Next-Generation Database Interfaces: A Survey of LLM-based Text-to-SQL

Zijin Hong[1], Zheng Yuan[2], Qinggang Zhang[2], Hao Chen[2], Junnan Dong[2], Feiran Huang[1], and Xiao Huang[2]*

[1]*Jinan University, Guangzhou, China*

[2]*The Hong Kong Polytechnic University, Hong Kong SAR, China*

hongzijin@stu2020.jnu.edu.cn, yzheng.yuan@connect.polyu.hk, qinggangg.zhang@connect.polyu.hk, sundaychenhao@gmail.com, hanson.dong@connect.polyu.hk, huangfr@jnu.edu.cn, xiaohuang@comp.polyu.edu.hk

***Abstract*—Generating accurate SQL from natural language questions (text-to-SQL) is a long-standing challenge due to the complexities in user question understanding, database schema comprehension, and SQL generation. Conventional text-to-SQL systems, comprising human engineering and deep neural networks, have made substantial progress. Subsequently, pre-trained language models (PLMs) have been developed and utilized for text-to-SQL tasks, achieving promising performance. As modern databases become more complex, the corresponding user questions also grow more challenging, causing PLMs with parameter constraints to produce incorrect SQL. This necessitates more sophisticated and tailored optimization methods, which, in turn, restricts the applications of PLM-based systems. Recently, large language models (LLMs) have demonstrated significant capabilities in natural language understanding as the model scale increases. Therefore, integrating LLM-based implementation can bring unique opportunities, improvements, and solutions to text-to-SQL research. In this survey, we present a comprehensive review of LLM-based text-to-SQL. Specifically, we propose a brief overview of the technical challenges and the evolutionary process of text-to-SQL. Then, we provide a detailed introduction to the datasets and metrics designed to evaluate text-to-SQL systems. After that, we present a systematic analysis of recent advances in LLM-based text-to-SQL. Finally, we discuss the remaining challenges in this field and propose expectations for future research directions.**

***Index Terms*—text-to-SQL, large language models, database, natural language understanding**

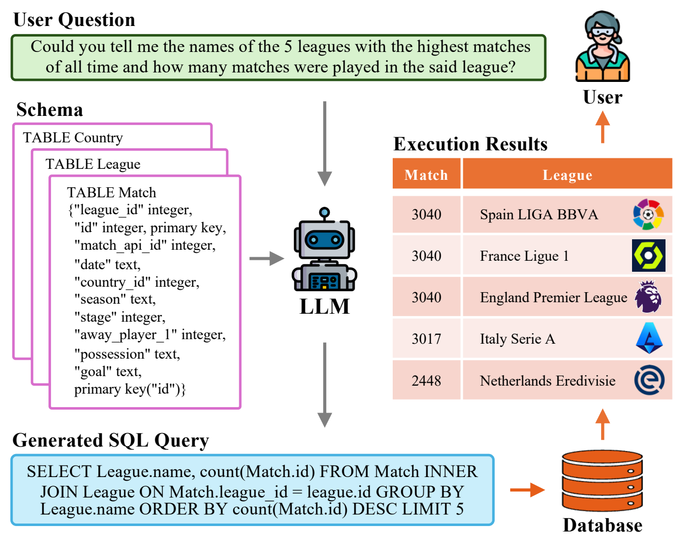

Fig. 1: An example for LLM-based text-to-SQL selected from the BIRD dataset. A user proposes a question about football leagues. The LLM takes the question and the schema of its corresponding database as the input and then generates an SQL query as the output. The SQL query can be executed in the database and retrieve the content *"The 5 leagues with the highest matches"* to answer the user question.

I. Introduction

TEXT-TO-SQL is a long-standing task in natural language processing research. It aims to convert (translate) natural language questions into database-executable SQL queries. Fig. 1 provides an example of a large language model-based (LLM-based) text-to-SQL system. Given a user question such as *"Could you tell me the names of the 5 leagues with the highest matches of all time and how many matches were played in the said league?"*, the LLM takes the question and its corresponding database schema as input and then generates an SQL query as output. This SQL query can be executed in the database to retrieve the relevant content to answer the user's question. The above system builds a natural language interface to the database (NLIDB) with LLMs. Since SQL remains one of the most widely used programming languages, with over half (51.52%) of professional developers using SQL in their work, it is notable that only around a third (35.29%) of those developers are systematically trained[1], the NLIDB enables non-skilled users to access structured databases like professional database engineers [1, 2] and also accelerates human-computer interaction [3]. Furthermore, amid the research hotspot of LLMs, text-to-SQL can provide a potential solution to the prevalent hallucination [4, 5] issue by incorporating realistic content from the database to fill the knowledge gaps of LLMs [6]. The significant value and potential for text-to-SQL have triggered a range of studies on its integration and optimization with LLMs [7–10]; consequently, LLM-based text-to-SQL remains a highly discussed research field within the NLP and database communities.

* Corresponding author.

[1] https://survey.stackoverflow.co/2023

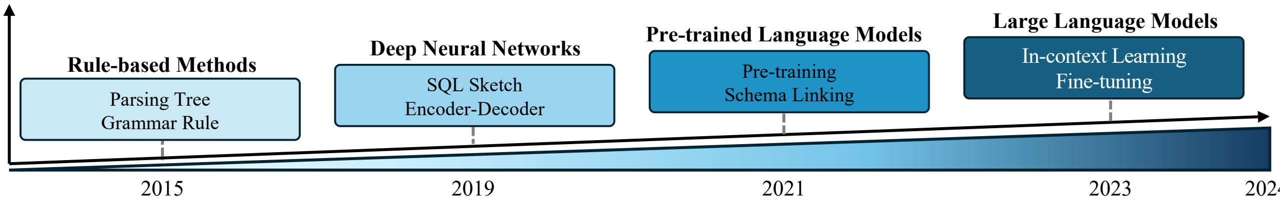

Fig. 2: A sketch of the evolutionary process for text-to-SQL research from the perspective of implementation paradigm. Each stage is presented with two representative implementation techniques. The timestamps for the stages are not exactly accurate; we set each timestamp according to the release time of the representative works of each implementation paradigm, with a margin of error of about one year before and after. The format is inspired from [29].

Previous studies have made notable progress in implementing text-to-SQL and have undergone a long evolutionary process. Early efforts were mostly based on well-designed rules and templates [11], specifically suitable for simple database scenarios. In recent years, with the heavy labor costs [12] brought by rule-based methods and the growing complexity of database environments [13–15], designing a rule or template for each scenario has become increasingly difficult and impractical. The development of deep neural networks has advanced the progress of text-to-SQL [16, 17], which can automatically learn a mapping from the user question to its corresponding SQL [18, 19]. Subsequently, pre-trained language models (PLMs) with strong semantic parsing capacity have become the new paradigm for text-to-SQL systems [20], boosting their performance to a new level [21–23]. Incremental research on PLM-based optimization, such as table content encoding [19, 24, 25] and pre-training [20, 26], has further advanced this field. Most recently, the LLM-based approaches implementing text-to-SQL through in-context learning (ICL) [8] and fine-tuning (FT) [10] paradigm, reaching state-of-the-art accuracy with the well-designed framework and stronger comprehension capability compared to PLMs.

The overall implementation details of LLM-based text-to-SQL can be divided into 3 aspects: **1. Question understanding**: The NL question is a semantic representation of the user's intention, which the corresponding generated SQL query is expected to align with; **2. Schema comprehension**: The schema provides the table and column structure of the database, and the text-to-SQL system is required to identify the target components that match the user question; **3. SQL generation**: This involves incorporating the above parsing and then predicting the correct syntax to generate executable SQL queries that can retrieve the required answer. The LLMs have proven to perform a good vanilla implementation [7, 27], benefiting from the more powerful semantic parsing capacity enabled by the richer training corpus [28, 29]. Further studies on enhancing the LLMs for question understanding [8, 9], schema comprehension [30, 31], and SQL generation [32] are being increasingly released.

Despite the significant progress made in text-to-SQL research, several challenges remain that hinder the development of robust and generalized text-to-SQL systems [73]. Related works in recent years have surveyed the text-to-SQL systems in deep learning approaches and provided insights into the previous deep neural network and PLM-based research [2, 29, 74]. In this survey, we aim to catch up with the recent advances and provide a comprehensive review of the current state-of-the-art models and approaches in LLM-based text-to-SQL. We begin by introducing the fundamental concepts and challenges associated with text-to-SQL, highlighting the importance of this task in various domains. We then delve into the evolution of the implementation paradigm for text-to-SQL systems, discussing the key advancements and breakthroughs in this field. After the overview, we provide a detailed introduction and analysis of the recent advances in text-to-SQL integrating LLMs. Specifically, the body of our survey covers a range of contents related to LLM-based text-to-SQL, including:

- Datasets and Benchmarks: We provide a detailed introduction to the commonly used datasets and benchmarks for evaluating LLM-based text-to-SQL systems. We discuss their characteristics, complexity, and the challenges they pose for text-to-SQL development and evaluation.
- Evaluation Metrics: We present the evaluation metrics used to assess the performance of LLM-based text-to-SQL systems, including both content matching-based and execution-based paradigms. We then briefly introduce the characteristics of each metric.
- Methods and Models: We present a systematic analysis of the different methods and models employed for LLM-based text-to-SQL, including in-context learning and fine-tuning-based paradigms. We discuss their implementation details, strengths, and adaptations specific to the text-to-SQL task from various implementation perspectives.
- Expectations and Future Directions: We discuss the remaining challenges and limitations of LLM-based text-to-SQL, such as real-world robustness, computational efficiency, data privacy, and extensions. We also outline potential future research directions and opportunities for improvement and optimization.

We hope this survey provides a clear overview of recent studies and inspires future research. Fig. 3 shows a taxonomy tree that summarizes the structure and contents of our survey.

II. Overview

Text-to-SQL is a task that aims to convert natural language questions into corresponding SQL queries that can be executed in a relational database. Formally, given a user question $\mathcal{Q}$ (also known as a user query, natural language question, etc.)