



Efficient Question Answering Based on Language Models and Knowledge Graphs

Fengying Li, Hongfei Huang, and Rongsheng Dong^(✉)

Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic
Technology, Guilin, China
{lfy, ccrsdong}@guet.edu.cn

Abstract. Knowledge graph question answering (Q&A) aims to answer questions through a knowledge base (KB). When using a knowledge base as a data source for multihop Q&A, knowledge graph Q&A needs to obtain relevant entities, their relationships and the correct answer, but often the correct answer cannot be obtained through the reasoning path because of absent relationships. Currently, using pre-trained language models (PLM) and knowledge graphs (KG) has a good effect on complex problems. However, challenging problems remain; the relationships between problems and candidate entities need to be better represented, and joint reasoning must be performed in the relationship graph based on problems and entities. To solve these problems, we expand the relational graph by adding tail entities to the list of preselected entities through reverse relations and then add the processed problems and entities to the problem subgraph. To perform inference on a relational graph, we design an attention-based neural network module. To calculate the loss of the model's inference process nodes, we use a modified Euclidean distance function as the loss function. To evaluate our model, we conducted experiments on the WebQSP and CWQ datasets, and the model obtained state-of-the-art results in both the KB-full and KB-half settings.

Keywords: Question answering · Knowledge graphs · Neural networks

1 Introduction

Question answering (Q&A) plays a central role in the field of artificial intelligence [1], and Q&A systems must be able to reason about the correct answer through the relevant knowledge base [2]. Multihop reasoning that uses knowledge bases as data sources requires subject word entities and multihop relationships from the questions and then reasons about the questions through multiple triples in the knowledge graph. For questions, it must enable an encoder that can be converted by language models (LM) to be recognized by machines [3,4], or it must have the ability to represent questions as structured knowledge graphs (KG), e.g. [5–7]. For instance, the entities in Freebase [3] and YAGO [4] can be

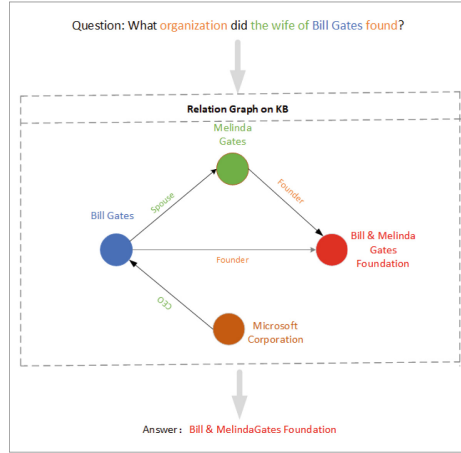


Fig. 1. A 2-hop problem on a relationship diagram, In the relationship diagram \langle Bill Gates, Spouse, Melinda Gates \rangle is a pair of relationships \langle Melinda Gates, Spouse Bill Gates \rangle is an inverse pair of relationships.

represented as nodes of knowledge graphs, and relations can be represented as edges. In recent years, Pre-trained language models have achieved good results in multihop Q&A. Although pre-trained language models have a very good representation of knowledge [5], they do not perform well in structural reasoning; furthermore, knowledge graphs can clearly explain the predicted results through inference paths [6], which are well suited for structural reasoning but lack the ability to provide full coverage of knowledge [7]. How to effectively combine pre-trained language models and knowledge graphs for Q&A remains an open and challenging problem.

A knowledge graph is a structured database that stores facts in the form of triples. Some of the larger publicly available knowledge graphs are Freebase [8], YAGO [9], NELL [13], Wikidata [14] built by Google in 2013, and DBpedia [15] extracted from Wikipedia entries given by Lehmann et al. In this paper, we focus on multihop Q&A on the problem relationship subgraph, which consists of entities and relationships, as shown in Fig. 1. Multihop Q&A currently has two dominant approaches: semantic parsing-based [2] and information retrieval-based approaches [16]. Both approaches first identify head entity in a question, connect them to entities in the knowledge base and then reason by executing the logical form of the parsed question or by extracting specific question subgraphs from the knowledge base to the answer in the subject entity domain. They used different working mechanisms to solve the knowledge base Q&A task, the former approach represented the problem through a logical symbolic model, and the latter approach constructed a subgraph specific to the problem, provided comprehensive relocation information to the problem and ranked the extracted entities based on their relevance to the problem [2]. In complex problems, more relations and topics increase the search space of potential logical forms used

for parsing, which will increase the computational cost significantly, while more relations and entities may prevent the ranking of all entities based on information retrieval methods, leading to incorrect results. Our model outperforms all current baseline models in terms of efficiency (as shown in Fig. 3).

Currently, multihop Q&A on the problem relationship subgraph has two main challenges [7].

- (1) How to form the corresponding problem-related subgraphs through incomplete knowledge graphs for the given problem text.
- (2) How to reason on the problem-related subgraphs.

To address these problems, we propose an end-to-end information retrieval model using an efficient inference model consisting of a pre-trained language model and a neural network (QA-Net),¹ which has the following advantages, Effectiveness: QA-Net achieves SOTA results on all relevant data sets Transparency: QA-Net is fully attention-based, so, we can easily understand the reasoning path of the problem. We obtain the problem text by pretraining the language model and then use it for relational reasoning by constructing a structured knowledge graph. The contributions of our paper are as follows.

- (1) We use the Roberta [9] model to embed the problem, expand the target entity list by reverse relationships, and embed the entity list by the TF-IDF [17] model (Rencode). In question reasoning, we join the Q&A context and question relationship subgraph for reasoning, which decreases the relationship gap between questions and entities.
- (2) To reason over the problem subgraph, we design an attention-based Net module for problem reasoning, add the previous problem and relation graph-generated values to the existing reasoning process by means of weights to strengthen the connection during reasoning and design a new computational function for node loss (Net).

2 Related Work

In this paper, We address multihop Q&A of a structural knowledge graph, which is constructed by extracting relevant entities and relationships from the knowledge base (as shown in Fig. 1). In previous work, PullNet [19] and QA-GNN [12] are similar to our model, with the former focusing on a hybrid form of labeled and textual relations and the latter focusing on constructing relational graph inference in multiple question and answer sessions. They first retrieve a subgraph of relations for a particular question and then use graph neural networks to reason implicitly about the answer to the question. These graph neural network-based methods are usually weak in interpretability because they cannot produce intermediate inference paths [20], which we believe is necessary in multihop Q&A.

¹ Source code will be made available post acceptance.

The graph form of Q&A that uses labels is called KGQA and is mainly grouped into two categories: semantic parsing [21, 22] and information retrieval [1, 2]. The former parses a question into logical form and queries the answer entity based on the knowledge base, while the latter retrieves subgraphs of a particular question and applies some ranking algorithm to select the most likely answer entity. Among these methods, VRN [24] and SRN [25] have good interpretability because they clarify the inference path through reinforcement learning. However, they suffer from convergence problems due to the very large knowledge base, which leads to models with very large search spaces. IRN [26] and ReifKB [27] learn the distribution of relations in the inference process but can only optimize the reasoning process through answer entities.

In single-hop problem inference, where the answers to questions can be retrieved directly from the knowledge base, pre-trained language models often outperform humans. For more challenging multihop problems, in the present work, Denis Lukovnikov et al. [28], used a transformer for a multitask model based on joint learning, but it could only handle simple Q&A tasks and used relationship prediction as a classification task. The CQA-NMT model uses a transformer for the prediction of variable-length paths in multihop Q&A but does not address the incompleteness of multihop Q&A in knowledge graphs. In this paper, we use the transformer model and TF-IDF [17] model to close the gap between the problem embedding space and the knowledge graph embedding space and add reverse relations and entities to the relationship graph to compensate for the incompleteness of knowledge graphs.

3 Methodology

We reason on a structured knowledge graph, where nodes denote entities and edges denote relationships. As shown in Fig. 1, given a question and answers, we need to reason on the knowledge graph to obtain the correct result. as shown in Fig. 3. First, we embed the problem using the Roberta model, add reverse relational entities to the list of problem-related entities and then perform entity embedding through the TF-IDF model (3.1). To find the correct inference path in the structured knowledge graph, we used the multilayer perceptron MLP for inference over the knowledge graph (3.2). To capture the relationship between the head node in the questions and other entities in the structured knowledge graph and to adjust our model, we designed the relevance score of the nodes and adjusted the calculation of the loss function (3.3) (Fig. 2).

3.1 Structured Knowledge Graph Embedding

To merge the problem and knowledge base into the same structure, we designed a joint reasoning space. We explicitly connect the problem and related entity nodes together to form a common graphical structure. Structured knowledge graphs are represented by \mathcal{G} , entities are represented by \mathcal{E} and edges connecting entities are represented by \mathcal{R} . We use q to represent the problem text, e to

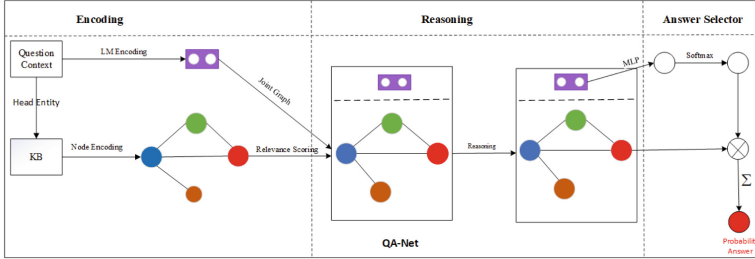


Fig. 2. The reasoning process of a 2-hop problem on QA-Net. For a two-hop problem, we start from the head entity (blue point) in the problem, and in the relationship diagram we are able to clearly know the first hop to the entity (green point) as well as the second hop entity (red point) (Color figure online)

represent the related entities, e_h to represent the head entity in the problem text, r to represent the relationship between entities, n to represent the number of entities, and $r_{i,j}$ to represent the relationship between the head entity e_i and the tail entity e_j . $r_{i,j}$ needs to be obtained from the entity relationship pairs in the knowledge base. For a multihop problem q , the correct problem path needs to be reasoned in the relational graph from the head entity e_h . The correct set of answer entities $Y = \{e_{Y^1}, \dots, e_{Y^{|Y|}}\}$ is found by reasoning on the path.

We encode the question q using the language model.

$$q^{LM} = Roberta(text(q)). \tag{1}$$

for each entity in all questions, in order for the model to have a better response to similar topic Q&A task, we use a TF-IDF model to quantify each entity in the knowledge base.

$$TF_i = \frac{n_i}{\sum_{w=0}^k n_w}, \tag{2}$$

$$IDF_i = \log(D/(1 + d_i)),$$

$$TI_i = TF_i * IDF_i.$$

where n_i denotes the number of occurrences of word i in the knowledge base, $\sum_{w=0}^k n_w$ denotes the number of occurrences of all words, and TF_i denotes the word frequency of entity i . D denotes the total number of problems, and d_i denotes the number of entities i contained in the list of target entities in the problem, TI_i is the $TF - IDF$ value for the numbered word i .

3.2 QA-Net Architecture

To obtain the correct answer to a multihop question, QA-Net starts from the head entity and goes through T-step reasoning, where at each step it looks at a

different part of the question and determines the most relevant content. Initially, to maintain the activation probability of the entities, QA-Net sets the head entity probability to 1 and the rest of the entity probabilities to 0. In each step, QA-Net calculates the relevance score of each node to represent its activated agent relationship, and then transmits the score of the entity in the activated relationship. Finally, we obtain the correct result by the softmax function. In each step, we use

$$h^{(t+1)} = f_t(h^t) + h^t. \quad (3)$$

to update h_t .

In the model calculation process, $h^t = [0, 1]^n$ denotes the score of entities in step t . $[0, 1]^n$ n -dimensional vectors have values from 0 to 1, h^0 is the initial value, and only the head entity e_h is 1.

$$\begin{aligned} q^{LM}(h_1, \dots, h_n) &= f_{enc}(q), \\ c \cdot q^t &= f^t(q^{LM}), \\ b^t &= Softmax(c \cdot q^t * q^{LM}), \\ q^t &= \frac{b^t}{\sum_{i=1}^{|n|} b_i^t * h_i}. \end{aligned} \quad (4)$$

where f_{enc} denotes the language model of the problem embedding (Roberta), n indicates the number of words in the question, h_i denotes the vector of each word in the problem f^t denotes the MLP of $R^D \rightarrow R^D$, and $D = 768$ denotes the embedding dimension of the problem.

In addition, during the model reasoning process, we need to calculate and update the relationship between nodes, And at each step we need to calculate the midweight of the problem to achieve the best results.

$$\begin{aligned} p_{i,j}^t &= f^t(r_{i,j,k}^t * e_h), \\ w_{i,j,k}^t &= Sigmoid(p_{i,j,k}^t), \\ a_i^t &= e_i^{t-1} w_{i,j,k}^t. \end{aligned} \quad (5)$$

where $r_{i,j,k}$ denotes the k -th relation sentence, e_h denotes the relationship vector of the head node, f^t is an MLP of $R^D \rightarrow R^R$, R denotes the number of relations, e_i^{t-1} denotes entity i , e_i^0 is the initial head node, and a_i^t is the target entity.

Finally, we need to find the weighted sum of a^1, a^2, \dots, a^t in each step t to determine the final output of the target entity.

$$\begin{aligned} c &= Softmax(f^t(q^{LM})), \\ a^* &= \sum_{t=1}^T c^t * a^t. \end{aligned} \quad (6)$$

where f^t is an MLP function of $R^D \rightarrow R^R$, $c \in [0, 1]^T$ denotes the probability distribution of the problem jumps, and a^* indicates the final target entity answer.

3.3 Training

Assuming that the set of answer entities to the question is $Y = e_{y^1}, \dots, e_{y^{|Y|}}$, then

$$y_i = \begin{cases} 1 & e_i \in Y \\ 0 & e_i \notin Y \end{cases} \quad (7)$$

To obtain better results, we designed an ideal equation for the model to weigh the loss between a^* and y .

$$\mathcal{L} = k * \frac{\sum_{i=1}^{|Y|} (y_i * a_i * \|a_i^* - y_i\|)}{\sum_{i=1}^{|Y|} (y_i * a_i)} \quad (8)$$

where y is the answer entity, a is the target entity, and K is a constant used to adjust the size of the loss function, the training process $k = 0.8$ in the training process, and the training effect is better. $\|a^* - y\|$ represents the square of the two vectors Euclidean distance.

In Eq. (6), since the model may become very large in each reasoning step, the values in a_i^t values may cause the gradient to explode when the number of jumps increases, thus affecting the accuracy of the model. Therefore we need to rectify the entity fraction after each step t to ensure that the value is in the range $[0, 1]$, while also retaining the differentiability of the operation. We have the following truncation function.

$$f(x) = \begin{cases} 1 & x \geq 1 \\ x & x < 1 \end{cases} \quad (9)$$

In step t , we use $f(x)$ to truncate each element in a^t .

4 Experiment

4.1 Dataset

WebQSP [8] is a small dataset containing 4737 Freebase-based natural language questions that can form a large-scale knowledge graph with millions of entities and triples. The questions in this dataset are one-hop and two-hop questions and can be answered by Freebase.

CompWebQ (CWQ) [9] is generated by WebQSP by extending the question entities and adding constraints to the answers with more jumps and constraints, and the questions require up to 4 jumps for reasoning in the knowledge graph. The dataset contains 34,654 questions and consists of four distinct question types: combinatorial (45%), connected (45%), comparative (5%), and top-level (5%) (Table 1).

Table 1. The training, validation and test set division of WebQSP and CWQ.

Dataset	Train	Dev	Test
WebQSP	2937	100	1569
CWQ	27623	3518	3513

4.2 Baseline

KVMemNN [29] is mainly intended to solve the problems that when Q&A applies a knowledge base, the knowledge base may be too limited, its schema for storing knowledge (schema) may not support certain types of Q&A, and the knowledge base is excessively sparse.

GraftNet [30] uses heuristics to extract a problem-specific subgraph from the entire relational graph, and then uses a neural network to reason about the answer.

PullNet [19] improves GraftNet by using a method that replaces the heuristic with a graph CNN to retrieve subgraphs.

EmbedKGQA [23] uses KGQA as a link prediction task and incorporates the knowledge graph embedding to solve the incompleteness of KGQA. EmbedKGQA embeds the questions and KG triples into the vector space, calculates the scores of the candidate answers by the score function and selects the highest scoring entity as the answer.

TransferNet [31] supports label and text relationships, and TransferNet maintains multishop relationship scores for entity score spreading. It starts from the subject entities of the problem and maintains a vector of entity scores. In each step, it processes the problem words and calculates the scores of the relationships between entities, then transforms these relationship scores into an adjacency matrix, and finally multiplies the entity score vector with the relationship score matrix. After repeating multiple steps, the target entities are obtained.

4.3 Results

In the entity set for word embedding, we perform frequency statistics on the entities in the knowledge graph, and then generate corresponding one-hot codes based on the statistical word frequencies to achieve the embedding of the entities. Moreover, we find that not only the reverse relationship will affect the experimental results. For some problems, we may need the tail entity in the knowledge graph to begin problem reasoning, so we will select some entities with the opposite relationship to add to the candidate entity list for training. On the WebQSP dataset, we set the reasoning step to $T = 2$, and we use Roberta as the question embedding model. We set the hidden layer size to $D = 768$. We optimize the model using RAdam [10] with a learning rate of 0.001, compute the final question answer by using a multilayer MLP to obtain the multihop reasoning and relations results, and finally select the final question answer by the sigmoid function. We obtained satisfactory results in less than 4h running on a single

Table 2. Model accuracy comparison on WebQSP, QA-Net is far more efficient than other baseline models on WebQSP.

Model	WebQSP KG-full	WebQSP KG-half
KV-Mem	46.7	32.7(31.6)
GraftNet	66.4	48.2(49.7)
PullNet	68.1	50.1(51.9)
EmbedKGQA	66.6	53.2
TransferNet	71.4	–
QA-Net (ours)	74.5	59.0

Table 3. Model accuracy comparison on CWQ, QA-Net also outperforms other baseline models on the CWQ.

Model	CWQ KG-full	CWQ KG-half
KV-Mem	21.1	14.8
GraftNet	32.8	26.1
PullNet	47.2	31.5
TransferNet	48.6	–
QA-Net (ours)	47.8	33.5

NVIDIA RTX8000. CWQ contains problems that require no more than four hops of reasoning and are of different types. We set the inference step to $T = 4$, and other settings are the same as those for WebQSP (Table 2 and 3).

4.4 Interpretability

Table 4. QA-Net ablation experiment; Renode denotes our processes at the node and Net denotes our neural network processes; “–” means no such module is available.

	WebQSP KG-full	WebQSP KG-half
QA-Net	74.5	59.0
QA-Net(-Renode)	74.1	56.8
QA-Net(-Net)	72.4	56.7
QA-Net(-Renode/-Net)	70.3	56.1

Table 4 shows the results of our ablation experiments on the modules of QA-Net. Each module has a positive contribution to the final results of the model. During the experiments we found that when we add reverse nodes to the list of preselected entities, the effect on the one-hop Q&A is very large, and the same

effect in the KG-half setting. At the same time, we find that when the knowledge graph becomes larger, the relevance score of nodes has a greater impact on the model, and it is worth considering how to better calculate the relevance between nodes.

4.5 Model Efficiency

From Fig. 3, we can see that on the WebQSP dataset, EmbedKGQA does not have a high model aggregation rate although it starts with a higher accuracy rate, instead our model has a better aggregation rate, while our model outperforms other baseline models with the same number of epochs.

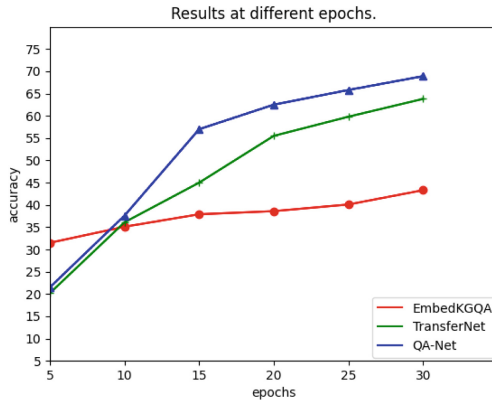


Fig. 3. Effect of different epoch numbers on model accuracy

5 Conclusion

In this paper, we propose an effective joint inference model QA-Net for knowledge graph Q&A, and we determine on how to form better subgraphs of related questions and how to combine questions and question subgraphs for reasoning. The problem embedding by pre-trained language models and TF-IDF embedding of entities are followed by joint reasoning over the problem subgraphs obtained from multilayer neural networks. The contributions of this paper are as follows: based on the Q&A problem and the knowledge graph entity relationship space, an attention-based Net module is designed for joint inference, while the information of inference in the previous step is retained by adding weights to the current inference process, and a new calculation function for node loss is designed. Our accuracy results on the WebQSP dataset surpasses the results of all baseline models to achieve state-of-the-art performance. Moreover, our model achieves good results on the CWQ dataset, proving the effectiveness of our method and illustrating the possibility of our model to be applied to larger knowledge bases.

Acknowledgements. This work is supported by National Nature Science Foundation of China (No.61762024, No.62062029) and Innovation Project of GUET Graduate Educatio (No.2022YCXS091)

References

1. Rajpurkar, P., Zhang, J., Lopyrev, K., et al.: Squad: 100,000+ questions for machine com-prehension of text. arXiv preprint [arXiv:1606.05250](https://arxiv.org/abs/1606.05250) (2016)
2. Zhao, W., Chung, T., Goyal, A., Metallinou, A.: Simple question answering with subgraph ranking and joint-scoring. arXiv preprint [arXiv:1904.04049](https://arxiv.org/abs/1904.04049) (2019)
3. Lan, Y., He, G., Jiang, J., et al.: A survey on complex knowledge base question answering: Methods, challenges and solutions. arXiv preprint [arXiv:2105.11644](https://arxiv.org/abs/2105.11644) (2021)
4. Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading Wikipedia to answer open-domain questions. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1870–1879, Vancouver, Canada. Association for Computational Linguistics (2017)
5. Jiang, K., Wu, D., Jiang, H.: FreebaseQA: a new factoid QA data set matching trivia-style question-answer pairs with freebase. 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), pp. 318–323, Minneapolis, Minnesota. Association for Computational Linguistics (2019)
6. Antoine, B., et al.: Large-scale simple question answering with memory networks. arXiv preprint [arXiv:1506.02075](https://arxiv.org/abs/1506.02075) (2015)
7. Yih, S.W., Chang, M.W., He, X., et al.: Semantic parsing via staged query graph generation: question answering with knowledge base. In: Proceedings of the Joint Conference of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing of the AFNLP (2015)
8. Devlin, J., Chang, M.W., Lee, K., et al.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
9. Liu, Z., et al.: A robustly optimized BERT pre-training approach with post-training. In: Proceedings of the 20th Chinese National Conference on Computational Linguistics (2021)
10. Kassner, N., Schütze, H.: Negated and misprimed probes for pre-trained language models: birds can talk, but cannot fly. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7811–7818. Association for Computational Linguistics (2020)
11. Feng, Y., Chen, X., Lin, B.Y., Wang, P., Yan, J., Ren, X.: Scalable multi-hop relational reasoning for knowledge-aware question answering. arXiv preprint [arXiv:2005.00646](https://arxiv.org/abs/2005.00646) (2020)
12. Yasunaga, M., Ren, H., Bosselut, A., et al.: QA-GNN: Reasoning with language models and knowledge graphs for question answering. arXiv preprint [arXiv:2104.06378](https://arxiv.org/abs/2104.06378) (2021)
13. Mitchell, T., et al.: Never-ending learning. *Commun. ACM* **61**(5), 103–115 (2018)
14. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledge-base. *Commun. ACM* **57**(10), 78–85 (2014)
15. Färber, M., et al.: Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. *Seman. Web* **9**(1), 77–129 (2018)

16. Sukhbaatar, S., Weston, J., Fergus, R.: End-to-end memory networks. In: *Advances in Neural Information Processing Systems*, vol. 28 (2015)
17. Martineau, J., Finin, T.: Delta tfidf: an improved feature space for sentiment analysis. In: *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 3. no. 1 (2009)
18. Yih, W., Richardson, M., Meek, C., et al.: The value of semantic parse labeling for knowledge base question answering. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 201–206 (2016)
19. Sun, H., Bedrax-Weiss, T., Cohen, W.W.: Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. arXiv preprint [arXiv:1904.09537](https://arxiv.org/abs/1904.09537) (2019)
20. Lin, B.Y., Chen, X., Chen, J., Ren, X.: KagNet: Knowledge-aware graph networks for commonsense reasoning. arXiv preprint [arXiv:1909.02151](https://arxiv.org/abs/1909.02151) (2019)
21. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic parsing on freebase from question-answer pairs. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1533–1544 (2013)
22. Saha, A., Ansari, G.A., Laddha, A., Sankaranarayanan, K., Chakrabarti, S.: Complex program induction for querying knowledge bases in the absence of gold programs. *Trans. Assoc. Comput. Linguist.* **7**, 185–200 (2019)
23. Zhang, Y., et al.: Variational reasoning for question answering with knowledge graph. In: *Thirty-second AAAI Conference on Artificial Intelligence* (2018)
24. Qiu, Y., Wang, Y., Jin, X., et al.: Stepwise reasoning for multi-relation question answering over knowledge graph with weak supervision. In: *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 474–482 (2020)
25. Zhou, M., Huang, M., Zhu, X.: An interpretable reasoning network for multi-relation question answering. arXiv preprint [arXiv:1801.04726](https://arxiv.org/abs/1801.04726) (2018)
26. Cohen, W.W., Sun, H., Hofer, R.A., Siegler, M.: Scalable neural methods for reasoning with a symbolic knowledge base. arXiv preprint [arXiv:2002.06115](https://arxiv.org/abs/2002.06115) (2020)
27. Lukovnikov, D., Fischer, A., Lehmann, J.: Pretrained transformers for simple question answering over knowledge graphs. In: Ghidini, C., et al. (eds.) *ISWC 2019*. LNCS, vol. 11778, pp. 470–486. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30793-6_27
28. Talmor, A., Berant, J.: The web as a knowledge-base for answering complex questions. arXiv preprint [arXiv:1803.06643](https://arxiv.org/abs/1803.06643) (2018)
29. Miller, A., Fisch, A., Dodge, J., et al.: Key-value memory networks for directly reading documents. arXiv preprint [arXiv:1606.03126](https://arxiv.org/abs/1606.03126) (2016)
30. Sun, H., Dhingra, B., Zaheer, M., et al.: Open domain question answering using early fusion of knowledge bases and text. arXiv preprint [arXiv:1809.00782](https://arxiv.org/abs/1809.00782) (2018)
31. Saxena, A., Tripathi, A., Talukdar, P.: Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4498–4507 (2020)
32. Shi, J., Cao, S., Hou, L., et al.: TransferNet: An effective and transparent frame-work for multi-hop question answering over relation graph. arXiv preprint [arXiv:2104.07302](https://arxiv.org/abs/2104.07302) (2021)