Scalable Database-Driven KGs can help Text-to-SQL

Zhongqiu Li[1,†], Zhenhe Wu[1,2,‡,†], Mengxiang Li[3], Zhongjiang He[1], Ruiyu Fang[1], Jie Zhang[1], Yu Zhao[1], Yongxiang Li[1], Zhoujun Li[2,*] and Shuangyong Song[1,*]

[1]*Institute of Artificial Intelligence (TeleAI), China Telecom Corp Ltd, Xicheng District, Beijing, China*

[2]*BeiHang University, No.37 XueYuan Road, HaiDian District, Beijing, China*

[3]*China Telecom Corporation, No.31 Financial Street, Xicheng District, Beijing, China*

Abstract

The text-to-SQL task aims to covert natural language questions into SQL queries. Large Language Models (LLMs) have demonstrated remarkable performance on this task, which relied on in-context learing or Supervised Fine-Tuning (SFT). However, the heterogeneity of database and the complexity of the knowledge acquisition process pose significant challenges in previous works. To address these, we propose a novel text-to-SQL framework that enhances the performance of LLMs through Knowledge Graphs (KGs). We construct the KGs based on schemas, which are structured representations of the relationships and attributes within the databases. Then, we utilize LLMs to extract descriptions and dependencies from historical queries, which are used to complete contextual knowledge in KGs. We leverage retrieval model to recall benefit nodes and edges from KGs and then employ LLMs to generate task-specific evidence. Based on the evidence and retrieved information, we define a unified KGs-based schema for LLMs to generate SQL queries. Our paper conducts experiments on public datasets BIRD and Spider, and the results indicate that our framework significantly improves the text-to-SQL performance.

Keywords

Text-to-SQL, Large Language Models, Knowledge Graph, Knowledge Generation

1. Introduction

Text-to-SQL task aims to generate Structured Query Language (SQL) queries for databases from natural language questions[1, 2]. The process facilitates non-expert data analysts in automatically extracting desired information from widely accessible databases using natural language. Database schema is the structural representation of database. It defines the organization of data storage, including tables, columns, data types, relationships and constraints within the database. In different databases, schemas also include descriptions, as well as domain dependencies related to columns. Advanced studies focus on in-context learning to guide LLMs to understand and generate SQL queries. These approaches typically involve providing questions and schemas into LLMs as context, while adapting domain knowledge through few-shot learning or Super-

Figure 1: Framework of the Scalable Database-Driven KGs for Text-to-SQL. *Schema*, *Data storage*, and *Query Views* are fundamental components of a database. First, we use LLMs to construct KGs from schema and data values. Then, we retrieve knowledge that is closely related to the user question and generate evidence. Finally, the selected knowledge is used to complete the SQL generation.

vised Fine-Tuning (SFT). However, practical applications encounter several challenges. Firstly, different database structures and application scenarios, leading to significant variations in data styles that complicate schema representation. Secondly, SQL queries often involve large tables with an excessive number of columns, exceeding the typical window size of LLMs.

Our paper converts schemas into nodes and edges within Knowledge Graphs (KGs). To enrich these KGs, we use LLMs to analyze database interaction history, extracting detailed descriptions and dependencies to complete the KGs. We then retrieve nodes and edges closely related to user queries from these KGs, generate relevant evidence, and use this information as context for generating precise SQL queries.

Li et al.[3] discover the critical importance of supplementary evidence when handling complex database values. Their work explicitly defines "evidenc" for the first time and annotates a standard dataset with evidence, namely BIRD. Evidence typically consists of concise statements that link multiple entity mentions in natural language to functions, tables, columns, and constraints in the database. It acts as an intermediary state between user questions and the database schema, substantially improving models' understanding of the database schema. However, the evidence they provide is manually annotated, which limits its applicability to other datasets. Nonetheless, evidence is extremely scarce across databases. To address this deficiency, we utilize LLMs to generate evidence through KGs-based schema, which is highly effective for SQL generation.

2. Our Approach

We introduce a scalable KGs-based text-to-SQL framework. As illustrated in Figure 1, our framework comprises four stages. The first stage is KGs construction, where we design two types of nodes and three types of predefined relationships based on database components. The second stage is knowledge completion, where we utilize LLMs to extract implicit knowledge from user historical questions and SQL pairs, in order to expand the attributes and dependencies in the KGs. The third stage is knowledge selection and generation. We design a retrieval model

to identify and select the most relevant knowledge from the expanded KGs, and then use this knowledge to generate evidence and context for SQL query formulation. The last stage is SQL generation, where we combine the user question and the previously obtained schema knowledge as context and use LLMs to generate SQL queries using a least-to-most method.

Knowledge Graph Construction. Our approach aims to construct the KGs based on tabular structures and columns. The graph mainly consists of two key entity types, i.e.,**Table** and **Column**. The former is defined by the attributes "Table Name" and "Table Description ". The latter is achieved by the attributes "Column Name", "Column Description", "Data Type", "Example Value", and "Value Explanation". We design three predefined relational categories: inter-table relationship, inclusion relationship between tables and columns, and foreign key relationship. Overall, we use the distinct operation of databases for extracting metadata (i.e., the above mentioned entity types and relational categories) in order to obtain structured database information for constructing KGs. Figure 1 illustrates our approach of using value examples as attributes for columns to manage the diversity in data storage formats among different databases. This diversity stems from the different conventions for data representation.

Knowledge Completion with LLMs. We propose a knowledge completion strategy based on LLMs that helps to enrich the attributes of nodes and the relationships between nodes in the knowledge graph. Our method collects tuples $\{q, s\}$ comprising user historical questions and corresponding SQL queries. These tuples are processed using LLMs to extract column knowledge as $c_d = \{c, d\}$ and functional dependencies as $c_r = \{c_h, r, c_t\}$. c denotes the column name and d represents its description in the question. c_h and c_r respectively denote the head and tail columns extracted from SQL, while r describes the relationship between c_h and c_r from question. Both c_d and c_r serve as supplementary updates to the graph. For reducing the addition of redundant information, we assess the relevance of existing column knowledge before updating and employ LLMs to determine the necessity of update.

Knowledge Selection and Generation. As illustrated in Figure 1, we design a retrieval model to recall columns from KGs, which is proved effective in Dense Passage Retrieval (DPR)[4]. We convert the candidate columns into continuous text by concatenating attributes of columns as $\{c = c_{name}||c_{description}||c_{example}||c_{explanation}\}$. Afterwards, we encode question q and column c using encoders to obtain embeddings O_q and O_c, which are utilized for the interaction. We calculate the final similarity score of O_q and O_c as follows:

$$O_{q/c} = \text{Normalize}\left(\text{CNN}\left(\text{BERT}_{Q/C}\left(q/c\right)\right)\right) \tag{1}$$

$$score_{q,t} = \sum_{i \in [|O_q|]} \max_{j \in [|O_c|]} O_q^i \cdot O_c^j \tag{2}$$

i and j are token embeddings indices of O_q and O_c. By setting a threshold of score, we disambiguate the candidates to select the final columns.

Through knowledge selection, we extract general knowledge from KGs. And then, we input the KGs-based schema, user question, and instructions into the LLMs to generate evidence, which is crucial for establishing the correlation between questions and database. We specifically design the evidence generation process which comprises three steps. First, we utilize LLMs to extract key terms from user questions. These key terms typically include important entities, attribute names, and query conditions. In practice, these terms may not always be explicit

Table 1
Execution Accuracy on BIRD and Spider.

Method	BIRD	Spider	
	dev	dev	test
GPT-4	47.97	81.18	83.40
GPT-4 + KGs	51.20	82.15	85.61
GPT-4 + KGs + evidence (generated)	54.33	**84.79**	**86.51**
GPT-4 + KGs + evidence (gold)	**57.17**	—	—

and clear, and can sometimes be ambiguous or non-standardized. Next, based on these terms, we use LLMs to enumerate normalized schema blocks such as related tables, columns, and database function expressions. LLMs map these terms to the database schema. During this mapping process, LLMs not only correspond the terms to standardized expressions but may also elucidate the dependencies in the query. For instance, they identify relevant tables and columns, determine logical relationships between these columns (such as ratios or comparisons), and may infer necessary data connections (such as joins) and filtering conditions. This detailed mapping by LLMs facilitates the precise extraction of results that meet the specified conditions, ensuring the accuracy and completeness of the query process. We summarize the results of this process into a concise sentence as evidence.

SQL Generation. We employ LLMs with in-context learning to address SQL generation[5, 6, 7, 8]. Specifically, our approach involves organizing nodes and relationships extracted from KGs into a redefined schema. We then input the redefined schema and generated evidence into the LLMs as context. The models use this contextual information to decompose the user's questions into several sub-questions and generate SQL queries for each sub-question using a least-to-most prompting strategy. To enhance the LLMs' SQL generation accuracy, we manually crafted three examples to guide the model through few-shot learning, improving the overall effectiveness of the final SQL query.

3. Experiments and Conclusion

We ran our experiments on two widely used benchmark SQL datasets: BIRD[3] and Spider 1.0[9]. We use GPT-4 to complete KGs and generate SQLs. In our paper, we fine-tune a BERT-based dual-tower model using training datasets. It is employed to generate embeddings for entities and relations within the KGs, thereby enhancing graph retrieval capabilities.

As indicated in Table 1, the row for *GPT-4* indicates using the origin database schema directly, rather than leveraging schema KGs. In comparison, by organizing schemas components into KGs and utilizing retrieval models to recall tables and columns, we observe a notable enhancement of 3.23% , 0.97% and 2.21% on BIRD and Spider. Secondly, we employ generated evidences for SQL generation. It contributes to additional improvement of 3.13% , 2.62% and 0.90%.

References

[1] B. Qin, B. Hui, L. Wang, M. Yang, J. Li, B. Li, R. Geng, R. Cao, J. Sun, L. Si, F. Huang, Y. Li, A survey on text-to-sql parsing: Concepts, methods, and future directions, CoRR abs/2208.13629 (2022).

[2] R. Sun, S. Ö. Arik, H. Nakhost, H. Dai, R. Sinha, P. Yin, T. Pfister, Sql-palm: Improved large language model adaptation for text-to-sql, CoRR abs/2306.00739 (2023).

[3] J. Li, B. Hui, G. Qu, J. Yang, B. Li, B. Li, B. Wang, B. Qin, R. Geng, N. Huo, X. Zhou, C. Ma, G. Li, K. C. Chang, F. Huang, R. Cheng, Y. Li, Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls (2024).

[4] V. Karpukhin, B. Oguz, S. Min, P. S. H. Lewis, L. Wu, S. Edunov, D. Chen, W. Yih, Dense passage retrieval for open-domain question answering, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 6769–6781.

[5] C. Tai, Z. Chen, T. Zhang, X. Deng, H. Sun, Exploring chain of thought style prompting for text-to-sql, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, Association for Computational Linguistics, 2023, pp. 5376–5393.

[6] D. Gao, H. Wang, Y. Li, X. Sun, Y. Qian, B. Ding, J. Zhou, Text-to-sql empowered by large language models: A benchmark evaluation, Proc. VLDB Endow. 17 (2024) 1132–1145.

[7] M. Pourreza, D. Rafiei, DIN-SQL: decomposed in-context learning of text-to-sql with self-correction, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023.

[8] B. Wang, C. Ren, J. Yang, X. Liang, J. Bai, Q. Zhang, Z. Yan, Z. Li, MAC-SQL: A multi-agent collaborative framework for text-to-sql, CoRR abs/2312.11242 (2023).

[9] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman, Z. Zhang, D. R. Radev, Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task, in: E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii (Eds.), Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, Association for Computational Linguistics, 2018, pp. 3911–3921.