CronKGQA Review

https://github.com/apoorvumang/CronKGQA

Jae Park

The paper investigates whether temporal KG Embeddings can be applied to the task of Temporal KGQA and perform better vs. non-temporal embeddings

<CRONQUESTIONS>

A new Temporal KGQA dataset that consists of **both** *KG with temporal annotations* **and** *a set of natural language questions requiring temporal reasoning*.

1. The associated KG must provide temporal annotations (Temporal KG)
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

Overview

0. **TComplEx KG Embedding**
1. **Temporal KG**
2. **Temporal QA dataset**

- Dataset based on WikiData.
  - Removed scholarly articles, proteins
  - Removed disambiguation, template, category, and project pages from wikipedia
  - Removed all facts for which the object was not an entity
  - Filtered out entities that had degree at least 5 and predicates that had at least 50 occurrences

  - 432k entities
  - 407 predicates
  - 1724 timestamps
  - Datum is a triple (subject, predicate, object) and a timestamp (begin, end) <- either can be unspecified
  - 7M train triples (10% contains partially specified temporal tuples)
  - from which 50k each for valid, test.

- Training and Test
  - With (subject, predicate, object, [begin, end]), sample a timestamp at random in range [begin, end].
  - For datum without a timestamp, sampled over the maximum date range
  - Then, rank the objects for a partial query (subject, predicate, ? , timestamp).
  - The final Temporal KG consists of 328k facts, out of which 5k are event facts.

Overview

0. TComplEx KG Embedding
1. **CRON Temporal KG**
2. Temporal QA dataset

- First take all the facts with temporal annotations from the WikiData dataset for TComplEx
  i.e., extract entities that have a “start time” and “end time” annotation.
  - a KG with 323k facts, 125k entities, 203 relations
  - However, this has missing entities (e.g., World War II) that has no start/end time
  - Add these set of entities in the format
    (*WWII*, *significant event*, *occurred*, *1939*, *1945*)
- The final Temporal KG consists of 328k facts, out of which 5k are event facts.
  - remove game shows, movies, television series
  - remove other entities with less than 50 associated facts

Overview

0. TComplEx KG Embedding
1. CRON Temporal KG
2. **Temporal QA dataset**

- Generate seed templates with the five most frequent **relations** from WikiData subset and five different **reasoning structure**
    - **relations**: *member of sports team, position held, award received, spouse, employer*
    - **reasoning structure**: Simple time, Simple entity, Before/After, First/Last, Time join

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.

- Using 30 unique seed templates (ex. Table 2)
    - Human annotators paraphrase the seed templates while the question meaning does not change
    - Resulted in 246 unique templates
    - Using monolingual paraphraser by Hu et al. (2019) resulted in 654 templates (machine paraphrases)
- 654 templates are filled using entity aliases from WikiData to generate 410k unique question-answer pairs
    - For train/test folds,
    - **paraphrases of train questions are not present in test questions**
    - there is no entity overlap between test questions and train questions. Event overlap is allowed
- Answer is either entity or time

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***

Overview

0. TComplEx KG Embedding
1. CRON Temporal KG
2. **Temporal QA dataset**

Simple reasoning

Complex reasoning

	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

Number of questions in the dataset across different types of reasoning required and different answer types

Simple reasoning: These questions require a single fact to answer, where the answer can be either an entity or a time instance e.x. 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 e.x. "*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 the dataset, all questions that are not "simple reasoning" questions are considered complex questions.

Dataset

```
{ 'question_text': ['When  Stevan Tommei  was playing their first game in   Juventus FC ',
                            (43461)                                     (14678)
                   "Which was the award   Narendra Kumar  received in  1985",
                                           (53755)                (127711)

                   "Theo Zwanziger received Commander's Cross of the Order of Merit of the Federal Republic of Germany in what year",

                   'The Member of the  Illinois House of Representatives after Paul Martin Simon was who'],
  'head': [   43461,   53755,   57197,   26484 ],
  'tail': [   14678,   53755,    3546, 108107 ],
  'time': [  125726, 127711,  125726,  125726  ],
  'answers_arr': [[127654], [67662], [127738], [88627]]
}
```

tkbc data id

125726 means no time specified

DistilBert Tokenizer

```
tensor([[  101,  2043, 26261,  6212,  3419, 26432,  2001,  2652,  2037,  2034, 2208,  1999, 22760,  4429,   102,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0],
        [  101,  2029,  2001,  1996,  2400,  6583,  7389,  7265,  9600,  2363, 1999,  3106,   102,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0],
        [  101, 14833,  1062,  7447,  5831,  4590,  2363,  3474,  1005,  1055, 2892,  1997,  1996,  2344,  1997,  7857,  1997,  1996,  2976,  3072, 1997,  2762,  1999,  2054,  2095,   102],
        [  101,  1996,  2266,  1997,  1996,  4307,  2160,  1997,  4505,  2044, 2703,  3235,  4079,  2001,  2040,   102,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0]]),
```

question tokenized

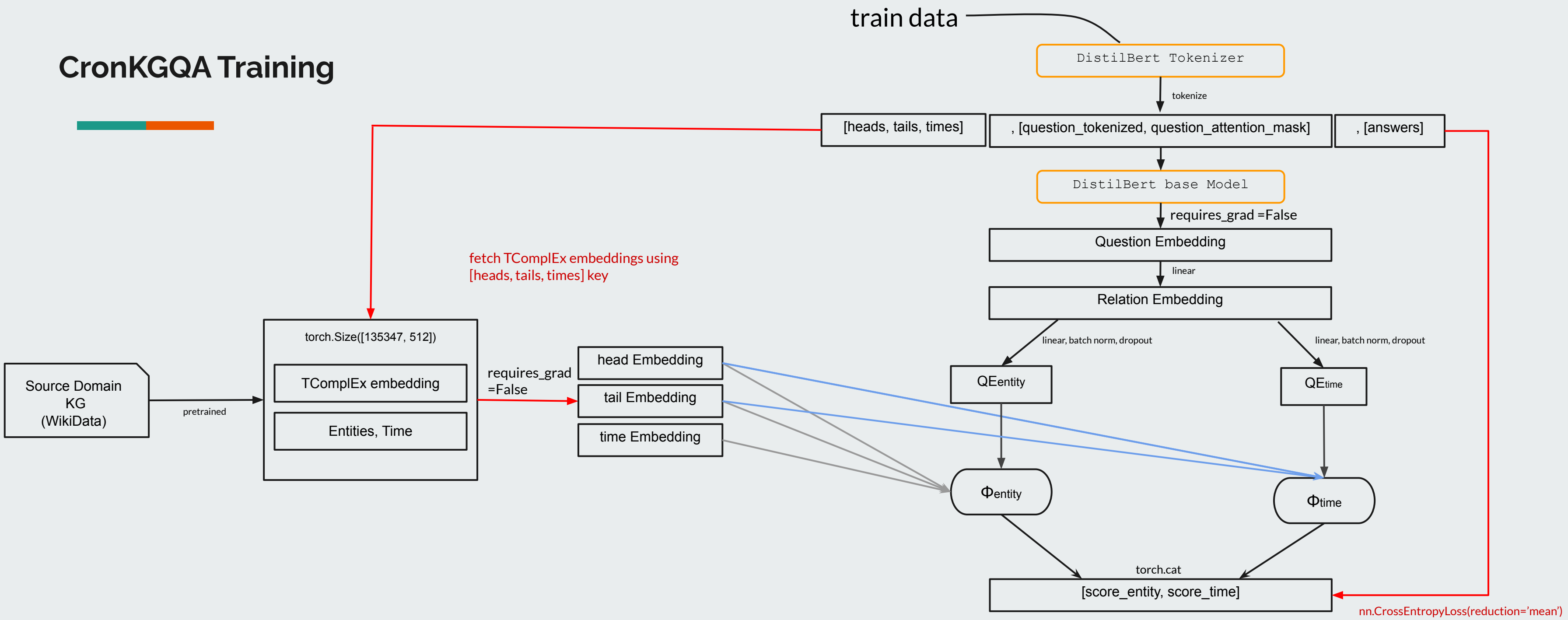
```
tensor([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]),
```

question attention mask

```
tensor([ 43461,   53755,   57197,   26484])
tensor([ 14678,   53755,    3546,  108107])
tensor([125726,  127711,  125726,  125726])
tensor([127654,   67662,  127738,   88627])
```

heads
tails
times
answers

CronKGQA Training



1. **Start with a temporal KG**
2. **Apply a time-agnostic or time-sensitive KG embedding algorithms (ComplEx, TComplEx, TimePlex)**
3. **Obtain entity, relation, and timestamp embeddings for the temporal KG**

- Using a pre-trained LM, CRONKGQA finds a question embedding *qe*. This is then projected to get two embeddings, *qeent* and *qetime*, which are question embeddings for entity and time prediction respectively.
- We extract a subject entity s and a timestamp t from the question. If either is missing, we use a dummy entity/time.

- Then, we calculate a score for each entity e ∈ E where E is the set of entities in the KG
- Entity scoring function: $\phi_{ent}(e) = \Re(\langle \boldsymbol{u}_s, \boldsymbol{qe}_{ent}, \boldsymbol{u}_e^{\star}, \boldsymbol{w}_t \rangle)$

- For each timestamp t ∈ T
- Time scoring function: $\phi_{time}(t) = \Re(\langle \boldsymbol{u}_s, \boldsymbol{qe}_{time}, \boldsymbol{u}_o^{\star}, \boldsymbol{w}_t \rangle)$

Result & Contribution

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**

Performance of baselines and the 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.*

While there exist some Temporal KGQA (TKGQA) datasets, they are all based on non-temporal KGs and have relatively few questions. The CRONQUESTIONS dataset consists of both a temporal KG as well as a large set of temporal questions requiring various structures of reasoning. It is experimentally shown 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.