Question Answering over Knowledge Graphs with Query Path Generation

Linqing Yang, Kecen Guo, Bo Liu(✉), Jiazheng Gong, Zhujian Zhang, and Peiyu Zhao

College of Information Science and Technology, Jinan University, Guangzhou, China
joesphy_yang@163.com, {gkc0421,gongjiazheng}@stu2020.jnu.edu.cn, ddxllb@163.com, {zzj2021,jnuzpy}@stu2021.jnu.edu.cn

Abstract. Knowledge graphs have been applied in question answering. Many researchers have proposed methods based on query graph generation, but there are some defects such as high cost of query graph generation and large search scope of knowledge graphs. Especially for the complex questions, which refer to those with multi-hop relations and constraints, there are problems such as incomplete search and inaccurate selection of answers. In order to solve the problems mentioned above, this paper proposes a staged query path generation method. The approach firstly takes the predicate sequence of the question in knowledge graphs as the breakthrough and constructs the core path. Then, the constraints are obtained by analyzing the question. And on this basis, the core path is extended to generate the query path. Finally, the final answer to the question is determined through the query path. Experimental results show that the Hit score of proposed approach is higher than that of many competitive state-of-the-art baselines.

Keywords: Question answering · Knowledge graphs · Query path generation · Predicate sequence · Complex questions

1 Introduction

Question answering (QA) is the advanced form of information retrieval, which aims at answering the questions in natural language. Moreover, question answering over knowledge graph (KGQA) has the following data advantages. Firstly, in the knowledge graph (KG), an entity is associated with other entities or its attribute values through edges with semantic information. Secondly, constructing the KG usually requires the participation of experts, so it has higher accuracy. Thirdly, the structured form of KG not only improves the retrieval efficiency of computer, but also makes it possible to locate answers accurately.

Supported by the National Key Research and Development Project of China (2018YFC2002500).

G. Memmi et al. (Eds.): KSEM 2022, LNAI 13368, pp. 146–158, 2022.
https://doi.org/10.1007/978-3-031-10983-6_12

Nowadays, methods of KGQA can be divided into 5 categories: template-based [2,3], query graph-based [7,10,14,16], network-based [8,11–13,15], question graph alignment-based [1] and embedding-based [4–6,9] methods. However, the above approaches have several shortcomings. More specifically, in the template-based approach, it's not possible to cover all situations with manually defined templates. In the query graph-based approach, establishing the relationship between the question and each candidate query graph has some defects such as high cost of query graph generation, large search scope of knowledge graphs and low search efficiency. In embedding-based approach, the black box has poor interpretability. In addition, for complex questions, which refer to those with multi-hop relations and constraints, there are problems such as incomplete search and inaccurate selection of answers.

The research goal of our paper is how to answer the natural language questions, especially for the complex questions with multi-hop relations and constraints. The proposed method can save the cost of query graph generation, improve the interpretability of model, reduce the search scope of KG, and improve the accuracy of QA.

The main contributions of this paper can be summarized as follows:

- This paper takes the predicate sequence of the question in KG as the breakthrough and proposes a staged query path generation method, including predicate sequence detector training model and query path generation and answer selection model.
- The predicate sequence detector can transform the question answering model from query graph level to predicate level. The QA model firstly learns not the features of the query graph, but the predicates corresponding to the question in the KG. Furthermore, associate questions with predicate sequences and extended triples in KG, rather than directly with query graphs.
- Our model can not only enhance the interpretability of QA and solve the problem of high cost of query graph generation, but also accurately understand the intention of the question, greatly narrow the range of answer choices and save the consumption of computing resources.

The paper is organized as follows: in Sect. 2, we introduce the related work of KGQA. We describe the proposed approach in Sect. 3. In Sect. 4, the results and analysis of experiment are described. After that, conclusions are given in Sect. 5.

2 Related Work

The section introduces the work of KGQA, including template-based, query graph-based, network-based, question graph alignment-based and embedding-based methods.

Template-based question answering rely on templates to translate natural language sentences into pre-defined logical forms [2,3]. Additionally, a staged query path generation method was proposed in work [7,14]. The work [10] proposed a question answering system with predicate constraints, including dictionary construction module and dictionary-based QA module. And the work [16] proposed a framework to answer natural language questions in a user-interactive manner while keeping the cost as low as possible.

Furthermore, the key-value memory network retrieves answers with data table and the table stores facts and text encoded as key-value pairs in work [8]. In order to solve the noise problem of natural language and multi-hop inference based on knowledge graphs, the work [15] introduced an end-to-end variational inference network, which could simultaneously locate the topic entity of the question and find the unknown inference steps leading to the answer based on the question-answer pairs. The work [12] proposed a GRAFT-Net model, which creates problem-specific subgraphs containing facts, entities, and textual sentences with heuristic method and performs reasoning with variant Convolutional Neural Network (CNN). PullNet was proposed in work [11] and the model could extract facts and sentences from data to create more relevant subgraphs and perform reasoning with graph CNN. The work [13] introduced a semantic fusion model, which uses Recurrent Neural Network (RNN) to build sequence annotation module and design dynamic candidate path generation algorithm to achieve multi-hop reasoning.

A novel framework for resource description framework (RDF) question answering based on data-driven graph similarity was proposed in [1]. And a method based on knowledge embedding for KGQA was introduced in [5]. In addition, the work [9] proposed EmbedKGQA model to solve multi-hop QA mission based on knowledge graph. The work [6] proposed the RceKGQA model, which introduced relational chain reasoning to improve the multi-hop reasoning.

3 Approach

3.1 Related Definition

Definition 1. *Knowledge graph (KG) is represented as a quadruple, namely* $KG = (E, R, P, PV)$*. Where* E *is the set of entities,* R *is the set of relations,* P *is the set of attributes and* PV *is the set of attribute values.*

Definition 2. *Triplet is the basic unit of KG, which consists of subject* s*, predicate* p *and object* o*, namely* $t = (s, p, o), s \in E, p \in R \cup P, o \in E \cup PV$*. In the KG, the subject and object of triplet correspond to nodes, and edges correspond to predicates.*

Definition 3. *Focus word is the entity linked to the topic entity mention in the question and is the starting point for finding answers in the KG.*

Definition 4. *Predicate sequence is the sequence of predicates on the path from the focus word to the answer in KG.*

Definition 5. *Core path is the subgraph of knowledge graph, including focus word, predicate sequence and the nodes linked by the predicate sequence.*

Definition 6. *Query path is a subgraph of knowledge graph, which is formed by linking one or more triples with the core path according to the constraints of the question. If the question has no constraints, the query path is equivalent to the core path.*

For example, Fig. 1 is an example of a subgraph of KG, where nodes represent entities and edges represent predicates of links between entities.

For the question "What is the name of Justin Bieber brother?", our method can obtain the following key information step by step. Suppose that the predicate "*/people/person/sibling_s*" is represented by "$Predicate_1$", the predicate "*people/sibling_relationship/sibling*" is represented by "$Predicate_2$" and the predicate "*/people/person/gender*" is represented by "$Predicate_3$". Similarly, the node "*Justin Bieber*" is represented by "$Node_1$", the node "*Jaxon Bieber*" is represented by "$Node_2$" and the node "*Jazmyn Bieber*" is represented by "$Node_3$".

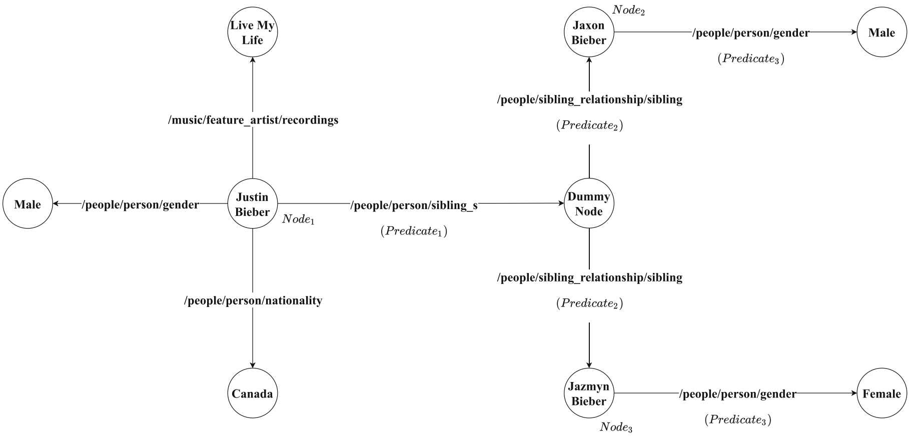

Fig. 1. Example of a subgraph of KG

Focus Word: "$Node_1$";

Predicate Sequence: "$[Predicate_1, Predicate_2]$";

Core Paths: "$Node_1 - Predicate_1 - Dummy\ Node - Predicate_2 - Node_2$" and "$Node_1 - Predicate_1 - Dummy\ Node - Predicate_2 - Node_3$";

Constraints on the Question: "*brother*";

In order to acquire the query path, the triplets $(Node_2, Predicate_3, Male)$ and $(Node_3, Predicate_3, Female)$ need to be linked into the two core paths.

Query Paths: "$Node_1 - Predicate_1 - Dummy\,Node - Predicate_2 - Node_2 - Predicate_3 - Male$" and "$Node_1 - Predicate_1 - Dummy\,Node - Predicate_2 - Node_3 - Predicate_3 - Female$";

Answer: "$Node_2$".

3.2 The Framework of KGQA Based on Query Path Generation

The process of KGQA in our paper is shown in Fig. 2 and the model framework is shown in Fig. 3. In this paper, the whole question answering over knowledge graph model mainly includes predicate sequence detector training model and query path generation and answer selection model.

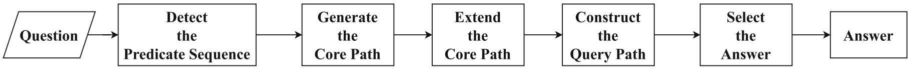

Fig. 2. Process of KGQA

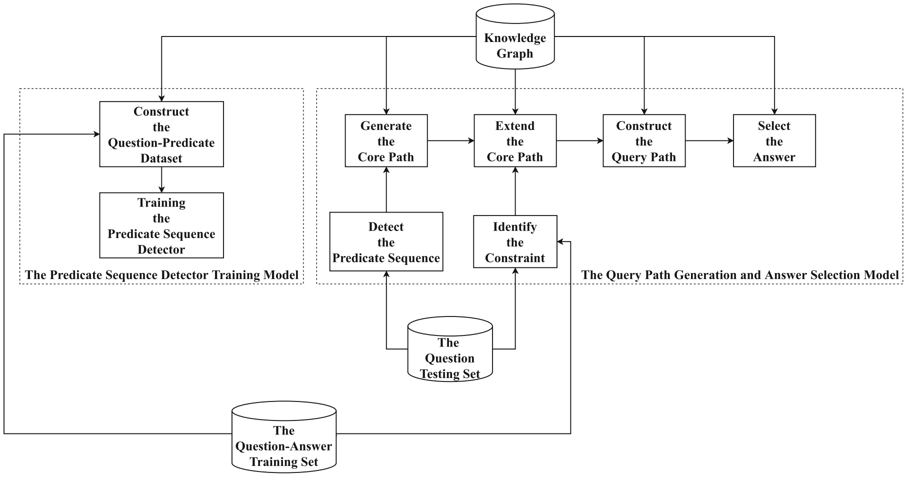
Knowledge Graph
Construct the Question-Predicate Dataset
Training the Predicate Sequence Detector
The Predicate Sequence Detector Training Model
Generate the Core Path
Extend the Core Path
Construct the Query Path
Select the Answer
Detect the Predicate Sequence
Identify the Constraint
The Query Path Generation and Answer Selection Model
The Question Testing Set
The Question-Answer Training Set

Fig. 3. The model framework of KGQA based on query path generation

Predicate Sequence Detector Training Model. The predicate sequence detector training model is mainly composed of constructing question-predicate sequence dataset module and training predicate sequence detector module. More specifically, the constructing question-predicate sequence dataset module takes the focus word, the answer and KG dataset as input, and outputs the predicate sequence. In addition, the training predicate sequence detector module takes question-predicate dataset as input, and outputs the predicate sequence detector.

In the predicate sequence detector training model, firstly the focus word and an answer of the question are extracted by question-answer training dataset, and a predicate sequence of this question is obtained after searching and filtering the

KG. By this way, the question-predicate sequence dataset is constructed. Then, the predicate sequence detector is trained with the above question-predicate sequence dataset based on RoBERTa model and Multi-Layer Perceptron (MLP).

Suppose that the predicate sequence detector is denoted as P-Detector. And the structure of P-Detector is shown in Fig. 4. In the training model, the input question goes through the Embedding module, the Encoding module and the Classifying module to get one or more predicates. For the design of P-Detector, both single-hop and multi-hop questions are considered. The question and the obtained previous predicate are inputted into P-Detector to predict the next hop predicate. Once the obtained predicate is empty, the prediction is terminated. It should be noted that in Fig. 4, the P-Detector outputs each predicate of the question in order. That is, if the first predicate is output, the corresponding input has only the question. If the subsequent predicate is output, it is used as input to P-Detector along with the question.

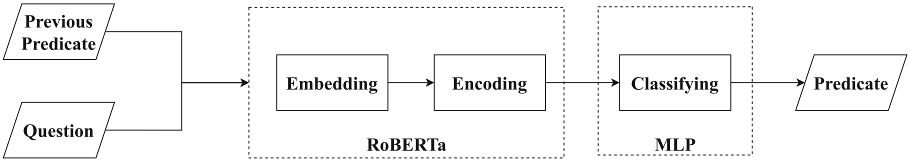

Fig. 4. The structure of P-Detector

Query Path Generation and Answer Selection Model. In the query path generation and answer selection model, firstly, the predicate sequence of question is identified by the trained P-Detector. Secondly, the core path is constructed through the focus word, the predicate sequence of the question and the nodes linked by the predicate sequence. Thirdly, the constraints are obtained by analyzing the question and the core path is extended to generate the query path based on the constraints. Finally, the candidate answers based on the query path are selected to determine the final answer to the question. Table 1 describes the algorithm of the query path generation and answer selection model, referred to as QPath-Answer.

Besides, the time complexity of QPath-Answer is $O(max(n^2, m^2, p * q^2))$. Where, n represents the length of the word sequence input by the P-Detector, m represents the sum of the word sequence length of the original problem and the problem with the focus word removed, p represents the number of query paths, which is equal to the number of candidate answers, and q represents the length of the word sequence input when the similarity calculation is completed using the RoBERTa classification task.

QPath-Answer Algorithm is explained as follows:

Line 1: Obtain the focus word of the question in KG. Since the focus words of the question have been provided in the experimental dataset, this paper does not study how to find the focus words.

Table 1. QPath-answer algorithm

```
Algorithm name: QPath-Answer
Input: The question of testing set ($question$) and dataset of knowledge graph ($KG$)
Output: The answer of question ($answer$)
1: $focus\_word$ = EXTRACT_FOCUS_WORD($question$)
2: $predicate\_sequence$ = P-Detector($question$)
3: $core\_path\_set$ = GENERATE($KG, focus\_word, predicate\_sequence$)
4: $constraints$ = IDENTIFY($question$)
5: $query\_path\_set$ = CONSTRUCT($KG, core\_path\_set, constraints$)
6: for $qp$ in $query\_path\_set$:
7:     answer.append(SELECT($qp$))
8: end for
```

Line 2: Detect the predicate sequence of the question with P-Detector.

Line 3: Generate the core path in the following form:

$Focus\,word - W_1 - node_1 - ... - node_{N-1} - W_N - node_N$;

And N is the number of predicates in the predicate sequences, W_i is the i-th predicate in the predicate sequence, and $node_i$ is the node in the core path. Note that there may be multiple core paths, and the $node_N$ in each core path is selected as the candidate answer.

Line 4: Identify the constraints on the question. The constraints considered in our paper include label value constraint, entity constraint, time constraint and ordinal constraint. The examples of constraints for questions are shown in Table 2. The constraint discrimination rules are as follows:

(1) If the question has a noun and the noun is closest to the interrogative word, as the same time, the noun indicates the entity label value in KG, then the entity label value indicated by the noun is the label value constraint of the question.
(2) If the question has a noun and the noun has obviously indication function in KG, then the noun is the entity constraint of the question.
(3) If there is a cardinal word in the question, the cardinal word is an explicit time constraint of the question. If there is a time indicator, the adverbial containing the time indicator is the adverbial time constraint. In addition, if the constraints are implicit in the tense of the question, then the tense of the question is the implicit time constraint.
(4) If the question has an ordinal word, then the ordinal word is the ordinal constraint.

Line 5: Construct the query path. The key is to decide whether to extend the core path based on the constraints of the question. If the constraint is empty or the constraint is label value data, then the core path is directly used as the query path without extension. However, if the constraint is entity data, time data or ordinal data, then the core path needs to be extended. In other words, the corresponding constraint in KG is identified and linked to the core path.

Table 2. Examples of constraints for questions

Type	Question	Constraint
Label Value	What country is the Grand Bahama in?	Country
Entity	What is the name of Justin Bieber brother?	Brother
Explicit Time	Who is the current president of the Dominican Republic in 2010?	2010
Adverbial Time	What did Abraham Lincoln do before he was president?	Before he was president
Implicit Time	Who does Joakim Noah play for?	“Does” implies the present
Ordinal	What is the name of the first Harry Potter literary series novel?	First

The linked triplet is called extended triplet, and the query path is obtained by linking extended triplets with the core path.

Line 6–8: Select the answer. The rules are as follows:

(1) For the unconstrained question, the candidate answers obtained in the core paths are determined as the final answers.
(2) For the question with the label value constraint, if the label value of a candidate answer is consistent with the constraint in its query path, the candidate answer is selected as a final answer.
(3) For the question with the entity constraint, the candidate answer of the query path where the determinative object is located is selected as a final answer. More specifically, the way to determine the determinative object is as follows: obtain each extended triplet in the query path, calculate the semantic similarity score between the object of each extended triplet and the question with the focus word removed, in this way, the object with the highest score is determinative object.
(4) For the question with the time constraint, the candidate answer of the query path where the determinative object resides is selected as a final answer. Specifically, the ways to determine the determinative object are as follows:

For the question with the explicit time constraint, the object of extended triplet is the determinative object if its time range contains the explicit time of the question.

For the question with the adverbial time constraint, firstly, the candidate answer with the highest semantic similarity with the adverbial clause of time is determined, then the time range of the object of the extended triplet corresponding to the above candidate answer is determined, and finally, the time range corresponding to the time indicator is inferred. The object of the extended triplet of the query path is the determinative object if the time range of the object contains the time range of the inference.

For the question with the implicit time constraint, the time range is inferred through the question tense. The object of the extended triplet of the query path is the determinative object if the time range of the object contains the time range of the inference.

(5) For the question with the ordinal constraint, the candidate answer of the query path where the determinative object resides is selected as the final answer. And the determinative object is determined by the ordering of the object of extended triplet in the query path.

4 Experimental Results and Analysis

4.1 Experimental Datasets

In the experiment, our datasets are divided into two types: KG dataset and QA dataset. Specifically, the KG dataset includes MetaKG and Freebase, and the QA dataset includes MetaQA (MQA) and WebQuestions Semantic Parses (WSP). In addition, the MQA dataset corresponds to MetaKG and the WSP dataset corresponds to Freebase.

The number of triplets of MetaKG is 134,741 with 43,234 entities and 9 relations. And the triple number of complete Freebase is 1.9 billion, while the number of triplets in Freebase ExQ selected in our experiment is 306,733,220 with 72,407,365 entities and 4,335 relations. In order to improve the efficiency of QA, the MetaKG and Freebase ExQ are imported into Neo4j database in our research. Moreover, the Meta KG is imported into Neo4j with the designed Cypher statement and Freebase ExQ is imported with the Freebase Neo4j Importer tool (https://github.com/kuzeko/neo4j-freebase).

In MQA, there are 329,282 training questions, 39,138 validation questions and 39,093 testing questions, including 1-hop, 2-hop and 3-hop relations. In WSP, there are 3,098 training questions and 1,639 testing questions, including 1-hop and multi-hop relations and the questions with constraints.

4.2 Baseline and Evaluation Metrics

Our model is compared with Bordes, Chopra, and Weston's QA system [4], KV-MemNN [8], VRN [15], GRAFT-Net [12], PullNet [11] and EmbedKGQA [9].

Hit evaluation metrics is used to evaluate the accuracy of QA. If the predicted answer is exactly the same as the ground truth answer, the result is a correct. Otherwise, the result is a incorrect. The calculation formula is shown in Eq. 1, where the *pos* refers to the number of questions answered correctly and *neg* refers to the number of questions answered incorrectly.

$$Hit = \frac{pos}{pos + neg} \tag{1}$$

4.3 Experimental Results and Analysis

The QA performance for MQA and WSP dataset is obtained with Hit evaluation metrics, as shown in Table 3 and Table 4.

According to the results in Table 3, the 1-hop questions get the Hit score of 93.9%. Compared with previous studies, although our method is not the best for 1-hop questions in MQA, it is the best for 2-hop and 3-hop questions.

According to the results in Table 4, the Hit score of our model is 71.1% on the WSP dataset, which is 4.4% higher than the second best model, PullNet.

Table 3. QA performance of MQA dataset

Model	1-hop Hit	2-hop Hit	3-hop Hit
Bordes, Chopra, and Weston's QA system	95.7%	81.8%	28.4%
KV-MemNN	96.2%	82.7%	48.9%
VRN	**97.5%**	89.9%	62.5%
GRAFT-Net	97.0%	94.8%	77.7%
PullNet	97.0%	**99.9%**	91.4%
EmbedKGQA	**97.5%**	98.8%	94.8%
Ours	93.9%	**99.9%**	**98.5%**

Table 4. QA performance of WSP dataset

Model	Hit
KV-MemNN	46.7%
GRAFT-Net	66.4%
PullNet	68.1%
EmbedKGQA	66.6%
Ours	**71.1**

4.4 Case Study

This section will describe three examples as follows:

(1) Question 1: “Who was vp for Richard Nixon?”
This is a question with multi-hop relations.
And the predicate sequence detector model can correctly identify its predicate sequence “[*government/us_president/vice_president*]” in KG. And the core path is generated correctly. Because no constraint is identified, the two core paths are the query paths. Therefore, the two candidate answers (Gerald Ford, Spiro) on the core paths are selected as the final answer, and the result is correct.

(2) Question 2: “Who did Samir Nasri play for before arsenal?”
This is a question with multi-hop relations and the adverbial time constraints. And the predicate sequence detector model can identify its predicate sequence “[*/sports/pro_athlete/teams*, */sports/sports_team_roster/team*]”. In addition to that four core paths are generated, whose candidate answers are “Arsenal F.C.”, “France national football team”, “Manchester City F.C.” and “Olympique de Marseille”. And moreover, it is identified that the question has the adverbial time constraint, and the constraint corresponding to the core paths: “[*/sports/sports_team_roster/from*, */sports/sports_team_roster/to*]”.
And the time “from 2008” and “to 2011” are determined by extending the core paths with time constraints. Then through the time indicator “before”, the time range is inferred to be before 2008, and the answer is “Olympique de Marseille”.

(3) Question 3: “What jobs did John Adams have before he was president?”
For this question, in the work [7], the author holds that their method can not find the query graph. In contrast, our model can find the core paths corresponding to this question. Although the constraint is misidentified in KG, four correct answers and one wrong answer are obtained.

5 Conclusion

This paper proposed a staged query path generation method for KGQA, especially for the complex questions with multi-hop relations and constraints. More specifically, our method mainly includes the predicate sequence detector training model and query path generation and answer selection model. Taking the predicate sequence of the question in KG as the breakthrough point, the question is associated with the predicate sequence and extended triplets in KG, rather than directly with the query graph. The process of QA is highly interpretable and it can accurately understand the intent of the question, greatly reduce the range of choices, and improve the efficiency of QA.

In this work, only 4 types of question constraints were studied. The next step will be to study the answers to the questions with comparative, superlative and aggregate constraints. And further explore how to accurately identify various constraints in natural language questions.

References

1. Bakhshi, M., Nematbakhsh, M., Mohsenzadeh, M., Rahmani, A.M.: Data-driven construction of SPARQL queries by approximate question graph alignment in question answering over knowledge graphs. Expert Syst. Appl. **146**, 113205 (2020)
2. Bast, H., Haussmann, E.: More accurate question answering on freebase. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, 19–23 October, 2015, pp. 1431–1440. ACM (2015)
3. Berant, J., Liang, P.: Semantic parsing via paraphrasing. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22–27, 2014, Baltimore, MD, USA, Volume 1: Long Papers, pp. 1415–1425. The Association for Computer Linguistics (2014)
4. Bordes, A., Chopra, S., Weston, J.: Question answering with subgraph embeddings. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 615–620. ACL (2014)
5. Huang, X., Zhang, J., Li, D., Li, P.: Knowledge graph embedding based question answering. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11–15, 2019, pp. 105–113. ACM (2019)
6. Jin, W., Yu, H., Tao, X., Yin, R.: Improving embedded knowledge graph multi-hop question answering by introducing relational chain reasoning. CoRR abs/2110.12679 (2021)
7. Lan, Y., Jiang, J.: 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, ACL 2020, Online, 5–10 July, 2020, pp. 969–974. Association for Computational Linguistics (2020)
8. Miller, A.H., Fisch, A., Dodge, J., Karimi, A., Bordes, A., Weston, J.: Key-value memory networks for directly reading documents. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, 1–4 November, 2016, pp. 1400–1409. The Association for Computational Linguistics (2016)
9. Saxena, A., Tripathi, A., Talukdar, P.P.: Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10, July 2020, pp. 4498–4507. Association for Computational Linguistics (2020)
10. Shin, S., Jin, X., Jung, J., Lee, K.: Predicate constraints based question answering over knowledge graph. Inf. Process. Manag. **56**(3), 445–462 (2019)
11. Sun, H., Bedrax-Weiss, T., Cohen, W.W.: 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 2019, Hong Kong, China, 3–7 November 2019, pp. 2380–2390. Association for Computational Linguistics (2019)
12. Sun, H., Dhingra, B., Zaheer, M., Mazaitis, K., Salakhutdinov, R., Cohen, W.W.: 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, Brussels, Belgium, October 31–4 November, 2018, pp. 4231–4242. Association for Computational Linguistics (2018)

13. Xiong, H., Wang, S., Tang, M., Wang, L., Lin, X.: Knowledge graph question answering with semantic oriented fusion model. Knowl. Based Syst. **221**, 106954 (2021)
14. Yih, W., Chang, M., He, X., Gao, J.: Semantic parsing via staged query graph generation: Question answering with knowledge base. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, 26–31 July, 2015, Beijing, China, Volume 1: Long Papers. pp. 1321–1331. The Association for Computer Linguistics (2015)
15. Zhang, Y., Dai, H., Kozareva, Z., Smola, A.J., Song, L.: 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, pp. 6069–6076. AAAI Press (2018)
16. Zheng, W., Cheng, H., Yu, J.X., Zou, L., Zhao, K.: Interactive natural language question answering over knowledge graphs. Inf. Sci. **481**, 141–159 (2019)