Constraint-based Multi-hop Question Answering with Knowledge Graph

Sayantan Mitra, Roshni Ramnani and **Shubhashis Sengupta**
Accenture Labs, Bengaluru
{sayantan.a.mitra, roshni.r.ramnani, shubhashis.sengupta}@accenture.com

Abstract

The objective of a Question-Answering system over Knowledge Graph (KGQA) is to respond to natural language queries presented over the KG. A complex question answering system typically addresses one of the two categories of complexity: questions with constraints and questions involving multiple hops of relations. Most of the previous works have addressed these complexities separately. Multi-hop KGQA necessitates reasoning across numerous edges of the KG in order to arrive at the correct answer. Because KGs are frequently sparse, multi-hop KGQA presents extra complications. Recent works have developed KG embedding approaches to reduce KG sparsity by performing missing link prediction. In this paper, we tried to address multi-hop constrained-based queries using KG embeddings to generate more flexible query graphs. Empirical results indicate that the proposed methodology produces state-of-the-art outcomes on three KGQA datasets.

1 Introduction

Multi-relational graph, also known as Knowledge Graph (KG) comprises of a large number (often, millions) of entities and relations represented in the form of triplets (entity -> relation -> entity). Some of the most widely used KGs include DBPedia (Lehmann et al., 2015), Freebase [1], YAGO (Suchanek et al., 2007), KENSHO[2] and NELL (Mitchell et al., 2018). In the recent years, Knowledge Graph question answering (KGQA) has emerged as a significant research field (Sun et al., 2018; Zhang et al., 2018; Bordes et al., 2014). Given a natural language question, a KGQA system derives the right answer by analyzing the question and mapping it to the underlying KG.

Early works of KGQA mainly focused on simple questions containing single relations (Yang et al., 2014; Hao et al., 2017; Dong et al., 2015). However, in the real world, questions are often complex and recent work focuses on addressing these complexities. The complexities in KGQA can broadly be divided into two types: (1) **Constraint based:** Single-relation questions with constraints. For example, in this query *"when did the 7th harry potter book come out?"* there is only one relation, *"published in"* between the answer entity and the entity, *"harry potter book"* but there is also a constraint *"7th"* which needs to be addressed. To handle these kind of questions, query graph generation methods have been proposed (Yih et al., 2015; Bao et al., 2016; Luo et al., 2018). These methods first identify the 1-hop paths and then apply constraints on them. (2) **Multi-hop based:** Questions with multi-hop answers. For example, consider this query *"What language is spoken where the capital city is Brussels?"* the answer is associated with entity *"Brussels"* through two hops of relations, namely, *"capital of"* and *"language spoken"*. For addressing such multi-hop questions, it is important to consider longer paths. One of the main challenges is increasing search space. It is important to restrict the multi-hop relations to be considered, otherwise the search space can grow exponentially with the length of the relation paths. For example, Chen et al. (2019) and Lan et al. (2019) proposed to consider only the best matching relations instead of all relations when extending a relation path. However, there exists little work to address both types of complexities together.

In this paper, we address both types of complex question answering - with constraints as well as multi-hop relations - together. We propose an embedding based graph query generation method by allowing longer relation paths. Instead of adding constraints after complete generation of all probable paths, we apply constraints on partial paths

[1] https://developers.google.com/freebase

[2] https://www.kaggle.com/kenshoresearch/kensho-derived-wikimedia-data

Proceedings of NAACL-HLT 2022: Industry Track Papers, pages 280 - 288
July 10-15, 2022 ©2022 Association for Computational Linguistics

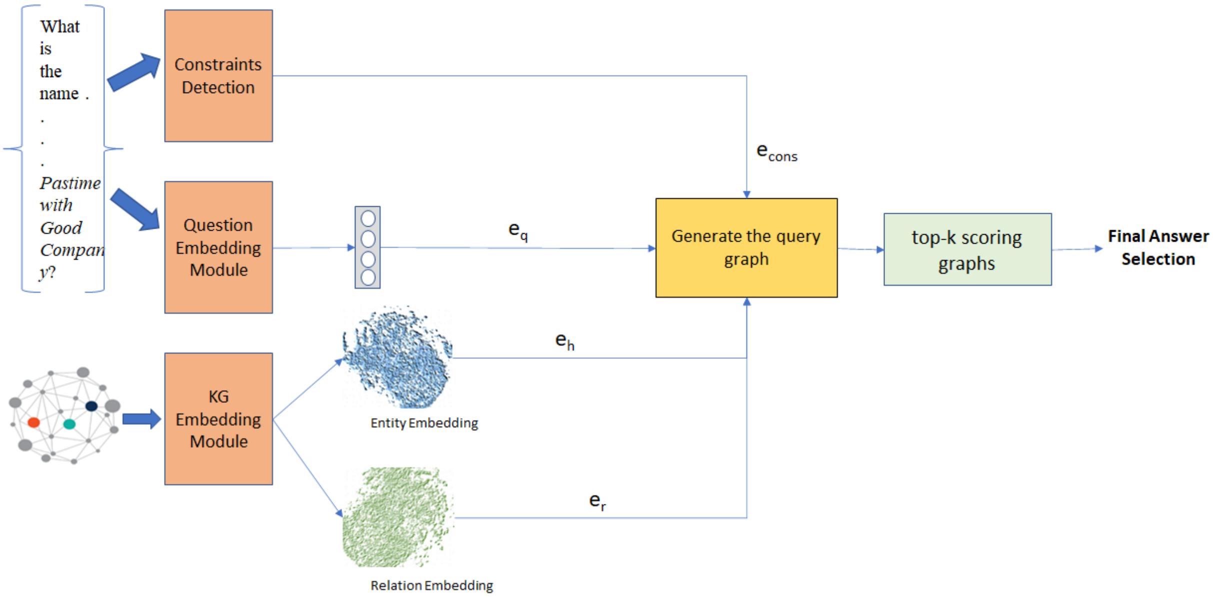

Figure 1: Overall representation of the proposed method for constraint based Multi-hop KGQA. It has five components: (1) KG Embedding Module which learns embeddings of all the entities and relations present in the KG, (2) Question Embedding Module which learns embedding for the given question, (3) Constraint Module for identifying constraints in the question (4) Graph Query Generation Module which generates the relevant query graphs based on the question, and (5) The fifth module selects top-k graphs and generates the final answer. Graph embeddings help our proposed method to effectively handle KG sparsity.

and explore the next path segments. This helps to lessen the query search space effectively. For the ComplexWebQuestions dataset, which has more number of complex questions; our method outperforms SOTA in terms of Prec@1 and F1. On other benchmark datasets as well our proposed approach achieves SOTA results. The overall representation of our proposed model is presented in Figure 1. We make the following contributions in this paper:

1. Our proposed method combines embeddings with query graphs to address constraint based multi-hop complex questions. To the best of our knowledge, this is the first attempt of combining embeddings with query graph to address all types of complex questions.

2. The proposed method leverages the requirement of answer selection from a pre-specified local neighborhood - an auxiliary constraint.

2 Related Work

2.1 Knowledge Graph Question Answering

Previous works (Li et al., 2018) used TransE (Bordes et al., 2013) graph embedding method to answer factoid based questions. However, it is a simple question answering method which works with 1-hop questions and furthermore, it requires ground-truth labeling for each question. Yih et al. (2015) and Bao et al. (2016) in their works used query graph based approaches to answer the questions. Yang et al. (2015) uses embedding based approach to co-related natural language questions to its corresponding logic forms. Different methodologies proposed in (Hao et al., 2017; Lukovnikov et al., 2017; Yin et al., 2016; Dai et al., 2016; Dong et al., 2015) use neural networks based approach. These neural networks are trained to learn a scoring function and rank the candidate answers based on these scores. There are other works (Mohammed et al., 2018; Ture and Jojic, 2017) which have formulated the QA task as a classification problem by using relations as a label. These approaches are not easily extendable to multi-hop settings.

2.2 Knowledge Graph Embedding

Real world KGs have the following limitations: (1) Most of them are often incomplete (Wang et al., 2017); (2) Real-world data is frequently dynamic and constantly changing(Cai et al., 2018). Therefore, KG completion is often formulated as the link-prediction problem(Arora, 2020). In the recent years, a lot of research has gone into link predictions in Knowledge Graphs using KG embeddings.

TransE (Bordes et al., 2013) generates high dimensional embeddings for entities in real space

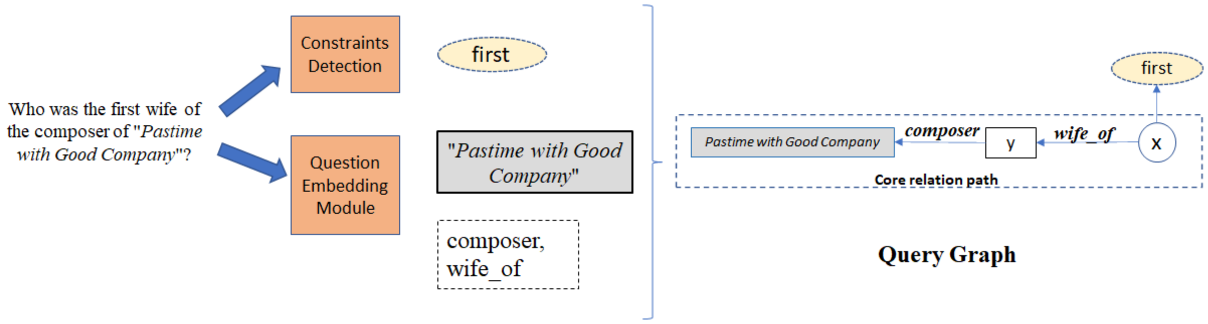
Who was the first wife of the composer of "Pastime with Good Company"?
Constraints Detection
Question Embedding Module
first
"Pastime with Good Company"
composer, wife_of
first
Pastime with Good Company
composer
y
wife_of
x
Core relation path
Query Graph

Figure 2: Constraint(s), topic entity and relations are detected for the given question. Assuming we start from the topic entity *"Pastime with Good Company"*, the core relation path is the path linking topic entity to the variable **X**. Here, only one constraint (*"first"*) is present, represented by shaded ellipse.

and TransE (Bordes et al., 2013) embeds entities in high-dimensional real space and translates between the head and tail entities. DistMult (Yang et al., 2015) and RESCAL (Nickel et al., 2011) construct KG embeddings by learning a score function that contains a bi-linear product of the vectors of the head and tail entities, as well as a relation matrix. These models, however, only consider each individual fact and neglect intrinsic relationships, thus cannot capture deeper semantics for better embedding. ComplEx (Trouillon et al., 2016), first presents complex vector space, which is capable of capturing both symmetric and antisymmetric relations. It uses tensor factorization to generate embeddings of relations and entities in complex space. The complex vectors can retain the benefit of dot product, that is linearity in both space and time complexity. This motivated us to use this embedding in our present work. RotatE (Sun et al., 2019) creates entity embeddings by projecting them in complex space, and relations are represented as complex plane rotations. InteractE (Vashishth et al., 2020) is an improvement over ConvE (Dettmers et al., 2018) method by increasing feature interaction. These models have high time complexity.

3 Background

3.1 Knowledge Graph

A knowledge graph is a set of triples represented by (h, r, t), where $r \in \mathcal{R}$ is the relation and $h, t \in \mathcal{E}$ are subject and object respectively. Here, $\mathcal{R}$ and $\mathcal{E}$ represents set of relations and entities respectively.

3.2 Link Prediction

Given an incomplete knowledge graph, the task of the link prediction is to predict the valid unknown links. This task is achieved by KG embedding models which assign a score $s = \phi(h, r, t) \in \mathcal{R}$ to the predicted links and validate whether the link is true. The goal of this model is to predict the missing links correctly.

3.3 Knowledge Graph Embeddings

Complex Embedding (Trouillon et al., 2016) is a latent factorization method that learns a large variety of symmetric and anti-symmetric relations in complex space. The complex vectors can retains the benefit of dot product, that is linearity in both space and time complexity. This motivated us to use this embedding in our present work. It is used to generate entity and relation embeddings in knowledge graphs. Given $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$ the complex embedding generates $e_h, e_r, e_t \in \mathbb{C}^d$ and defines a scoring function:

$$\phi(h, r, t) = Re(\sum_{k=1}^{d} e_h^{(k)} e_r^{(k)} \bar{e}_t^{(k)}) \qquad (1)$$

For all correct triplets $\phi(h, r, t) > 0$ and for others $\phi(h, r, t) < 0$. Re stands for the real part of the complex numbers.

3.4 Graph Query

For a given question Q, the task of the KGQA is to find an answer a such that $a \in \mathcal{E}$.

In Figure 2, we show the query graph (Bao et al., 2016; Yih et al., 2015; Luo et al., 2018) for the input question *Who was the first wife of the composer of "Pastime with Good Company"?* A query graph broadly have four parts: (i) A **grounded entity**, a head/ topic entity (for e.g *"Pastime with Good Company"*) which is explicitly mentioned in the question. It is represented by a shaded recatangle in Fig 2; (ii) A **lambda variable** (**X** in Figure 2) is the actual answer to the input question; (iii) An **existential entity**, intermediate node/nodes (**y** in Figure 2) between grounded entity and lambda variable; and (iv) An **aggregation function** (*argmin/count*) is the constraint imposed on the lambda variable. In Fig 2, *first* is the constraint on the **lambda variable**, it is internally mapped to *argmin* (described under Section 4.6). The edges of the graph represent the relations $r \in \mathcal{R}$. The **core relation path** is the path connecting the **topic entity** to the **lambda variable** *X*.

4 Method

4.1 Problem Statement

For a given natural language question q having relations r ($r \in \mathcal{R}$), entities e_h ($e_h \in \mathcal{E}$) and zero/more constraint(s), the task is to identify the answer e_t, where $e_t \in \mathcal{E}$. As an external knowledge source, Knowledge graph $\mathcal{G}$ is used. It is the set of available facts represented by triples $\mathcal{K}$, where $\mathcal{K} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. Here, $\mathcal{R}$ is set of relations and $\mathcal{E}$ is set of entities.

4.2 Overview of the Proposed Method

Our proposed method uses graph embeddings to answer complex questions. It begins by learning a KG representation in the embedding space. For a given question, it then learns the question embedding and also identifies the topic entities. For relation extraction, we use the training questions and their answers to learn the linking model. For learning the temporal constraints and superlative linking, we simply use regular expressions and a superlative word list(Luo et al., 2018) . The superlative words are manually mapped to the aggregation functions: *argmin* & *argmax*. Finally it combines these embedding and constraints to predict the answer.

4.3 KG Embedding Module

For KG embedding we used complex embeddings (Trouillon et al., 2016) for all $h, t \in \mathcal{E}$ and all $r \in \mathcal{R}$ such that $e_h, e_r, e_t \in \mathbb{C}^d$. The entity embeddings are used to learn a triple scoring function between topic entity, question and answer entity. The selected triplets are used to generate the query graphs. The entity and relation embeddings learned here are kept fixed and used for fine-tuning subsequent steps. For our work we have used latest dump of Freebase[3] as our Knowledge graph for all the datasets.

4.4 Question Embedding Module

This module is used to map the natural language questions to a fixed dimension vector $e_q \in \mathbb{C}^d$. We have used ROBERTa[4] (Liu et al., 2019) model to generate q into vector of dimension 768. The generated vector is passed through three fully-connected linear layers with ReLU activation and a dropout of 0.1 in each layer and finally projected to a complex space $\mathbb{C}^d$.

For a question q, topic entity $h \in \mathcal{E}$ and set of answers $\mathcal{A} \subseteq \mathcal{E}$, the question embedding is learned such that

$$\phi(e_h, e_q, e_a) > 0 \; \forall a \in \mathcal{A} \tag{2}$$

$$\phi(e_h, e_q, e_{\bar{a}}) < 0 \; \forall \bar{a} \notin \mathcal{A} \tag{3}$$

where, ϕ is defined in equation 1 and e_h, e_a are entity embeddings. The model is learned by minimizing the binary cross entropy loss between the $sigmoid$ of the scores and the target answer labels. When the entities are large we do label smoothing.

4.5 Relation matching

For relation matching we learn a scoring function $S_r(r, q)$ similar to PullNet (Sun et al., 2019) and rank the relations $r \in \mathcal{R}$ for question q. Let $\bar{q} = \{< s > w_1, w_2 \dots w_{|q|} < /s >\}$, word sequence in question q and h_r be the relation embedding, then the scoring function is defined as follows:

$$h_q = ROBERTa(\bar{q}) \tag{4}$$

$$S_r(r, q) = sigmoid(h_q^T h_r) \tag{5}$$

$ROBERTa(.)$ returns the last hidden layer output of ROBERTa model. We select those relations where $S_r > 0.5$ it is denoted as $\mathcal{R}_a$.

4.6 Query Graph Generation

After extracting the entities and relation(s) in the previous steps, the constraint(s) are detected and are manually mapped to the aggregation functions by the method described under Section 4.2. To generate the query graph g, $\{extend, aggregate\}$ actions are applied. An **extend** action extends the core path by one or more relation in $\mathcal{R}$. An **aggregate** action attaches the detected aggregation

[3] https://developers.google.com/freebase/

[4] The pre-trained ROBERTa base model could be found at https://huggingface.co/models?search=roberta

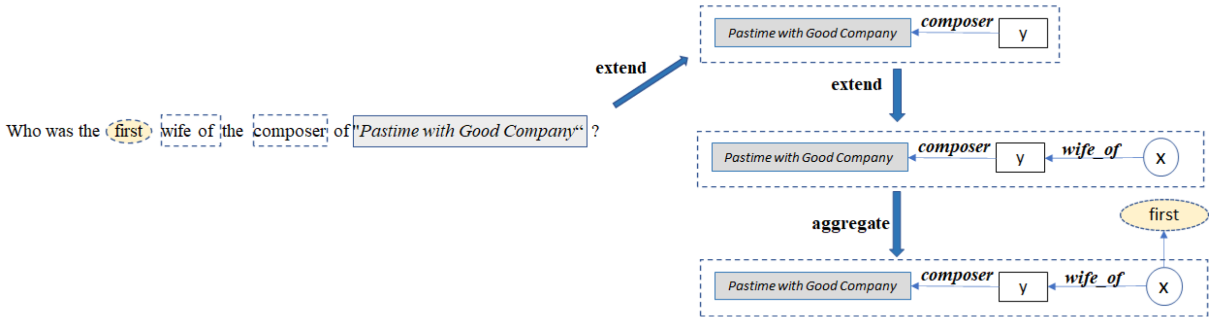
Who was the first wife of the composer of "Pastime with Good Company" ?
extend
Pastime with Good Company
composer
y
extend
Pastime with Good Company
composer
y
wife_of
x
aggregate
first
Pastime with Good Company
composer
y
wife_of
x

Figure 3: Examples of the Query Graph generation by extend and aggregate actions.

function to either a **lambda variable** or an existential variable.

In Figure 3, we start with the ground entity *"Pastime with Good Company"* and apply **extend** action to find the temporal entity **y** (here it is Henry VIII) connected by the relation *"composer"*. As there is no constraint attached with the above relation, we again apply **extend** action and find the *"lambda variable"* attached with **y** with the relation *"wife_of"*. Here, the constraint *"first"* is associated with this relation and is mapped to the aggregation function *argmin*. We apply this constraint on the *"lambda variable"* (**X**) and select the final answer (here the answer is Catherine of Aragon).

We start with **extend** action, then apply **aggregate** action, this significantly reduces the search space. We repeat the steps till we generate the query graph.

4.7 Answer Selection Module

Sometimes, the previous step may generate a set of query graphs instead of a single graph. In that case we select the best answer by this module. Let $\mathcal{R}_g$ represent the set of relations for each query graph g. We also have a set of relations $\mathcal{R}_a$ extracted from equation 5. For g, we calculate a relation score as:

$$RelScore_g = |\mathcal{R}_a \cap \mathcal{R}_g| \qquad (6)$$

We combine $RelScore_g$ with Complex score to find the answer entity:

$$e_{ans} = \arg\max_{a' \in \mathcal{N}_g} \phi(e_h, e_q, e_{a'}) + \gamma * RelScore_g$$

Here, γ is a tunable parameter. We select the entity with highest score (e_{ans}) as the answer.

5 Experiments

In this section, we first describe the datasets used for evaluating our method and the SOTA models. Finally we describe the results, ablation study and error analysis.

Question Type	CWQ	WQSP
1-hop w/o cons.	0.10%	71.30%
1-hop w/ cons.	35.90%	28.20%
2-hop w/o cons.	33.50%	0.0%
2-hop w/ cons.	30.50%	0.50%

Table 1: Statistics for CWQ and WQSP datasets. *cons.* stands for constraints.

5.1 Datasets

We evaluate our method on the following datasets: WebQuestions Semantic Parses (WQSP)(Yih et al., 2015), ComplexQuestions (CQ) (Bao et al., 2016) and ComplexWebQuestions (CWQ) (Talmor and Berant, 2018). In Table 1, we have listed the statistics for each dataset. CQ dataset does not provide ground truth query graphs so we could not collect similar statistics. It has been observed that major questions are 1-hop in CQ dataset.

Method	CWQ (Prec@1 / F1)	WQSP (F1)	CQ (F1)
Yih et al. (2015)	NA	69.0	NA
Luo et al. (2018)	NA	NA	40.9
Bao et al. (2016)	NA	NA	42.8
Lan et al. (2019)	39.3/36.5	67.9	NA
Bhutani et al. (2019)	40.8/33.9	60.3	NA
Chen et al. (2019)	30.5/29.8	68.5	35.3
Ansari et al. (2019)	NA	72.6	NA
Lan and Jiang (2020)	44.1/40.4	74.0	43.3
Proposed Method	46.3/41.9	77.8	45.9

Table 2: Comparison of the results between our proposed method and other state of the art methods.

5.2 Results

We compare the results of our proposed model with the following existing works. We first compare with the methods which use staged graph query but cannot handle multi-hop questions (Yih et al., 2015; Bao et al., 2016; Luo et al., 2018). Next, we compare with the method proposed by Lan et al. (2019), it handles constraints and consider multi-hop but does not use any strategy to reduce the search space. We further compare with (Chen et al., 2019), which does not handle constraint but uses beam search with a beam size of 1 to handle multi-hop questions. Bhutani et al. (2019) uses a strategy to decompose complex questions into simple questions and achieved SOTA results on CWQ dataset in terms of Prec@1. Ansari et al. (2019) proposed a method which generates query programs from question token by token. Finally, we compare our method with Lan and Jiang (2020), their method handles both multi-hop and constrained based complex question and uses beam search with beam size 3 to reduce the search space.

The overall comparison results with the SOTA models in shown in Table 2. From the table we can see that our model outperforms other methods on CWQ dataset in terms of both Prec@1 and F1. Our models shows an improvement of 2.2% in terms of Prec@1 and 1.5% in terms of F1 compared to the best SOTA model. This validates our claim that our proposed method can effectively handle the complex questions with both constraints and multiple hops. In WQSP dataset, the percentage of constrained based questions are low specially for multi hops (0.5% only, shown in Table 1). For this reason, our model not only outperforms all other SOTA models but also displays around 74% F1 score which is highest in comparison to other two datasets (CWQ and CQ). CQ dataset contains only single hop constrained based questions. In this dataset also our model outperforms other models in terms of F1 by 2.5%. This shows the effectiveness of our model in terms of handling only constraint based question. Overall the results in Table 2 shows the robustness and efficiency of our proposed model.

Method	**CWQ (Prec@1 / F1)**
SOTA	44.1 / 40.4
w GRU	43.3 / 38.6
w/o extend	26.4 / 22.8
w/o connect	36.8 / 32.3
w/o aggregate	43.8 / 39.5
Freebase-50 (avg.)	27.7 / 22.8
TransE	43.8 / 39.8
TransH	44.5 / 40.8

Table 3: Ablation study on CWQ dataset.

5.3 Ablation Study

We performed ablation study to better understand our model. To show that the performance of our model is not mainly due to use of ROBERTa (Liu et al., 2019) we replaced it with simple GRU model and conducted the experiments. The results in Table 3 shows that GRU based version of our methodology shows comparable results with SOTA in terms of both Prec@1 and F1. This verifies that performance of our method is not mainly because of the use of ROBERTa. To show the importance of each actions in the query graph, we have created three variants of our proposed method by eliminating one of the actions from each of them. From the results in Table 3, we can see that aggregate action has the least effect among the three and extend action have the most effect on the performance of the proposed method. The best answer is obtained when all the three actions are used together. To show the effectiveness of the KG embedding module we have randomly removed 50% of the relations from the KG (Freebase) and created a new KG Freebase-50 and then execute our algorithm on this new KG. We reported this step 10 times and reported the average results of our model in Table 3 in terms of Prec@1 and F1. From the results we can see that with the reduced KG our model performs similar to that of the *w/o extend* approach. This shows that our model is able to predict the missing links correctly but failed to apply constraints effectively due to sparse KG.

Further, to show the effect of embedding model on our proposed method, we have created two new variants of our proposed model using TransE (Bordes et al., 2013) and TransH(Wang et al., 2014) KG embeddings. The results are shown in Table 3. From the Table it can be seen that both the models produce comparable results with respect to SOTA models. This shows that the performance of our proposed method is not mainly dependent on the type of KG embeddings used.

6 Conclusion

In this paper, we propose an embedding based query graph generation method to address complex questions (constrained multiple hops queries). Often KGs are incomplete or sparsely populated, and this poses additional challenges for complex KGQA methods. By using KG embedding, the proposed methodology effectively address this KG sparsity problem by predicting missing links without the use of secondary corpus. By strategically incorporating constraints into the query graphs, we are able to restrict the search space. Experiments showed that our proposed method outperforms all other SOTA methods on all the datasets (CWQ, WQSP and CQ). In future, we would like include a module to handle abbreviation errors.

References

Ghulam Ahmed Ansari, Amrita Saha, Vishwajeet Kumar, Mohan Bhambhani, Karthik Sankaranarayanan, and Soumen Chakrabarti. 2019. Neural program induction for kbqa without gold programs or query annotations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4890–4896. International Joint Conferences on Artificial Intelligence Organization.

Siddhant Arora. 2020. A survey on graph neural networks for knowledge graph completion. *CoRR*, abs/2007.12374.

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.

Nikita Bhutani, Xinyi Zheng, and H V Jagadish. 2019. Learning to answer complex questions over knowledge bases with query composition. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 739–748, New York, NY, USA. Association for Computing Machinery.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620, Doha, Qatar. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

HongYun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. 2018. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637.

Zi-Yuan Chen, Chih-Hung Chang, Yi-Pei Chen, Jijnasa Nayak, and Lun-Wei Ku. 2019. UHop: An unrestricted-hop relation extraction framework for knowledge-based question answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 345–356, Minneapolis, Minnesota. Association for Computational Linguistics.

Zihang Dai, Lei Li, and Wei Xu. 2016. CFO: Conditional focused neural question answering with large-scale knowledge bases. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 800–810, Berlin, Germany. Association for Computational Linguistics.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over Freebase with multi-column convolutional neural networks. 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 260–269, Beijing, China. Association for Computational Linguistics.

Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 221–231, Vancouver, Canada. Association for Computational Linguistics.

Yunshi Lan and Jing Jiang. 2020. Query graph generation for answering multi-hop complex questions from knowledge bases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 969–974, Online. Association for Computational Linguistics.

Yunshi Lan, Shuohang Wang, and Jing Jiang. 2019. Multi-hop knowledge base question answering with an iterative sequence matching model. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 359–368.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia – a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6:167–195. 2.

Dingcheng Li, Jingyuan Zhang, and Ping Li. 2018. Representation learning for question classification via topic sparse autoencoder and entity embedding. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 126–133.

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. *arXiv e-prints*, page arXiv:1907.11692.

Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. 2017. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1211–1220, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Kangqi Luo, Fengli Lin, Xusheng Luo, and Kenny Zhu. 2018. Knowledge base question answering via encoding of complex query graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2185–2194, Brussels, Belgium. Association for Computational Linguistics.

T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2018. Never-ending learning. *Commun. ACM*, 61(5):103–115.

Salman Mohammed, Peng Shi, and Jimmy Lin. 2018. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 291–296, New Orleans, Louisiana. Association for Computational Linguistics.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 809–816, Madison, WI, USA. Omnipress.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 697–706, New York, NY, USA. Association for Computing Machinery.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2380–2390, Hong Kong, China. Association for Computational Linguistics.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242, Brussels, Belgium. Association for Computational Linguistics.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. *arXiv e-prints*, page arXiv:1902.10197.

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.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 2071–2080. JMLR.org.

Ferhan Ture and Oliver Jojic. 2017. No need to pay attention: Simple recurrent neural networks work! In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2866–2872, Copenhagen, Denmark. Association for Computational Linguistics.

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 34th AAAI Conference on Artificial Intelligence*, pages 3009–3016. AAAI Press.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 1112–1119. AAAI Press.

Min-Chul Yang, Nan Duan, Ming Zhou, and Hae-Chang Rim. 2014. Joint relational embeddings for knowledge-based question answering. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 645–650, Doha, Qatar. Association for Computational Linguistics.

Min-Chul Yang, Do-Gil Lee, So-Young Park, and Hae-Chang Rim. 2015. Knowledge-based question answering using the semantic embedding space. *Expert Systems with Applications*, 42(23):9086–9104.

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.

Wenpeng Yin, Mo Yu, Bing Xiang, Bowen Zhou, and Hinrich Schütze. 2016. Simple question answering by attentive convolutional neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1746–1756, Osaka, Japan. The COLING 2016 Organizing Committee.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *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*, pages 6069–6076. AAAI Press.