A Dynamic Answering Path Based Fusion Model for KGQA

Mingrong Tang[1], Haobo Xiong[1], Liping Wang[1(✉)], and Xuemin Lin[1,2]

[1] Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
{51184501148,51194501192}@stu.ecnu.edu.cn, lipingwang@sei.ecnu.edu.cn
[2] The University of New South Wales, Sydney, Australia
lxue@cse.unsw.edu.au

Abstract. The Knowledge Graph Question Answering (KGQA) task is useful for information retrieval systems, intelligent customer service systems, etc., which has attracted the attention of a large number of researchers. Although the performance of KGQA has been further improved by introducing the Deep Learning models, there are still some difficulties to be solved, such as the representation of questions and answers, the efficient construction way of candidate path set, etc. In this paper, we propose a complete approach for KGQA task. Firstly, we devise a novel candidate path generation process, which effectively improves computation performance by reducing the number of candidate paths corresponding to a question and at the same time guarantees the accuracy of results. Secondly, considering the textual expression diversity of questions and stochastic of candidate paths, we present four models to learn semantic features of Chinese sequence with different focuses. Finally, in order to combine the advantages of each presented model, we propose a dedicated fusion policy which can get the most suitable path from the path set predicted by our presented models. We conduct experiments on Chinese Knowledge Base Question Answering (CKBQA) dataset. Experiment results show that our approach achieves better performance than the best one published in CCKS2019 competition.

Keywords: Knowledge Graph Question Answering · Bert · Path generation · Model fusion

1 Introduction

With emergence of many high-quality and large-scale knowledge graphs, such as Freebase, Wikidata, YaGo, etc., the research about KGQA is becoming more and more popular. KGQA can return one or more nodes in the knowledge graph as correct answers for a question described by natural language, which can greatly reduce the difficulty of query and improve the usability of knowledge graph. Recently, KGQA has been widely applied in information retrieval systems, intelligent customer service systems, etc., which has become the main branch of study on knowledge graph.

G. Li et al. (Eds.): KSEM 2020, LNAI 12274, pp. 235–246, 2020.
https://doi.org/10.1007/978-3-030-55130-8_21

At present, the mainstream methods for KGQA can be grouped into the following two categories: semantic parsing-based methods and information retrieval-based methods. Semantic parsing-based methods [3,16] aim to convert a natural language question into an intermediate logical form, and then into a structured query statement that can be executed in the knowledge graph. Information retrieval-based methods [20] refer to extracting mentions in the question, linking mentions to corresponding topic entities in the knowledge graph, and then taking the connected entities as candidate answers, lastly ranking candidate answers for choosing the right answer. With the rapid development of Deep Learning, the neural network-based methods make the performance of the KGQA task get further improvement and no longer be limited by rules and templates. The neural network-based methods [1,2] mainly learn the semantic vectors of questions and candidate answers, and then model complex question-answering tasks into relatively simple similarity computation tasks. For those studies, researchers usually make an effort to effectively represent questions and candidate answers in a vector space. As a result, the technology of semantic information extraction and representation has been successively applied to KGQA including word embedding models such as word2vec [13] and Glove [14], and pre-trained language representation models such as GPT [15], Bert [4] and XLnet [21].

Challenges. For Knowledge Graph Question Answering task, there are two main challenges. One is how to efficiently construct a candidate answer set. When knowledge graph is relatively dense, there are lots of candidate paths related to the topic entity causing a lot of redundant calculations. As a result, it is difficult to compute the similarity between a question and all candidate paths. To solve above problem, the answer paths are usually limited to two hops in previous work. The other is how to effectively learn the representation of questions and candidate answers. To obtain more semantic information, most of the previous works represent answers from different aspects [8], but generally we can only represent answers from one aspect of answer path limited by the uncertainty of knowledge graph.

In current research, as the expression of questions and candidate paths is diverse, it's very difficult to learn all the semantic features between questions and paths through only one model. Moreover, for the dynamic candidate path generation process studied in this paper, there is a new challenge in the fusion processing. Specifically, for a question, the candidate path set from different models is different, therefore, we can't fuse results by calculating the average score of every candidate paths directly.

To handle above challenges, the corresponding solutions are proposed. Firstly, inspired by beam search [19] idea, we present a process for dynamically expanding candidate path set. In this way, path set is expanded hop by hop, and the number of candidate paths corresponding to a question is reduced effectively. Secondly, to extract semantic features of questions and candidate paths from different focuses, we design four models incorporating Bert including Basic global model, Path diversity model, Implicit relation model, and Path decomposition model. At last, we propose a suitable model fusion policy, which gets the candidate path with the highest score as the final right answer path by two steps including obtaining the top m paths of each model and calculating the average score of the candidate paths after filling in the missing values.

Contributions. Our contributions are summarized as follows:

1) We present a novel dynamic candidate path generation process named DPG, which iteratively generates candidate path hop by hop, and automatically terminate when postconditions are satisfied.
2) Considering the textual expression diversity of questions and stochastic of candidate paths, we present four models to learn semantic representations of Chinese sequence with different focuses including Basic global model, Path diversity model, Implicit relation model, and Path decomposition model.
3) To combine the advantages of each presented model, we propose a path similarity-based fusion policy which gets the most suitable path from the differentiated path set predicted by proposed models.
4) We construct extensive experiments and evaluate our approach on the CKBQA dataset. The experiment results demonstrate that our approach achieves better performance than the best one published in CCKS2019 competition.

2 Related Work

Semantic Parsing-Based Methods. [12] parses a natural language question into a new semantic representation form called DCS. This method can solve the problem of missing labeled data, but requires more professional linguistic knowledge and is difficult to generalize. [10] parses a question into a query subgraph, and then performs subgraph matching to obtain the correct answer. This method can understand certain implicit relationships but still needs to construct a mapping dictionary and constraint rules. With the popularity of the encoder-decoder models in the field of translation, these models are introduced to the question answering task which converts a question answering task to a translation task. Through the encoding and decoding model, [6] translates questions into logical forms that can be executed in the knowledge graph and effectively deal with the conversational question answering task. With the development of knowledge graph embedding technology, [18] introduces knowledge graph embedding technology to solve the problem of entity and relationship mapping and then construct graph-structured queries. In addition, [7] believes that improving the accuracy of the semantic parsing-based method should not be completely dependent on the design of algorithms and proposes a system named DialSQL, which can identify potential errors in a generated query and return simple multiple-choice Questions over multiple turns to users, and then use the user feedback to modify the query. In general, the semantic parsing-based method can resolve lots of simple and complex questions, but it involves much traditional linguistic knowledge, which is difficult to understand.

Information Retrieval-Based Methods. [22] firstly gets a topic entity in the question, then gets candidate answers related to the topic entity, and ranks them based on their relationship. In order to reason over knowledge graph, [23] proposes an end-to-end variational learning model that can handle noise in problems and learns multi-hop inference simultaneously. With the development of representation learning technology, more and more researchers use the method of vector modeling to deal with the question answering task. This method converts a question answering task into a similarity computation task between the question vector and the candidate answer vector. [5] uses a multi-column convolutional neural network to conduct distributed representation learning of questions and answers from three aspects including answer path, answer context, answer type so that the representation can contain more effective features than the previous. [17] uses tree-structured long short-term memory networks to learn the representation of sequences, which is helpful to analyze the implicit relationship and intent. Considering that most of the works pay more attention to the representation of the candidate answer end and ignore the question representation, [8] introduces an end-to-end neural network model to dynamically represent questions according to the diverse candidate answer aspects via cross-attention.

Based on information retrieval methods, we first locate a topic entity in the knowledge graph by named entity recognition and then dynamically generate and rank candidate paths hop by hop. To rank paths, we design four models which can learn from multi focuses. Finally, we fuse the outputs of our models and get the most suitable answer path.

3 Our Approach

Usually, the knowledge graph (KG) is regarded as a directed graph, in which the nodes are real-world entities and the edges are their relations. In KG, each directed edge, along with its head entity and tail entity, make up a triple, expressed as (h, r, t), which is also named as a fact. The knowledge graph question answering task (KGQA) aims to input a natural language question and then return an entity set A as right answers. The architecture of our approach for KGQA just as shown in Fig. 1. The proposed solution can be divided into three steps: candidate generation, similarity computation and results fusion. The following sections explain the details of each step.

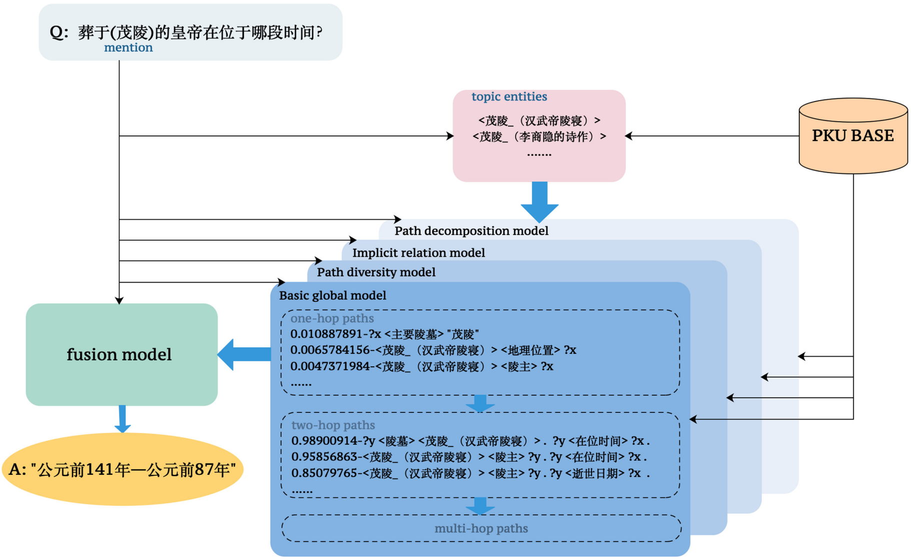


Fig. 1. The overview of the dynamic answering path based fusion model.

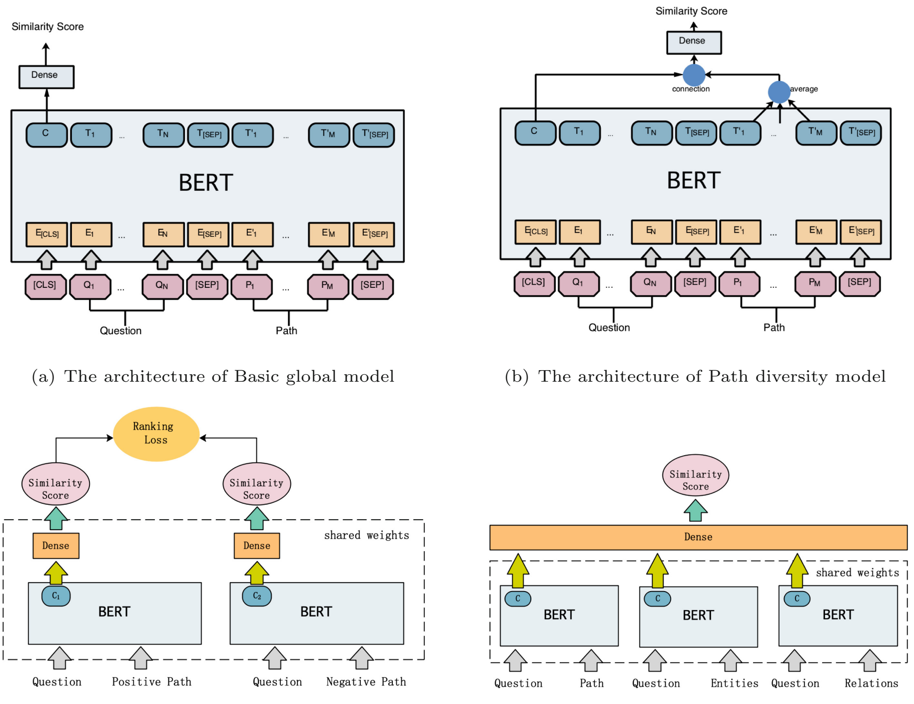
3.1 Candidate Generation

Ideally, we use all entities in the knowledge graph as candidate answers, and rank them. However, in practice, this is very time-consuming and unnecessary, so we develop a process for dynamic candidate path generation named DPG.

We first obtain a topic entity set E by named entity recognition and then build the candidate answer path set P according to the topic entities located in the knowledge graph. For the named entity recognition task, we first perform BIO sequence labeling by utilizing Bert, bidirectional LSTM [9] and CRF [11], and then link the identified mentions to topic entities through accurate matching and fuzzy matching. After obtaining the topic entity set, we dynamically generate candidate paths P in an expanded strategy. We first add candidate paths of one hop around the topic entity into set P, then calculate the similarity score. After that, we select top k paths in one-hop set to generate two-hop candidate paths and calculate their similarity score. By the same method, the hop of candidate path can be continuously increased, and the candidate path set P can be expanded by new generated paths. When hop count threshold is reached or scores of all new generated paths are no longer increased, the expansion process is terminated.

3.2 Similarity Computation Model

Considering the diversity of questions and candidate paths, we devise four models to compute similarity scores of questions and paths from different focuses. As a pre-trained language representation model, Bert achieves state-of-the-art results



(a) The architecture of Basic global model (b) The architecture of Path diversity model

(c) The architecture of Implicit relation model (d) The architecture of Path decomposition model

Fig. 2. The overview of our proposed models. (a) is Basic global model which takes into account the global representation of questions and paths. (b) is Path diversity model which pays more attention on the expression of paths. (c) is Implicit relation model which is used to extract implicit relation. (d) is Path decomposition model which can learn weights of entity and relation.

on many natural language processing tasks, including the similarity computation task. So we develop our similarity computation model based on Bert to handle the downstream question answering task.

Basic Global Model. This is the simplest similarity computation model based on Bert which takes into account the global representation of questions and paths. In this model, we only get the hidden representation vector from Bert corresponding to first input token ([CLS]) into next layer. The architecture of Basic global model is shown in Fig. 2(a). This model can handle simple questions with one-hop answer path like question "清华大学的校训是什么？", but limited by input information, it usually fails in some complex question with multi-hop answering path like question "维力医疗的高管中谁是复旦大学毕业的？".

Path Diversity Model. For a question with extra semantic content, the expression of paths is usually diverse. In some cases, the expression of answer paths is sequential such as path "< 维力医疗 > < 独立董事 > ?y . ?y < 上任时间 > ?x ."

for question “维力医疗的独立董事的上任时间？”, and in a few cases, the expression of answer paths can be reversed such as path “?y <陵墓><茂陵_（汉武帝陵寝）>. ?y <在位时间> ?x .” for question “葬于茂陵的皇帝在位于哪段时间？”. Generally, it is easy to deal with the first type of questions, but the second type of questions are difficult to answer. To deal with the complexity of answer path, we propose a new model named Path diversity model, in which we pay more attention to the expression of paths by feeding the hidden representation vector of path token into the next layer. We can see the detail in Fig. 2(b).

Implicit Relation Model. Implicit relation extraction has always been a challenge in KGQA tasks. For the question “《红楼梦》中贾琏妻子的丫鬟是谁？”, the right answer path is “?y <丈夫><贾琏_（小说《红楼梦》人物）>. ?y <丫鬟> ?x .”. In practice, the simple model is hard to give the relation “<丈夫>” a high score. For such complex questions, we propose a model named Implicit relation model by employing the triplet loss function presented below.

$$Loss = max(s(q,n) - s(q,p) + \gamma, 0) \tag{1}$$

where $s(q,p)$ calculates the similarity score between a question and a positive candidate. $s(q,n)$ calculates the similarity score between a question and a negative candidate. The γ is a positive number range from 0 to 1 that ensures a margin between positive candidates and negative candidates. With this loss function, the inputs of Implicit relation model are question, positive path, and negative path. The architecture of model is shown in Fig. 2(c).

Path Decomposition Model. Considering the distinct importance of entity and relation, we design a model named Path decomposition model to extract the similarity features of questions and paths, questions and entities, questions and relationships respectively, and then automatically learn their weights. The architecture of this model is shown in Fig. 2(d).

3.3 Results Fusion

The four models devised above cover the different conditions in KGQA tasks. To achieve the most suitable answer, a path similarity based fusion policy is proposed. Ideally, we expect to get four scores for a candidate path from four models and select the candidate path with the highest average score as the final right answer path. However, the candidate paths generated by our proposed process DPG are related to the requirement of similarity computation model. In other words, the top m paths are different for the same question under different models. To solve the above problem, we propose a novel solution for the best path decision.

Firstly, for each question, top m candidate paths in each model are selected, here m is a parameter. Secondly, for each candidate path absented in model i but recommended by other models, the prediction from model i to the candidate path is missing. We define Eq. 2 to count the missing scores which denotes evaluation score of model i to the candidate path.

$$Score = max(min(M_i, Q_j) * k_1 - k_2, 0) \tag{2}$$

where $min(M_i, Q_j)$ represents the minimum score in top m path set given by model i to question j, and k_1 and k_2 are adjustable parameters. Thirdly, after filling in the missing values, we calculate average score of each candidate path from different models. Finally, we use the candidate path with the highest average score as the final right answer path of the question.

4 Experiments

We use PKU BASE as our knowledge graph. It has 66,191,767 triples with 25,437,419 nodes and 408,261 relations. To evaluate our proposed models, we do experiments on Chinese Knowledge Base Question Answering (CKBQA) dataset which includes 2998 samples for training, 766 samples for validating, and 766 samples for testing. To compare with others, we use average F1 score as the final evaluation matrix.

4.1 Setting

In the candidate path generation step, considering that most of the right answer paths for questions in CKBQA dataset are in two hops, we set the hop threshold to 2. In the similarity computation step, we use Adam as our optimizer to minimize question-path pairs training loss. We set the batch size to 64, set the learning rate to 1e-5, and set the max length of model input sequences to 100. What's more, we early stop when the F1 score on the validation set reaches the maximum. Differently, the γ in loss function of Implicit relation model is set to 0.9. In the results fusion step, we set the parameter k_1 to 0.7 and k_2 to 0.2. We feed top 5 paths from each similarity model into the fusion model.

4.2 Results

The Effectiveness of Candidate Generation Process. In order to evaluate the effectiveness of our candidate generation process named DPG, we set the hop threshold to two and compare it with a commonly used fixed candidate path generation process named FPG. In general method FPG, we first construct a candidate set in which all paths are within two hops and include only one entity, then rank paths to select the correct one, and finally restrict the correct answer. In our method, we dynamically generate candidate paths, which can have multiple entities, and then rank the candidate paths to obtain correct path and correct answer without any constrains. We can see the results based on Basic global model in Table 1. The experiment results demonstrate that our process in generating candidate paths is effective. It reduces the number of candidate paths from 4081 to 1203 and increases the F1 score from 61.80% to 67.65%.

The Effectiveness of Fusion Policy. We conduct multiple experiments to evaluate the effectiveness of our fusion policy. What's more, in order to reduce the impact of mistake during named entity recognition stage, we also create an

Table 1. The evaluation of different candidate generation process.

Process	Avg count	Macro precision	Macro recall	Avg F1
FPG	4081	62.31	63.94	61.80
DPG	**1203**	**68.07**	**69.11**	**67.65**

Table 2. The evaluation of results fusion policy.

Model	Linking method	Macro precision	Macro recall	Avg F1
Basic global model	Ent linking	68.07	69.11	67.65
Path diversity model	Ent linking	64.13	64.75	63.63
Implicit relation model	Ent linking	61.97	61.82	61.20
Path decomposition model	Ent linking	65.37	65.33	64.56
Basic global model	Path linking	60.02	60.49	59.56
Fusion model	-	**75.21**	**76.13**	**74.63**

elastic search (ES) index on triple which mask head entity or tail entity to find one-hop paths. When we search a question on the index, we can obtain one-hop candidate paths easily. The next steps for expanding candidate path set P are the same as described above. We evaluate the performance of proposed approaches by employing the macro precision, macro recall, and average F1 evaluation matrix. The results are shown in Table 2.

In the table, the ent linking means that we get one-hop candidate paths by topic entities, and the path linking means that we get one-hop paths by searching for questions on the triple-based ES index. To make up for the error of the named entity recognition, we build a candidate path set by linking question to path and rank paths by the simplest Basic global model. In addition, we also list results with a variety of hit rates for reference, as shown in Fig. 3. From the results, we observe that the F1 score has been significantly improved by fusing the five result files, which demonstrates our fusion policy can combine the advantages of each model.

The Effectiveness of Our Proposed Approach. We conduct experiments on CKBQA dataset which comes from the CCKS2109 KGQA competition. We compare the final results with the top four teams in the competition to illustrate the effectiveness of our approach. We can see the results in Table 3.

The top four teams in the competition all fine-tune the pre-trained language model Bert on the KGQA task and achieve promised results. Differently, the rank one uses a dictionary to segment the question to recall candidate entities, then they extracted 39 features in the similarity computation stage. In addition, in order to reduce the variance, 10-fold cross-validation was employed. The system of the first team has achieved good results but is too cumbersome. Compared with the rank one, our approach achieves better performance and it is more concise and interpretable.

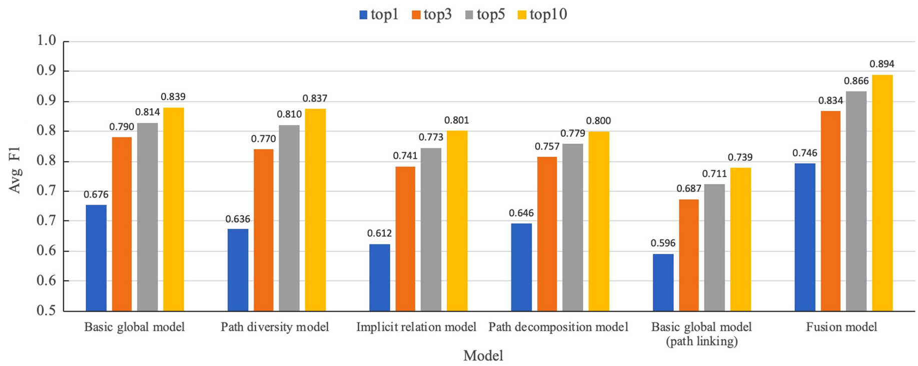


Fig. 3. The evaluation of model results with different statistical scope.

Table 3. The evaluation results on CKBQA dataset.

Teams	Avg F1
Rank 1	73.54
Rank 2	73.08
Rank 3	70.45
Rank 4	67.60
Our approach	**74.63**

5 Conclusion

In this paper, we propose a dynamic answering path based fusion model for the Knowledge Graph Question Answering task, and make a significant improvement on CKBQA dataset compared with representative ones. We propose a novel dynamic candidate path generation process which effectively reduces the number of candidate paths and makes the hop of answer paths not limited to a fixed number. Moreover, we design four independent models to learn the similarity representation of questions and candidate paths from different focuses. Lastly, the advantages of models are combined by an effective fusion policy which solves the problem of missing path scores caused by the dynamic path generation process. The experiment results demonstrate that our approach is effective at all stages of KGQA and we achieve better performance than the best one published in CCKS2019 competition.

Acknowledgment. This work was supported by project 2018AAA0102502.

References

1. 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), Doha, Qatar, pp. 615–620. Association for Computational Linguistics, October 2014
2. Bordes, A., Weston, J., Usunier, N.: Open question answering with weakly supervised embedding models. In: Proceedings of the 2014th European Conference on Machine Learning and Knowledge Discovery in Databases-Volume Part I, pp. 165–180 (2014)
3. Cai, Q., Yates, A.: Large-scale semantic parsing via schema matching and lexicon extension. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Sofia, Bulgaria, pp. 423–433. Association for Computational Linguistics, August 2013
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. 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. 4171–4186 (2019)
5. Dong, L., Wei, F., Zhou, M., Xu, K.: Question answering over Freebase with multicolumn 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), Beijing, China, pp. 260–269. Association for Computational Linguistics, July 2015
6. Guo, D., Tang, D., Duan, N., Zhou, M., Yin, J.: Dialog-to-action: conversational question answering over a large-scale knowledge base. In: NeurIPS, pp. 2946–2955 (2018)
7. Gur, I., Yavuz, S., Su, Y., Yan, X.: DialSQL: dialogue based structured query generation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers, Melbourne, Australia, pp. 1339–1349. Association for Computational Linguistics, July 2018
8. Hao, Y., et al.: 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), Vancouver, Canada, pp. 221–231. Association for Computational Linguistics, July 2017
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
10. Hu, S., Zou, L., Yu, J.X., Wang, H., Zhao, D.: Answering natural language questions by subgraph matching over knowledge graphs. IEEE Trans. Knowl. Data Eng. **30**(5), 824–837 (2018)
11. Lafferty, J., Mccallum, A., Pereira, F.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning, pp. 282–289 (2001)
12. Liang, P., Jordan, M., Klein, D.: Learning dependency-based compositional semantics. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Oregon, USA, pp. 590–599. Association for Computational Linguistics, June 2011
13. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)

14. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp. 1532–1543. Association for Computational Linguistics, October 2014
15. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
16. Reddy, S., Täckström, O., Collins, M., Kwiatkowski, T., Das, D., Steedman, M., Lapata, M.: Transforming dependency structures to logical forms for semantic parsing. Trans. Assoc. Comput. Linguist. **4**, 127–140 (2016)
17. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory 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), Beijing, China, pp. 1556–1566. Association for Computational Linguistics, July 2015
18. Wang, R., Wang, M., Liu, J., Chen, W., Cochez, M., Decker, S.: Leveraging knowledge graph embeddings for natural language question answering. In: Li, G., Yang, J., Gama, J., Natwichai, J., Tong, Y. (eds.) Database Systems for Advanced Applications, pp. 659–675. Springer International Publishing, Cham (2019)
19. Wiseman, S., Rush, A.M.: Sequence-to-sequence learning as beam-search optimization. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1296–1306 (2016)
20. Xu, K., Feng, Y., Reddy, S., Huang, S., Zhao, D.: Enhancing freebase question answering using textual evidence. CoRR abs/1603.00957 (2016). http://arxiv.org/abs/1603.00957
21. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: XLNET: generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems, pp. 5754–5764 (2019)
22. Yao, X., Van Durme, B.: Information extraction over structured data: question answering with Freebase. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Baltimore, Maryland, pp. 956–966. Association for Computational Linguistics, June 2014
23. Zhang, Y., Dai, H., Kozareva, Z., Smola, A.J., Song, L.: Variational reasoning for question answering with knowledge graph. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)