



LLM-Based SPARQL Generation with Selected Schema from Large Scale Knowledge Base

Shuangtao Yang^(✉), Mao Teng, Xiaozheng Dong, and Fu Bo

Lenovo Knowdee (Beijing) Intelligent Technology, Haidian District, Beijing, China
{yangst, Maoteng, dongxz}@knowdee.com

Abstract. Knowledge base question answering (KBQA) aims to answer natural language questions using structured knowledge bases. Common approaches include semantic parsing-based approaches and retrieval-based approaches. However, both approaches have some limitations. Retrieval-based methods struggle with complex reasoning requirements. Semantic parsing approaches have a complex reasoning process and cannot tolerate errors in earlier steps when generating the final logical form. In this paper, we proposed a large language model (LLM)-based SPARQL generation model, which accepts multiple candidate entities and relations as inputs, reducing the reliance on mention extraction and entity linking performance, and we found an entity combination strategy based on mentions, which can produce multiple SPARQL queries for a single question to boost the chances of finding the correct answer. Finally, our model achieves state-of-the-art performance in the CCKS2023 CKBQA competition, F1 score is 75.63%.

Keywords: KBQA · Large Language Model · SPARQL Generation

1 Introduction

Knowledge based question answering (KBQA) [1] has recently gained research interest, as it provides an intuitive way to access factual knowledge. The KBQA system makes use of structured knowledge bases such as Freebase, Wikidata, and DBpedia, which have logically organized entities and relations. A knowledge base typically contains a large number of triples, which can be represented as (*head, relation, tail*), the head refers to main entity, the tail refers to another entity or a literal value, and the relation is a directed relationship between head and tail [2]. KBQA systems can infer answers to questions by matching relevant entities and relations.

The existing KBQA approaches can be divided into two main categories: retrieval-based methods [3–9] and semantic parsing-based methods [10–23]. Retrieval-based methods directly represent and rank entities parsed from the input question. Among them, some methods first extract a subgraph containing only question-relevant entities from the knowledge base before performing reasoning. By narrowing the focus to a subset of KB, these methods can reduce the space for reasoning and be more efficient, while still struggling with complex questions. In contrast, semantic parsing-based (SP-based) methods parse a question into a logic form like SPARQL [10], Lambda-DCS [11], and

KoPL [12] that can be executed against the KB. However, these methods rely heavily on expensive annotations of intermediate logic forms and tend to be limited to narrow domains. With the advance of pre-trained language models (PLMs), many works have reformulated the semantic parsing task as a sequence-to-sequence (Seq2Seq) logical expression generation problem, which directly translate natural language queries into logical forms.

More recently, Large language models(LLM) have made significant advancements in natural language processing (NLP), such as GPT-3 [24], PaLM [25], LLaMA [26], which has proven to be an effective technique for improving performance on a wide range of language tasks [27]. Considering the large scale of the knowledge graph to process, containing 66,630,393 triplets, 11,327,935 entities, and 408,794 relations, we adopt a semantic parsing-based method with LLM as CKBQA solution. Like traditional semantics-based approaches, our method adopts a staged pipeline architecture. Traditional semantic parsing pipeline comprises mention extraction, entity linking, and SPARQL generation. However, for extremely large knowledge graphs, SPARQL generation performance by traditional semantic parsing pipeline often decreases substantially due to error propagation across pipeline. Due to the outstanding capabilities of large language models (LLMs) [20], we proposed large language model (LLM)-based SPARQL generation model that accepts multiple candidate entities and relations as inputs, which helps to reduce the reliance on mention extraction and entity linking performance. We incorporate an entity relation selection model into the pipeline to prune noisy inputs for the generation model. Additionally, we implement an entity combination strategy based on mentions, which can produce multiple SPARQL queries for a single question to boost the chances of finding the correct answer.

The main contributions of this paper are summarized below:

- This work represents the first attempt at leveraging large pre-trained language models (LLM) for SPARQL generation to address Chinese knowledge graph question answering, achieving top-1 ranking performance in the CCKS2023 CKBQA competition.
- We propose an effective SPARQL generation method based on large language models, utilizing mention extraction, entity linking, attribute selection models, and entity combination to provide high-quality inputs for the language models, significantly improving SPARQL generation quality. The model process is shown in Fig. 1.
- Ablation experiments were conducted to assess the importance of each module in SPARQL generation for our approach.

2 Related Work

Retrieval-Based Methods. Zhang et al. proposed a subgraph retriever (SR) separate from the subsequent reasoner for KBQA. The SR was designed as an efficient dual-encoder capable of updating the question representation when expanding the path, as well as determining when to stop the expansion [3]. He et al. proposed a teacher-student approach for multi-hop KBQA. The teacher network utilized bidirectional reasoning to produce reliable intermediate supervision signals that improved the reasoning of the student network and reduced spurious reasoning [4].

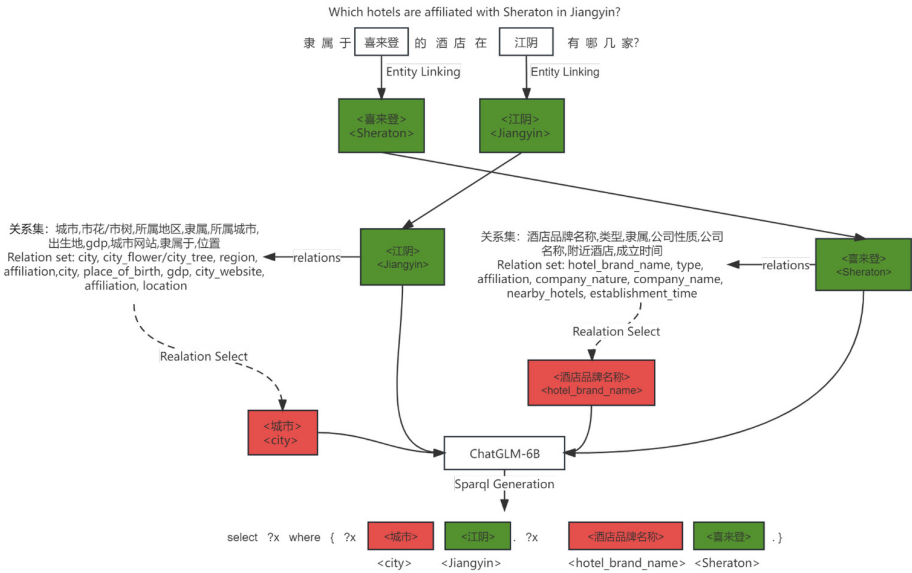


Fig. 1. SPAEQL generation with selected entity and relations. Mentions (highlighted in white boxes) need to be linked to entites which are from knowledge base. There are two entities (highlighted in green boxes), we need to obtain all the relations or attributes of each entity, and then use attribute/relation rank model to sort them. The selected entities (in green boxes) and relations (in red boxes) will as input to the SPARQL generation model. The given textual mentions can be utilized to construct focused SPARQL queries, incorporating the most relevant entities and relations.

Semantic Parsing-Based Methods. Purkayastha et al. [13] used a Seq2Seq model to generate SPARQL query sketch, and then apply entity and relation linkers to fill in the sketch and produce a complete SPARQL query. Lambda-DCS (lambda dependency-based compositional semantics) [11] is a tree-structured logical Forms, which propose to reduce the complexity in compositionally creating the logical form of a sentence. Cao et al. [12] first parse the original question into the skeleton of KoPL program, a sequence of symbolic functions, and then train an argument parser to retrieve corresponding arguments of these functions.

Seq2Seq Methods. Nie et al. proposed a unified intermediate representation (GraphQ IR) that bridges the semantic gap between natural language queries and formal graph query languages. GraphQ IR can produce intermediate representation sequences using composition rules consistent with English to capture natural language semantics while maintaining fundamental graph structures [14]. Cao et al. proposed a Line Graph Enhanced Text-to-SQL (LGESQL) to extract relational features from text without having to construct meta paths. The Line Graph representation allowed messages to propagate more efficiently by considering not just connections between nodes, but also the topology of directed edges [15]. Das et al. first identify different queries with semantically equivalent components, and then construct a new logical form by combining these matching components from the discovered queries [21]. Huang et al. utilize a large model-based

algorithm to identify entities and relations within a question, and then generate a query structure with placeholders, which are then populated in a post-processing step [22]. Xiong et al. utilize advanced generative pre-trained language models to generate questions from logical form and then make predictions, the auto-prompter has the ability to paraphrase predicates in a consistent and fluent manner [23].

LLM-Based Methods. LLM with billions of parameters have achieved state-of-the-art results on many NLP benchmarks by learning powerful contextual representations of language from large amounts of text data. One key development in LLM is the use of self-attention mechanism [28] and transformer architectures [29]. Another important development is the use of pre-training, where models are first trained on massive datasets and then fine-tuned on downstream tasks. LLM transfers broad linguistic knowledge that significantly improves performance across many language understanding tasks. One remarkable recent development is the launch of ChatGPT [30], a conversational AI system powered by LLMs. ChatGPT has gained widespread public attention for its ability to engage in surprisingly natural conversations, which highlight the substantial progress LLMs have made in language understanding and generation that allows them to partake in coherent human-like dialogue.

3 Method

As shown in Fig. 2, The methodology we propose comprises four fundamental components: 1) extracting textual mentions from the input, 2) linking mentions to entities in the knowledge graph, 3) selecting relevant attributes and relations from these entities, and 4) combining these entities to generate SPARQL queries. The specific implementations of each module will be described fully in subsequent sections. The complete descriptions of the individual modules' specific implementations will be provided in subsequent sections.

3.1 Mention Extraction

Mention Extraction is the task of identifying the mention span of all entities in the question [31]. Each such span is referred to as an entity mention. The word or sequence of words that refers to an entity is also known as the surface form of the entity. An utterance may contain multiple entity, often also consisting of more than one word. Additionally, a broader classification of entities, such as person, location, and organization, can sometimes be assigned.

Our mention extraction model architecture is composed of a BERT encoder with a token-level classifier on top followed by a Linear-Chain CRF. We first use BERT to encode user question and outputs a sequence of encoded token representations, then a classification model projects each token's encoded representation to the tag space. We also frame mention extraction as a generative task, and attempt to extract mentions using ChatGLM [35].

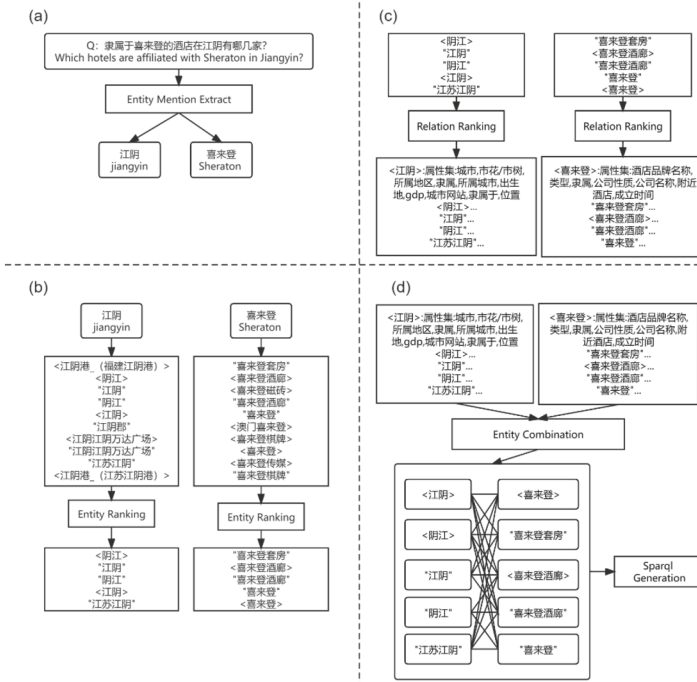


Fig. 2. Method Flow: (a) Mention extraction is carried out on the question. (b) For each mention, top-5 candidate entities are selected from the knowledge graph by using Elasticsearch and rules, and then ranked by the entity linking model. (c) Relation selection is applied to choose the most relevant relations for entities from the previous entity linking results. (d) Candidate entities across different mentions are combined and fed into the SPARQL generation model to produce multiple SPARQL queries.

3.2 Entity Linking

The task of Entity Linking involves establishing connections between annotated mentions in a given utterance and their corresponding entities within a knowledge base [32–34]. This task was addressed by using popular knowledge bases such as DBpedia, Freebase or Wikipedia. Entity linking serves as a bridge between textual spans and structured entities within a knowledge base, thereby will be beneficial to various downstream tasks like question answering and knowledge extraction. EL aims to link entity mentions in unstructured text to their corresponding entities in a designated knowledge base.

Our entity linking model is trained to assign a score to each candidate entity as shown in (1). Specifically, given the question q and the candidate entity e_text , we use a BERT-based encoder to generate a score indicating the confidence of the link [29].

$$el_score = sigmoid(AVG(BERT([q : e_text]))) \tag{1}$$

For every mention, we will select top-5 entities according to their linking confidence scores for the next phase.

3.3 Entity Attribute/Relation Select

When given an entity and its relations, Entity Attribute/Relation Select model can select question related relations, thus the model also learns to score each entity relation. Specifically, given the question q and the candidate entity relations r_text , we also use a BERT-based encoder to generate a correlation score between the question and either entity attributes or relations.

$$es_score = sigmoid(AVG(BERT([q : r_text]))) \quad (2)$$

The entity and its relation are represented by triples consisting of the entity, relationship, and tail. For each entity, we will select top-5 relationships based on their correlation score for the next phase.

3.4 Entity Combination and SPARQL Generation

After the phase of entity linking and entity attribute/relation select, we have obtained top-5 entities for each mention, and each entity has top-5 relations, as key supporting evidence. In the SPARQL generation stage, we attempted different methods.

Method 1: The question and all key supporting evidence from different mentions are concatenated as input to the SPARQL generation model, resulting in a single expression.

Method 2: Entities from different elements are combined and concatenated within each combination to generate multiple SPARQL queries, which can be executed against the KB. Unlike Method 1, the approach will produce multiple SPARQL expressions.

Taking the question “What are the hotels affiliated with Sheraton in Jiangyin?” as an example, after mention extraction, entity linking, and entity attribute/relation selection, we obtained the most relevant knowledge related to this question from the knowledge graph. In Method 1, we filled all relevant information into the prompt, obtaining the complete prompt as shown below.

请根据问题:\”隶属于喜来登的酒店在江阴有哪几家?”,和候选实体信息:[0] 名称: < 喜来登 >,属性集:酒店品牌名称,类型,隶属,公司性质,公司名称,附近酒店,成立时间;[1]名称: < 江阴 >,属性集:城市,市花/市树,所属地区,隶属,所属城市,出生地,gdp,城市网站,隶属于,位置,市长,所在城市,适用地区,分布区域,所属地区,著名景点,位于,行政区域,属于,家乡;[2]名称: < 江阴黄嘉喜来登酒店 >,属性集:实体名称,酒店品牌名称,酒店入住开始时间,是否有鲜花店,是否有酒吧,是否有接机服务,是否有接机服务-营业时间,是否有接送服务-营业时间,是否有温泉,是否有桑拿浴室,是否有允许带宠物,是否有茶室,是否有会议室,是否有桌球室,是否有管家服务,是否有熨衣服务,是否有图书馆,是否有wifi服务,是否有游戏室,是否有礼宾服务;[3]名称: < 镇江富力喜来登酒店 >,属性集:实体名称,酒店品牌名称,房型名称,是否有鲜花店,是否有桑拿浴室,酒店入住开始时间,是否有允许带宠物,是否有温泉,是否有高尔夫球场,是否有保龄球馆,是否有租车服务,是否有大堂吧,是否有多功能厅,是否有网球场,是否有婚宴服务,是否有叫醒服务,是否有礼宾服务,是否有KTV,是否有图书馆,是否有会议室;[4]名称:\”喜来登”,属性集:中文名称,公司名称,对应查询图谱的sparql的语句为:

Please follow the question: \”Which hotels are affiliated to Sheraton in Jiangyin?”, and candidate entity information: [0] Name: < Sheraton >, attribute set: hotel brand

name, type, affiliation, company nature, company Name, nearby hotels, establishment time; [1] Name: < Jiangyin >, attribute set: city, city flower/city tree, region, affiliation, city, place of birth, gdp, city website, affiliation, location, city Length, city, applicable area, distribution area, belonging area, famous scenic spot, location, administrative area, belonging to, hometown; [2] name: < Jiangyin Huangjia Sheraton Hotel >, attribute set: entity name, hotel brand name, hotel Check-in start time, whether there is a flower shop, whether there is a bar, whether there is a pick-up service, whether there is a pick-up service-opening hours, whether there is a pick-up service-opening hours, whether there is a hot spring, whether there is a sauna, whether pets are allowed, whether there is a tea room, whether there is a meeting room, whether there is a billiard room, whether there is a butler service, whether there is an ironing service, whether there is a library, whether there is wifi service, whether there is a game room, whether there is a concierge service; [3] name: < Sheraton Zhenjiang Hotel >, attribute set: entity name, hotel brand name, room type name, whether there is a flower shop, whether there is a sauna, hotel check-in start time, whether pets are allowed, whether there is a hot spring, whether there is a golf course, Is there a bowling alley, is there a car rental service, is there a lobby bar, is there a multi-function hall, is there a tennis court, is there a wedding banquet service, is there a wake-up call service, is there a concierge service, is there a KTV, is there a book Museum, whether there is a conference room; [4] Name: \“Sheraton\”, attribute set: Chinese name, company name, the Sparql statement corresponding to the graph is:

In Method 2, we combined entity information from different mentions, then filled each combination into the prompt, so we could obtain multiple prompts to generate SPARQL statements. For the combination < Sheraton >, < Jiangyin >, the complete prompt is shown below.

请根据问题:\“隶属于喜来登的酒店在江阴有哪几家\”,和候选实体信息:[0]名称: < 喜来登 >,属性集:酒店品牌名称,类型,隶属,公司性质,公司名称,附近酒店,成立时间;[1]名称: < 江阴 >,属性集:城市,市花/市树,所属地区,隶属,所属城市,出生地,gdp,城市网站,隶属于,位置,市长,所在城市,适用地区,分布区域,所属地区,著名景点,位于,行政区域,属于,家乡,对应查询图谱的Sparql的语句为:

Please follow the question: \“Which hotels are affiliated to Sheraton in Jiangyin?\”, and candidate entity information: [0] Name: < Sheraton >, attribute set: hotel brand name, type, affiliation, company nature, company Name, nearby hotels, establishment time; [1] Name: < Jiangyin >, attribute set: city, city flower/city tree, region, affiliation, city, place of birth, gdp, city website, affiliation, location, city Length, city, applicable area, distribution area, area, famous scenic spot, location, administrative area, belonging, hometown, the Sparql statement corresponding to the graph is:

After obtaining the complete prompt, we feed it to the LLM to generate SPARQL. We select ChatGLM-6B¹ [35] as the SPARQL generation models. ChatGLM-6B is a pre-trained large language model with 6.2 billion parameters, based on the General Language Model (GLM) architecture. ChatGLM-6B was trained on around 1 trillion tokens of Chinese and English corpus, with additional supervised fine-tuning, feedback

¹ <https://github.com/THUDM/ChatGLM-6B>.

bootstrap, and reinforcement learning using human feedback. This enables ChatGLM-6B to generate answers that are aligned with human preference, with fluency in both English and Chinese.

We use low-rank adaptation (LoRA) to finetune ChatGLM-6B for SPARQL Generation [36]. The parameter settings used for LoRA fine-tuning are shown in Table 1.

Table 1. Knowledge Graph Information.

Item	Quantity
lora_alpha	32
lora_dropout	0.1
lora_rank	8
lora_target	query_key_value

4 Results

The key statistics for the knowledge graph and training data used in this work are presented in Table 2. The knowledge graph contains 66,630,393 triplets, 11,327,935 entities, and 408,794 relations. The training data is comprised of 7,625 examples.

Table 2. Knowledge Graph and Data Information.

Item	Quantity
Triplet	66,630,393
Entity	11,327,935
Relation	408,794
Train Case	7,625

4.1 Mention Extraction Result

We compared several mention extraction methods on CKBQA dataset, including BERT + CRF, Roberta + CRF, and ChatGLM-6b(LoRA). As shown in Table 3, the ChatGLM-6b(LoRA) model achieved the highest F1 score.

4.2 Entity Linking Result

We compared Bert and Roberta on entity linking task. As shown in the Table 4, RoBerta achieved the higher F1 score of 94.48%, compared to 93.64% for Bert. This indicates RoBerta is more effective for this entity linking task, outperforming Bert by 0.84% in terms of F1 score.

Table 3. Mention Extraction Result

Method	F1 Performance (100%)
Bert + CRF	84.5
Roberta + CRF	85.2
ChatGLM-6b(LoRA)	89.4

Table 4. Entity Linking Result

Method	F1 Performance (100%)
Bert	93.64
RoBerta	94.48

4.3 Entity Attribute/Relation Select

F1 scores for Bert and RoBerta models on an entity attribute/relation selection task are presented in Table 5. RoBerta model achieved the higher F1 score of 95.17%, compared to 94.12% for Bert model, indicating RoBerta is more effective for extracting entity attributes and relations, outperforming Bert model by 1.05% based F1 evaluation metric.

Table 5. Entity Attribute/Relation Select Result

Method	Performance (100%)
Bert	94.12
RoBerta	95.17

4.4 Entity Combination and SPARQL Generation

At this stage, we compared the impact of different entity combination methods on SPARQL generation. Using the same ChatGLM-6B model and LoRA fine-tuning parameters, we trained and fine-tuned two SPARQL generation models with different entity combination approaches. Table 6 shows the performance of the two entity combination methods on the training and validation set, which is evaluated using ROUGE-1, ROUGE-2, ROUGE-L [37]. To evaluate the correctness of SPARQL, we introduced the Pass Rate metric. ChatGLM-6b-Method2 achieved higher scores across all metrics, with notably large improvements in ROUGE-2 (90.11% vs 85.96%) and Pass rate (68.9% vs 61.5%). This suggests that ChatGLM-6b-Method2 is more effective for SPARQL generation.

The pass rate metric measures the ratio of generated SPARQL queries that are syntactically valid and return correct answers on test set.

Table 6. Entity Combination and SPARQL Generation Result.

Method	Acc Performance (100%)			
	ROUGE-1	ROUGE -2	ROUGE -L	Pass Rate
ChatGLM-6b-Method1	91.72	85.96	89.09	61.5
ChatGLM-6b-Method2	94.89	90.11	91.72	68.9

4.5 End to End Performance

We conducted ablation experiments to evaluate the importance of each module in our pipeline. The results of these experiments are shown in Table 7.

Table 6 shows the incremental impact of on KBQA system from adding different knowledge graph components. We evaluated five system variations (V1-V5) on the CKBQA training dataset.

System V1 uses only a mention extraction (ME) model and SPARQL generation (SG) module, achieving an F1 score of 45.11%. The lack of entity linking, relation selection, and entity combining modules limits its performance. By analyzing the generated SPARQL, we found that errors often occur due to inconsistent entity formats with the knowledge base, making it impossible to obtain answers through SPARQL.

System V2 adds an entity linking (EL) module using RoBerta, improving performance to 66.45% F1. Linking mentions to knowledge graph entities provides useful contextual information.

System V3 further incorporates an entity attribute/relation selection (ERS) module based on Roberta. This model eliminates interference from irrelevant attributes and relationship of entities in the input, increasing F1 to 69.23%.

System V4 adds an entity combination (EC) module. Through this module, we can assemble entity information from different mentions to generate multiple SPARQL queries. Concurrently, we can determine the relevance of each SPARQL query based on relatedness between entities. The most relevant SPARQL that can retrieve results from the knowledge graph is selected as the final generated query. By utilizing this method, we improved the performance of our system to 73.93% F1 score.

Even after System V4, we still found a limited number of questions for which it was not possible to generate an accurate SPARQL query that could retrieve answers from the knowledge graph. Therefore, we supplemented with an additional KBQA method based on triple retrieval. By integrating this approach, we further improved our system's score to 75.63%.

Table 7. Knowledge Graph Information. **ME** means Mention Extract Model; **EL** means Entity Linking Model; **ERS** means Entity attribute/relation Select Model and **EC** means Entity Combination Module and **SG** means SPARQL Generation, and **Retrieval** means Retrieval Method For KBQA.

System	ME	EL	ERS	EC	SG	Retrieval	Acc (100%)
V1	✓	×	×	×	✓	×	45.11
V2	✓	✓	×	×	✓	×	60.45
V3	✓	✓	✓	×	✓	×	69.23
V4	✓	✓	✓	✓	✓	×	73.93
V5	✓	✓	✓	✓	✓	✓	75.63

5 Conclusion

In this paper, we proposed large language model (LLM)-based SPARQL generation model, which accepts multiple candidate entities and relations as inputs, reducing the reliance on mention extraction and entity linking performance. And we found an entity combination strategy based on mentions, which can produce multiple SPARQL queries for a single question to boost the chances of finding the correct answer. Finally, we get 1st place in CCKS2023 CKBQA competition with F1 score of 75.63%. In the future, we will delve into research on SPARQL query generation with large language models, especially focus on multiple hops and multi constraints query.

References

1. Zhang, J., Chen, B., Zhang, L., et al.: Neural, symbolic and neural-symbolic reasoning on knowledge graphs. *AI Open* **2**, 14–35 (2021). <https://doi.org/10.1016/j.aiopen.2021.03.001>
2. Ye, X., Yavuz, S., Hashimoto, K., et al.: RnG-KBQA: Generation Augmented Iterative Ranking for Knowledge Base Question Answering. *arXiv e-prints* <https://doi.org/10.48550/arXiv.2109.08678> (2021)
3. Zhang, J., Zhang, X., Yu, J., et al.: Subgraph Retrieval Enhanced Model for Multi-hop Knowledge Base Question Answering (2022). <https://doi.org/10.48550/arXiv.2202.13296>
4. He, G., Lan, Y., Jiang, J., et al.: Improving multi-hop knowledge base question answering by learning intermediate supervision signals. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 553–561 (2021)
5. Chen, Y., Wu, L., Zaki, M.J.: Bidirectional Attentive Memory Networks for Question Answering Over Knowledge Bases. *arXiv preprint* [arXiv:1903.02188](https://arxiv.org/abs/1903.02188) (2019)
6. Saxena, A., Tripathi, A., Talukdar, P.: Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4498–4507 (2020)
7. Xu, K., Lai, Y., Feng, Y., et al.: Enhancing key-value memory neural networks for knowledge based question answering. *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. 2937–2947 (2019)

8. 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)
9. 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)
10. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of SPARQL. *ACM Trans. Database Syst.* **34**(3), 1–45 (2009)
11. Liang, P.: Lambda dependency-based compositional semantics. *Computer Science* (2013). [https://doi.org/10.48550/arXiv.1309.4408.](https://doi.org/10.48550/arXiv.1309.4408)]
12. Cao, S., et al.: KQapro: a dataset with explicit compositional programs for complex question answering over knowledge base. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pp. 6101–6119 (2022)
13. Purkayastha, S., Dana, S., Garg, D., Khandelwal, D., Bhargava, G.S.: A deep neural approach to KGQA via SPARQL Silhouette generation. In: *2022 International Joint Conference on Neural Networks. IJCNN, IEEE*, pp. 1–8 (2022)
14. Nie, L., et al: GraphQ IR: Unifying the Semantic Parsing of Graph Query Languages with One Intermediate Representation. ArXiv, [arXiv:2205.12078](https://arxiv.org/abs/2205.12078) (2022)
15. Cao, R., Chen, L., Chen, Z., Zhao, Y., Zhu, S., Yu, K.: LGESQL: Line graph enhanced text-to-SQL model with mixed local and non-local relations. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1–6, 2021, Association for Computational Linguistics*, pp. 2541–2555 (2021)
16. Das, R., Zaheer, M., Thai, D., et al.: Case-Based Reasoning for Natural Language Queries Over Knowledge Bases. arXiv preprint [arXiv:2104.08762](https://arxiv.org/abs/2104.08762) (2021)
17. Kapanipathi, P., Abdelaziz, I., Ravishankar, S., et al.: Leveraging Abstract Meaning Representation for Knowledge Base Question Answering. arXiv preprint [arXiv:2012.01707](https://arxiv.org/abs/2012.01707) (2020)
18. Lan, Y., Jiang, J.: Query Graph Generation for Answering Multi-Hop Complex Questions from Knowledge Bases. *Association for Computational Linguistics* (2020)
19. Sun, Y., Zhang, L., Cheng, G., et al.: SPARQA: skeleton-based semantic parsing for complex questions over knowledge bases. *Proceedings of the AAAI Conference on Artificial Intelligence* **34**(05), 8952–8959 (2020)
20. Qiu, Y., Wang, Y., Jin, X., et al.: Stepwise reasoning for multi-relation question answering over knowledge graph with weak supervision. *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 474–482 (2020)
21. Das, R., Zaheer, M., Thai, D., et al.: Case-based reasoning for natural language queries over knowledge bases. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9594–9611 (2021)
22. Huang, X., Kim, J.J., Zou, B.: Unseen entity handling in complex question answering over knowledge base via language generation. *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 547–557 (2021)
23. Xiong, G., Bao, J., Zhao, W., et al.: AutoQGS: auto-prompt for low-resource knowledge-based question generation from SPARQL. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 2250–2259 (2022)
24. Floridi, L., Chiriatti, M.: GPT-3: Its Nature, Scope, Limits, and Consequences. [2023–08–17]. <https://doi.org/10.1007/s11023-020-09548-1>
25. Chowdhery, A., Narang, S., Devlin, J., et al.: PaLM: Scaling Language Modeling with Pathways (2022). <https://doi.org/10.48550/arXiv.2204.02311>
26. Touvron, H., Lavril, T., Izacard, G., et al.: Llama: Open and Efficient Foundation Language Models. arXiv preprint [arXiv:2302.13971](https://arxiv.org/abs/2302.13971) (2023)

27. Min, B., Ross, H., Sulem, E., et al.: Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey. arXiv preprint [arXiv:2111.01243](https://arxiv.org/abs/2111.01243) (2021)
28. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. *Advances in Neural Information Processing Syst.* **30** (2017)
29. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NaaCL-HLT*, **1**, p. 2 (2019)
30. Team O A I. ChatGPT: Optimizing Language Models for Dialogue (2022)
31. Zhao, W.X., Zhou, K., Li, J., et al.: A Survey of Large Language Models. arXiv preprint [arXiv:2303.18223](https://arxiv.org/abs/2303.18223) (2023)
32. Lin, Y., Ji, H., Huang, F., et al.: A joint neural model for information extraction with global features. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7999–8009 (2020)
33. Wu, L., Petroni, F., Josifoski, M., et al.: Scalable Zero-Shot Entity Linking with Dense Entity Retrieval. arXiv preprint [arXiv:1911.03814](https://arxiv.org/abs/1911.03814) (2019)
34. Li, B.Z., Min, S., Iyer, S., et al.: Efficient One-Pass End-to-End Entity Linking for Questions. arXiv preprint [arXiv:2010.02413](https://arxiv.org/abs/2010.02413) (2020)
35. Du, Z., Qian, Y., Liu, X., et al.: Glm: General Language Model Pretraining with Autoregressive Blank Infilling. arXiv preprint [arXiv:2103.10360](https://arxiv.org/abs/2103.10360) (2021)
36. Hu, E.J., Shen, Y., Wallis, P., et al.: Lora: Low-Rank Adaptation of Large Language Models. arXiv preprint [arXiv:2106.09685](https://arxiv.org/abs/2106.09685) (2021)
37. Lin, C.Y.: Rouge: a package for automatic evaluation of summaries. *Text Summarization Branches Out*, pp. 74–81 (2004)