Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
() : –
© TÜBİTAK
doi:

Text-to-SQL: A methodical review of challenges and models

Ali Buğra KANBUROĞLU[1*], **F. Boray TEK**[1,2],
[1]Department of Computer Engineering, Işık University, İstanbul, Turkiye
[2]Department of Artificial Intelligence and Data Engineering, İstanbul Technical University, İstanbul, Turkiye

Received: 18.10.2023 • **Accepted/Published Online:** 23.03.2024 • **Final Version:** ..201

Abstract: This survey focuses on Text-to-SQL, automated translation of natural language queries into SQL queries. Initially, we describe the problem and its main challenges. Then, by following the PRISMA systematic review methodology, we survey the existing Text-to-SQL review papers in the literature. We apply the same method to extract proposed Text-to-SQL models and classify them with respect to used evaluation metrics and benchmarks. We highlight the accuracies achieved by various models on Text-to-SQL datasets and discuss execution-guided evaluation strategies. We present insights into model training times and implementations of different models. We also explore the availability of Text-to-SQL datasets in non-English languages. Additionally, we focus on large language model (LLM) based approaches for the Text-to-SQL task, where we examine LLM-based studies in the literature and subsequently evaluate the LLMs on the cross-domain Spider dataset. Finally, we conclude with a discussion of future directions for Text-to-SQL research, identifying potential areas of improvement and advancements in this field.

Key words: Text-to-SQL, large language model, natural language processing, deep learning

1. Introduction

Relational databases are frequently used in the IT industry to organize and maintain data and information from various fields and domains. To retrieve information from these databases, professionals require machine-readable instructions in the form of programs. Structured Query Language (SQL) is the most common language used to access and query data in relational databases. Therefore, experts or professionals with SQL knowledge are required to access database information and obtain meaningful results. These experts or professionals must also understand the existing schemas and tables of the database and create appropriate queries.

If successfully applied, envisioned Text-to-SQL systems can help end-users translate natural language input into queries without the need for SQL knowledge to query necessary information from the database. Some data applications, such as ThoughtSpot[1], have already claimed the capability to translate natural language given by users into SQL queries and display or visualize the results. According to Gartner Inc., chatbots will become the primary customer service channel for roughly a quarter of organizations by 2027. Thus, end-users will be able to access the desired information with the help of assistant applications such as the ones shipped with mobile operating systems.

Several deep learning-based studies in the literature have addressed different aspects of the Text-to-SQL problem. For example; Seq2SQL [1] and SQLNet [2] focused on the ordering issue that refers to the challenge of

*Correspondence: bugra.kanburoglu@isik.edu.tr
[1]https://www.thoughtspot.com

correctly identifying the order in which different components of a SQL query should appear when generating the query from natural language text. Also, SyntaxSQLNet [3] and RYANSQL [4] worked on complex and cross-domain Text-to-SQL task. TypeSQL [5] assigns types to words in SQL queries for better understanding of rare entities and numbers, while DialSQL [6] is a dialogue-based framework that generates accurate SQL queries by incorporating user interaction to identify and correct potential errors. Additionally, IRNet [7] addressed the mismatch problem, which refers to the challenge of accurately aligning the elements of a natural language sentence with their corresponding components in the SQL query. Moreover, GNN [8], RAT-SQL [9] and SADGA [10] considered the schema representation problem that refers to the challenge of effectively representing the structure of a database schema in a way that can be understood and used by a model to generate accurate SQL queries.

In this study, we conducted a methodical review of Text-to-SQL research published between 2018 and 2023 using a Preferred Reporting Items for Systemic Review and Meta-Analysis (PRISMA) [11] approach. Our contributions can be summarized as follows:

- Methodical Review: We conducted a PRISMA-based methodical review of Text-to-SQL studies, including literature surveys. Additionally, we incorporated a review of studies and datasets related to the conversion of non-English languages to SQL.
- Challenges and issues: We identified several significant issues and challenges related to Text-to-SQL studies.
- Benchmarks and evaluation metrics: We examined single domain and cross-domain benchmarks, related evaluation metrics, and the accuracies of the Text-to-SQL methods on these benchmarks.
- Large language model (LLM) based Text-to-SQL: We reviewed recently developed LLM-based Text-to-SQL studies. Additionally, we evaluated and compared three general purpose LLM models on complex cross-domain Spider dataset.
- Future directions: We discuss potential future research directions in the Text-to-SQL field.

The main goal of this paper is to provide a methodical survey of the Text-to-SQL problem. To achieve this goal, we have organized the paper in a way that reflects this objective. In Section 2, we define the problem of translating natural language to SQL and present a formal description of the problem. We also discuss the challenges and issues that Text-to-SQL methods face. Section 3 reviews earlier surveys and literature reviews related to Text-to-SQL, and also provides information on the number of documents searched from two scientific databases. Section 4 focuses on the most commonly used benchmarks and evaluation metrics in Text-to-SQL studies. Section 5 provides a summary of the accuracies of Text-to-SQL methods on different benchmarks. Section 6 reports on the implementation information of Text-to-SQL methods, reproducibility, the popularity on Github, and the Text-to-SQL initiatives announced by the industry, along with insights into training times. Additionally, Section 7 presents studies on different languages. Section 8 explains the LLM-based models for Text-to-SQL. In Section 9, we analyze the findings of our study in detail. Section 10 then concludes the paper by summarizing the key points.

2. Problem definition

A query refers to a question or request for data, which can be expressed in natural language. For example, "Retrieve the ids of all authors." would translate to the SQL query which is "`SELECT id FROM author`." SQL

utilizes keywords that match the given natural language. In this example, `SELECT` is used to retrieve data from the database, `FROM` specifies the table being used, and `WHERE` is used to establish a condition. In the following example, we give a typical SQL query.

SELECT attributes FROM tables WHERE conditions

There are various structures that are used when constructing an SQL query, such as `JOIN`, `NESTED`, `GROUP BY`, and `ORDER BY`, as in [12]. The process of converting natural language to SQL is referred to in literature with different terms, such as natural language to SQL (NL2SQL), Text-to-SQL, or natural language interface to databases (NLIDB). In this study, we use the term Text-to-SQL. Text-to-SQL is the task of automatically translating natural language (NL) into SQL, as shown in Figure 1.

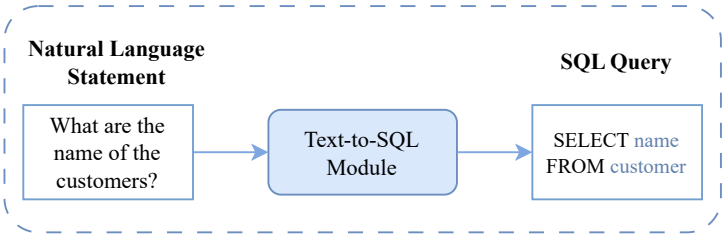

Figure 1. A simple Text-to-SQL diagram.

The challenge of Text-to-SQL can also be regarded as a semantic parsing problem, which is concerned with the conversion of a natural language query into a machine-understandable semantic representation, also known as a logical form. Natural language refers to the way people communicate with each other through speech and text [13], e.g., menus, emails, web pages. SQL has basic syntax and language elements such as; SQL language characters, tokens, separators, keywords, identifiers [12].

2.1. Challenges and issues

There are several significant challenges and issues associated with Text-to-SQL that must be addressed. We describe some of the important ones below.

One key challenge is the **ordering issue** also known as the "order-matters" problem [2], where the order of predicates within the WHERE clause in SQL queries does not affect the resultant execution outcomes thus the same query can be expressed in different orders. The SQL queries "SELECT * FROM Employees WHERE Salary >50000 AND Department = 'IT"' and "SELECT * FROM Employees WHERE Department = 'IT' AND Salary >50000" yield the same result. Solutions like Seq2SQL [1], SQLNet [2], and IncSQL [14] employ different approaches, such as reinforcement learning, sequence-to-set and sequence-to-action models, to tackle this issue.

Another challenge is the **complex and cross-domain Text-to-SQL task**, which involves handling complex SQL queries across diverse domains and databases. The Spider [15] Text-to-SQL dataset is a prime example of such complexity, requiring models to generalize across different subjects such as movie databases, geography, and sports. SyntaxSQLNet [3] and RYANSQL [4] address this challenge through syntax tree networks and sketch-based slot-filling methods.

The lack of information challenge arises due to the lack of domain-specific knowledge and rare entities in natural language queries, leading to inaccurate translations. In a query asking for the highest-scoring player in a sports database without specifying the sport, the model might struggle without additional context. TypeSQL [5] assigns types to words as entities, while DialSQL [6] incorporates user interaction to enhance query accuracy by identifying and revising potential errors.

In some cases, there is a **mismatch problem** [7] where SQL column names do not align with their natural language descriptions, confusing, especially in GROUP BY queries. In a GROUP BY query, the natural language might refer to "total sales" while the SQL column name is "revenue". STAMP [16] and IRNet [7] address this by considering the structure of table and the syntax of SQL and schema linking module.

The **lexical problem** [7] arises in cross-domain benchmarks like Spider and WikiSQL, where a significant portion of words in the development set is absent in the training set. This poses a challenge as models lack accurate representations for these out-of-domain (OOD) words. For example; a model trained on a dataset of scientific articles may struggle with terms specific to a dataset about movie reviews. IRNet tackles this problem with schema linking solutions.

Lastly, the **schema representation problem** pertains to generalizing models to unseen database schemas in cross-domain Text-to-SQL tasks. This challenge involves encoding schema information, including table and column details, and building a link between natural language and database schema. Adapting a model trained on a dataset about e-commerce to generate queries for a medical database, which requires understanding different table structures and relationships. GNN [8], RAT-SQL [9] and SADGA [10] models are presented to address this challenge.

3. Related works

This section presents a review of previous survey and review studies that have provided valuable descriptions and number of studies related to NL2SQL and Text-to-SQL keywords reported by Scopus and Web of Science in the last six years.

3.1. Literature surveys

In this section, we followed the PRISMA approach [11] to identify survey and review studies that discussed Text-to-SQL or NL2SQL. We conducted our search using Web of Science, Scopus, and Google Scholar from 2018 to 2023, using the search terms "(nl2sql OR text-to-sql) AND (survey OR review)". Figure 2 illustrates the PRISMA flow chart of our review process. As eligibility criteria, we removed duplicates, eliminated studies not available in English, and studies that were not survey or review. As a result, 9 studies remained.

Iacob et al. [17] delved into architecture decisions for Text-to-SQL models, focusing on encoder and decoder choices, while also drawing insights from classical methods and nondeep learning solutions. They evaluated model performance on the Spider dataset, highlighting the resurgence of natural language interfaces to databases. Kalajdjieski et al. [18] conducted a comprehensive review of Text-to-SQL methods, models, and datasets, encompassing a wide range of architectures, including CNNs, RNNs, pointer networks, and reinforcement learning. They introduced various Text-to-SQL datasets and evaluation metrics that consider execution and logical form accuracy. Kim et al. [19] provided a taxonomy-based review of NL2SQL methods, comparing eleven methods against multiple benchmarks. They introduced a validation tool to measure the quality of NL2SQL models accurately by considering the semantic equivalence of SQL queries. Majhadi and Machkour [20] presented an overview of NLIDB models, discussing their architectures and experimental results.

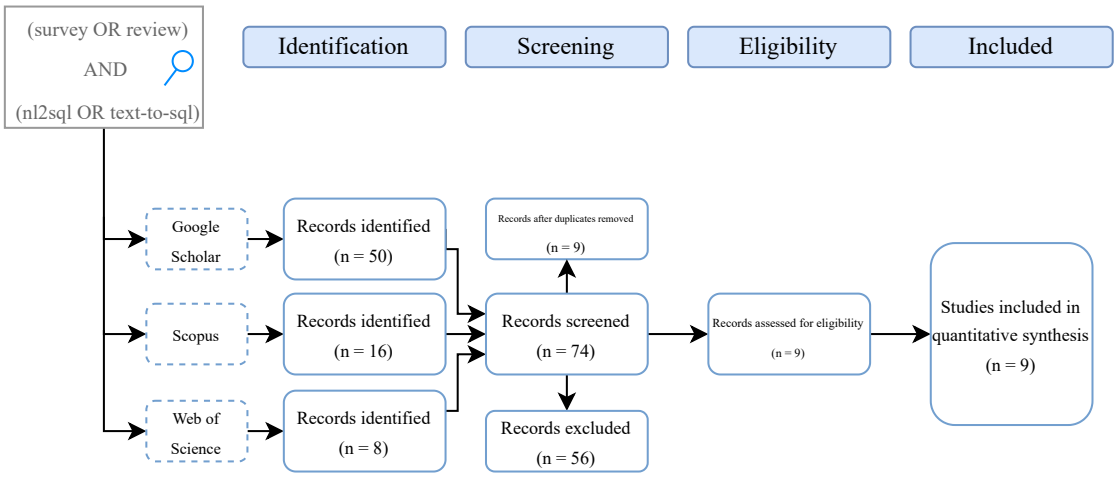

Figure 2. The PRISMA flow chart for research strategy.

They emphasized the need for improved accuracy in NLIDB systems. Ahkouk et al. [21] summarized Text-to-SQL techniques for nested queries, highlighting the potential for generating high-quality queries through human function imitation. Abbas et al. [22] conducted a detailed review of deep learning-based NLIDB systems, comparing WikiSQL and Spider datasets and identifying challenges in dataset quality and condition accuracy. Baig et al. [23] categorized NL2SQL frameworks based on implementation techniques and evaluated their performance on the SPIDER dataset, with RAT-SQL using BERT achieving the highest accuracy. Deng et al. [24] reviewed Text-to-SQL datasets and deep-learning-based methods, categorizing them into different groups and discussing common evaluation metrics. Qin et al. [25] provided a comprehensive review of deep learning approaches for Text-to-SQL, categorizing them into context-dependent and context-independent methods. They explored encoding and decoding techniques, including pretrained language models like BERT, and discussed potential future directions for Text-to-SQL. Katsogiannis-Meimarakis and Koutrika [26] presented a detailed taxonomy that decomposes the Text-to-SQL problem into different subproblems and compares various approaches of neural Text-to-SQL methods by explaining the available benchmarks and evaluation methods.

In comparison to existing surveys on Text-to-SQL, our paper contributes a methodical examination of the field by adopting the PRISMA systematic review methodology. We systematically survey the existing review papers, categorize Text-to-SQL models based on proposed methodologies, and classify them concerning the evaluation metrics and benchmarks used. We provide a detailed analysis of the accuracies achieved by various models on Text-to-SQL datasets and delve into execution-guided evaluation strategies. Furthermore, our paper places a specific emphasis on large language models (LLM) based approaches for the Text-to-SQL task, evaluating these models on the cross-domain Spider dataset. We extend our investigation to explore the availability of Text-to-SQL datasets in non-English languages. Moreover, we enhance the industrial perspective in our paper by incorporating insights from studies conducted by companies. Besides, we provide valuable insights into the efficiency of Text-to-SQL models by reporting their training times on well-established benchmarks such as WikiSQL and Spider. While existing surveys have provided valuable insights into specific aspects of Text-to-SQL, our paper stands out in its systematic approach, presenting a consolidated view of the field and identifying potential future directions for research in this domain.

3.2. Number of studies on Text-to-SQL

Web of Science[2] and Scopus[3] are two major databases used for research references that complement each other. Therefore, we conducted keyword searches in both databases for studies published between 2018 and 2023. The period between 2018 and 2023 was selected for analysis as there was a surge of interest in Text-to-SQL studies in 2018, which coincided with an increase in deep learning research during those years. Figure 3 shows the number of research studies found in Scopus and Web of Science databases using NL2SQL or Text-to-SQL keywords. The number of research papers has significantly increased since 2018, from 9 to 268 according to the Scopus results between 2018 and 2023.

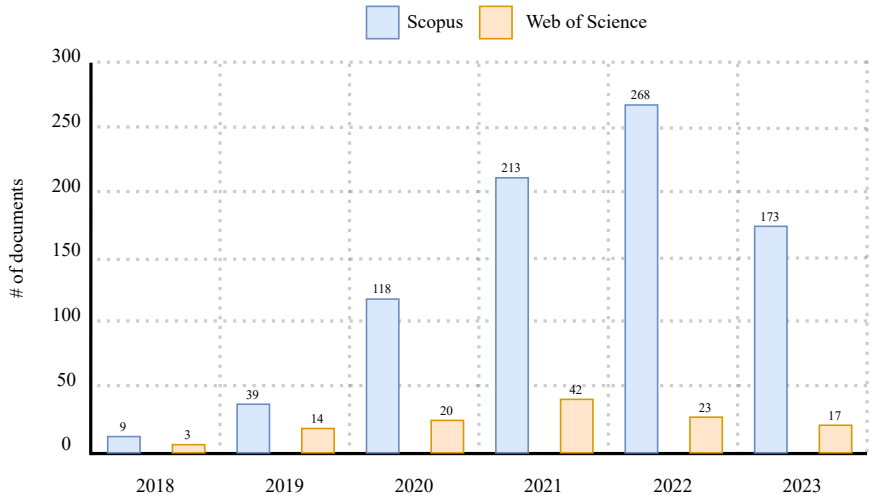

Figure 3. Number of documents searched from Scopus and Web of Science databases including NL2SQL or Text-to-SQL from 2018-01-01 to 2023-10-01.

4. Evaluation

4.1. Benchmarks

Early datasets were created within a single domain, whereas newer datasets are developed across multiple domains. We compared the datasets based on their structures, which include JOIN, NESTED, GROUP BY, and ORDER BY. Our findings can be seen in Table 1. Below we examine these benchmarks in detail.

4.1.1. Single domain benchmarks

In this section, we present popular single domain benchmarks [27–29, 32]. The Airline Travel Information System (ATIS) [27, 28] dataset is a benchmark for evaluating Text-to-SQL models within the context of airline travel, consisting of 5280 natural language and 947 SQL query pairs, with a database schema of 25 tables representing the ATIS database. This dataset is unique in that it focuses on a single domain, which presents specific challenges for Text-to-SQL models in handling the domain-specific language and query patterns. It includes simple queries as well as JOIN and NESTED queries, but does not include queries with GROUPING or ORDERING, and has been widely used for Text-to-SQL model evaluation. Similarly, the Microsoft Academic Search (MAS)

[2] https://www.webofscience.com
[3] https://www.scopus.com/

Table 1. Comparison of Text-to-SQL datasets.

Dataset	Year	Domain	Join	Nested	Group	Order	# NL	# SQL
ATIS [27, 28]	1990	Single	+	+	-	-	5280	947
MAS [29]	2014	Single	+	+	+	-	196	196
WikiSQL [1]	2017	Cross	-	-	-	-	80,654	77,840
Spider [15]	2018	Cross	+	+	+	+	10,181	5693
SparC [30]	2019	Cross	+	+	+	+	4298	12,726
CoSQL [31]	2019	Cross	+	+	+	+	3007	15,598
FIBEN [32]	2020	Single	+	+	+	+	300	237

[29] dataset provides a collection of 196 queries extracted from an academic databases, categorized as easy, medium, and complex, and accompanied by a schema of 17 tables. These queries include JOIN, NESTED, and GROUPING, but not ORDERING, with a strict format of natural language queries starting with the word "return". Lastly, the FIBEN [32] dataset, designed for finance-related natural language querying, describes financial transactions from public companies, featuring 300 natural language queries paired with 237 distinct SQL targets, including joins and nested queries.

4.1.2. Cross domain benchmarks

There are various cross domain benchmarks [1, 15, 30, 31]. WikiSQL [1] is the largest human handwritten semantic parsing dataset, containing 80,654 examples of natural language questions and SQL queries extracted from Wikipedia tables, although these queries are relatively simple, lacking JOIN, NESTED, GROUPING, or ORDERING. Spider [15], a complex and cross-domain semantic parsing dataset, consists of 10,181 questions and 5693 unique complex SQL queries across 200 databases, including NESTED queries and ORDERING/GROUPING components. SparC [30] offers 4298 question sequences with over 12,000 questions and SQL queries querying 200 complex databases, covering various SQL structures. Lastly, CoSQL [31], a conversational Text-to-SQL corpus, contains 3007 dialogues with more than 30,000 turns, encompassing 10,000 expert-labeled SQL queries, spanning 200 databases across 138 domains, notable for its extensive dialogue context and large natural language vocabulary compared to other datasets.

4.2. Metrics

Next, we examined evaluation metrics in combination with benchmark datasets and Text-to-SQL models. Our findings are summarized in Table 2. In this table, evaluation metrics are shown as abbreviations. EX refers to execution accuracy, LF refers to logical-form accuracy and EM refers to exact matching.

Execution accuracy [1] compares the synthesized (generated) query and the ground truth query in terms of their results, typically by executing both queries on the same database and comparing their outputs. Logical-form accuracy [1], on the other hand, focuses on exact string matching between synthesized and ground truth queries but penalizes correct results without exact string matches. Finally, exact matching [15] calculates the average exact match between synthesized and ground truth queries for different SQL components, such as SELECT, WHERE, GROUP BY, ORDER BY, and KEYWORDS, checking whether the component groups match precisely.

Table 2. Text-to-SQL models with their datasets and used evaluation metrics.

Dataset	Evaluation metric	Models
WikiSQL	EX	Seq2SQL [1], SQLNet [2], PTMAML [33], COARSE2FINE [34], STAMP [16], TypeSQL [5], IncSQL [14], SQLOVA [35], X-SQL [36], BRIDGE [37], HydraNet [38], IE-SQL [39], SDSQL [40], SeaD [41], RAT-SQL [9]
	LFA	Seq2SQL [1], SQLNet [2], PTMAML [33], STAMP [16], TypeSQL [5], IncSQL [14], SQLOVA [35], X-SQL [36], HydraNet [38], IE-SQL [39], SDSQL [40], SeaD [41], RAT-SQL [9]
Spider	EM	SyntaxSQLNet [3], EditSQL [42], GNN [8], IRNet [7], ValueNet [43], RAT-SQL [9]

5. Accuracies on Text-to-SQL datasets

In this section, we compare the accuracies of the reviewed methods with different evaluation metrics. Table 3 and Table 4 present accuracies for the WikiSQL and Spider which are the two most popular datasets in terms of evaluation metrics. We have highlighted the best results in bold face for each dataset. For WikiSQL, the term "LF" refers to logical form accuracy and "EX" refers to execution accuracy. For Spider, "EM" denotes exact match accuracy. Accuracies are shown only for the dev set of Spider since both dev and test sets were published for WikiSQL while only the dev set was published for Spider.

Table 3. Accuracies on the Spider dev set.

Model	Dev Set
	EM Accuracy
SyntaxSQLNet [3]	24.8
GNN [8]	40.7
IRNet [7]	61.9
RAT-SQL [9]	62.7
PHOTON [44]	63.2
RYANSQL [4]	70.6
SADGA [10]	73.4
ShadowGNN [45]	72.3
LGESQL [46]	75.1
S^2SQL [47]	76.4
UniSAr [48]	70.0
RASAT [49]	75.3
T5-SR-3b [50]	**79.9**

The best-performing model on the WikiSQL dataset based on the execution accuracy metric is SeaD, with an execution accuracy of 92.9 on the development set and 93.0 on the test set. On the other hand, the best model based on the logical form accuracy metric is IE-SQL, with a logical form accuracy of 87.9 on the development set and 87.8 on the test set. Based on the evaluation using exact match accuracy on the Spider dataset, the best model on the development set is T5-SR-3b, which combined a seq2seq-oriented decoding strategy with the pretrained seq2seq model T5, with a score of 79.9.

5.1. Execution-guided decoding

While neural network models for Text-to-SQL have achieved remarkable progress, they still face challenges in generating accurate and executable queries at run-time. This section explores execution-guided decoding (EG) as a promising technique to address these limitations and improve Text-to-SQL performance.

In a SQL query, each component (SELECT, AGGREGATION, WHERE) is predicted independently, leading to the potential generation of invalid combinations. For example, a string-type column is not allowed to be combined with an aggregation operator such as "min", or condition operator like "less-than". To address these issues, Wang et al. [51] proposed Execution-guided decoding (EG), which executes the predicted SQL query at run-time, correcting any errors or empty outputs returned by the database engine.

1. If the query execution results in an error, it can be identified and excluded as an invalid prediction.
2. Queries that return no results can be identified and filtered out as irrelevant.
3. EG identifies and excludes combinations of operators and data types that are semantically invalid (e.g., string-type column with less-than operator).
4. By filtering out invalid and erroneous predictions, EG leads to more accurate and executable SQL queries.

Therefore, execution-guided decoding represents a promising approach for improving the accuracy and robustness of Text-to-SQL models. By leveraging the power of run-time query execution and incorporating the semantics of SQL, EG can significantly enhance the performance of Text-to-SQL systems and lead to more reliable and efficient data retrieval. Table 4 presents the results with and without EG applied. In this table, we provide the results before and after the execution-guided decoding in the models. We observe an increase in both LF and EX metrics on both the development and test sets.

Table 4. Dev and test accuracies on WikiSQL dataset with and without EG applied.

Model	Without EG				With EG			
	Dev		Test		Dev		Test	
	LF	EX	LF	EX	LF	EX	LF	EX
Coarse2Fine [34]	72.9	79.2	71.7	78.4	76.0	84.0	75.4	83.8
IncSQL [14]	76.1	82.5	75.5	81.6	51.3	87.2	51.1	87.1
SQLova [35]	81.6	87.2	80.7	86.2	84.2	90.2	83.6	89.6
X-SQL [36]	83.8	89.5	83.3	88.7	86.2	92.3	86.0	91.8
HydraNet [38]	83.6	89.1	83.8	89.2	86.6	92.4	86.5	92.2
IE-SQL [39]	84.6	88.7	84.6	88.8	**87.9**	92.6	**87.8**	92.5
BRIDGE [37]	**86.2**	91.7	**85.7**	91.1	86.8	92.6	86.3	91.9
SDSQL [40]	86.0	**91.8**	85.6	**91.4**	86.7	92.5	86.6	92.4
SeaD [41]	84.0	90.2	84.7	90.1	87.3	**92.8**	87.1	**92.7**

6. Implementations

In this section, we examine the implementation details of the Text-to-SQL methods. First of all, there are rule-based and deep learning-based methods in Text-to-SQL methods. Open-source code resources were searched for all these methods and information was gathered about the programming languages in which the methods were implemented and the frameworks they used. We have identified open-source methods, their frameworks, and

activity. We listed top 15 Text-to-SQL methods as shown in Table 5 where SQLOVA, Seq2SQL and SQLNet models have garnered the highest number of Github stars, indicating their popularity among users. The majority used the PyTorch framework with Python programming language. Table summarizes the activity and public attention to the open source code by using Github stars. GitHub stars is an easy metric to measure how popular an open-source project is. In addition, we assessed the reproducibility of these methods by checking the README files and issues of the repositories because reproducibility is a crucial aspect of open-source projects, ensuring that others can replicate and verify the results. We examined the GitHub repositories of the models provided in the table. Within the README file of each repository, we conducted searches for the keywords "reproduce, reproducible, reproducibility" to determine whether the results of the mentioned model are reproducible. Additionally, we assessed discussions related to reproducibility by examining both open and closed issues in the repository. For repositories where reproducibility was not explicitly mentioned in the README file but was confirmed through the examination of open and closed issues, we made our determination accordingly.

Table 5. The number of GitHub stars for Text-to-SQL models.

Model	Year	Framework	Github Stars	Reproducibility
SQLOVA [35]	2019	PyTorch	621	NO
Seq2SQL [1]	2017	PyTorch	409	NO
SQLNet [2]	2017	PyTorch	409	NO
RAT-SQL [9]	2020	PyTorch	369	YES
UniSAr [48]	2022	PyTorch	326	YES
IRNet [7]	2019	PyTorch	244	NO
BRIDGE [37]	2020	PyTorch	204	NO
EditSQL [42]	2019	PyTorch	189	YES
Coarse2Fine [34]	2018	PyTorch	167	YES
LGESQL [46]	2021	PyTorch	135	YES
GNN [8]	2019	PyTorch	134	YES
PT-MAML [33]	2018	TensorFlow	128	NO
SyntaxSQLNet [3]	2018	PyTorch	125	NO
TypeSQL [5]	2018	PyTorch	107	NO
HydraNet [38]	2020	PyTorch	61	YES

In recent years, academic research has become increasingly important to companies, with many tech industry leaders placing a high value on turning academic knowledge into practical products. Analyzing the activity of the tech industry in this area can provide insights into its reception and efforts toward academic research. IBM Research has contributed to this domain with publications on models like ATHENA, TEMPLAR, and ATHENA++. Meanwhile, SalesForce Research has published Seq2SQL, EditSQL, BRIDGE, and PHOTON. Microsoft Research has introduced many different Text-to-SQL models, including PTMAML, STAMP, IncSQL, IRNet, X-SQL, HydraNet, RAT-SQL, and UniSAr. On the other hand, Alibaba Group has been active in this space with the publication of models such as SDSQL, S^2SQL, and HIE-SQL.

Moreover, some Text-to-SQL models reported their training times on popular benchmarks such as WikiSQL and Spider. For example, the BRIDGE model was trained on the WikiSQL dataset for about 6 h using an NVIDIA A100 GPU, and on the Spider dataset for about 51.5 h using the same GPU. The SeaD model was trained on the WikiSQL benchmark using NVIDIA V100 GPUs for approximately 3 h. The UNISAR model, on the other hand, was trained on Spider, CoSQL, and SParC datasets jointly with four V100-16G GPUs for around 10 h.

7. Text-to-SQL on different languages

Ethnologue[4] data indicates that there are over 7000 known living languages in the world, highlighting the potential need for Text-to-SQL systems that can operate in languages other than English. However, the datasets commonly used in the Text-to-SQL problem were created for English language. To address this, non-English datasets were also searched for in this study. Searching the Scopus database using the keywords "Non-English, Dataset" and "Text-to-SQL, NL2SQL" for the past five years yielded 77 results. Examining these results, we found datasets in Chinese and Vietnamese languages. For instance, Min et al. [52] presented the CSpider dataset by translating all English questions in the Spider dataset into Chinese. One annotator translated each question, and then a second translator cross-checked and corrected it, and a third annotator verified it. Sun et al. [53] introduced TableQA, a large-scale cross-domain Natural Language to SQL dataset in the Chinese language. This dataset consists of 64,891 questions and 20,311 unique SQL queries over 6000 tables collected from public financial reports. Nguyen et al. [54] presented ViText2SQL, the first public large-scale dataset for Vietnamese Text-to-SQL semantic parsing tasks, consisting of about 10,000 question and SQL query pairs. They translated all English questions and the database tables and columns in SQL queries from the original Spider dataset into Vietnamese. Wang et al. [55] presented DuSQL, a large-scale and pragmatic Chinese Text-to-SQL dataset. It contains 200 databases, 813 tables, and 23,797 question and SQL pairs, built using human-computer collaboration. Guo et al. [56] presented CHASE, a large-scale and pragmatic Chinese dataset for cross-database context-dependent Text-to-SQL. It consists of 17,940 questions with their SQL queries over 280 databases. Huang et al. [57] presented SeSQL, a large-scale session-level Chinese Text-to-SQL dataset. It contains 5028 unique questions over 201 databases, with 27,012 annotated questions with their SQL queries. Compared to the CHASE dataset, SeSQL contains more question/query rounds per session.

8. LLM-based Text-to-SQL

Recently, there have been significant advancements in the field of Text-to-SQL, particularly in the context of large language models (LLMs). Liu et al. [58] conducted an extensive analysis of ChatGPT's Text-to-SQL capabilities, achieving an execution accuracy of 70.1% on the Spider dataset. In addition, Jiang et al. [59] introduced StructGPT, a general framework aimed at improving the zero-shot reasoning abilities of LLMs over structured data. Through experiments on different datasets, they showed that their approach can significantly improve the zero-shot performance of LLMs. Specifically, on the Spider dataset, StructGPT achieved 74.8% execution accuracy. Pourreza and Rafiei [60] proposed a novel approach called DIN-SQL based on few-shot prompting to address the Text-to-SQL task. Their method decomposes the problem into multiple steps. This approach consistently demonstrated remarkable performance improvements across different LLMs. Specifically, on the Spider dataset, DIN-SQL model achieved an execution accuracy of 75.6% with CodeX Davinci and an impressive 82.8% with GPT-4. Sun et al. [61] contributed to this evolving landscape with SQL-PaLM, an LLM-based Text-to-SQL model leveraging PaLM-2 [62]. Their research demonstrated substantial advancements, particularly in Few-shot SQL-PaLM, which achieved an execution accuracy of 82.7% on the Spider dataset. Fine-tuned SQL-PaLM further improved upon this performance, reaching an accuracy of 82.8%.

In the current study, we conducted an evaluation of general purpose large language models (LLMs) in the context of Text-to-SQL, measuring their performance in terms of execution accuracy (EX). We used the Hugging Face's training framework for Llama-2 and directly gave prompts on ChatGPT. ChatGPT demonstrated a

[4] https://www.ethnologue.com

remarkable execution accuracy of 73.83. In contrast, Llama-2-7b [63], another LLM, achieved a comparatively lower execution accuracy of 34.20. Additionally, fine-tuning Llama-2-7b on the Spider training set led to an improved execution accuracy of 46.60. These results give valuable insights into the performance of these models and the impact of fine-tuning on their effectiveness in the challenging task of generating SQL queries from natural language text. In Table 6, we present the results of our evaluation for ChatGPT, Llama-2-7b, and fine-tuned Llama-2-7b.

Table 6. Execution accuracy (EX) of different LLMs on the Spider development set.

Model	Execution accuracy (EX)
ChatGPT	73.83
Llama-2-7b	34.20
Fine-tuned Llama-2-7b	46.60

ChatGPT demonstrates an impressive ability in executing the Text-to-SQL task. Specifically, the execution accuracy achieved by ChatGPT was measured at 73.83%, outperforming several Text-to-SQL models such as; UniSAr [48], SADGA [10] and RYANSQL [4]. This result strongly confirms ChatGPT's competitive advantage in accurately translating natural language queries into SQL statements.

We explored some of the issues encountered in the Text-to-SQL domain, specifically on ChatGPT. First, we investigated the ordering issue, where ChatGPT generated different variations of a given query without altering the execution result. We examined how these variations could potentially address the ordering problem. Second, we addressed the mismatch problem, where ChatGPT generated queries that included entities not mentioned in the natural language input. While ChatGPT successfully produced various variations that did not affect the execution result in the case of the ordering issue, the results were less promising in tackling the mismatch problem. For future work, we can explore how large language models (LLMs) can effectively resolve the issues and challenges encountered in the Text-to-SQL domain.

9. Discussion

In this study, we investigated the challenges involved in Text-to-SQL. A PRISMA-based systematic review approach was conducted to search for relevant literature surveys, resulting in the identification of nine literature surveys. The distribution of papers related to NL2SQL and Text-to-SQL was analyzed over the years, revealing a noticeable growth trend since 2018. We discussed the benchmarks used both single domain and cross domain. We found that the most commonly used Text-to-SQL datasets are WikiSQL and Spider. We examined the evaluation metrics used by these datasets, and found that the most frequently used metrics are execution accuracy (EX) and logical form accuracy (LF) in the WikiSQL dataset and exact match accuracy (EM) in the Spider dataset. We also examined the accuracy performance of the methods in WikiSQL and Spider datasets in Table 4 and Table 3, respectively. The SeaD model that is a seq-to-seq model with schema-aware denoising achieved the highest execution accuracy (92.9 on development set, 93.0 on test set) on the WikiSQL dataset, while the IE-SQL model which is an extraction-linking approach to Text-to-SQL task had the highest logical form accuracy (87.9 on development set, 87.8 on test set). On the Spider dataset, the T5-SR-3b model had the highest exact match accuracy, scoring 79.9 on the development set. We found that the accuracy of Text-to-SQL models tends to improve with the use of execution-guided decoding (EG) approach. For instance, on the development set of WikiSQL dataset, the logical form accuracy (LF) of IE-SQL model increased from 84.6

to 87.9, and the execution accuracy (EX) of SeaD model increased from 90.2 to 92.8. This suggests that EG approach is effective in improving the performance of these models. We reviewed the implementation, training times, and the number of stars available on Github for the methods, which signifies popularity of the problem and also presented the studies of some companies in the industrial sense which shows the industrial attention and requirements, feasibility of Text-to-SQL. Additionally, we examined Text-to-SQL datasets that have been developed in languages other than English, may be limited, mostly from East Asian countries. However, there is a need for more datasets in other languages. However, large language models (LLMs) with their massive cross-languages training sets and acquired few-shot learning capabilities can potentially help address this limitation. Our experiments with ChatGPT, which is based on GPT-3.5, adeptly generated diverse query versions, indicating that the ordering issue posed no significant challenge. However, it exhibited less success in addressing the mismatch problem. This suggests that future work could delve into exploring how LLMs might effectively resolve these challenges.

10. Conclusion

We presented a methodical survey about Text-to-SQL. We have defined the Text-to-SQL problem and discussed the challenges and issues associated with it. We have also presented the literature surveys on Text-to-SQL and reported benchmarks in different languages. Moreover, we have compared these benchmarks, presented the Text-to-SQL methods used on these benchmarks, and explained the evaluation metrics used with these methods. Furthermore, we have provided the accuracy results of Text-to-SQL methods on two popular benchmarks and compared these methods based on the frameworks used. We have also reported some of the methods studied in an industrial context. Moreover, we have reviewed the LLM-based Text-to-SQL studies and conducted an evaluation for different LLMs. Lastly, we have analyzed the number of published papers in this field by years from scientific databases. We hope that this paper will benefit all researchers, practitioners, and educators in the community.

Future research should prioritize the development of more comprehensive datasets to boost the performance and adaptability of Text-to-SQL models, with a pressing need for expansion particularly in non-English languages. The current emphasis on non-English languages is likely to grow, with research focusing on the development of Text-to-SQL models that can effectively understand and generate SQL queries in multiple languages. This involves not only translation but also addressing language-specific nuances and variations in database structures. Future research might delve deeper into improving the semantic parsing capabilities of Text-to-SQL models, including handling ambiguous queries, understanding context more effectively, and accurately capturing user intent. To enhance user understanding and increase trustworthiness, future models could be designed with a stronger focus on providing explanations for generated queries. Research may explore techniques for generating human-readable explanations alongside SQL queries. In addition, as a multimodal Text-to-SQL approach, combining text input with visual representations or other modalities (e.g., tables, charts) could improve model understanding and query generation. Addressing ethical concerns and biases is crucial in any AI system, including Text-to-SQL models. Future research could focus on developing models that are more transparent, fair, and considerate of privacy concerns, especially when dealing with sensitive data. Leveraging pretrained knowledge graphs, ontologies, and semantic resources can enrich the model's understanding of entities, relationships, and domain-specific concepts, contributing to more accurate and context-aware query generation. Additionally, incorporating human interaction in the design of Text-to-SQL models could potentially enhance their performance by leveraging human expertise and feedback during the training process. Exploring the use

of teacher-student network with knowledge distillation methods, such as the one proposed by Hinton et al. [64], could lead to more efficient and accurate Text-to-SQL models. Further research is needed to enhance the performance of Text-to-SQL models, making them suitable for deployment and end-user testing in real-world scenarios.

References

[1] Zhong V, Xiong C, Socher R. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. arXiv preprint arXiv:1709.00103. 2017. https://doi.org/10.48550/arXiv.1709.00103

[2] Xu X, Liu C, Song D. Sqlnet: Generating structured queries from natural language without reinforcement learning. arXiv preprint arXiv:1711.04436. 2017. https://doi.org/10.48550/arXiv.1711.04436

[3] Yu T, Yasunaga M, Yang K, Zhang R, Wang D et al. SyntaxSQLNet: Syntax Tree Networks for Complex and Cross-Domain Text-to-SQL Task. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP); Brussels, Belgium; 2018. pp. 1653-1663. https://doi.org/10.18653/v1/D18-1193

[4] Choi D, Shin MC, Kim E, Shin DR. Ryansql: Recursively applying sketch-based slot fillings for complex text-to-sql in cross-domain databases. Computational Linguistics 2021; 47 (2): 309-332. https://doi.org/10.48550/arXiv.2004.03125

[5] Yu T, Li Z, Zhang Z, Zhang R, Radev D. TypeSQL: Knowledge-Based Type-Aware Neural Text-to-SQL Generation. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers); New Orleans, Louisiana; 2018. pp. 588-594. https://doi.org/10.18653/v1/N18-2093

[6] Gür I, Yavuz S, Su Y, Yan X. Dialsql: Dialogue based structured query generation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Melbourne, Australia; 2018. pp. 1339-1349. https://doi.org/10.18653/v1/P18-1124

[7] Guo J, Zhan Z, Gao Y, Xiao Y, Lou JG et al. Towards Complex Text-to-SQL in Cross-Domain Database with Intermediate Representation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics; Florence, Italy; 2019. pp. 4524-4535. https://doi.org/10.18653/v1/P19-1444

[8] Bogin B, Berant J, Gardner M. Representing Schema Structure with Graph Neural Networks for Text-to-SQL Parsing. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics; Florence, Italy; 2019. pp. 4560-4565. https://doi.org/10.18653/v1/P19-1448

[9] Wang B, Shin R, Liu X, Polozov O, Richardson M. RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; Online; 2020. pp. 7567-7578. https://doi.org/10.18653/v1/2020.acl-main.677

[10] Cai R, Yuan J, Xu B, Hao Z. Sadga: Structure-aware dual graph aggregation network for text-to-sql. Advances in Neural Information Processing Systems 2021; 34: 7664-7676.

[11] Liberati A, Altman DG, Tetzlaff J, Mulrow C, Gøtzsche PC et al. The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: explanation and elaboration. Annals of internal medicine 2009; 151 (4): W-65.

[12] Date, CJ. A Guide to the SQL Standard. Addison-Wesley Longman Publishing Co., Inc., 1989.

[13] Brownlee J. Deep learning for natural language processing: develop deep learning models for your natural language problems. Machine Learning Mastery, 2017.

[14] Shi T, Tatwawadi K, Chakrabarti K, Mao Y, Polozov O et al. Incsql: Training incremental text-to-sql parsers with non-deterministic oracles. arXiv preprint arXiv:1809.05054. 2018. https://doi.org/10.48550/arXiv.1809.05054

[15] Yu T, Zhang R, Yang K, Yasunaga M, Wang D et al. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing; Brussels, Belgium; 2018. pp. 3911-3921. https://doi.org/10.18653/v1/D18-1425

[16] Sun Y, Tang D, Duan N, Ji J, Cao G et al. Semantic Parsing with Syntax-and Table-Aware SQL Generation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Melbourne, Australia; 2018. pp. 361-372. https://doi.org/10.18653/v1/P18-1034

[17] Iacob RCA, Brad F, Apostol ES, Truică CO, Hosu IA et al. Neural approaches for natural language interfaces to databases: A survey. In: proceedings of the 28th International Conference on Computational Linguistics; Barcelona, Spain (Online); 2020. pp. 381-395. https://doi.org/10.18653/v1/2020.coling-main.34

[18] Kalajdjieski J, Toshevska M, Stojanovska F. Recent Advances in SQL Query Generation: A Survey. arXiv preprint arXiv:2005.07667. 2020. https://doi.org/10.48550/arXiv.2005.07667

[19] Kim H, So BH, Han WS, Lee H. Natural language to SQL: Where are we today?. Proceedings of the VLDB Endowment 2020; 13 (10): 1737-1750. https://doi.org/10.14778/3401960.3401970

[20] Majhadi K, Machkour M. The history and recent advances of Natural Language Interfaces for Databases Querying. In: E3S Web of Conferences (Vol. 229, p. 01039); Agadir, Morocco; 2021. https://doi.org/10.1051/e3sconf/202122901039

[21] Ahkouk K, Mustapha M, Khadija M, Rachid M. A review of the Text to SQL Frameworks. In: Proceedings of the 4th International Conference on Networking, Information Systems & Security; New York, NY, United States; 2021. pp. 1-6. https://doi.org/10.1145/3454127.3457619

[22] Abbas S, Khan MU, Lee SUJ, Abbas A, Bashir AK. A review of nlidb with deep learning: findings, challenges and open issues. IEEE Access 2022; 10: 14927-14945. doi: 10.1109/ACCESS.2022.3147586

[23] Baig MS, Imran A, Yasin AU, Butt AH, Khan MI. Natural Language to SQL Queries: A Review. International Journal of Innovations in Science and Technology, 2022; 4 (1): 147-162.

[24] Deng N, Chen Y, Zhang Y. Recent Advances in Text-to-SQL: A Survey of What We Have and What We Expect. In: Proceedings of the 29th International Conference on Computational Linguistics; Gyeongju, Republic of Korea; 2022. pp. 2166-2187.

[25] Qin B, Hui B, Wang L, Yang M, Li J et al. A survey on text-to-sql parsing: Concepts, methods, and future directions. arXiv preprint arXiv:2208.13629. 2022. https://doi.org/10.48550/arXiv.2208.13629

[26] Katsogiannis-Meimarakis G, Koutrika G. A survey on deep learning approaches for text-to-SQL. The VLDB Journal 2023; 32 (4): 905-936. doi: 10.1007/s00778-022-00776-8

[27] Price P. Evaluation of spoken language systems: The ATIS domain. In: Speech and Natural Language: Proceedings of a Workshop; Hidden Valley, Pennsylvania; 1990.

[28] Iyer S, Konstas I, Cheung A, Krishnamurthy J, Zettlemoyer L. Learning a Neural Semantic Parser from User Feedback. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Vancouver, Canada; 2017. pp. 963-973. https://doi.org/10.18653/v1/P17-1089

[29] Li F, Jagadish HV. Constructing an interactive natural language interface for relational databases. Proceedings of the VLDB Endowment 2014; 8 (1): 73-84. https://doi.org/10.14778/2735461.2735468

[30] Yu T, Zhang R, Yasunaga M, Tan YC, Lin XV et al. SParC: Cross-Domain Semantic Parsing in Context. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics; Florence, Italy; 2019. pp. 4511-4523. https://doi.org/10.18653/v1/P19-1443

[31] Yu T, Zhang R, Er H, Li S, Xue E et al. CoSQL: A Conversational Text-to-SQL Challenge Towards Cross-Domain Natural Language Interfaces to Databases. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); Hong Kong, China; 2019. pp. 1962-1979. https://doi.org/10.18653/v1/D19-1204

[32] Sen J, Lei C, Quamar A, Özcan F, Efthymiou V et al. Athena++ natural language querying for complex nested sql queries. Proceedings of the VLDB Endowment 2020; 13 (12): 2747-2759. https://doi.org/10.14778/3407790.3407858

[33] Huang PS, Wang C, Singh R, Yih WT, He X. Natural Language to Structured Query Generation via Meta-Learning. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers); New Orleans, Louisiana; 2018. pp. 732-738. https://doi.org/10.18653/v1/N18-2115

[34] Dong L, Lapata M. Coarse-to-Fine Decoding for Neural Semantic Parsing. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Melbourne, Australia; 2018. pp. 731-742. https://doi.org/10.18653/v1/P18-1068

[35] Hwang W, Yim J, Park S, Seo M. A comprehensive exploration on wikisql with table-aware word contextualization. arXiv preprint arXiv:1902.01069. 2019. https://doi.org/10.48550/arXiv.1902.01069

[36] He P, Mao Y, Chakrabarti K, Chen W. X-SQL: reinforce context into schema representation. arXiv preprint arXiv:1908.08113. 2019. https://doi.org/10.48550/arXiv.1908.08113

[37] Lin XV, Socher R, Xiong C. Bridging Textual and Tabular Data for Cross-Domain Text-to-SQL Semantic Parsing. In: Findings of the Association for Computational Linguistics: EMNLP 2020; Online; 2020. pp. 4870-4888. https://doi.org/10.18653/v1/2020.findings-emnlp.438

[38] Lyu Q, Chakrabarti K, Hathi S, Kundu S, Zhang J et al. Hybrid ranking network for text-to-sql. arXiv preprint arXiv:2008.04759. 2020. https://doi.org/10.48550/arXiv.2008.04759

[39] Ma J, Yan Z, Pang S, Zhang Y, Shen J. Mention Extraction and Linking for SQL Query Generation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); Online; 2020. pp. 6936-6942. https://doi.org/10.18653/v1/2020.emnlp-main.563

[40] Hui B, Shi X, Geng R, Li B, Li Y et al. Improving text-to-sql with schema dependency learning. arXiv preprint arXiv:2103.04399. 2021. https://doi.org/10.48550/arXiv.2103.04399

[41] Xu K, Wang Y, Wang Y, Wang Z, Wen Z et al. SeaD: End-to-end Text-to-SQL Generation with Schema-aware Denoising. In: Findings of the Association for Computational Linguistics: NAACL 2022; Seattle, United States; 2022. pp. 1845-1853. https://doi.org/10.18653/v1/2022.findings-naacl.141

[42] Zhang R, Yu T, Er H, Shim S, Xue E et al. Editing-Based SQL Query Generation for Cross-Domain Context-Dependent Questions. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); Hong Kong, China; 2019. pp. 5338-5349. https://doi.org/10.18653/v1/D19-1537

[43] Brunner U, Stockinger K. Valuenet: A natural language-to-sql system that learns from database information. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE); Chania, Greece; 2021. pp. 2177-2182. https://doi.org/10.1109/ICDE51399.2021.00220

[44] Zeng J, Lin XV, Hoi SC, Socher R, Xiong C et al. Photon: A Robust Cross-Domain Text-to-SQL System. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations; Online; 2020. pp. 204-214. https://doi.org/10.18653/v1/2020.acl-demos.24

[45] Chen Z, Chen L, Zhao Y, Cao R, Xu Z et al. ShadowGNN: Graph Projection Neural Network for Text-to-SQL Parser. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; Online; 2021. pp. 5567-5577. https://doi.org/10.18653/v1/2021.naacl-main.441

[46] Cao R, Chen L, Chen Z, Zhao Y, Zhu S et al. 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 (Volume 1: Long Papers); Online; 2021. pp. 2541-2555. https://doi.org/10.18653/v1/2021.acl-long.198

[47] Hui B, Geng R, Wang L, Qin B, Li Y et al. S2SQL: Injecting Syntax to Question-Schema Interaction Graph Encoder for Text-to-SQL Parsers. In: Findings of the Association for Computational Linguistics: ACL 2022; Dublin, Ireland; 2022. pp. 1254-1262. https://doi.org/10.18653/v1/2022.findings-acl.99

[48] Dou L, Gao Y, Pan M, Wang D, Che W et al. UniSAr: a unified structure-aware autoregressive language model for text-to-SQL semantic parsing. International Journal of Machine Learning and Cybernetics 2023; 14 (12): 4361-4376. https://doi.org/10.1007/s13042-023-01898-3

[49] Qi J, Tang J, He Z, Wan X, Cheng Y et al. RASAT: Integrating Relational Structures into Pretrained Seq2Seq Model for Text-to-SQL. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing; Abu Dhabi, United Arab Emirates; 2022. pp. 3215-3229. https://doi.org/10.18653/v1/2022.emnlp-main.211

[50] Li Y, Su Z, Li Y, Zhang H, Wang S et al. T5-SR: A Unified Seq-to-Seq Decoding Strategy for Semantic Parsing. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); Rhodes Island, Greece; 2023. pp. 1-5. https://doi.org/10.1109/ICASSP49357.2023.10096172

[51] Wang C, Tatwawadi K, Brockschmidt M, Huang PS, Mao Y et al. Robust text-to-sql generation with execution-guided decoding. arXiv preprint arXiv:1807.03100. 2018. https://doi.org/10.48550/arXiv.1807.03100

[52] Min Q, Shi Y, Zhang Y. A Pilot Study for Chinese SQL Semantic Parsing. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); Hong Kong, China; 2019. pp. 3652-3658. https://doi.org/10.18653/v1/D19-1377

[53] Sun N, Yang X, Liu Y. Tableqa: a large-scale Chinese text-to-sql dataset for table-aware sql generation. arXiv preprint arXiv:2006.06434. 2020. https://doi.org/10.48550/arXiv.2006.06434

[54] Nguyen AT, Dao MH, Nguyen DQ. A Pilot Study of Text-to-SQL Semantic Parsing for Vietnamese. In: Findings of the Association for Computational Linguistics: EMNLP 2020; Online; 2020. pp. 4079-4085. https://doi.org/10.18653/v1/2020.findings-emnlp.364

[55] Wang L, Zhang A, Wu K, Sun K, Li Z et al. DuSQL: A large-scale and pragmatic Chinese text-to-SQL dataset. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); Online; 2020. pp. 6923-6935. https://doi.org/10.18653/v1/2020.emnlp-main.562

[56] Guo J, Si Z, Wang Y, Liu Q, Fan M et al. Chase: A Large-Scale and Pragmatic Chinese Dataset for Cross-Database Context-Dependent Text-to-SQL. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers); Online; 2021. pp. 2316-2331. https://doi.org/10.18653/v1/2021.acl-long.180

[57] Huang S, Wang L, Li Z, Liu Z, Dou C et al. SeSQL: A High-Quality Large-Scale Session-Level Chinese Text-to-SQL Dataset. In: CCF International Conference on Natural Language Processing and Chinese Computing; 2023. pp. 537-550. $https://doi.org/10.1007/978-3-031-44693-1\_42$

[58] Liu A, Hu X, Wen L, Yu PS. A comprehensive evaluation of ChatGPT's zero-shot Text-to-SQL capability. arXiv preprint arXiv:2303.13547. 2023. https://doi.org/10.48550/arXiv.2303.13547

[59] Jiang J, Zhou K, Dong Z, Ye K, Zhao WX et al. Structgpt: A general framework for large language model to reason over structured data. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP); Singapore; 2023. pp. 9237-9251. https://doi.org/10.18653/v1/2023.emnlp-main.574

[60] Pourreza M, Rafiei D. Din-sql: Decomposed in-context learning of text-to-sql with self-correction. arXiv preprint arXiv:2304.11015. 2023. https://doi.org/10.48550/arXiv.2304.11015

[61] Sun R, Arik SO, Nakhost H, Dai H, Sinha R et al. SQL-PaLM: Improved Large Language Model Adaptation for Text-to-SQL. arXiv preprint arXiv:2306.00739. 2023. https://doi.org/10.48550/arXiv.2306.00739

[62] Anil R, Dai AM, Firat O, Johnson M, Lepikhin D et al. Palm 2 technical report. arXiv preprint arXiv:2305.10403. 2023. https://doi.org/10.48550/arXiv.2305.10403

[63] Touvron H, Martin L, Stone K, Albert P, Almahairi A et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288. 2023. https://doi.org/10.48550/arXiv.2307.09288

[64] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531. 2015. https://doi.org/10.48550/arXiv.1503.02531