Question Answering Over Temporal Knowledge Graphs

Apoorv Saxena
Indian Institute of Science
Bangalore
apoorvsaxena@iisc.ac.in

Soumen Chakrabarti
Indian Institute of Technology
Bombay
soumen@cse.iitb.ac.in

Partha Talukdar
Google Research
India
partha@google.com

Abstract

Temporal Knowledge Graphs (Temporal KGs) extend regular Knowledge Graphs by providing temporal scopes (e.g., start and end times) on each edge in the KG. While Question Answering over KG (KGQA) has received some attention from the research community, QA over Temporal KGs (Temporal KGQA) is a relatively unexplored area. Lack of broad-coverage datasets has been another factor limiting progress in this area. We address this challenge by presenting CRONQUESTIONS, the largest known Temporal KGQA dataset, clearly stratified into buckets of structural complexity. CRONQUESTIONS expands the only known previous dataset by a factor of $340\times$. We find that various state-of-the-art KGQA methods fall far short of the desired performance on this new dataset. In response, we also propose CRONKGQA, a transformer-based solution that exploits recent advances in Temporal KG embeddings, and achieves performance superior to all baselines, with an increase of 120% in accuracy over the next best performing method. Through extensive experiments, we give detailed insights into the workings of CRONKGQA, as well as situations where significant further improvements appear possible. In addition to the dataset, we have released our code as well.

1 Introduction

Temporal Knowledge Graphs (Temporal KGs) are multi-relational graph where each edge is associated with a time duration. This is in contrast to a regular KG where no time annotation is present. For example, a regular KG may contain a fact such as (*Barack Obama*, *held position*, *President of USA*), while a temporal KG would contain the start and end time as well — (*Barack Obama*, *held position*, *President of USA*, *2008*, *2016*). Edges may be associated with a set of non-contiguous

time intervals as well. These temporal scopes on facts can be either automatically estimated (Taluk-dar et al., 2012) or user contributed. Several such Temporal KGs have been proposed in the literature, where the focus is on KG completion (Dasgupta et al. 2018; García-Durán et al. 2018; Leetaru and Schrodt 2013; Lacroix et al. 2020; Jain et al. 2020).

The task of Knowledge Graph Question Answering (KGQA) is to answer natural language questions using a KG as the knowledge base. This is in contrast to reading comprehension-based question answering, where typically the question is accompanied by a context (e.g., text passage) and the answer is either one of multiple choices (Rajpurkar et al., 2016) or a piece of text from the context (Yang et al., 2018). In KGQA, the answer is usually an entity (node) in the KG, and the reasoning required to answer questions is either single-fact based (Bordes et al., 2015), multi-hop (Yih et al. 2015, Zhang et al. 2017) or conjunction/comparison based reasoning (Talmor and Berant, 2018). Temporal KGQA takes this a step further where:

1. The underlying KG is a Temporal KG.
2. The answer is either an entity or time duration.
3. Complex temporal reasoning might be needed.

KG Embeddings are low-dimensional dense vector representations of entities and relations in a KG. Several methods have been proposed in the literature to embed KGs (Bordes et al. 2013, Trouillon et al. 2016, Vashishth et al. 2020). These embeddings were originally proposed for the task of KG completion i.e., predicting missing edges in the KG, since most real world KGs are incomplete. Recently, however, they have also been applied to the task of KGQA where they have been shown to increase performance the settings of both of complete and incomplete KGs (Saxena et al. 2020; Sun et al. 2020).

Dataset	KG	Temporal facts	Question Types			# questions
			Multi-Entity	**Multi-Relation**	**Temporal**	
SimpleQuestions	FreeBase	✗	✗	✗	0%	108k
MetaQA	MetaQA KG	✗	✗	✓	0%	400k
WebQuestions	FreeBase	✗	✗	✓	<16%	5,810
ComplexWebQuestions	FreeBase	✗	✓	✓	-	35k
TempQuestions	FreeBase	✗	✓	✓	100%	1,271
CRONQUESTIONS (ours)	WikiData	✓	✓	✓	100%	410k

Table 1: KGQA dataset comparison. Statistics about percentage of temporal questions for WebQuestions are taken from Jia et al. (2018a). We do not have an explicit number of temporal questions for ComplexWebQuestions, but since it is constructed automatically using questions from WebQuestions, we expect the percentage to be similar to WebQuestions (16%). Please refer to Section 2.1 for details.

Temporal KG embeddings are another upcoming area where entities, relations and timestamps in a temporal KG are embedded in a low-dimensional vector space (Dasgupta et al. 2018, Lacroix et al. 2020, Jain et al. 2020, Goel et al. 2019). Here too, the main application so far has been temporal KG completion. In our work, we investigate whether temporal KG Embeddings can be applied to the task of Temporal KGQA, and how they fare compared to non-temporal embeddings or off-the-shelf methods without any KG Embeddings.

In this paper we propose CRONQUESTIONS, a new dataset for Temporal KGQA. CRONQUES-TIONS consists of both a temporal KG and accompanying natural language questions. There were three main guiding principles while creating this dataset:

1. The associated KG must provide temporal annotations.
2. Questions must involve an element of temporal reasoning.
3. The number of labeled instances must be large enough that it can be used for training models, rather than for evaluation alone.

Guided by the above principles, we present a dataset consisting of a Temporal KG with 125k entities and 328k facts, along with a set of 410k natural language questions that require temporal reasoning.

On this new dataset, we apply approaches based on deep language models (LM) alone, such as T5 (Raffel et al., 2020), BERT (Devlin et al., 2019), and KnowBERT (Peters et al., 2019), and also hybrid LM+KG embedding approaches, such as Entities-as-Experts (Févry et al., 2020) and Em-bedKGQA (Saxena et al., 2020). We find that these baselines are not suited to temporal reasoning. In response, we propose CRONKGQA, an enhancement of EmbedKGQA, which outperforms

baselines across all question types. CRONKGQA achieves very high accuracy on simple temporal reasoning questions, but falls short when it comes to questions requiring more complex reasoning. Thus, although we get promising early results, CRONQUESTIONS leaves ample scope to improve complex Temporal KGQA. Our source code along with the CRONQUESTIONS dataset can be found at https://github.com/apoorvumang/CronKGQA.

2 Related work

2.1 Temporal QA data sets

There have been several KGQA datasets proposed in the literature (Table 1). In SimpleQuestions (Bordes et al., 2015) one needs to extract just a single fact from the KG to answer a question. MetaQA (Zhang et al., 2017) and WebQuestionsSP (Yih et al., 2015) require multi-hop reasoning, where one must traverse over multiple edges in the KG to reach the answer. ComplexWebQuestions (Talmor and Berant, 2018) contains both multi-hop and conjunction/comparison type questions. However, none of these are aimed at temporal reasoning, and the KG they are based on is non-temporal.

Temporal QA datasets have mostly been studied in the area of reading comprehension. One such dataset is TORQUE (Ning et al., 2020), where the system is given a question along with some context (a text passage) and is asked to answer a multiple choice question with five choices. This is in contrast to KGQA, where there is no context, and the answer is one of potentially hundreds of thousands of entities.

TempQuestions (Jia et al., 2018a) is a KGQA dataset specifically aimed at temporal QA. It consists of a subset of questions from WebQuestions, Free917 (Cai and Yates, 2013) and Complex-Questions (Bao et al., 2016) that are temporal in

Reasoning	Example Template	Example Question
Simple time	When did {head} hold the position of {tail}	*When did Obama hold the position of President of USA*
Simple entity	Which award did {head} receive in {time}	*Which award did Brad Pitt receive in 2001*
Before/After	Who was the {tail} {type} {head}	*Who was the President of USA before Obama*
First/Last	When did {head} play their {adj} game	*When did Messi play their first game*
Time join	Who held the position of {tail} during {event}	*Who held the position of President of USA during WWII*

Table 2: Example questions for different types of temporal reasoning. {head}, {tail} and {time} correspond to entities/timestamps in facts of the form (head, relation, tail, timestamp). {event} corresponds to entities in event facts eg. *WWII*. {type} can be one of before/after and {adj} can be one of first/last. Please refer to Section 3.2 for details.

nature. They gave a definition for "temporal question" and used certain trigger words (for example 'before', 'after') along with other constraints to filter out questions from these datasets that fell under this definition. However, this dataset contains only 1271 questions — useful only for evaluation — and the KG on which it is based (a subset of FreeBase (Bollacker et al., 2008)) is not a temporal KG. Another drawback is that FreeBase has not been under active development since 2015, therefore some information stored in it is outdated and this is a potential source of inaccuracy.

2.2 Temporal QA algorithms

To the best of our knowledge, recent KGQA algorithms (Miller et al. 2016; Sun et al. 2019; Cohen et al. 2020; Sun et al. 2020) work with *non-temporal KGs*, i.e., KGs containing facts of the form (subject, relation, object). Extending these to *temporal KGs* containing facts of the form (subject, relation, object, start time, end time) is a non-trivial task. TEQUILA (Jia et al., 2018b) is one method aimed specifically at temporal KGQA. TEQUILA decomposes and rewrites the question into non-temporal sub-questions and temporal constraints. Answers to sub-questions are then retrieved using any KGQA engine. Finally, TEQUILA uses constraint reasoning on temporal intervals to compute final answers to the full question. A major drawback of this approach is the use of pre-specified templates for decomposition, as well as the assumption of having temporal constraints on entities. Also, since it is made for non-temporal KGs, there is no direct way of applying it to temporal KGs where facts are temporally scoped.

3 CRONQUESTIONS: The new Temporal KGQA dataset

CRONQUESTIONS, our Temporal KGQA dataset consists of two parts: a KG with temporal annotations, and a set of natural language questions

requiring temporal reasoning.

3.1 Temporal KG

To prepare our temporal KG, we started by taking all facts with temporal annotations from the WikiData subset proposed by Lacroix et al. (2020). We removed some instances of the predicate "*member of sports team*" in order to balance out the KG since this predicate constituted over 50 percent of the facts. Timestamps were discretized to years. This resulted in a KG with 323k facts, 125k entities and 203 relations.

However, this filtering of facts misses out on important world events. For example, the KG subset created using the aforementioned technique contains the entity *World War II* but no associated fact that tells us when *World War II* started or ended. This knowledge is needed to answer questions such as "*Who was the President of the USA during World War II?*" To overcome this shortcoming, we first extracted entities from WikiData that have a "start time" and "end time" annotation. From this set, we then removed entities which were game shows, movies or television series (since these are not important world events, but do have a start and end time annotation), and then removed entities with less than 50 associated facts. This final set of entities was then added as facts in the format (*WWII, significant event, occurred, 1939, 1945*). The final Temporal KG consisted of 328k facts out of which 5k are event-facts.

3.2 Temporal Questions

To generate the QA dataset, we started with a set of templates for temporal reasoning. These were made using the five most frequent relations from our WikiData subset, namely
- *member of sports team*
- *position held*
- *award received*
- *spouse*

Template	*When did {head} play in {tail}*
Seed Qn	*When did **Messi** play in **FC Barcelona***
Human Paraphrases	*When was **Messi** playing in **FC Barcelona*** *Which years did **Messi** play in **FC Barcelona*** *When did **FC Barcelona** have **Messi** in their team* *What time did **Messi** play in **FC Barcelona***
Machine Paraphrases	*When did **Messi** play for **FC Barcelona*** *When did **Messi** play at **FC Barcelona*** *When has **Messi** played at **FC Barcelona***

Table 3: Slot-filled paraphrases generated by humans and machine. Please refer to Section 3.2 for details.

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

Table 4: Number of questions in our dataset across different types of reasoning required and different answer types. Please refer to Section 3.2.1 for details.

- *employer*

This resulted in 30 unique seed templates over five relations and five different reasoning structures (please see Table 2 for some examples). Each of these templates has a corresponding procedure that could be executed over the temporal KG to extract all possible answers for that template. However, similar to Zhang et al. (2017), we chose not to make this procedure a part of the dataset, to remove unwelcome dependence of QA systems on such formal candidate collection methods. This also allows easy augmentation of the dataset, since only question-answer pairs are needed.

In the same spirit as ComplexWebQuestions, we then asked human annotators to paraphrase these templates in order to generate more linguistic diversity. Annotators were given slot-filled templates with dummy entities and times, and asked to rephrase the question such that the dummy entities/times were present in the paraphrase and the question meaning did not change. This resulted in 246 unique templates.

We then used the monolingual paraphraser developed by Hu et al. (2019) to automatically generate paraphrases using these 246 templates. After verifying their correctness through annotators, we ended up with 654 templates. These templates were

then filled using entity aliases from WikiData to generate 410k unique question-answer pairs.

Finally, while splitting the data into train/test folds, we ensured that

1. Paraphrases of train questions are not present in test questions.
2. There is no entity overlap between test questions and train questions. Event overlap is allowed.

The second requirement implies that, if the question "*Who was president before Obama*" is present in the train set, the test set cannot contain any question that mentions the entity '*Obama*'. While this policy may appear like an overabundance of caution, it ensures that models are doing temporal reasoning rather than guessing from entities seen during training. Lewis et al. (2020) noticed an issue in WebQuestions where they found that almost 30% of test questions overlapped with training questions. The issue has been seen in the MetaQA dataset as well, where there is significant overlap between test/train entities and test/train question paraphrases, leading to suspiciously high performance on baseline methods even with partial KG data (Saxena et al., 2020), which suggests that models that apparently perform well are not necessarily performing the desired reasoning over the KG.

A drawback of our data creation protocol is that question/answer pairs are generated automatically. Therefore, the question distribution is artificial from a semantic perspective. (ComplexWebQuestions has a similar limitation.) However, since developing models that are capable of temporal reasoning is an important direction for natural language understanding, we feel that our dataset provides an opportunity to both train and evaluate KGQA models because of its large size, notwithstanding its lower-than-natural linguistic variety. In Section 6.4, we show the effect that training data size has on model performance.

Summarizing, each of our examples contains

1. A paraphrased natural language question.
2. A set of entities/times in the question.
3. A set of 'gold' answers (entity or time).

The entities are specified as WikiData IDs (e.g., *Q219237*), and times are years (e.g., *1991*). We include the set of entities/times in the test questions as well since similar to other KGQA datasets (MetaQA, WebQuestions, ComplexWebQuestions) and methods that use these datasets (PullNet, EmQL), entity linking is considered as a separate problem and complete entity linking is as-

sumed. We also include the seed template and head/tail/time annotation in the train fold, but omit these from the test fold.

3.2.1 Question Categorization

In order to aid analysis, we categorize questions into "simple reasoning" and "complex reasoning" questions (please refer to Table 4 for the distribution statistics).

Simple reasoning: These questions require a single fact to answer, where the answer can be either an entity or a time instance. For example the question *"Who was the President of the United States in 2008?"* requires a single fact to answer the question, namely (*Barack Obama*, *held position*, *President of USA*, *2008*, *2016*)

Complex reasoning: These questions require multiple facts to answer and can be more varied. For example *"Who was the first President of the United States?"* This requires reasoning over multiple facts pertaining to the entity *"President of the United States"*. In our dataset, all questions that are not "simple reasoning" questions are considered complex questions. These are further categorized into the types "before/after'', "first/last" and "time join" — please refer Table 2 for examples of these questions.

4 Temporal KG Embeddings

We investigate how we can use KG embeddings, both temporal and non-temporal, along with pre-trained language models to perform temporal KGQA. We will first briefly describe the specific KG embedding models we use, and then go on to show how we use them in our QA models. In all cases, the scores are turned into suitable losses with regard to positive and negative tuples in an incomplete KG, and these losses minimized to train the entity, time and relation representations.

4.1 ComplEx

ComplEx (Trouillon et al., 2016) represents each entity e as a complex vector $\boldsymbol{u}_e \in \mathbb{C}^D$. Each relation r is represented as a complex vector $\boldsymbol{v}_r \in \mathbb{C}^D$ as well. The score ϕ of a claimed fact (s, r, o) is

$$\phi(s, r, o) = \Re(\langle \boldsymbol{u}_s, \boldsymbol{v}_r, \boldsymbol{u}_o^\star \rangle)$$
$$= \Re\left(\sum_{d=1}^D \boldsymbol{u}_s[d]\boldsymbol{v}_r[d]\boldsymbol{u}_o[d]^\star \right) \quad (1)$$

where $\Re(\cdot)$ denotes the real part and $c^\star$ is the complex conjugate. Despite further developments, ComplEx, along with refined training protocols

(Lacroix et al., 2018) remains among the strongest KB embedding approaches (Ruffinelli et al., 2020).

4.2 TComplEx, TNTComplEx

Lacroix et al. (2020) took an early step to extend ComplEx with time. Each timestamp t is also represented as a complex vector $\boldsymbol{w}_t \in \mathbb{C}^D$. For a claimed fact (s, r, o, t), their TComplEx scoring function is

$$\phi(s, r, o, t) = \Re(\langle \boldsymbol{u}_s, \boldsymbol{v}_r, \boldsymbol{u}_o^\star, \boldsymbol{w}_t \rangle) \quad (2)$$

Their TNTComplEx scoring function uses two representations of relations r: $\boldsymbol{v}_r^{\mathrm{T}}$, which is sensitive to time, and $\boldsymbol{v}_r$, which is not. The scoring function is the sum of a time-sensitive and a time-insensitive part: $\Re(\langle \boldsymbol{u}_s, \boldsymbol{v}_r^{\mathrm{T}}, \boldsymbol{u}_o^\star, \boldsymbol{w}_t \rangle + \langle \boldsymbol{u}_s, \boldsymbol{v}_r, \boldsymbol{u}_o^\star, \boldsymbol{1} \rangle)$.

4.3 TimePlex

TimePlex (Jain et al., 2020) augmented ComplEx with embeddings $\boldsymbol{u}_t \in \mathbb{C}^D$ for discretized time instants t. To incorporate time, TimePlex uses three representations for each relation r, viz., $(\boldsymbol{v}_r^{\mathrm{SO}}, \boldsymbol{v}_r^{\mathrm{ST}}, \boldsymbol{v}_r^{\mathrm{OT}})$ and writes the base score of a tuple (s, r, o, t) as

$$\phi(s, r, o, t) = \langle \boldsymbol{u}_s, \boldsymbol{v}_r^{\mathrm{SO}}, \boldsymbol{u}_o^\star \rangle + \alpha \langle \boldsymbol{u}_s, \boldsymbol{v}_r^{\mathrm{ST}}, \boldsymbol{u}_t^\star \rangle$$
$$+ \beta \langle \boldsymbol{u}_o, \boldsymbol{v}_r^{\mathrm{OT}}, \boldsymbol{u}_t^\star \rangle + \gamma \langle \boldsymbol{u}_s, \boldsymbol{u}_o, \boldsymbol{u}_t^\star \rangle, \quad (3)$$

where α, β, γ are hyperparameters.

5 CRONKGQA: Our proposed method

We start with a temporal KG, apply a time-agnostic or time-sensitive KG embedding algorithm (ComplEx, TComplEx, or TimePlex) to it, and obtain entity, relation, and timestamp embeddings for the temporal KG. We will use the following notation.

- $\mathcal{E}$ is the matrix of entity embeddings
- $\mathcal{T}$ is the matrix of timestamp embeddings
- $\mathcal{E}.\mathcal{T}$ is the concatenation of $\mathcal{E}$ and $\mathcal{T}$ matrices. This is used for scoring answers, since the answer can be either an entity or timestamp.

In case entity/timestamp embeddings are complex valued vectors in $\mathbb{C}^D$, we expand them to real valued vectors of size $2D$, where the first half is the real part and the second half is the complex part of the original vector.

We first apply EmbedKGQA (Saxena et al., 2020) directly to the task of Temporal KGQA. In its original implementation, EmbedKGQA uses ComplEx (Section 4.1) embeddings and can only deal with non-temporal KGs and single entity questions. In order to apply it to CRONQUESTIONS, we set the first entity encountered in the question as the
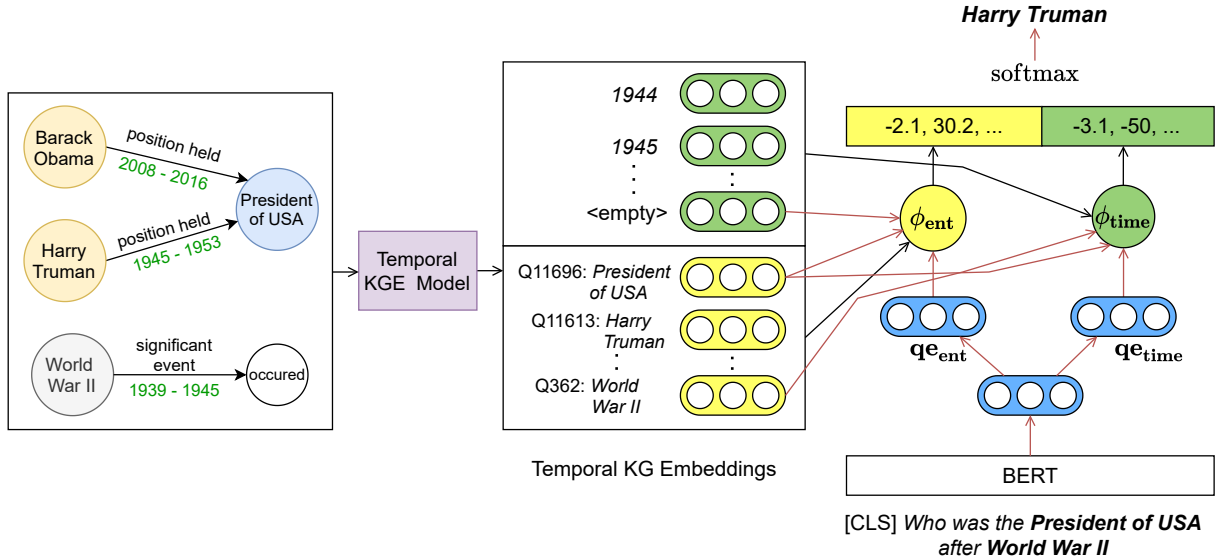
Figure 1: The CRONKGQA method. (i) A temporal KG embedding model (Section 4) is used to generate embeddings for each timestamp and entity in the temporal knowledge graph (ii) BERT is used to get two question embeddings: qe_{ent} and qe_{time}. (iii) Embeddings of entity/time mentions in the question are combined with question embeddings using equations 4 and 5 to get score vectors for entity and time prediction. (iv) Score vectors are concatenated and softmax is used get answer probabilities. Please refer to Section 5 for details.

"*head entity*" needed by EmbedKGQA. Along with this, we set the entity embedding matrix $\mathcal{E}$ to be the ComplEx embedding of our KG entities, and initialize $\mathcal{T}$ to a random learnable matrix. EmbedKGQA then performs prediction over $\mathcal{E}.\mathcal{T}$.

Next, we modify EmbedKGQA so that it can use temporal KG embeddings. We use TComplEx (Section 4.2) for getting entity and timestamp embeddings. CRONKGQA (Figure 1) utilizes two scoring functions, one for predicting entity and one for predicting time. Using a pre-trained LM (BERT in our case) CRONKGQA finds a question embedding qe. This is then projected to get two embeddings, qe_{ent} and qe_{time}, which are question embeddings for entity and time prediction respectively.

Entity scoring function: We extract a subject entity s and a timestamp t from the question. If either is missing, we use a dummy entity/time. Then, using the scoring function $\phi(s, r, o, t)$ from equation 2, we calculate a score for each entity $e \in \mathbf{E}$ as

$$\phi_{ent}(e) = \Re(\langle \boldsymbol{u}_s, \boldsymbol{qe}_{ent}, \boldsymbol{u}_e^{\star}, \boldsymbol{w}_t \rangle) \quad (4)$$

where $\mathbf{E}$ is the set of entities in the KG. This gives us a score for each entity being an answer.

Time scoring function: Similarly, we extract a subject entity s and object entity o from the question, using dummy entities if none are present. Then, using 2, we calculate a score for each times-

tamp $t \in \mathbf{T}$ as

$$\phi_{time}(t) = \Re(\langle \boldsymbol{u}_s, \boldsymbol{qe}_{time}, \boldsymbol{u}_o^{\star}, \boldsymbol{w}_t \rangle) \quad (5)$$

The scores for all entities and times are concatenated, and softmax is used to calculate answer probabilities over this combined score vector. The model is trained using cross entropy loss.

6 Experiments and diagnostics

In this section, we aim to answer the following questions:
1. How do baselines and CRONKGQA perform on the CRONQUESTIONS task? (Section 6.2.)
2. Do some methods perform better than others on specific reasoning tasks? (Section 6.3.)
3. How much does the training dataset size (number of questions) affect the performance of a model? (Section 6.4.)
4. Do temporal KG embeddings confer any advantage over non-temporal KG embeddings? (Section 6.5.)

6.1 Other methods compared

It has been shown by Petroni et al. (2019) and Raffel et al. (2020) that large LMs, such as BERT and its variants, capture real world knowledge (collected from their massive, encyclopedic training corpus) and can directly be applied to tasks such as QA. In these baselines, we do not specifically feed our version of the temporal KG to the model —

Model	Hits@1					Hits@10				
	Overall	Question Type		Answer Type		Overall	Question Type		Answer Type	
		Complex	Simple	Entity	Time		Complex	Simple	Entity	Time
BERT	0.071	0.086	0.052	0.077	0.06	0.213	0.205	0.225	0.192	0.253
RoBERTa	0.07	0.086	0.05	0.082	0.048	0.202	0.192	0.215	0.186	0.231
KnowBERT	0.07	0.083	0.051	0.081	0.048	0.201	0.189	0.217	0.185	0.23
T5-3B	0.081	0.073	0.091	0.088	0.067	-	-	-	-	-
EmbedKGQA	0.288	0.286	0.29	0.411	0.057	0.672	0.632	0.725	0.85	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**

Table 5: Performance of baselines and our methods on the CRONQUESTIONS dataset. Methods above the midrule do not use any KG embeddings, while the ones below use either temporal or non-temporal KG embeddings. Hits@10 are not available for T5-3B since it is a text-to-text model and makes a single prediction. Please refer to Section 6.2 for details.

we instead expect the model to have the real world knowledge to compute the answer.

BERT: We experiment with BERT, RoBERTa (Liu et al., 2019) and KnowBERT (Peters et al., 2019) which is a variant of BERT where information from knowledge bases such as WikiData and WordNet has been injected into BERT. We add a prediction head on top of the [CLS] token of the final layer and do a softmax over it to predict the answer probabilities.

T5: In order to apply T5 (Raffel et al., 2020) to temporal QA, we transform each question in our dataset to the form '*temporal question:* ⟨question⟩?'. For evaluation there are two cases:

1. Time answer: We do exact string matching between T5 output and correct answer.
2. Entity answer: We compare the system output to the aliases of all entities in the KG. The entity having an alias with the smallest edit distance (Levenshtein, 1966) to the predicted text output is taken as the predicted entity.

Entities as experts: Févry et al. (2020) proposed EaE, a model which aims to integrate entity knowledge into a transformer-based language model. For temporal KGQA on CRONQUES-TIONS, we assume that all grounded entity and time mention spans are marked in the question[1]. We will refer to this model as **T-EaE-add**. We try another variant of EaE, **T-EaE-replace**, where instead of adding the entity/time and BERT token embeddings, we replace the BERT embeddings with the entity/time embeddings for entity/time mentions.[2]

---

[1]This assumption can be removed by using EaE's early transformer stages as NE spotters and disambiguators.

[2]Appendix A.1 gives details of our EaE implementation.

6.2 Main results

Table 5 shows the results of various methods on our dataset. We see that methods based on large pre-trained LMs alone (BERT, RoBERTa, T5), as well as KnowBERT, perform significantly worse than methods that are augmented with KG embeddings (temporal or non-temporal). This is probably because having KG embeddings specific to our temporal KG helps the model to focus on those entities/timestamps. In our experiments, BERT performs slightly better than KnowBERT, even though KnowBERT has entity knowledge in its parameters. T5-3B performs the best among the LMs we tested, possibly because of the large number of parameters and pre-training.

Even among methods that use KG embeddings, CRONKGQA performs the best on all metrics, followed by T-EaE-replace. Since EmbedKGQA has non-temporal embeddings, its performance on questions where the answer is a time is very low — comparable to BERT — which is the LM used in our EmbedKGQA implementation.

Another interesting thing to note is the performance on simple reasoning questions. CRONKGQA far outperforms baselines for simple questions, achieving close to 0.99 hits@1, which is much lower for T-EaE (0.329). We believe there might be a few reasons that contribute to this:

1. There is the *inductive bias* of combining embeddings using TComplEx scoring function in CRONKGQA, which is the same one used in creating the entity and time embeddings, thus making the simple questions straightforward to answer. However, not relying on a scoring function means that T-EaE can be extended to any KG embedding, whereas CRONKGQA cannot.
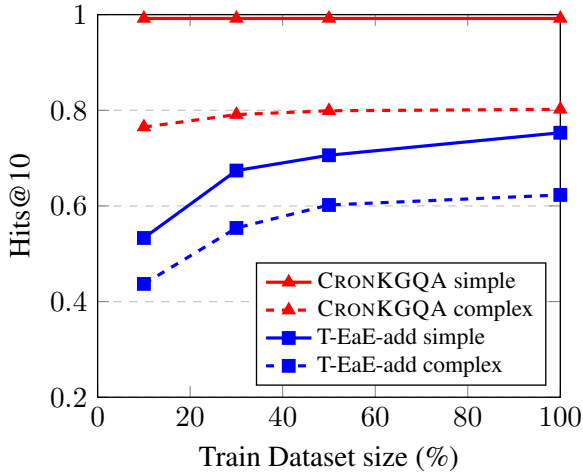
Figure 2: Model performance (hits@10) vs. training dataset size (percentage) for CRONKGQA and T-EaE-add. Solid line is for simple reasoning and dashed line is for complex reasoning type questions. For each dataset size, models were trained until validation hits@10 did not increase for 10 epochs. Please refer to Section 6.4 for details.

2. Another contributing reason could be that there are fewer parameters to be trained in CRONKGQA while a 6-layer Transformer encoder needs to be trained from scratch in T-EaE. Transformers typically require large amounts of varied data to train successfully.

6.3 Performance across question types

Table 6 shows the performance of KG embedding based models across different types of reasoning. As stated above in Section 6.2, CRONKGQA performs very well on simple reasoning questions (simple entity, simple time). Among complex question types, all models (except EmbedKGQA) perform the best on time join questions (e.g., '*Who played with Roberto Dinamite on the Brazil national football team*'). This is because such questions typically have multiple answers (such as all the players when *Roberto Dinamite* was playing for Brazil), which makes it easier for the model to make a correct prediction. In the other two question types, the answer is always a single entity/time. Before/after questions seem most challenging for all methods, with the best method achieving only 0.288 hits@1.

6.4 Effect of training dataset size

Figure 2 shows the effect of training dataset size on model performance. As we can see, for T-EaE-add,

increasing the training dataset size from 10% to 100% steadily increases its performance for both simple and complex reasoning type questions. This effect is somewhat present in CRONKGQA for complex reasoning, but not so for simple reasoning type questions. We hypothesize that this is because T-EaE has more trainable parameters — it has a 6-layer transformer that needs to be trained from scratch — in contrast to CRONKGQA that needs to merely fine tune BERT and train some shallow projection layers. These results affirm our hypothesis that having a large, even if synthetic, dataset is useful for training temporal reasoning models.

6.5 Temporal vs. non-temporal KG embeddings

We conducted further experiments to study the effect of temporal vs. non-temporal KG embeddings. We replaced the temporal entity embeddings in T-EaE-replace with ComplEx embeddings, and treated timestamps as regular tokens (not associated with any entity/time mentions). CRONKGQA-CX is the same as EmbedKGQA. The results can be seen in Table 7. As we can see, for both CRONKGQA and T-EaE-replace, using temporal KGE (TComplex) gives a significant boost in performance compared to non-temporal KGE (ComplEx). CRONKGQA receives a much larger boost in performance compared to T-EaE-replace, probably because the scoring function has been modeled after TComplEx and not ComplEx, while there is no such embedding-specific engineering in T-EaE-replace. Another observation is that questions having temporal answers achieve very low accuracy (0.057 and 0.062 respectively) in both CRONKGQA-CX and T-EaE-replace-CX, which is much lower than what these models achieve with TComplEx. This shows that having temporal KG embeddings is essential for achieving good performance for KG embedding-based methods.

7 Conclusion

In this paper we introduce CRONQUESTIONS, a new dataset for Temporal Knowledge Graph Question Answering. While there exist some Temporal KGQA datasets, they are all based on non-temporal KGs (e.g., Freebase) and have relatively few questions. Our dataset consists of both a temporal KG as well as a large set of temporal questions requiring various structures of reasoning. In order to develop such a large dataset, we used a synthetic

	Before/ After	First/ Last	Time Join	Simple Entity	Simple Time	All
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**

Table 6: Hits@1 for different reasoning type questions. 'Simple Entity' and 'Simple Time' correspond to simple question type in Table 5 while the others correspond to complex question type. Please refer to section 6.3 for more details.

Question Type	CRONKGQA		T-EaE-replace	
	CX	TCX	CX	TCX
Simple	0.29	0.987	0.248	0.329
Complex	0.286	0.392	0.247	0.257
Entity Answer	0.411	0.699	0.347	0.318
Time Answer	0.057	0.549	0.062	0.231
Overall	0.288	0.647	0.247	0.288

Table 7: Hits@1 for CRONKGQA and T-EaE-replace using ComplEx(CX) and TComplEx(TCX) KG embeddings. Please refer to Section 6.5 for more details.

generation procedure, leading to a question distribution that is artificial from a semantic perspective. However, having a large dataset provides an opportunity to train models, rather than just evaluate them. We experimentally show that increasing the training dataset size steadily improves the performance of certain methods on the TKGQA task.

We first apply large pre-trained LM based QA methods on our new dataset. Then we inject KG embeddings, both temporal and non-temporal, into these LMs and observe significant improvement in performance. We also propose a new method, CRONKGQA, that is able to leverage Temporal KG Embeddings to perform TKGQA. In our experiments, CRONKGQA outperforms all baselines. These results suggest that KG embeddings can be effectively used to perform temporal KGQA, although there remains significant scope for improvement when it comes to complex reasoning questions.

Acknowledgements

References

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-based question answering with knowledge graph. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2503–2514, Osaka, Japan. The COLING 2016 Organizing Committee.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*, pages 1–9.

Qingqing Cai and Alexander Yates. 2013. Large-scale semantic parsing via schema matching and lexicon extension. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 423–433, Sofia, Bulgaria. Association for Computational Linguistics.

William W. Cohen, Haitian Sun, R. Alex Hofer, and Matthew Siegler. 2020. Scalable neural methods for reasoning with a symbolic knowledge base.

Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. HyTE: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2001–2011, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. *arXiv preprint arXiv:2004.07202*.

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

Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2019. Diachronic embedding for temporal knowledge graph completion.

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved lexically constrained decoding for translation and monolingual rewriting. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics.

Prachi Jain, Sushant Rathi, Mausam, and Soumen Chakrabarti. 2020. Temporal Knowledge Base Completion: New Algorithms and Evaluation Protocols. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3733–3747, Online. Association for Computational Linguistics.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018a. Tempquestions: A benchmark for temporal question answering. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 1057–1062, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018b. Tequila. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. *arXiv preprint arXiv:2004.04926*.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. *arXiv preprint arXiv:1806.07297*.

Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10 (8), pages 707–710. Soviet Union.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2020. Question and answer test-train overlap in open-domain question answering datasets.

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

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. Torque: A reading comprehension dataset of temporal ordering questions.

Matthew E. Peters, Mark Neumann, Robert L. Logan IV au2, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases?

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. 2020. You can teach an old dog new tricks! on training knowledge graph embeddings. In *International Conference on Learning Representations*.

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

Haitian Sun, Andrew O. Arnold, Tania Bedrax-Weiss, Fernando Pereira, and William W. Cohen. 2020. Faithful embeddings for knowledge base queries.

Haitian Sun, Tania Bedrax-Weiss, and William W. Cohen. 2019. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Partha Pratim Talukdar, Derry Wijaya, and Tom Mitchell. 2012. Coupled temporal scoping of relational facts. In *Proceedings of WSDM 2012*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning (ICML)*.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar. 2020. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34 (03), pages 3009–3016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. 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 (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China. Association for Computational Linguistics.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. 2017. Variational reasoning for question answering with knowledge graph.

A   Appendix

A.1   Entities as Experts (EaE)

The model architecture follows Transformer (Vaswani et al., 2017) interleaved with an entity memory layer. It has two embedding matrices, for tokens and entities. It works on the input sequence

x as follows.

$$X^0 = \text{TokenEmbed}(x)$$
$$X^1 = \text{Transformer}_0(X^0, \text{num\_layers} = l_0)$$
$$X^2 = \text{EntityMemory}(X^1)$$
$$X^3 = \text{LayerNorm}(X^2 + X^1)$$
$$X^4 = \text{Transformer}_1(X^3, \text{num\_layers} = l_1)$$
$$X^5 = \text{TaskSpecificHeads}(X^4)$$

(6)

The whole model (transformers, token and entity embeddings, and task-specific heads) is trained end to end using losses for entity linking, mention detection and masked language modeling.

A.2   EaE for Temporal KGQA

CRONQUESTIONS does not provide a text corpus for training language models. Therefore, we use BERT (Devlin et al., 2019) for Transformer_0 as well as TokenEmbed (eqn. 6). For EntityMemory, we use TComplEx/TimePlex embeddings of entities and timestamps that have been pre-trained using the CRONQUESTIONS KG (please refer to Section 4 for details on KG embeddings). The modified model is as follows:

$$X^1 = \text{BERT}(x)$$
$$X^2 = \text{EntityTimeEmbedding}(X^1)$$
$$X^3 = \text{LayerNorm}(X^2 + X^1)$$
$$X^4 = \text{Transformer}_1(X^3, \text{num\_layers} = 6)$$
$$X^5 = \text{PredictionHead}(X^4)$$

(7)

For simplicity, we assume that all grounded entity and time mention spans are marked in the question, i.e., for each token, we know. which entity or timestamp it belongs to (or if it doesn't belong to any). Thus, for each token x_i in the input x,

- $X^1[i]$ contains the contextual BERT embedding of x_i
- For $X^2[i]$ there are 3 cases.
  - x_i is a mention of entity e. Then $X^2[i] = \mathcal{E}[e]$.
  - x_i is a mention of timestamp t. Then $X^2[i] = \mathcal{T}[t]$.
  - x_i is not a mention. Then $X^2[i]$ is the zero vector.

PredictionHead takes the final output from Transformer_1 of the token corresponding to the [CLS] token of BERT as the predicted answer embedding. This answer embedding is scored against $\mathcal{E}.\mathcal{T}$ using dot product to get a score for each possible answer, and softmax is taken to get answer probabilities. The model is trained on the QA dataset using cross-entropy loss. We will refer

to this model as **T-EaE-add** since we are taking element-wise sum of BERT and entity/time embeddings.

T-EaE-replace Instead of adding entity/time and BERT embeddings, we replace the BERT embeddings with the entity/time embeddings for entity/time mentions. Specifically, before feeding to Transformer_1 in step 4 of eqn. 7,

1. if x_i is not an entity or time mention, $X^3[i] = \text{BERT}(X^1[i])$
2. if x_i is an entity or time mention, $X^3[i] = \text{EntityTimeEmbedding}(X^1[i])$

The rest of the model remains the same.

A.3 Examples

Tables 8 to 12 contain some example questions from the validation set of CRONQUESTIONS, along with the top 5 predictions of the models we experimented with. T5-3B has a single prediction since it is a text-to-text model.

Question	Who held the position of Prime Minister of Sweden before 2nd World War
Question Type	Before/After
Gold answer(s)	Per Albin Hansson
BERT	Emil Stang, Sr., Sigurd Ibsen, Johan Nygaardsvold, Laila Freivalds, J. S. Woodsworth
KnowBERT	Benito Mussolini, Östen Undén, Hans-Dietrich Genscher, Winston Churchill, Lutz Graf Schwerin von Krosigk
T5-3B	bo osten unden
EmbedKGQA	**Per Albin Hansson**, Tage Erlander, Carl Gustaf Ekman, Arvid Lindman, Hjalmar Branting
T-EaE-add	**Per Albin Hansson**, Manuel Roxas, Arthur Sauvé, Konstantinos Demertzis, Karl Renner
T-EaE-replace	**Per Albin Hansson**, Tage Erlander, Arvid Lindman, Valère Bernard, Vladko Maček
CRONKGQA	**Per Albin Hansson**, Tage Erlander, Arvid Lindman, Carl Gustaf Ekman, Hjalmar Branting

Table 8: Before/After reasoning type question.

Question	When did Man on Wire receive Oscar for Best Documentary Feature
Question Type	Simple time
Gold answer(s)	2008
BERT	1995, 1993, 1999, 1991, 1987
KnowBERT	1993, 1996, 1994, 2006, 1995
T5-3B	1997
EmbedKGQA	2017, **2008**, 2016, 2013, 2004
T-EaE-add	**2008**, 2009, 2005, 1999, 2007
T-EaE-replace	2009, **2008**, 2005, 2006, 2007
CRONKGQA	**2008**, 2007, 2009, 2002, 1945

Table 9: Simple reasoning question with time answer.

Question	Who did John Alan Lasseter work with while employed at Pixar
Question Type	Time join
Gold answer(s)	Floyd Norman
BERT	Tim Cook, Eleanor Winsor Leach, David R. Williams, Robert M. Boynton, Jules Steeg
KnowBERT	1994, 1997, Walt Disney Animation Studios, Christiane Kubrick, 1989
T5-3B	john alan lasseter
EmbedKGQA	John Lasseter, **Floyd Norman**, Duncan Marjoribanks, Glen Keane, Theodore Ty
T-EaE-add	John Lasseter, Anne Marie Bardwell, Will Finn, **Floyd Norman**, Rejean Bourdages
T-EaE-replace	John Lasseter, Will Finn, **Floyd Norman**, Nik Ranieri, Ken Duncan
CRONKGQA	John Lasseter, **Floyd Norman**, Duncan Marjoribanks, David Pruiksma, Theodore Ty

Table 10: Time join type question.

Question	*Where did John Hubley work before working for Industrial Films*
Question Type	Before/After
Gold answer(s)	The Walt Disney Studios
BERT	**The Walt Disney Studios**, Warner Bros. Cartoons, Pixar, Microsoft, United States Navy
KnowBERT	École Polytechnique, Pitié-Salpêtrière Hospital, **The Walt Disney Studios**, Elisabeth Buddenbrook, Yale University
T5-3B	london film school
EmbedKGQA	**The Walt Disney Studios**, Collège de France, Warner Bros. Cartoons, University of Naples Federico II, ETH Zurich
T-EaE-add	**The Walt Disney Studios**, Fleischer Studios, UPA, Walter Lantz Productions, Wellesley College
T-EaE-replace	**The Walt Disney Studios**, City College of New York, UPA, Yale University, Indiana University
CRONKGQA	**The Walt Disney Studios**, UPA, Saint Petersburg State University, Warner Bros. Cartoons, Collège de France

Table 11: Before/After reasoning type question.

Question	*The last person that Naomi Foner Gyllenhaal was married to was*
Question Type	First/Last
Gold answer(s)	Stephen Gyllenhaal
BERT	1928, Jennifer Lash, Stephen Mallory, Martin Landau, Bayerische Verfassungsmedaille in Gold
KnowBERT	Nadia Benois, Eugenia Zukerman, Germany national football team, Talulah Riley, Lola Landau
T5-3B	gyllenhaal
EmbedKGQA	**Stephen Gyllenhaal**, Naomi Foner Gyllenhaal, Wolfhard von Boeselager, Heinrich Schweiger, Bruce Paltrow
T-EaE-add	**Stephen Gyllenhaal**, Marianne Zoff, Cotter Smith, Douglas Wilder, Gerd Vespermann
T-EaE-replace	**Stephen Gyllenhaal**, Hetty Broedelet-Henkes, Naomi Foner Gyllenhaal, Miles Copeland, Jr., member of the Chamber of Representatives of Colombia
CRONKGQA	**Stephen Gyllenhaal**, Antonia Fraser, Bruce Paltrow, Naomi Foner Gyllenhaal, Wolfhard von Boeselager

Table 12: First/Last reasoning type question.