering via subgraph reasoning. Knowl. Based Syst. **251**, 109134 (2022). https://doi.org/10.1016/j.knosys.2022.109134, https://doi.org/10.1016/j.knosys.2022.109134
5. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) 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). pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/n19-1423, https://doi.org/10.18653/v1/n19-1423
6. Ding, Z., Ma, Y., He, B., Han, Z., Tresp, V.: A simple but powerful graph encoder for temporal knowledge graph completion. In: NeurIPS 2022 Temporal Graph Learning Workshop (2022), https://openreview.net/forum?id=DYG8RbgAIo
7. Galárraga, L.A., Teflioudi, C., Hose, K., Suchanek, F.M.: AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In: Schwabe, D., Almeida, V.A.F., Glaser, H., Baeza-Yates, R., Moon, S.B. (eds.) 22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013. pp. 413–422. International World Wide Web Conferences Steering Committee / ACM (2013). https://doi.org/10.1145/2488388.2488425, https://doi.org/10.1145/2488388.2488425
8. Han, Z., Chen, P., Ma, Y., Tresp, V.: Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021), https://openreview.net/forum?id=pGIHq1m7PU
9. Han, Z., Ding, Z., Ma, Y., Gu, Y., Tresp, V.: Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In: Moens, M., Huang, X., Specia, L., Yih, S.W. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. pp. 8352–8364. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.emnlp-main.658, https://doi.org/10.18653/v1/2021.emnlp-main.658
10. Ji, H., Ke, P., Huang, S., Wei, F., Zhu, X., Huang, M.: Language generation with multi-hop reasoning on commonsense knowledge graph. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference

on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020. pp. 725–736. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.emnlp-main.54, https://doi.org/10.18653/v1/2020.emnlp-main.54

11. Jia, Z., Abujabal, A., Roy, R.S., Strötgen, J., Weikum, G.: Tempquestions: A benchmark for temporal question answering. In: Champin, P., Gandon, F., Lalmas, M., Ipeirotis, P.G. (eds.) Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018. pp. 1057–1062. ACM (2018). https://doi.org/10.1145/3184558.3191536, https://doi.org/10.1145/3184558.3191536
12. Jia, Z., Pramanik, S., Roy, R.S., Weikum, G.: Complex temporal question answering on knowledge graphs. In: Demartini, G., Zuccon, G., Culpepper, J.S., Huang, Z., Tong, H. (eds.) CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021. pp. 792–802. ACM (2021). https://doi.org/10.1145/3459637.3482416, https://doi.org/10.1145/3459637.3482416
13. Jin, W., Khanna, R., Kim, S., Lee, D., Morstatter, F., Galstyan, A., Ren, X.: Forecastqa: A question answering challenge for event forecasting with temporal text data. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) 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. pp. 4636–4650. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.acl-long.357, https://doi.org/10.18653/v1/2021.acl-long.357
14. Jin, W., Qu, M., Jin, X., Ren, X.: Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020. pp. 6669–6683. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.emnlp-main.541, https://doi.org/10.18653/v1/2020.emnlp-main.541
15. Jung, J., Jung, J., Kang, U.: Learning to walk across time for interpretable temporal knowledge graph completion. In: Zhu, F., Ooi, B.C., Miao, C. (eds.) KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021. pp. 786–795. ACM (2021). https://doi.org/10.1145/3447548.3467292, https://doi.org/10.1145/3447548.3467292
16. Lacroix, T., Obozinski, G., Usunier, N.: 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 (2020), https://openreview.net/forum?id=rke2P1BFwS
17. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach (2019). https://doi.org/10.48550/ARXIV.1907.11692, https://arxiv.org/abs/1907.11692
18. Liu, Y., Ma, Y., Hildebrandt, M., Joblin, M., Tresp, V.: Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twel-

veth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022. pp. 4120–4127. AAAI Press (2022), https://ojs.aaai.org/index.php/AAAI/article/view/20330

19. Mavromatis, C., Subramanyam, P.L., Ioannidis, V.N., Adeshina, A., Howard, P.R., Grinberg, T., Hakim, N., Karypis, G.: Tempoqr: Temporal question reasoning over knowledge graphs. 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. pp. 5825–5833. AAAI Press (2022), https://ojs.aaai.org/index.php/AAAI/article/view/20526
20. Meilicke, C., Chekol, M.W., Fink, M., Stuckenschmidt, H.: Reinforced anytime bottom up rule learning for knowledge graph completion (2020). https://doi.org/10.48550/ARXIV.2004.04412, https://arxiv.org/abs/2004.04412
21. Saxena, A., Chakrabarti, S., Talukdar, P.P.: Question answering over temporal knowledge graphs. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) 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. pp. 6663–6676. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.acl-long.520, https://doi.org/10.18653/v1/2021.acl-long.520
22. Saxena, A., Tripathi, A., Talukdar, P.: 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. pp. 4498–4507. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.412, https://aclanthology.org/2020.acl-main.412
23. Shang, C., Wang, G., Qi, P., Huang, J.: Improving time sensitivity for question answering over temporal knowledge graphs. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022. pp. 8017–8026. Association for Computational Linguistics (2022), https://aclanthology.org/2022.acl-long.552
24. Talmor, A., Berant, J.: The web as a knowledge-base for answering complex questions. In: Walker, M.A., Ji, H., Stent, A. (eds.) Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers). pp. 641–651. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/n18-1059, https://doi.org/10.18653/v1/n18-1059
25. Trivedi, R., Dai, H., Wang, Y., Song, L.: Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. Proceedings of Machine Learning Research, vol. 70, pp. 3462–3471. PMLR (2017), http://proceedings.mlr.press/v70/trivedi17a.html
26. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: Balcan, M., Weinberger, K.Q. (eds.) Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016. JMLR Workshop and Conference Proceed-

ings, vol. 48, pp. 2071–2080. JMLR.org (2016), http://proceedings.mlr.press/v48/trouillon16.html

27. Vrandecic, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014). https://doi.org/10.1145/2629489, https://doi.org/10.1145/2629489
28. Yih, W., Chang, M., He, X., Gao, J.: Semantic parsing via staged query graph generation: Question answering with knowledge base. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers. pp. 1321–1331. The Association for Computer Linguistics (2015). https://doi.org/10.3115/v1/p15-1128, https://doi.org/10.3115/v1/p15-1128
29. Zhang, Y., Dai, H., Kozareva, Z., Smola, A.J., Song, L.: Variational reasoning for question answering with knowledge graph. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. pp. 6069–6076. AAAI Press (2018), https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16983
30. Zhu, C., Chen, M., Fan, C., Cheng, G., Zhang, Y.: Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021. pp. 4732–4740. AAAI Press (2021), https://ojs.aaai.org/index.php/AAAI/article/view/16604