A Subgraph Retrieval Method for Complex Questions Based on Hybrid Semantics and Path Representation

Jifei Hao[1*], and *Bo* Cheng[2]

[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Beijing University of Posts and Telecommunications, Beijing, China

Abstract. Current subgraph retrieval methods generally fall into two categories: those that rely on semantic matching, which use only surface-level semantic information of relations and lack flexibility; and those based on personalized PageRank algorithms, which fail to leverage the semantic connection between the relation and the question, rendering them susceptible to noisy data. To address these issues, this paper introduces a novel retrieval model that employs hybrid semantics of relations and path representations. Specifically, hybrid semantics involves merging relational and entity information within a knowledge graph to extract the deep semantics of relations and enhance semantic representation by integrating it with the explicit descriptive text of the relations. Path representation merges the semantics of the current relation with those of preceding ones to form a complete path representation. This representation is then semantically matched with the question to compute a score, which determines whether the relation should form part of the subgraph. We integrated our subgraph retrieval model with the Neural Symbolic Machine (NSM) reasoning model and evaluated it on the publicly available CWQ and WebQSP datasets. The experimental results demonstrate that our method performs exceptionally well on these datasets, validating the efficacy of utilizing deep semantics and path representations for the retrieval of subgraphs in response to complex questions.

1 INTRODUCTION

Knowledge Base Question Answering [1] aims to find answers to natural language questions from structured knowledge bases, such as Freebase [2] and DBPedia [3]. Earlier research has been more oriented towards simple questions [4–6] (for example, "Which country was Yao Ming born in?"), where the answer can be retrieved through a single hop ("Yao Ming, born, China"). Compared to such simple questions, complex questions provide a greater scope for research. These types of questions typically require starting from a topic entity and traversing paths of two or more hops to deeply mine the multiple, nuanced relationships within the knowledge graph to match the target node.

In the realm of complex Knowledge Base Question Answering (KBQA), prevalent strategies encompass semantic parsing-based (SP-based) techniques [7–9] and information retrieval-based (IRbased) approaches [10–12]. The former involves converting natural language questions into logical expressions, such as SPARQL, and retrieving the answers. This method requires domain experts to spend a significant amount of time and effort to complete the annotation of intermediate logical forms. Therefore, we have focused our attention on the IR-based methods [10–12], which involve a two-step process: subgraph retrieval and answer reasoning. First, a smaller subgraph relevant to the question is generated, and then easoning is performed on this subgraph to obtain the final answer. The quality of the subgraph is crucial, as a smaller subgraph may exclude the correct answer, while a larger subgraph could introduce too many noise nodes.

Existing subgraph retrieval algorithms can be categorized into two types: Personalized PageRank (PPR) [13] and semantic matching[11]. The PageRank algorithm assesses the importance of nodes based on link structure, but it performs poorly when dealing with largescale graphs that are sparse and have low connectivity. To address this, the PPR algorithm incorporates user historical behavior and preferences to achieve a personalized ranking of retrieval results. Some current work [12, 14] directly applies the PPR algorithm for subgraph pruning without utilizing the semantic information of relationships and questions, making it susceptible to the influence of noise nodes. Semantic matching algorithms decide whether to include a current relation in the subgraph by performing a semantic match between the relation and the question.Sun et al. [11] introduced an innovative subgraph retriever intricately interwoven with the reasoning process. This retriever dynamically chooses new relations that are contextually relevant to the question at each step. Subsequently, the reasoner determines the entities within these new relations that warrant extension into the evolving subgraph. However, this method only utilizes the

[*] Corresponding author: 1036705119@bupt.edu.cn

surface meaning of relations and does not fully excavate the hidden meanings of relations, such as how the same relation can often have different meanings in different contexts. In addition, the approach only performs semantic matching of the question with each relation, ignoring the semantics of the path.

To address the aforementioned issues, we propose a novel trainable subgraph retrieval model that utilizes the mixed semantics and path information of relations to more fully leverage relational and path information. Firstly, we will utilize the head and tail entities of a relation for textual concatenation with the relation itself, for example, "person.born.country", and input it into a language model to mine the deeper semantics of the relation. Then, we prepare explicit descriptive text for each relation, such as: "Awarded: Granted an award or honor for a particular achievement, performance, or contribution. It is usually a recognition of an individual or group’s talent or effort." This is semantically integrated with the deep semantics of the relation to obtain the mixed semantics of the relation, making full use of the semantic information of the relation. Finally, we integrate the semantics of the current relation with its preceding relations to construct a path representation. This path representation is used to perform semantic matching with the question and to calculate scores. The relations with the top-k scores are added to the subgraph, completing the subgraph construction. Our contributions are as follows:

- We propose a trainable subgraph retrieval model that fully mines the semantic information of relations and utilizes path semantics for semantic matching.
- The subgraphs generated by our retrieval model have a higher probability of containing the answer and a higher answer coverage rate while ensuring a smaller size.
- We combined this subgraph retrieval model with the NSM reasoner and achieved good performance.

2 RELATED WORKS

The current mainstream technologies for complex KBQA (Knowledge Base Question Answering) are divided into methods based on SP (Semantic Parsing) and those based on IR (Information Retrieval). The former [15–19] involves transforming natural language questions into a structured query language like SQL, such as SPARQL, and retrieving answers. This method requires domain experts to expend a considerable amount of time and energy on annotating the intermediate logical forms. Hence, we have focused our attention on information retrieval-based approaches. A current challenge faced by this method is how to obtain subgraphs with high answer coverage and existence rates while ensuring that the subgraphs are kept small.

IR-based methods [4, 10–13, 20, 21] aim to obtain smaller subgraphs related to the question and perform reasoning on these subgraphs to arrive at the final answer. Subgraph retrieval is a core step since smaller subgraphs might exclude the correct answer, while larger subgraphs may introduce too many noise nodes. Current subgraph retrieval algorithms are divided into those based on a personalized PageRank algorithm (PPR) [13] and semantic matching algorithms. PPR algorithms are often applied directly to subgraph pruning, but their failure to utilize the semantic information of relations and questions makes them vulnerable to noise nodes. Semantic matching algorithms, such as PullNET [11], SRN [22], IRN [23], and UHop [24], select new relations relevant to the question at each step, with the reasoner then determining which entities in these new relations should be extended into the subgraph. However, they only utilize the surface meaning of relations, and their methods of semantic matching are somewhat unreasonable. Moreover, these subgraph retrieval models are intertwined with the reasoning process, which makes it difficult to integrate with complex reasoners [25] to better answer complex questions.

3 METHOD

During the process of knowledge graph question answering, it is necessary to build a relevant subgraph for each question and reason over this subgraph to obtain the answer. To further improve the performance of the generated subgraphs, we propose a trainable subgraph retrieval model, which is divided into two steps: model training and subgraph generation. The overall process is illustrated in Fig. 1.

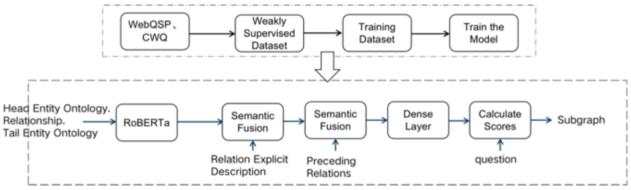

Fig. 1. Subgraph retrieval overall process.

3.1 Model Training

The data in the publicly available datasets CWQ [26] and WebQSP[17] exist in the form of question-answer pairs, leading to a lack of effective supervision signals when training subgraphs. To address this, for each question, we use the shortest path from the topic entity to the answer as a weak supervision signal because the shortest path is easier to obtain compared to the subgraph. At the same time, to further expand the volume of weak supervision data, we decompose each path $□ = (□_1, □_2, \ldots, □_□)$ into n (question, relation) examples. For example, $(□, [□_1]), (□, [□_1; □_2]) \ldots (□, [□_1; □_2; \ldots; □_□])$. After preparing the weak supervision signals, we constructed the training dataset with a negative sampling rate of 1:15. It is important to note that during the training process, we tried various loss functions and ultimately decided to use the cross-entropy loss function for model training. Its calculation formula is as follows.

$$loss(x, class) = -x[class] + \left(\log(\sum_j e^{x[j]}) \right) \quad (1)$$

Here, ⍰ represents the array of results output by the model, and ⍰⍰⍰⍰⍰ is the index where the positive sample is located, with all other elements being negative samples. The purpose of this formula is to make the score of the positive sample as high as possible and the score of the negative samples as low as possible, aligning with our training expectations.

3.2 Subgraph Generation

Firstly, to address the issue that existing subgraph retrieval algorithms do not fully exploit the semantic information of relations, we concatenate the ontology of head and tail entities with the relation into text and input it into a language model to mine the deep meanings of the relation in different contexts. For example, for the triple (Yao Ming, born, China), we obtain the ontologies of the head and tail entities of the relation "born", which are "person" and "country", and concatenate the texts to form "person.born.country". We then input this into a pre-trained model to obtain its embedding vector.

$$\mathbf{r}_{ij} = RoBERTa\left(h_i . r_{ij} . t_j\right) \tag{2}$$

Here, h_i refers to the ontology of the head entity, r_{ij} refers to the current relation, and t_j refers to the ontology of the tail entity. The output $\mathbf{r}_{ij}$ is the embedding vector representation of the relation.

Existing work[11, 14, 27] has demonstrated the advantages of introducing explicit descriptive text for entities. We hypothesize that introducing explicit descriptive text for relations and semantically enhancing them can also aid in subgraph retrieval. Therefore, we have prepared explicit descriptive texts for the relations involved in the retrieval process. For example, "awarded: to be given a prize or honor for some achievement, performance, or contribution. It is usually a recognition of an individual's or group's talent or effort." We then input this description into a language model to obtain its embedding vector representation. Finally, we merge it with the hidden meaning of the relation for semantic fusion, obtaining a hybrid semantic representation of the relation. The computation formula is as follows.

$$\mathbf{h}_{ij} = \left[RoBERTa(desc); \mathbf{r}_{ij}\right] \tag{3}$$

Here, $\mathbf{r}_{ij}$ represents the implicit semantic meaning of the relation in context, $desc$ is the explicit descriptive text of the relation, and $\mathbf{h}_{ij}$ represents the hybrid semantic meaning of the relation. We fuse the embedding vectors of both to fully utilize the semantic information of the relation.

Existing research typically involves semantic matching between the current relation and the question and calculating a score to decide whether to include the relation in the subgraph. However, we believe that the decision to add a current relation to the subgraph should be based more reasonably on the degree of match between the overall path information, after adding the current relation, and the question, rather than just the semantic match between the current relation and the question. For this purpose, we have fused the hybrid semantics of the current relation with the hybrid semantics of its preceding relations to construct a semantic representation of the overall path.

$$\mathbf{p} = \left[\mathbf{h}_{t(i-1)}; \mathbf{h}_{ij}\right] \tag{4}$$

Here, $\mathbf{h}_{t(i-1)}$ represents the hybrid semantic meaning of the preceding relation, $\mathbf{h}_{ij}$ represents the hybrid semantic meaning of the current relation, and the output p represents the semantic meaning of the path.

It should be noted that when obtaining the hybrid semantics of relationships and the path representation, the dimensionality of the embedding vectors keeps increasing. To address this, we introduced a fully connected layer to perform dimensionality reduction on the path semantics. This allows for a more compact and manageable representation that can be more effectively used in subsequent similarity calculations with the question embedding.

$$\mathbf{p} = ReLu\left(\mathbf{W}_r(\mathbf{p})\right) \tag{5}$$

In the end, we calculate the cosine similarity between the current path representation and the embedding vector of the question to obtain the overall path score after the inclusion of the current relation. We then select the paths with the highest scores to add to the subgraph, thus completing the subgraph retrieval for the current hop.

$$s = \mathbf{Q}^T \mathbf{p} \tag{6}$$

In this process, $\mathbf{Q}^T$ represents the embedding vector of the question, and the output s is the score for the current path. This score is used to determine how well the path aligns with the question, with higher scores indicating a better match. Paths with the highest scores are considered most relevant to the question and are therefore selected for inclusion in the subgraph.

4 EXPERIMENTS

In this phase, we conducted numerous experiments to assess our subgraph retrieval model. We designed a variety of experiments to verify three key questions: (1) Whether our subgraph retrieval model produces smaller and higher quality subgraphs. (2) Whether our model improves the final QA (Question Answering) performance. (3) Whether each intermediate step in the model is effective.

4.1 Experimental Settings

Datasets. To evaluate our model, we conduct experiments on publicly available datasets: WebQSP [17] and CWQ[26]. The WebQSP dataset contains a large number of relatively simple questions, while the CWQ dataset is composed of complex questions. Specific data for these datasets are presented in Table 1.

Table 1. Dataset information.

Dataset	Train	Validation	Test
CWQ	27639	3519	3531
WebQSP	2848	250	1639

Evaluation Metrics. When evaluating the performance of the subgraph retrieval model, we are particularly concerned with the Answer Existence Rate (ER) and Answer Coverage Rate (CR) of the subgraphs. The Answer Existence Rate refers to the ratio of the number of subgraphs that contain at least one answer to the total number of subgraphs, which indicates the probability that at least one correct answer exists in each subgraph and can be used to represent the accuracy of the retrieval. The Answer Coverage Rate refers to the ratio of the number of answer entities in the subgraph to the total number of entities, which measures whether the subgraph covers all possible answers to the question, thereby assessing the completeness of the information. Combined, these two metrics can be used to evaluate the accuracy and completeness of the subgraph.

$$CR = \frac{1}{D}\sum_{x=1}^{D}\frac{M_A^X}{N_A^X}, ER = \frac{D_E}{D} \tag{7}$$

D and D_E represent the total number of subgraphs and the number of subgraphs containing the answer, respectively, while N_A^X and M_A^X respectively indicate the number of answer entities in the set for the x sample and the corresponding number of answer entities contained in the subgraph.

When evaluating the final retrieval performance, we are particularly concerned with the values of hits@1 and F1. Hits@1 refers to the hit rate of the top scoring node from the inference model, and F1 refers to the harmonic mean of precision and recall.

$$r = \frac{a}{c}, p = \frac{c}{n}, F1 = \frac{2}{\frac{1}{p} + \frac{1}{r}} \tag{8}$$

Here, a represents the actual number of answers, n is the number of candidate nodes output by the graph retrieval model, and c is the number of correct answers output by the model. r and p represent precision and recall, respectively.

Baseline. When evaluating the performance of subgraph retrieval, since subgraph retrieval algorithms based on semantic matching are all intertwined with the inference process, we only compare our subgraph retrieval model with subgraphs generated by the NSM+h[12] algorithm and the original PPR[14] algorithm's subgraph retrieval model, to verify that our model contains more answer nodes while being smaller. When evaluating the final question answering performance, we combine the subgraph retrieval model with the NSM inference model. We conducted comparative experiments with IR-based algorithms such as KV-Mem [20], GraftNet [14], EmbedKGQA [28], NSM [12], and PullNet [11].

4.2 Retriever Evaluation

To verify that our subgraph retrieval model includes more answers while ensuring the subgraphs generated are smaller, we conducted comparative experiments with the PPR algorithm [14] on the publicly available CWQ and WebQSP datasets. During the experiments, we continually increased the size of the subgraph to find the optimal answer coverage rate.

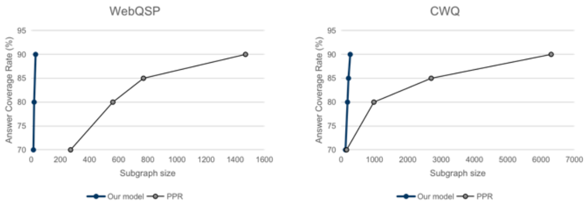

Fig. 2. The comparison results of answer coverage rates.

In Fig. 2, the horizontal axis represents the subgraph size, and the vertical axis represents the answer coverage rate. As can be seen from the graph, with the same subgraph size, our subgraph retrieval model has a higher answer coverage rate and includes more answers. The fundamental reason for this is that the PPR algorithm does not make use of the semantic information of questions and relations, resulting in the generated subgraph nodes having a degree of randomness. Consequently, it is necessary to continuously enlarge the subgraph to increase the answer coverage rate.

For a more comprehensive assessment of subgraph retrieval performance, we extended our comparison to include the Practical PageRank (PPR) algorithm [12, 14], evaluating both answer existence rates and coverage rates on the CWQ datasets.

Table 2. Experimental results of subgraph retrieval model.

Model	ER	CR
our subgraph	**89.7%**	**89.9%**
Raw PPR	77.6%	75.5%
NSM+h subgraph	79.3%	77.5%

From Table 2, it can be seen that our subgraph retrieval model has improved the answer existence rate by 10.4% and the answer coverage rate by 12.4% compared to the existing PPR algorithm, resulting in a significant enhancement in performance.

4.3 Overall QA Evaluation

To validate the performance improvement of our proposed subgraph retrieval model on the final QA, we combined the subgraph retrieval model with the NSM inference model and conducted comparative experiments on the publicly available CWQ and WebQSP datasets. When verifying the results, we are more concerned with the Hits@1 and F1 scores.

Table 3. Final QA performance.

Model	WebQSP		CWQ	
	Hits@1	F1	Hits@1	F1
KV-Mem[20]	46.6%	34.5%	18.4%	15.7 %
GRAF-Net[14]	66.4%	60.4%	%	32.7 %
Embed KGQA[28]	66.6%	-	-	-
NSM[12]	68.5%	62.8%	46.3%	42.4 %
PullNet[11]	68.1%	-	45.9%	-
Our subgraph model + NSM	**69.8%**	**64.6%**	**49.3%**	**45.8 %**

From Table 3, we can see that there is a significant improvement in the final retrieval performance on the CWQ dataset, with the Hits@1 value and F1 value increasing by 3% and 3.4% respectively. There is a smaller performance improvement on the WebQSP dataset, with the Hits@1 value and F1 value increasing by 1.3% and 1.8% respectively. This is because existing algorithms already perform quite well on simple questions and are mostly hindered by complex question answering, whereas our research is more oriented towards solving complex problems.

4.4 Ablation Experiment

In order to verify the effectiveness of each intermediate link in the model, we use the complete subgraph retrieval model combined with the NSM inference model as the baseline. Experiments are conducted by removing the path representation module, the hybrid semantic module, and the hidden semantic module, respectively.

Table 4. The results of ablation experiment.

Model	Hits@1	F1
Our subgraph + NSM(baseline)	**49.3%**	**45.8%**
w/o path representation	48.9%	45.1%
w/o hybrid semantics	48.4%	44.2%
w/o hidden semantics	46.7%	43.5%

From Table 4, we can see the effectiveness of each component of our model. Among them, the module for mining hidden semantics of relationships has the most significant impact on performance improvement, increasing the Hits@1 value by 2.6% and the F1 value by 2.3%. The path representation module has a relatively smaller impact on performance improvement, increasing the Hits@1 value by 0.4% and the F1 value by 0.7%.

5 CONCLUSION

In this paper, we proposed a novel subgraph retrieval algorithm for complex Knowledge Base Question Answering (KBQA) to address the limitations of existing methods. Our approach utilizes hybrid semantics of relations and path representations to enhance the quality of the retrieved subgraphs. We introduced explicit descriptive text for relations, mined deep semantics using language models, and integrated semantic information within paths for better subgraph construction.

The experimental results on the CWQ and WebQSP datasets demonstrated the effectiveness of our subgraph retrieval model. The model consistently outperformed existing methods in terms of answer existence rate and coverage rate, generating smaller yet more informative subgraphs. When integrated with the Neural Symbolic Machine (NSM) reasoning model, our approach achieved significant improvements in Hits@1 and F1 scores, showcasing its effectiveness in enhancing the overall question-answering performance.

Ablation studies further validated the importance of each component in our model, with the hidden semantics module contributing significantly to the performance boost. The proposed algorithm provides a valuable contribution to the field of complex KBQA by efficiently retrieving subgraphs that contain relevant information for answering intricate questions.

In conclusion, our trainable subgraph retrieval model, utilizing hybrid semantics and path representations, demonstrates superior performance in constructing informative subgraphs for complex KBQA. The integration of this model with reasoning systems holds promise for advancing the state-of-the-art in question-answering over knowledge graphs. Future work may explore extensions of our approach to handle diverse and evolving knowledge graphs, ensuring adaptability to a wide range of real-world applications.

This work was supported in part by the National Natural Science Foundation of China under grant U22A201339623720586192100З, 61972043, the Zhejiang Lab under grant 2021PD0AB02, the Fundamental Research Funds for the Central Universities under Grant 2020XD-A07-1.

References

1. C. Unger, A. Freitas, and P. Cimiano. An introduction to question answering over linked data. In *Reasoning Web*, 2014.
2. K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Sigmod Conference, 2008.*
3. J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6:167–195, 2015.

4. L. Dong, F. Wei, M. Zhou, and K. Xu. Question answering over freebase with multi-column convolutional neural networks. In *ACL*, 2015.

5. S. Hu, L. Zou, J. X. Yu, H. Wang, and D. Zhao. Answering natural language questions by subgraph matching over knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, 30:824–837, 2018.

6. Y. Lan, S. Wang, and J. Jiang. Knowledge base question answering with a matching-aggregation model and question-specific contextual relations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27:1629–1638, 2019.

7. C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. 2020.

8. Y. Lan and J. Jiang. Query graph generation for answering multi-hop complex questions from knowledge bases. In *Meeting of the Association for Computational Linguistics*, 2020.

9. R. Das, M. Zaheer, D. N. Thai, A. Godbole, E. Perez, J. Y. Lee, L. Tan, L. Polymenakos, and A. McCallum. Case-based reasoning for natural language queries over knowledge bases. In *EMNLP*, 2021.

10. Y. Lan, S. Wang, and J. Jiang. Knowledge base question answering with topic units. In *Twenty-Eighth International Joint Conference on Artificial Intelligence IJCAI-19*, 2019.

11. H. Sun, T. Bedrax-Weiss, and W. W. Cohen. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. 2019.

12. G. He, Y. Lan, J. Jiang, W. X. Zhao, and J. R. Wen. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. 2021.

13. T. H. Haveliwala. Topic-sensitive pagerank: a context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796, 2003.

14. H. Sun, B. Dhingra, M. Zaheer, K. Mazaitis, R. Salakhutdinov, and W. W. Cohen. 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*, 2018.

15. J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of EMNLP*, 2013.

16. L. Chen, J. Berant, Q. Le, K. D. Forbus, and L. Ni. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, 2017.

17. W. t. Yih, M. Richardson, C. Meek, M.-W. Chang, and J. Suh. 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)*, 2016

18. W. T. Yih, M. Richardson, C. Meek, M. W. Chang, and J. Suh. The value of semantic parse labeling for knowledge base question answering. *In Meeting of the Association for Computational Linguistics*, 2016.

19. X. Ye, S. Yavuz, K. Hashimoto, Y. Zhou, and C. Xiong. Rng-kbqa: Generation augmented iterative ranking for knowledge base question answering. In *ACL*, 2022.

20. A. Miller, A. Fisch, J. Dodge, A. H. Karimi, A. Bordes, and J. Weston. Key-value memory networks for directly reading documents. In *Conference on Empirical Methods in Natural Language Processing*, 2016.

21. K. Xu, S. Reddy, Y. Feng, S. Huang, and D. Zhao. Question answering on freebase via relation extraction and textual evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.

22. Yunqi Qiu, Yuanzhuo Wang, Xiaolong Jin, and Kun Zhang. 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*, page 474–482, 2020.

23. Mantong Zhou, Minlie Huang, and Xiaoyan Zhu. An interpretable reasoning network for multirelation question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*, page 2010–2022, 2018.

24. Zi-Yuan Chen, Chih-Hung Chang, Yi-Pei Chen, Jijnasa Nayak, and Lun-Wei Ku. Uhop: An unrestricted-hop relation extraction framework for knowledge-based question answering. In *Proceedings of NAACL-HLT*, page 345–356, 2019.

25. Yu Feng, Jing Zhang, Gaole He, Wayne Xin Zhao, Lemao Liu, Quan Liu, Cuiping Li, and Hong Chen. A pretraining numerical reasoning model for ordinal constrained question answering on knowledge base. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, page 1852–1861, 2021.

26. A. Talmor and J. Berant. 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)*, 2018.

27. Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. Improving question answering over incomplete kbs with knowledge-aware reader. In *ACL*, 2019.

28. Yu Chen, Lingfei Wu, and Mohammed J Zaki. Bidirectional attentive memory networks for

question answering over knowledge bases. In *Proceedings of NAACLHLT*, page 2913–2923, 2019.