CoopKG: An Academic Knowledge Graph for Question Answering Systems

Muyuan Niu and Cong Tian(✉)

School of Computer Science and Technology, Xidian University, Xi'an, China
mynew@stu.xidian.edu.cn, ctian@mail.xidian.edu.cn,

Abstract. The exponential growth of academic information on the Internet has created a pressing need for efficient methods to query, analyze, and reveal its potential value. However, the challenges in collecting, organizing and effectively utilizing the data remain significant obstacles. We construct CoopKG, a new academic knowledge graph (KG) that integrates extensive data on academic papers, researchers, and research projects. CoopKG is stored in a Neo4j database and is linked to comprehensive profile documents and portrait images of researchers. To fully leverage CoopKG, we develop a Knowledge Graph Question Answering (KGQA) system that utilizes innovative techniques with large language models (LLMs) to accurately convert a natural language question into Cypher Query Language (Text-to-CQL) for retrieving relevant information, thereby generating highly readable answers. Additionally, we integrate extensions such as graph visualization, data export, online search, and AI-assisted reading to enhance the system's usability. We conduct extensive experiments on open-source LLMs, demonstrating that on the Text-to-CQL task, our method can improve the logical accuracy by up to 5.71% and the execution accuracy by up to 5.29% on fine-tuned models.

Keywords: Knowledge Graph · Question Answering · Large Language Model

1 Introduction

Currently, Knowledge Graph Question Answering (KGQA) systems have undergone substantial development, aiding users in discovering and retrieving knowledge from large-scale structured graph data. However, research on KGQA systems in the academic domain remains underdeveloped. Existing academic KGs like Microsoft Academic Graph (MAG) [11], Aminer [12], and AceKG [13] focus on paper-related data but neglect researcher profiles and project information, restricting collaboration discovery. Furthermore, these KGs lack Chinese data and robust, user-friendly question-answering systems. MAG uses Bing Dialog for query processing but does not support natural language queries. AceKG enhances multi-hop question answering through KG embeddings; however, frequent data updates result in significant training costs. Aminer offers a natural language interface, but struggles with name abbreviation errors. This study introduces CoopKG, a Chinese academic knowledge graph that incorporates information on articles, researchers, and extensive project data sourced from multiple publicly available academic websites. CoopKG enhances diversity through comprehensive researcher profiles and portrait images. To facilitate efficient natural language querying, we develop a new KGQA system for CoopKG.

KGQA research currently centers on two core issues: knowledge retrieval [16] and semantic parsing [1]. Knowledge retrieval focuses on identifying the most relevant entities, relationships, or triples in the KG based on questions. Previous work aligns natural language questions with KG through named entity recognition(NER) [4], entity linking [6], or subgraph retrieval [18]. However, as the scale of the KG increases, the efficiency of retrieval noticeably decreases. Semantic parsing converts natural language questions into graph database queries for information retrieval. For example, data sets such as SpCQL [2], which maps natural language questions to CQL, have been extensively developed to enhance the capabilities of LLMs through fine-tuning. Other research generates a logical form, which is then converted into a SPARQL [8] query to execute in the KG. Despite these advantages, semantic parsing still faces challenges, including limited research on Chinese language parsing, a lack of effective integration with LLMs, and the increased complexity introduced by the generation of intermediate logical forms.

To overcome these challenges, we develop a KGQA system using open-source LLMs to query and analyze CoopKG. By combining manually crafted and automatically generated training samples, we reduce manual effort while improving sample diversity. We propose a verification and rewriting method that, combined with fine-tuned LLMs, generates CQL for execution in a graph database to retrieve information, thereby improving accuracy and reducing system complexity. We also handle entity alignment between the generated CQL and CoopKG while integrating several system extensions. To validate the effectiveness of our approach, we conduct extensive experiments on LLMs, such as ChatGLM3-6B [17] and Baichuan2-7B-Chat [15]. The results demonstrate that our method significantly improves system performance in the Text-to-CQL task. In summary, this paper makes the following contributions.

- We construct CoopKG, which offers more extensive content than other academic knowledge graphs by integrating researchers' profile documents, portrait images, and detailed information on their project involvements.
- We present a KGQA system that integrates a novel Text-to-CQL method, enabling efficient natural language question-answering on CoopKG. The system includes a user-friendly interface with extensions such as graph visualization, data export, online search, and AI-assisted reading.
- We evaluate the effectiveness of our training data generation method for Text-to-CQL tasks on open-source lightweight LLMs. By incorporating our validation and rewriting method, the experimental results show a further improvement in performance.

2 Academic Knowledge Graph

2.1 Properties of CoopKG

This section provides an overview of CoopKG, focusing on its key properties.

Scale. We gather information on 133,335 researchers from Chinese universities and collected data on 117,501 projects in the past five years. Furthermore, we collect 10,776,725 paper records from various academic platforms. The graph contains 69,216,382 relationships. In addition, we collect and organize 109,306 researcher profile documents and 68,334 portrait images.

Structure. CoopKG consists of three types of nodes: Researcher, Project, and Paper. These nodes represent researchers, projects, and papers, respectively, each with its associated attributes. Additionally, relationships among the different node types are established based on their interconnections. The detailed structure of CoopKG is illustrated in Figure 1.

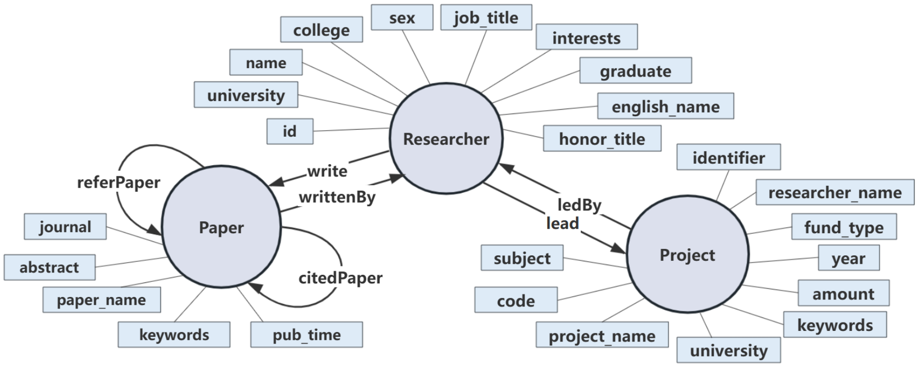


Fig. 1. Overview of CoopKG Structure

Comprehensiveness. To ensure the precision of CoopKG, we collect faculty information from the official websites of various universities in 71 universities. For research papers, we collect data from MAG, the DBLP database [5], and Baidu Scholar. In addition, to improve the completeness and accuracy of the information on projects, we collect data from multiple Chinese academic websites. By combining automated scripts with manual verification, we fill in most missing fields, thus improving data integrity.

Diversity. We clean and organize the personal profile webpages of the collected researchers, storing the information as profile documents. In addition, we download and store publicly available portrait images of researchers. These files are linked to the Researcher node ID attribute in CoopKG. Furthermore, the nodes in CoopKG contain comprehensive attributes, facilitating user queries and analyses.

2.2 Data Collection and Cleaning

We collect researcher information from university websites, including the name, affiliated college, personal homepage, and images. We then download the homepage content and store it as profile documents. The paper data are gathered from MAG, DBLP, and Baidu Scholar. We also collect citation references and associate them with the researchers. The project data are gathered from Chinese academic websites, organized by university, and linked to researcher information. To process the profile documents, we identify and remove HTML tags and code using scripts. Three experts spend 45 days reviewing these documents with the assistance of LLMs, identifying duplicated information and garbled text. After sorting the files by size and discarding those that are too small or lack meaningful content, we retain a total of 109,306 profile documents. Additionally, we clean the researcher profile images, resulting in a total of 68,334 portrait images.

3 KGQA System

3.1 Text-to-CQL Model Fine-Tuning

We use open source lightweight LLMs, which require fewer computational resources and are more conducive to optimization compared to large-scale models like ChatGPT [7]. However, achieving optimal performance with these models is challenging through prompt engineering [14] alone, making effective fine-tuning essential.

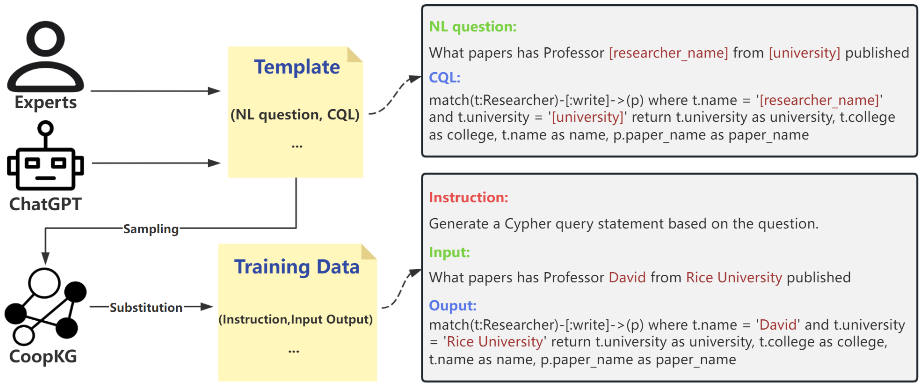
We employ a template-based approach to generate training data, with templates created both manually and automatically. The expert-crafted templates are based on (natural language question, CQL) pairs and cover 70 question types, with 5 examples for each type. Task descriptions are then added to these templates, and key elements such as paper titles and researcher names are designated with specific placeholders. The automatically generated templates, created by ChatGPT, consist of an additional 5 templates for each of the 70 question types. This process produces a total of 700 templates.

To incorporate information from CoopKG, we apply random sampling within the Neo4j database to substitute placeholders in the templates. This process can be repeated to generate more training samples. Finally, we fine-tune the LLMs using the LoRA method [3] on these samples. The process is shown in Figure 2.

3.2 Validation and Rewriting Method

We fine-tune the Text-to-JSON models using data sourced from CoopKG for the validation method. The models are trained by substituting the CQL in the Text-to-CQL templates with JSON-format data that were extracted from the natural language question. Text-to-CQL and Text-to-JSON models are referred as Model_CQL and Model_JSON, respectively.

Validation. We generate CQL and JSON from the original question using Model_CQL and Model_JSON, denoting the CQL as MG_CQL and the JSON



NL question:

What papers has Professor [researcher_name] from [university] published

CQL:

```
match(t:Researcher)-[:write]->(p) where t.name = '[researcher_name]'
and t.university = '[university]' return t.university as university, t.college
as college, t.name as name, p.paper_name as paper_name
```

Instruction:

Generate a Cypher query statement based on the question.

Input:

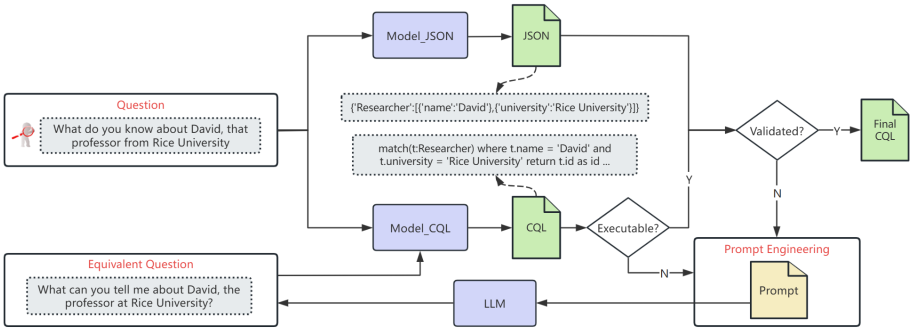
What papers has Professor David from Rice University published

Ouput:

```
match(t:Researcher)-[:write]->(p) where t.name = 'David' and t.university
= 'Rice University' return t.university as university, t.college as college,
t.name as name, p.paper_name as paper_name
```

Fig. 2. The process of generating training data through sampling and substitution

as MG_JSON, as shown in Figure 3. Next, we prepend the 'EXPLAIN' keyword to MG_CQL to generate its execution plan and execute this plan in the Neo4j database to verify its executability. If the plan is not executable, it indicates that MG_CQL is incorrect, and we execute the rewriting method. Conversely, We compare the entities and relationships in MG_CQL with those in MG_JSON. If the comparison is consistent, MG_CQL is validated as correct. Otherwise, we consider it erroneous and execute the rewriting method.



What do you know about David, that professor from Rice University

```
{'Researcher':[{'name':'David'},{'university':'Rice University'}]}
```

```
match(t:Researcher) where t.name = 'David' and
t.university = 'Rice University' return t.id as id ...
```

What can you tell me about David, the professor at Rice University?

Fig. 3. Workflow of the Verification and Rewriting Method

Rewriting. We construct a one-shot prompt using a vector search to find the closest question of the manually crafted template and its corresponding ChatGPT-generated template. The entities and relationships in both questions are then replaced by those of MG_JSON. This prompt is utilized to assist the LLM in generating equivalent questions, which are subsequently converted into CQL. The generated CQL undergoes validation, and if invalid, equivalent questions are iteratively generated until a valid CQL is produced or the maximum number of iterations is reached.

3.3 Entity Alignment with CoopKG

Users may input university abbreviations, such as 'PKU' for Peking University, which can cause mismatches with CoopKG, leading to empty search results. To address this, we fine-tune an LLM to convert abbreviations in user queries into their full names. The training data is constructed by replacing the placeholders in template questions with university abbreviations and their corresponding full names to form question-answer pairs.

Similarly, college name abbreviations present challenges, as they are numerous and subject to frequent changes. To handle this, we store college information from CoopKG in a Faiss database and replace abbreviations in user queries with the closest match from the database, ensuring accuracy of the generated CQL. Updates to CoopKG and the database are sufficient when a college name changes, avoiding the need for full model retraining.

3.4 System Extensions

Utilization of Portrait Images and Profile Documents Once the system generates the CQL, if rule-based matching determines that the query seeks to obtain detailed information about the Researcher nodes, we concatenate the Researcher node IDs from the query results with the designated root directory of the file storage, as specified in the configuration, to retrieve the corresponding profile documents and portrait images. The query results, together with the profile documents, are then supplied to the LLM to assist in generating the response, while the portrait images are displayed on the web page.

Graph Visualization and Data Export The system retrieves the CQL associated with the answer from the question-answer records and replaces the returned items of this CQL with the IDs assigned by the Neo4j database. It then executes the updated CQL to obtain these IDs, which are returned in an array format. Finally, the system utilizes multi-hop CQL to identify entities and relationships related to the retrieved IDs. The resulting entities and relationships are visualized as a graph on the web page.

When the user clicks the data export button, the system retrieves and re-executes the relevant CQL from the execution logs, then exports the results as an Excel file.

Online Search and AI-Assisted Reading The data collection process faces challenges due to incomplete information. To address this, we implement an online search feature, sourcing researcher data from Baidu Baike, paper data from the arXiv API, and project data via Google Search. By integrating online data, the system enhances answer relevance, accuracy, and query handling.

Table 1. Overview of Template-Based Datasets Generation.

Dataset	Template Type	Template Count	Task	Iteration Count	Size
TC_20H	Manual	350	Text-to-CQL	20	7000
TC_40H	Manual	350	Text-to-CQL	40	14000
TC_20M	Mixed	700	Text-to-CQL	20	14000
TJ_40M	Mixed	700	Text-to-JSON	40	28000
TJ_100M	Mixed	700	Text-to-JSON	100	70000

An extension is integrated to support common file formats, including text, Word, Markdown, and Excel. To manage file content, we use a content-overlapping segmentation strategy and store the segmented data with their vectorized representations. User queries related to document content are addressed by retrieving relevant information using the BM25 algorithm [9] and vector similarity. Subsequently, this information is analyzed and summarized by an LLM to generate the final answer.

4 Experimental Evaluation

4.1 Experimental Setup

Datasets In Chapter 3, we introduce a method for generating training samples. In experiments, we use this method to develop various training datasets for Text-to-CQL and Text-to-JSON tasks, as outlined in Table 1. To improve the accuracy of entity and relationship recognition, we increase the number of sampling iterations to enhance the quality of the Text-to-JSON dataset.

We also develop a Text-to-CQL test dataset. We manually craft 2 templates for generating CQL from natural language question for each of the 70 question types, yielding a total of 140 templates. By conducting a sampling and substitution process over 10 iterations, we generate 1,400 test samples to assess the effectiveness of our method for the Text-to-CQL task.

Metrics Akin to the evaluation metrics used in Text-to-SQL tasks, we adopt two measures, Logical Accuracy and Execution Accuracy, to evaluate the Text-to-CQL task.

Logical Accuracy compares the model-generated CQL with the logical form of gold CQL. It may contain false positives caused by conditional order. It is calculated as:

$$ACC_{\text{LX}} = \frac{\text{number of CQL with correct logic form}}{\text{total number of CQL}} \tag{1}$$

Execution Accuracy compares the execution results of CQL generated by the model with the execution results of gold CQL. It is calculated as:

$$ACC_{\text{EX}} = \frac{\text{number of CQL with correct execution result}}{\text{total number of CQL}} \tag{2}$$

Details All experiments are performed on a single Tesla V100 (32GB) GPU. The fine-tuning process for all LLMs employs the LoRA method with a LoRA-rank set to 8. Optimization utilizes the Adam optimizer, with the learning rate set to 1e-4, a batch size of 32, and 5 training epochs.

Baselines We fine-tune the base models, Baichuan2-7B-Chat [15], Llama2-Chinese-7B-Chat [10], and ChatGLM3-6B [17], using the TC_20H, TC_40H, and TC_20M datasets, and subsequently evaluate their performance on the Text-to-CQL task. To further evaluate the effectiveness of our proposed validation and rewriting method in the Text-to-CQL task, we also fine-tune these base models on the TJ_40M and TJ_100M datasets. In subsequent experiments, models fine-tuned on the TC_20M dataset will serve as baselines.

4.2 Supervised Fine-Tuning experiments

Table 2. Model Performance with Template-Generated Training Datasets (%)

Model	dataset	ACC_{EX}	ACC_{LX}
Baichuan2-7B-Chat	TC_20H	61.14	39.07
	TC_40H	86.00	74.93
	TC_20M	89.50	81.29
Llama2-Chinese-7b-Chat	TC_20H	88.00	80.14
	TC_40H	90.79	84.43
	TC_20M	91.72	86.43
Chatglm3-6B	TC_20H	73.64	47.36
	TC_40H	88.35	77.21
	TC_20M	85.00	72.36

We fine-tune the LLMs used in our experiments on the training datasets, followed by a comprehensive evaluation of the resulting models on the test dataset. The detailed experimental results are shown in Table 2. A comparison of the fine-tuning results on the TC_20H and TC_40H datasets demonstrates that our method effectively generates high-quality training datasets, significantly improving the accuracy of the fine-tuned models. Additionally, comparative experiments on the TC_20H and TC_20M datasets show that augmenting manually crafted training data with automatically generated data further enhance model performance.

4.3 Ablation experiments

We employ the model fine-tuned on the TC_20M dataset as the baseline and denote the method that integrates the baseline model with our validation and rewriting method for generating CQL as VARE.

Table 3. Performance Comparison of Baseline, and VARE Methods (%)

model	method	TJ_40M		TJ_100M	
		ACC_{EX}	ACC_{LX}	ACC_{EX}	ACC_{LX}
Baichuan2-7B-Chat	Baseline	89.50(−)	81.29(−)	89.50(−)	81.29(−)
	VARE	90.79(+1.29)	82.86(+1.57)	91.00(+1.50)	83.79(+2.50)
Llama2-Chinese-7b-Chat	Baseline	91.72(−)	86.43(−)	91.72(−)	86.43(−)
	VARE	93.14(+1.42)	88.29(+1.86)	93.29(+1.57)	88.36(+1.93)
Chatglm3-6B	Baseline	85.00(−)	72.36(−)	85.00(−)	72.36(−)
	VARE	89.86(+4.86)	76.57(+4.21)	90.29(+5.29)	78.07(+5.71)

The validation and rewriting method is applied to different datasets and LLMs. The experimental results demonstrate that our method achieves the highest accuracy in most cases. The VARE method significantly outperforms the baseline model, with a maximum increase of approximately 5.71% in ACC_{LX} and 5.29% in ACC_{EX}. Furthermore, applying our method to Chatglm3-6B yielded evaluation metrics comparable to those of Baichuan2-7B-Chat, showing that our approach effectively mitigates the impact of differences in model parameters. Additionally, a comparison of the experimental results between the TJ_40M and TJ_100M datasets reveals that improving the performance of the validation and rewriting method leads to further enhancement of the accuracy in the Text-to-CQL task.

5 CONCLUSION

In this study, we construct CoopKG, which includes detailed information about papers, researchers, and projects. We develop a KGQA system to effectively query and analyze CoopKG by natural language questions, employing a novel Text-to-CQL method. System extensions such as knowledge graph visualization, data export, online search, and AI-assisted reading are integrated into the system. We conduct comprehensive experiments on our proposed Text-to-CQL method to validate its effectiveness. The results demonstrate that, regardless of the open-source LLMs used, the accuracy of the generated outputs is significantly improved.

Acknowledgments. This research is supported by the National Natural Science Foundation of China (62192734, 62192730).

References

1. 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)

2. Guo, A., Li, X., Xiao, G., Tan, Z., Zhao, X.: Spcql: A semantic parsing dataset for converting natural language into cypher. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 3973–3977 (2022)
3. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
4. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of naacL-HLT. vol. 1, p. 2. Minneapolis, Minnesota (2019)
5. Ley, M.: Dblp: some lessons learned. Proceedings of the VLDB Endowment **2**(2), 1493–1500 (2009)
6. Li, B.Z., Min, S., Iyer, S., Mehdad, Y., Yih, W.t.: Efficient one-pass end-to-end entity linking for questions. arXiv preprint arXiv:2010.02413 (2020)
7. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in neural information processing systems **35**, 27730–27744 (2022)
8. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of sparql. ACM Transactions on Database Systems (TODS) **34**(3), 1–45 (2009)
9. Robertson, S., Zaragoza, H., et al.: The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends® in Information Retrieval **3**(4), 333–389 (2009)
10. Roumeliotis, K.I., Tselikas, N.D., Nasiopoulos, D.K.: Llama 2: Early adopters' utilization of meta's new open-source pretrained model (2023)
11. Sinha, A., Shen, Z., Song, Y., Ma, H., Eide, D., Hsu, B.J., Wang, K.: An overview of microsoft academic service (mas) and applications. In: Proceedings of the 24th international conference on world wide web. pp. 243–246 (2015)
12. Wan, H., Zhang, Y., Zhang, J., Tang, J.: Aminer: Search and mining of academic social networks. Data Intelligence **1**(1), 58–76 (2019)
13. Wang, R., Yan, Y., Wang, J., Jia, Y., Zhang, Y., Zhang, W., Wang, X.: Acekg: A large-scale knowledge graph for academic data mining. In: Proceedings of the 27th ACM international conference on information and knowledge management. pp. 1487–1490 (2018)
14. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., Schmidt, D.C.: A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:2302.11382 (2023)
15. Yang, A., Xiao, B., Wang, B., Zhang, B., Bian, C., Yin, C., Lv, C., Pan, D., Wang, D., Yan, D., et al.: Baichuan 2: Open large-scale language models. arXiv preprint arXiv:2309.10305 (2023)
16. Yao, Y., Zeng, Y., Zhong, N., Huang, X.: Knowledge retrieval (kr). In: IEEE/WIC/ACM International Conference on Web Intelligence (WI'07). pp. 729–735. IEEE (2007)
17. Zeng, A., Liu, X., Du, Z., Wang, Z., Lai, H., Ding, M., Yang, Z., Xu, Y., Zheng, W., Xia, X., et al.: Glm-130b: An open bilingual pre-trained model. arXiv preprint arXiv:2210.02414 (2022)
18. Zhang, J., Zhang, X., Yu, J., Tang, J., Tang, J., Li, C., Chen, H.: Subgraph retrieval enhanced model for multi-hop knowledge base question answering. arXiv preprint arXiv:2202.13296 (2022)