Natural Language to SQL: State of the Art and Open Problems

Yuyu Luo[*]
HKUST (GZ)
yuyuluo@hkust-gz.edu.cn

Guoliang Li
Tsinghua University
liguoliang@tsinghua.edu.cn

Ju Fan
Renmin University of China
fanj@ruc.edu.cn

Chengliang Chai
Beijing Institute of Technology
ccl@bit.edu.cn

Nan Tang
HKUST (GZ)
nantang@hkust-gz.edu.cn

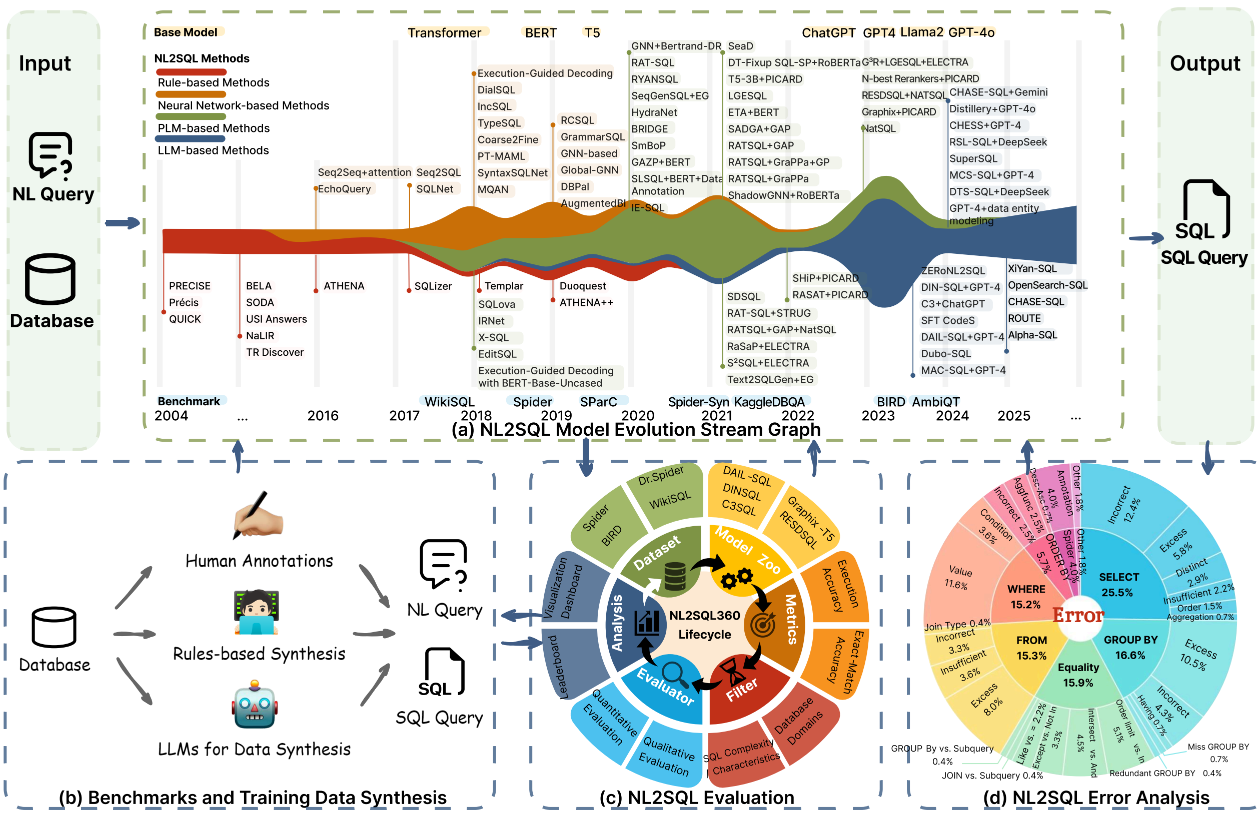

Figure 1: An Overview of the Tutorial: The Lifecycle of the NL2SQL Task (https://github.com/HKUSTDial/NL2SQL_Handbook).

ABSTRACT

Translating users' natural language queries (NL) into SQL queries (*i.e.,* NL2SQL) can significantly reduce barriers to accessing relational databases and support various commercial applications. The performance of NL2SQL has been greatly improved with the emergence of large language models (LLMs). In this context, it is crucial to assess our current position, determine the NL2SQL solutions that should be adopted for specific scenarios by practitioners, and identify the research topics that researchers should explore next.

In this tutorial, we will provide a comprehensive overview of NL2SQL techniques, covering every aspect of its lifecycle, from the collection and synthesis of training data, recent advancements in NL2SQL translation techniques using LLMs and agents, debugging NL2SQL processes, to multi-angle and scenario-based evaluation of NL2SQL methods. We conclude by highlighting the research challenges and open problems in NL2SQL.

PVLDB Reference Format:
Yuyu Luo, Guoliang Li, Ju Fan, Chengliang Chai, and Nan Tang. Natural Language to SQL: State of the Art and Open Problems. PVLDB, 18(12): 5466 - 5471, 2025.
doi:10.14778/3750601.3750696

Proceedings of the VLDB Endowment, Vol. 18, No. 12 ISSN 2150-8097.
doi:10.14778/3750601.3750696

[*]Yuyu Luo is the corresponding author.

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/HKUSTDial/NL2SQL_Handbook.

1 INTRODUCTION

Natural Language to SQL (*i.e.,*NL2SQL), which translates natural language queries (NL) into executable SQL queries, significantly lowers the barriers for users to access relational databases [6, 15, 18, 39, 42–44, 49, 62]. Recent advances in language models have notably expanded the capabilities and adoption of NL2SQL techniques, prompting database vendors to integrate NL2SQL solutions as essential offerings [40]. Thus, understanding the core methods, recent innovations, and practical challenges of NL2SQL has become increasingly critical.

In this tutorial, we will systematically review recent NL2SQL techniques through a new framework, as shown in Figure 1. We will first review four major categories of representative methods in the past decade (see Figure 1(a)). We then zoom in on the recent advances of tunable pre-trained language models (PLMs) and large language models (LLMs) for the NL2SQL translation. Then, the performance of learning-based NL2SQL models is highly dependent on the quality of the training data. Therefore, we will summarize available benchmarks and discuss how to collect and synthesize high-quality training data (see Figure 1(b)). In addition, NL2SQL model evaluation is crucial for optimizing and selecting models. We will discuss multi-angle evaluation and scenario-based evaluation for the NL2SQL task (see Figure 1(c)). Furthermore, the NL2SQL model may generate incorrect SQL queries that are not equivalent to the NL queries, such as selecting the wrong columns in the SELECT clause. As shown in Figure 1(d), we analyze common NL2SQL errors and categorize them into seven types of SQL errors and annotation errors in benchmarks (*e.g.,* BIRD). Undoubtedly, it is crucial to detect whether the generated SQL are correct, to trace back to the reasons if they are incorrect, and then to correct them, as this can enhance the trustworthiness of the NL2SQL solution. We will introduce the NL2SQL debugging problem and preliminary solutions.

1.1 Tutorial Overview

We will give a 3-hour lecture-style tutorial.

Part I: Problem Definition and Preliminaries.

(i) Problem and Challenges: We will begin by introducing the motivation and problem definition of NL2SQL. Next, we will elaborate on the key challenges faced by researchers and practitioners.

(ii) Literature Review on PLMs, LLMs, and Agents: We will provide an in-depth review of the literature on PLMs, LLMs and LLM Agents. We will examine their evolution, capabilities, and applications in NL2SQL and related tasks, highlighting their potential to address existing challenges and advance the state of the art [57, 64, 68, 71].

Part II: NL2SQL Solutions with PLMs and LLM Agents

(i) PLM-based NL2SQL Solutions:. We then elaborate on PLM-based NL2SQL architectures and methods. Specifically, we will elaborate on data-centric approaches, including high-quality training data synthesis [19, 45, 65, 69], and model-centric methods, focusing on the model design perspective [18, 33, 36, 54].

(ii) LLM-based NL2SQL *Solutions:* We will cover how to harness the LLMs for the NL2SQL task using prompt engineering techniques [9, 16, 49]. We will then introduce how to further improve LLM-based NL2SQL by leveraging the supervised fine-tuning [16, 30], multi-agent framework [60], and agentic workflow [31].

(iii) Modularized NL2SQL *Solutions:* Modularized NL2SQL solutions use distinct modules for specific sub-tasks (e.g., schema linking), offering better flexibility, adaptability, and error handling [30, 46]. We will introduce the key designs of these solutions [17, 30] and examine how LLM agents can augment them [48, 56].

Part III. Benchmarks and Evaluation.

(i) Benchmarks: We will categorize available benchmarks and highlight their limitations [8, 28, 35, 41].

(ii) Multi-angle and Scenario-based Evaluations: We will first review existing evaluation methods [13]. Then, we will discuss the importance of multi-angle, scenario-based evaluation for model selection and training data synthesis [7, 30, 45, 69].

(iii) Training Data Synthesis: We will also discuss how to automatically synthesize high-quality training data to enhance model training and facilitate domain adaptation [32].

Part IV. Debugging and Open Problems.

(i) NL2SQL *Debugging:* We will first introduce the NL2SQL debugging problem. Next, we will discuss the design goals, choices, and current progress toward a robust NL2SQL debugger [40, 41].

(ii) Open Problems: We will discuss key research opportunities.

1.2 Our Distinction

Differences from Existing Tutorials. Our tutorial distinguishes itself from existing tutorials [23, 24, 37, 47] in three aspects.

(1) *Comprehensive Lifecycle Review.* We systematically review *the entire lifecycle of* NL2SQL *problem*, as shown in Figure 1. This lifecycle includes training data collection and synthesis methods (Figure 1(b)), various NL2SQL translation methodologies (Figure 1(a)), highlighting the importance of *evaluating* NL2SQL *methods through a multifaceted approach* (Figure 1(c)), and NL2SQL debugging techniques (Figure 1(d)).

(2) *Focus on LLM-based and Modularized Solutions.* We explore *LLM-based* methods, discuss the design of *modularized* solutions, and emphasize the latest advancements in *LLM agents* for NL2SQL.

(3) *Introducing the NL2SQL Debugging Problem.* We highlight the emerging NL2SQL *debugging problem* and its challenges.

Target Audience. This tutorial is designed for a diverse group of VLDB attendees, including researchers, developers, practitioners, and students. *Researchers* will derive insights from the pros and cons of existing NL2SQL techniques and explore new topics and research problems. *Developers and practitioners* will deepen their understanding of the core techniques behind NL2SQL solutions, enabling them to select or enhance NL2SQL systems that are best suited to their specific applications and business needs. *Students* will be introduced to essential techniques and research topics within the NL2SQL field, laying a solid foundation for their research. The tutorial will be self-contained, but we assume some familiarity with SQL, database, and language models terminology.

2 TUTORIAL OUTLINE

2.1 Background

Problem Description. Given a natural language query (NL) and a database consisting of tables $\{T_1, \ldots, T_n\}$, the goal of NL2SQL (*a.k.a.*

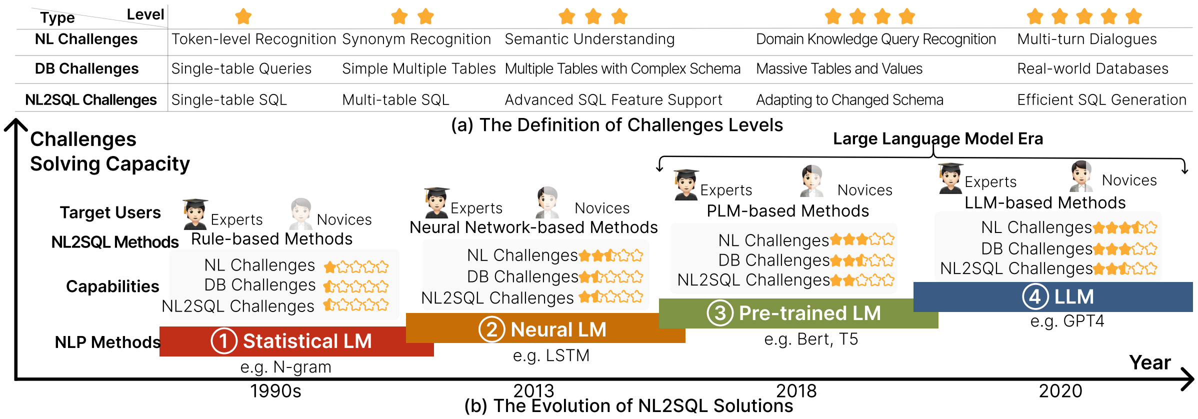

Figure 2: The Evolution of NL2SQL Solutions from the Perspective of Language Models.

Text-to-SQL) is to generate an SQL query that accurately represents the semantics of the original NL.

NL2SQL Task Challenges. There are several key challenges:

*(**NL Challenges**) Ambiguous or Underspecified* NL *Queries.* Natural language queries may lack sufficient detail or contain ambiguities, making it difficult to infer the precise intent.

*(**DB Challenges**) Complex and Ambiguous Database Schemas.* Real-world databases often feature complex structures and ambiguous relationships. In addition, incomplete, inconsistent, or noisy data further increases the difficulty of aligning NL queries with the underlying database content.

*(**NL2SQL Translation Challenges**) Intent Alignment and Generating Semantically Equivalent SQL.* Unlike flexible NL queries, SQL queries must follow strict syntax, demanding precise translations for executable queries. A single NL query can map to multiple valid SQL queries, creating ambiguity in determining the most appropriate output. Furthermore, NL2SQL translation must account for schema dependencies, as variations in schema design can produce different SQL queries for the same NL query, requiring models to generalize across diverse real-world schemas effectively.

Difficulty Levels *vs.* The Evolution of NL2SQL Solutions. We categorize the challenges of NL2SQL into five distinct levels, as depicted in Figure 2(a). The first three levels include challenges that have been resolved or are actively being tackled, showcasing the steady progress in NL2SQL capabilities. The fourth level focuses on current challenges addressed by LLM-based solutions, while the fifth level outlines future challenges, reflecting our vision for advancing NL2SQL over the next five years [40]. As depicted in Figure 2(b), NL2SQL solutions have evolved significantly over time.

2.2 PLM-based NL2SQL Methods

With the introduction of Transformer [59] [66] around 2017, pre-trained language models (PLMs) such as T5 significantly advanced NL2SQL capabilities [33, 36, 54].

Recent works primarily focus on two aspects: (i) developing new model architectures and learning strategies [2, 14, 18, 22, 26, 33, 36, 51, 54, 61], such as SC-Prompt's divide-and-conquer approach with hybrid prompt-tuning [18]; and (ii) acquiring high-quality training data through automatic or semi-automatic synthesis and augmentation methods, aiming at improving model performance, robustness, and domain adaptability [19, 21, 45, 65, 69].

2.3 LLM-based NL2SQL Methods

Recently, the emergence of large language models like ChatGPT and GPT-4 has triggered a new wave of solutions. These LLM-based NL2SQL methods have become the most representative solutions in the current NL2SQL landscape [3, 4, 27, 30, 53, 55, 67, 70].

Prompting-based Methods. We will first show how prompt engineering techniques can harness the capabilities of LLMs for the NL2SQL task [16, 49]. We then highlight their challenges in handling large and complex database schemas and incur significant monetary costs when relying on closed-source LLMs. Finally, we will share insights into developing cost-effective NL2SQL solutions, such as EllieSQL [72], which employs complexity-aware routing to enhance cost-efficiency by assigning queries to suitable generators.

Supervised Fine-tuning Methods. We will then take a close look at how to leverage the supervised fine-tuning technique to further enhance LLM-based NL2SQL methods, which involves training the LLM on a curated dataset of (NL, SQL) pairs to improve its accuracy and reliability in specific scenarios [16, 30, 34].

LLM Agents for NL2SQL. Finally, we discuss the integration of LLM agents into the NL2SQL pipeline, examining how these agents leverage advanced reasoning, multi-step problem-solving, and decision-making capabilities to handle complex queries across diverse domains [7, 31, 48, 56].

2.4 Modularized NL2SQL Solutions

Recent studies are exploring the decomposition of end-to-end NL2SQL into several steps, aiming to define the design space for modularized NL2SQL solutions [10, 17, 30, 33, 50, 52, 60].

Key Modules in NL2SQL Solutions. Recent NL2SQL methods typically rely on ***language models*** (*e.g.*, GPT-4o, LLaMA) as their backbone for interpreting natural language queries and database schemas. A crucial step is ***schema linking***, which explicitly maps elements of the NL query to database schema components [29, 33]. Additionally, incorporating ***database content*** further improves schema understanding and query accuracy. During SQL generation, most methods adopt ***output refinement*** strategies, such as constrained decoding (*e.g.,* PICARD [54]) and heuristic prompting techniques such as Self-Consistency [11, 38, 63].

Multi-Agent Framework for NL2SQL. We have already discussed how to decompose NL2SQL tasks into subtasks. Intuitively, we can deploy LLM agents to specifically tackle various sub-tasks, thereby enhancing the overall performance of NL2SQL tasks. The key challenges of this framework lie in defining appropriate sub-tasks, customizing different LLM-based agents for each specific task, and ensuring effective collaboration among them [60]. A prominent example is Alpha-SQL [58], which proposes a planning-centric autonomous agent framework that combines LLMs with Monte Carlo Tree Search (MCTS). This agent dynamically selects and activates appropriate modules, such as schema linking and SQL generation, based on contextual reasoning and execution-based feedback.

2.5 Benchmarks and Multi-Angle Evaluations

Benchmarks. With advancements in NL2SQL, various datasets have been developed to address the evolving challenges in the field. Key benchmarks include BIRD [35], Spider [66], Dr.Spider [5], AmbiQT [1], ScienceBenchmark [69], among others [8, 25, 28]. These can be used to train and evaluate NL2SQL models, including assessing robustness (Dr.Spider) and the ability to handle ambiguous NL (AmbiQT). There is also a line of work emphasizing the critical role of synthesized training data in the NL2SQL task [20, 21].

Metrics. Typical metrics for evaluating NL2SQL effectiveness include Execution Accuracy and Exact Match Accuracy [66]. Recently, SuperSQL proposed Query Variance Testing [30] to further assess model robustness under variations in natural language queries.

Evaluation Toolkits. Effectively evaluating NL2SQL methods and guiding users toward suitable models for specific scenarios remains challenging [12]. We briefly summarize existing benchmarks and metrics for NL2SQL evaluation, followed by recent tools enabling fine-grained evaluation and model comparison [30].

2.6 NL2SQL Results Debugging

NL2SQL solutions can definitely produce incorrect SQL queries. Detecting and repairing these SQL queries is crucial for developing a trustworthy NL2SQL solution. To this end, NL2SQL results debugging is an option. The key task is to detect whether the generated SQL queries are semantically equivalent to the NL query [40].

To understand the types of errors present in SQL queries generated by existing NL2SQL methods, NL2SQL-BUGs [41] adopts a two-level taxonomy to systematically classify semantic errors, covering 9 main categories and 31 subcategories. NL2SQL-BUGs also proposes a benchmark for semantic error detection and uses it to test current LLMs. This analysis can help in building a robust NL2SQL results debugger. We will also discuss the design choices and current progress toward a robust NL2SQL results debugger.

2.7 Research Opportunities

We summarize open problems to further advance NL2SQL methods:

Multi-Database NL2SQL Problem. Real-world applications often require queries that span multiple databases with heterogeneous schemas. Key challenges include how to dynamically select relevant databases, accurately integrate their diverse schemas, effectively aggregate query results, and adapt queries across domains.

Trustworthy and Interpretable NL2SQL. Existing NL2SQL methods often produce inaccurate or unreliable queries due to ambiguous natural language inputs and inconsistent schemas, hindering user trust. Key challenges include how to automatically clarify ambiguous queries, transparently interpret SQL query logic, and provide interactive debugging support to improve overall reliability.

Interactive NL2SQL Systems. Complex database tasks often require expert construction of sophisticated SQL queries. Key challenges include how to enable users to interactively and incrementally build queries, combining automatic SQL generation with expert-driven adjustments seamlessly.

Cost-effective NL2SQL Solutions. Although powerful, LLM-based NL2SQL approaches incur substantial computational costs and inference delays due to extensive token consumption. Key challenges include how to reduce inference expenses through modularized designs, multi-agent collaboration, and adaptive training-data generation driven by model feedback.

3 BIOGRAPHY

Yuyu Luo is an Assistant Professor at The Hong Kong University of Science and Technology (Guangzhou), with an affiliated position at the HKUST. He received his PhD from Tsinghua University in 2023. His research interests include NL2SQL and LLMs for Agents for Databases. He has received the Best-of-SIGMOD 2023 Papers.

Guoliang Li is a full professor in the Department of Computer Science, Tsinghua University. His research interests mainly include data cleaning and integration and machine learning for databases. He got VLDB 2017 early research contribution award, TCDE 2014 Early Career Award, VLDB 2023 Industry Best Paper Runner-up, Best of SIGMOD 2023, SIGMOD 2023 research highlight award, DASFAA 2023 Best Paper Award, and CIKM 2017 Best Paper Award.

Ju Fan is a Professor at the DEKE Lab, MOE China, and School of Information, Renmin University of China. He received his PhD from Tsinghua University in 2013 and received the ACM China Rising Star Award and the 2023 SIGMOD Research Highlight Award. Dr. Fan's main research interests are NL2SQL and database systems.

Chengliang Chai is an Associate Professor at School of Computer Science & Technology, Beijing Institute of Technology, China. He received his PhD from Tsinghua University in 2020 and received the ACM China Doctoral Dissertation Award and Best-of-SIGMOD 2023 Papers. Dr. Chai's main research interest is database systems.

Nan Tang is an Associate Professor at The Hong Kong University of Science and Technology (Guangzhou), with an affiliated position at the HKUST. He has received the VLDB 2010 Best Paper Award, the 2023 SIGMOD Research Highlight Award, and the Best-of-SIGMOD 2023. His main research interests are NL2SQL and data-centric AI.

ACKNOWLEDGMENTS

This paper was supported by National Key R&D Program of China (2023YFB4503600, 2024YFC3308200); the NSF of China (62402409, 62525202, 62232009, 62436010, and 62441230); Guangdong Basic and Applied Basic Research Foundation (2023A1515110545); Guangzhou Basic and Applied Basic Research Foundation (2025A04J3935); Guangzhou-HKUST(GZ) Joint Funding Program (2025A03J3714); Guangdong Provincial Project (2023CX10X008); Beijing Nova Program; and CCF-Baidu Open Fund (CCF-Baidu202402).

REFERENCES

[1] Adithya Bhaskar, Tushar Tomar, Ashutosh Sathe, and et al. 2023. Benchmarking and Improving Text-to-SQL Generation under Ambiguity. In *EMNLP*.

[2] Ursin Brunner and Kurt Stockinger. 2021. ValueNet: A Natural Language-to-SQL System that Learns from Database Information. In *ICDE*. IEEE, 2177–2182.

[3] Hasan Alp Caferoğlu and Özgür Ulusoy. 2024. E-SQL: Direct Schema Linking via Question Enrichment in Text-to-SQL. arXiv:2409.16751 [cs.CL] https://arxiv.org/abs/2409.16751

[4] Zhenbiao Cao, Yuanlei Zheng, Zhihao Fan, Xiaojin Zhang, Wei Chen, and Xiang Bai. 2024. RSL-SQL: Robust Schema Linking in Text-to-SQL Generation. arXiv:2411.00073 [cs.CL] https://arxiv.org/abs/2411.00073

[5] Shuaichen Chang, Jun Wang, Mingwen Dong, Lin Pan, and et al. 2023. Dr.Spider: A Diagnostic Evaluation Benchmark towards Text-to-SQL Robustness. In *ICLR*.

[6] Ziru Chen, Shijie Chen, Michael White, Raymond J. Mooney, and et al. 2023. Text-to-SQL Error Correction with Language Models of Code. In *ACL*.

[7] Yeounoh Chung, Gaurav T. Kakkar, Yu Gan, Brenton Milne, and Fatma Ozcan. 2025. Is Long Context All You Need? Leveraging LLM's Extended Context for NL2SQL. arXiv:2501.12372 [cs.DB] https://arxiv.org/abs/2501.12372

[8] Xiang Deng, Ahmed Hassan Awadallah, Christopher Meek, and et al. 2021. Structure-Grounded Pretraining for Text-to-SQL. In *NAACL-HLT*.

[9] Xuemei Dong, Chao Zhang, Yuhang Ge, and et al. 2023. C3: Zero-shot Text-to-SQL with ChatGPT. *arXiv preprint arXiv:2307.07306* (2023).

[10] Ben Eyal, Moran Mahabi, Ophir Haroche, Amir Bachar, and Michael Elhadad. 2023. Semantic Decomposition of Question and SQL for Text-to-SQL Parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 13629–13645. https://doi.org/10.18653/v1/2023.findings-emnlp.910

[11] Yuankai Fan, Zhenying He, Tonghui Ren, Can Huang, Yinan Jing, Kai Zhang, and X. Sean Wang. 2024. Metasql: A Generate-then-Rank Framework for Natural Language to SQL Translation. arXiv:2402.17144 [cs.DB] https://arxiv.org/abs/2402.17144

[12] Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir R. Radev. 2018. Improving Text-to-SQL Evaluation Methodology. In *ACL (1)*. Association for Computational Linguistics, 351–360.

[13] Avrilia Floratou, Fotis Psallidas, Fuheng Zhao, Shaleen Deep, Gunther Hagleither, Wangda Tan, and et al. 2024. NL2SQL is a solved problem... Not!. In *CIDR*.

[14] Han Fu, Chang Liu, Bin Wu, Feifei Li, Jian Tan, and Jianling Sun. 2023. CatSQL: Towards Real World Natural Language to SQL Applications. *Proc. VLDB Endow.* 16, 6 (Feb. 2023), 1534–1547. https://doi.org/10.14778/3583140.3583165

[15] Yujian Gan, Xinyun Chen, Jinxia Xie, Matthew Purver, John R Woodward, John Drake, and Qiaofu Zhang. 2021. Natural SQL: Making SQL easier to infer from natural language specifications. *arXiv preprint arXiv:2109.05153* (2021).

[16] Dawei Gao, Haibin Wang, Yaliang Li, and et al. 2024. Text-to-sql empowered by large language models: A benchmark evaluation. *Proc. VLDB Endow.* (2024).

[17] Zihui Gu and et al. 2023. Interleaving Pre-Trained Language Models and Large Language Models for Zero-Shot NL2SQL Generation. *CoRR* 2306.08891 (2023).

[18] Zihui Gu, Ju Fan, Nan Tang, and et al. 2023. Few-shot Text-to-SQL Translation using Structure and Content Prompt Learning. *Proc. ACM Manag. Data* (2023).

[19] Yiqun Hu, Yiyun Zhao, Jiarong Jiang, and et al. 2023. Importance of Synthesizing High-quality Data for Text-to-SQL Parsing. In *ACL (Findings)*.

[20] Yiqun Hu, Yiyun Zhao, Jiarong Jiang, Wuwei Lan, Henghui Zhu, Anuj Chauhan, Alexander Hanbo Li, Lin Pan, Jun Wang, Chung-Wei Hang, Sheng Zhang, Jiang Guo, Mingwen Dong, Joseph Lilien, Patrick Ng, Zhiguo Wang, Vittorio Castelli, and Bing Xiang. 2023. Importance of Synthesizing High-quality Data for Text-to-SQL Parsing. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 1327–1343. https://doi.org/10.18653/v1/2023.findings-acl.86

[21] Zezhou Huang, Shuo Zhang, Kechen Liu, and Eugene Wu. 2024. Data-Centric Text-to-SQL with Large Language Models. In *NeurIPS 2024 Third Table Representation Learning Workshop*. https://openreview.net/forum?id=gDKIjZcg93

[22] Binyuan Hui, Ruiying Geng, Lihan Wang, Bowen Qin, Yanyang Li, Bowen Li, Jian Sun, and Yongbin Li. 2022. S^2SQL: Injecting Syntax to Question-Schema Interaction Graph Encoder for Text-to-SQL Parsers. In *Findings of the Association for Computational Linguistics: ACL 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 1254–1262. https://doi.org/10.18653/v1/2022.findings-acl.99

[23] George Katsogiannis-Meimarakis and et al. 2023. Natural Language Interfaces for Databases with Deep Learning. *Proc. VLDB Endow.* (2023).

[24] George Katsogiannis-Meimarakis and Georgia Koutrika. [n.d.]. A Deep Dive into Deep Learning Approaches for Text-to-SQL Systems. In *SIGMOD 2021*.

[25] Chia-Hsuan Lee, Oleksandr Polozov, and Matthew Richardson. [n.d.]. KaggleDBQA: Realistic Evaluation of Text-to-SQL Parsers. In *ACL/IJCNLP 2021*.

[26] Dongjun Lee. 2019. Clause-Wise and Recursive Decoding for Complex and Cross-Domain Text-to-SQL Generation. arXiv:1904.08835 [cs.CL] https://arxiv.org/abs/1904.08835

[27] Dongjun Lee, Choongwon Park, Jaehyuk Kim, and Heesoo Park. 2024. MCS-SQL: Leveraging Multiple Prompts and Multiple-Choice Selection For Text-to-SQL Generation. arXiv:2405.07467 [cs.CL] https://arxiv.org/abs/2405.07467

[28] Fangyu Lei, Jixuan Chen, Yuxiao Ye, Ruisheng Cao, Dongchan Shin, Hongjin Su, Zhaoqing Suo, Hongcheng Gao, Wenjing Hu, Pengcheng Yin, Victor Zhong, Caiming Xiong, Ruoxi Sun, Qian Liu, Sida Wang, and Tao Yu. 2024. Spider 2.0: Evaluating Language Models on Real-World Enterprise Text-to-SQL Workflows. arXiv:2411.07763 [cs.CL] https://arxiv.org/abs/2411.07763

[29] Wenqiang Lei, Weixin Wang, Zhixin Ma, Tian Gan, Wei Lu, Min-Yen Kan, and Tat-Seng Chua. 2020. Re-examining the Role of Schema Linking in Text-to-SQL. In *EMNLP (1)*. Association for Computational Linguistics, 6943–6954.

[30] Boyan Li, Yuyu Luo, and et al. 2024. The Dawn of Natural Language to SQL: Are We Fully Ready? *Proc. VLDB Endow.* (2024).

[31] Boyan Li, Jiayi Zhang, Ju Fan, Yanwei Xu, Chong Chen, Nan Tang, and Yuyu Luo. 2025. Alpha-SQL: Zero-Shot Text-to-SQL using Monte Carlo Tree Search. arXiv:2502.17248 [cs.DB] https://arxiv.org/abs/2502.17248

[32] Haoyang Li, Shang Wu, Xiaokang Zhang, Xinmei Huang, Jing Zhang, Fuxin Jiang, Shuai Wang, Tieying Zhang, Jianjun Chen, Rui Shi, Hong Chen, and Cuiping Li. 2025. OmniSQL: Synthesizing High-quality Text-to-SQL Data at Scale. arXiv:2503.02240 [cs.CL] https://arxiv.org/abs/2503.02240

[33] Haoyang Li, Jing Zhang, Cuiping Li, and Hong Chen. 2023. RESDSQL: Decoupling Schema Linking and Skeleton Parsing for Text-to-SQL. arXiv:2302.05965 [cs.CL]

[34] Haoyang Li, Jing Zhang, Hanbing Liu, Ju Fan, Xiaokang Zhang, Jun Zhu, Renjie Wei, Hongyan Pan, Cuiping Li, and Hong Chen. 2024. CodeS: Towards Building Open-source Language Models for Text-to-SQL. arXiv:2402.16347 [cs.CL] https://arxiv.org/abs/2402.16347

[35] Jinyang Li and et al. 2023. Can LLM Already Serve as A Database Interface? A BIg Bench for Large-Scale Database Grounded Text-to-SQLs. *CoRR* (2023).

[36] Jinyang Li and et al. 2023. Graphix-t5: Mixing pre-trained transformers with graph-aware layers for text-to-sql parsing. *arXiv:2301.07507* (2023).

[37] Yunyao Li and Davood Rafiei. 2017. Natural Language Data Management and Interfaces: Recent Development and Open Challenges. In *SIGMOD Conference*.

[38] Zhishuai Li, Xiang Wang, Jingjing Zhao, Sun Yang, Guoqing Du, Xiaoru Hu, Bin Zhang, Yuxiao Ye, Ziyue Li, Rui Zhao, and Hangyu Mao. 2024. PET-SQL: A Prompt-Enhanced Two-Round Refinement of Text-to-SQL with Cross-consistency. arXiv:2403.09732 [cs.CL] https://arxiv.org/abs/2403.09732

[39] Aiwei Liu, Xuming Hu, Li Lin, and Lijie Wen. 2022. Semantic Enhanced Text-to-SQL Parsing via Iteratively Learning Schema Linking Graph. In *KDD*.

[40] Xinyu Liu, Shuyu Shen, Boyan Li, Peixian Ma, Runzhi Jiang, Yuxin Zhang, Ju Fan, Guoliang Li, Nan Tang, and Yuyu Luo. 2024. A Survey of NL2SQL with Large Language Models: Where are we, and where are we going? arXiv:2408.05109 [cs.DB] https://arxiv.org/abs/2408.05109

[41] Xinyu Liu, Shuyu Shen, Boyan Li, Nan Tang, and Yuyu Luo. 2025. NL2SQL-BUGs: A Benchmark for Detecting Semantic Errors in NL2SQL Translation. arXiv:2503.11984 [cs.DB] https://arxiv.org/abs/2503.11984

[42] Yuyu Luo, Xuedi Qin, Nan Tang, and Guoliang Li. 2018. DeepEye: Towards Automatic Data Visualization. In *ICDE*. IEEE Computer Society, 101–112.

[43] Yuyu Luo, Nan Tang, Guoliang Li, Chengliang Chai, Wenbo Li, and Xuedi Qin. 2021. Synthesizing Natural Language to Visualization (NL2VIS) Benchmarks from NL2SQL Benchmarks. In *SIGMOD Conference*. ACM, 1235–1247.

[44] Yuyu Luo, Nan Tang, Guoliang Li, Jiawei Tang, Chengliang Chai, and Xuedi Qin. 2022. Natural Language to Visualization by Neural Machine Translation. *IEEE Trans. Vis. Comput. Graph.* 28, 1 (2022), 217–226.

[45] Pingchuan Ma and Shuai Wang. 2021. MT-Teql: Evaluating and Augmenting Neural NLIDB on Real-world Linguistic and Schema Variations. *VLDB* (2021).

[46] Karime Maamari, Fadhil Abubaker, Daniel Jaroslawicz, and Amine Mhedhbi. 2024. The Death of Schema Linking? Text-to-SQL in the Age of Well-Reasoned Language Models. *CoRR* abs/2408.07702 (2024).

[47] Fatma Özcan, Abdul Quamar, Jaydeep Sen, and et al. 2020. State of the art and open challenges in natural language interfaces to data. In *ACM SIGMOD*.

[48] Mohammadreza Pourreza, Hailong Li, Ruoxi Sun, Yeounoh Chung, Shayan Talaei, Gaurav Tarlok Kakkar, Yu Gan, Amin Saberi, Fatma Ozcan, and Sercan Ö. Arik. 2024. CHASE-SQL: Multi-Path Reasoning and Preference Optimized Candidate Selection in Text-to-SQL. *CoRR* abs/2410.01943 (2024).

[49] Mohammadreza Pourreza and Davood Rafiei. 2023. Din-sql: Decomposed in-context learning of text-to-sql with self-correction. *arXiv:2304.11015* (2023).

[50] Mohammadreza Pourreza and Davood Rafiei. 2024. DTS-SQL: Decomposed Text-to-SQL with Small Large Language Models. arXiv:2402.01117 [cs.CL] https://arxiv.org/abs/2402.01117

[51] Jiexing Qi, Jingyao Tang, Ziwei He, Xiangpeng Wan, Yu Cheng, Chenghu Zhou, Xinbing Wang, Quanshi Zhang, and Zhouhan Lin. 2022. RASAT: Integrating Relational Structures into Pretrained Seq2Seq Model for Text-to-SQL. arXiv:2205.06983 [cs.CL] https://arxiv.org/abs/2205.06983

[52] Nitarshan Rajkumar and et al. 2022. Evaluating the Text-to-SQL Capabilities of Large Language Models. *CoRR* abs/2204.00498 (2022).

[53] Tonghui Ren, Yuankai Fan, Zhenying He, Ren Huang, Jiaqi Dai, Can Huang, Yinan Jing, Kai Zhang, Yifan Yang, and X. Sean Wang. 2024. PURPLE: Making

a Large Language Model a Better SQL Writer. arXiv:2403.20014 [cs.DB] https://arxiv.org/abs/2403.20014

[54] Torsten Scholak and et al. 2021. PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models. arXiv:2109.05093 [cs.CL]

[55] Chang-Yu Tai, Ziru Chen, Tianshu Zhang, Xiang Deng, and Huan Sun. 2023. Exploring Chain of Thought Style Prompting for Text-to-SQL. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 5376–5393. https://doi.org/10.18653/v1/2023.emnlp-main.327

[56] Shayan Talaei, Mohammadreza Pourreza, Yu-Chen Chang, Azalia Mirhoseini, and Amin Saberi. 2024. CHESS: Contextual Harnessing for Efficient SQL Synthesis. *CoRR* abs/2405.16755 (2024).

[57] Nan Tang, Chenyu Yang, Ju Fan, Lei Cao, Yuyu Luo, and Alon Y. Halevy. 2024. VerifAI: Verified Generative AI. In *CIDR*. www.cidrdb.org.

[58] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, and et al. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[60] Bing Wang, Changyu Ren, Jian Yang, and et al. 2023. MAC-SQL: A Multi-Agent Collaborative Framework for Text-to-SQL. *CoRR* abs/2312.11242 (2023).

[61] Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020. RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers. In *ACL*. Association for Computational Linguistics, 7567–7578.

[62] Lihan Wang and et al. 2022. Proton: Probing Schema Linking Information from Pre-trained Language Models for Text-to-SQL Parsing. In *KDD*.

[63] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. arXiv:2203.11171 [cs.CL] https://arxiv.org/abs/2203.11171

[64] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 1800, 14 pages.

[65] Nathaniel Weir, Prasetya Ajie Utama, Alex Galakatos, and et al. 2020. DBPal: A Fully Pluggable NL2SQL Training Pipeline. In *SIGMOD Conference*.

[66] Tao Yu and et al. 2018. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In *EMNLP*.

[67] Hanchong Zhang, Ruisheng Cao, Hongshen Xu, Lu Chen, and Kai Yu. 2024. CoE-SQL: In-Context Learning for Multi-Turn Text-to-SQL with Chain-of-Editions. arXiv:2405.02712 [cs.CL] https://arxiv.org/abs/2405.02712

[68] Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. 2024. AFlow: Automating Agentic Workflow Generation. *CoRR* abs/2410.10762 (2024).

[69] Yi Zhang, Jan Deriu, and et al. 2023. ScienceBenchmark: A Complex Real-World Benchmark for Evaluating Natural Language to SQL Systems. *VLDB* (2023).

[70] Yuxin Zhang, Meihao Fan, Ju Fan, Mingyang Yi, Yuyu Luo, Jian Tan, and Guoliang Li. 2025. Reward-SQL: Boosting Text-to-SQL via Stepwise Reasoning and Process-Supervised Rewards. *CoRR* abs/2505.04671 (2025).

[71] Yizhang Zhu, Shiyin Du, Boyan Li, Yuyu Luo, and Nan Tang. 2024. Are Large Language Models Good Statisticians? *CoRR* abs/2406.07815 (2024).

[72] Yizhang Zhu, Runzhi Jiang, Boyan Li, Nan Tang, and Yuyu Luo. 2025. EllieSQL: Cost-Efficient Text-to-SQL with Complexity-Aware Routing. arXiv:2503.22402 [cs.DB] https://arxiv.org/abs/2503.22402