Texila International Journal of Academic Research
ISSN : 2520-3088
DOI: 10.21522/TIJAR.2014.11.03.Art009

On Deep Learning Approaches for Single-Turn Text-to-SQL Parsing: Concepts, Methods, and Future Directions

Nathan Manzambi Ndongala
Department of Computer Science, Texila American University, Guyana

Abstract

This literature review delves into deep learning techniques for text-to-SQL parsing, exploring datasets, evaluation metrics, models, and methodologies. The study aims to provide a comprehensive overview of the field, analyzing strengths, weaknesses, accuracy, practical applications, and scalability of various approaches. By examining the current landscape and future directions, this work serves as a valuable resource for researchers, industry professionals, and enthusiasts interested in Natural Language Processing and neural semantic parsing.

Keywords: *Deep Learning, Natural Language Processing, Neural Semantic Parsing, Pretrained Language Model, Prompt Engineering, Single-Turn Text-To-SQL, Seq2seq, Transformers.*

Introduction

The field of Natural Language Interfaces to Databases (NLIDB) plays a crucial role in bridging the gap between human users and complex database systems. One of the key tasks within NLIDB is Text-to-SQL, which involves transforming natural language queries into executable SQL queries that can retrieve information from databases. This task is essential for enabling users to interact with databases using everyday language, eliminating the need for knowledge of complex query languages.

Text-to-SQL has garnered significant attention due to its practical applications in various domains, including information retrieval, data analysis, and decision-making processes. By enabling users to express their information needs in natural language, Text-to-SQL systems enhance the accessibility and usability of databases for a wide range of users, including those without technical expertise in SQL query writing.

Advancements in deep learning [1, 2, 11–15, 3–10][16–18][19–21] have revolutionized the Text-to-SQL task [22–24], leading to the development of sophisticated models capable of understanding the semantic nuances of natural language queries and generating accurate SQL representations. These models leverage techniques such as Seq-to-seq frameworks, Transformer-based approaches, pre-trained models, and Prompt Engineering strategies to improve the accuracy and efficiency of Text-to-SQL systems.

Despite advancements in text-to-SQL models, there is a lack of comprehensive analysis comparing different architectures and methodologies. This hinders a holistic understanding of the field and future model development. In addition, the diversity and complexity of datasets used for training and evaluating text-to-SQL models vary significantly, affecting model performance and generalization ability. In this work, we try to respond to these questions:

1. what are the best practices for dataset curation and utilization in the text-to-SQL domain?
2. What are the strengths and weaknesses of the current text-to-SQL models and architectures, and how do different

Received: 07.06.2024 ***Accepted: 09.07.2024*** ***Published on: 30.07.2024***
****Corresponding Author: nathanmanzambi@yahoo.fr***

methodologies compare in terms of accuracy, scalability, and practical applications?

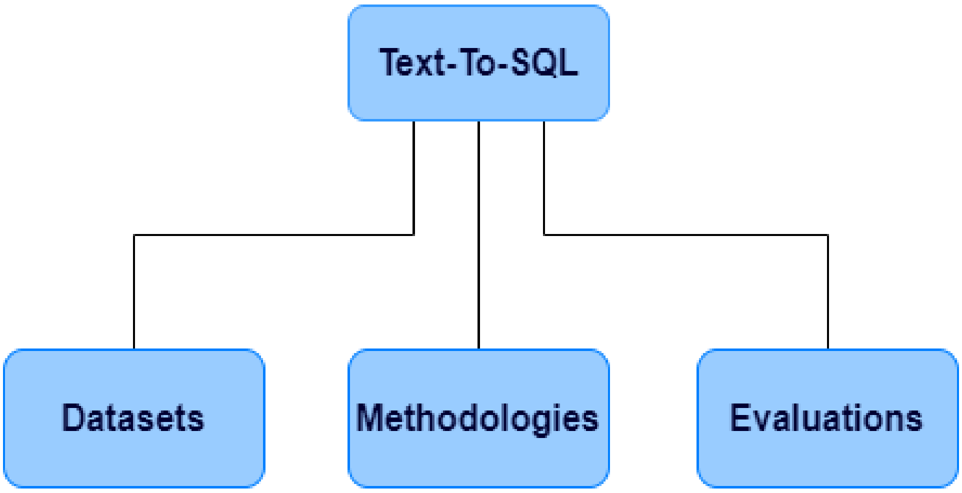

Figure 1: High-level Topology for Text-to-SQL

This work aims to review deep learning techniques used in the field of Text-to-SQL i.e. the dataset, the evaluation, and models. Our objective is to

1. create a high-level map (Fig1.) that providing a comprehensive review of text-to-SQL.
2. provide an analysis of each component of this map.
3. discuss the future direction in the field of text-to-SQL.

Initially, we suggest **4D as the** best practice for dataset curation, encompassing **D**iversity of Schema, **D**iversity (Complexity) of Queries, **D**ata Size and Quality, and **D**omain Specificity, to evaluate a text-to-SQL dataset. Subsequently, we perform a **SWAPS analysis**, focusing on **S**trengths, **W**eaknesses, **A**ccuracy, **P**ractical Applications, and **S**calability of various text-to-SQL deep learning approaches. This dual analysis framework serves as a valuable resource for academia, industry professionals, and individuals interested in delving into the realms of Natural Language Processing (NLP), neural semantic parsing, or Text-to-SQL. By offering insights into dataset characteristics and deep learning model assessments, this resource aims to guide advancements in the field and facilitate informed decision-making for researchers, practitioners, and enthusiasts seeking to explore the intricacies of NLIDB technologies.

The rest of this paper is organized as follows: In the second section, we formally define the text-to-SQL problem and examine the prevalent architecture in deep learning approaches, specifically the sequence-to-sequence (seq-to-seq) architecture. We elucidate the encoder-decoder framework, detailing each constituent component: the encoder, the attention mechanism, and the decoder. The third section provides an in-depth analysis of the datasets employed in single-turn text-to-SQL tasks. We discuss their characteristics, advantages, and domains, distinguishing between single-domain and cross-domain datasets.

The fourth section scrutinizes the evaluation metrics used in text-to-SQL challenges, focusing on the two principal metrics: exact match accuracy (**EM**) and Execution accuracy (**EX**). Achieving high exact-match accuracy is challenging due to the complex syntax and structure of SQL queries. Small variations or even semantically equivalent queries with different syntax can result in a lower score. On the other hand, EM requires access to the actual database and the ability to execute queries, which might not always be feasible or scalable.

Additionally, databases must be kept consistent to ensure fair evaluation.

The fifth section categorizes and examines the methodologies utilized in text-to-SQL, which we have grouped into four categories: seq-to-seq models, transformer-based models, pre-trained text-to-SQL models, prompt engineering techniques, particularly in the context of emerging Large Language Models (LLMs).

The final section before the conclusion outlines potential future research directions, informed by our investigation and a review of relevant literature.

Background

Problem Formulation

Given a natural language question **Q=q1…q**$_{|Q|}$, a database schema **S = <C, T>** with columns **C= {c1, …, c**$_{|C|}$ **}** and tables **T={ t1,…, t**$_{|T|}$ **}.** The problem of text-to-SQL involves converting natural language queries expressed in text into executable SQL queries that can be used to retrieve information from a database.

The objective of text-to-SQL is to predict the SQL query ***y*** from the input **<Q, S>**

The tasks and challenges are to build a model that:

1. Understands **Semantic Understanding**: The model needs to understand the semantic meaning of the natural language query. This involves parsing the input to identify entities, relationships, and conditions.
2. Generates **SQL Query**: Once the semantic understanding is achieved, the model must generate the corresponding SQL query. This includes selecting the appropriate tables, columns, conditions, and other SQL syntax elements.
3. Handles **Ambiguity**: Natural language queries can be ambiguous. The model should handle ambiguity and uncertainty to provide accurate and contextually appropriate SQL queries.
4. Manages **Variability in Language**: Users can express the same query in various ways. The model should be robust to different ways of asking the same question.

The most used model in such a challenge is an encoder and decoder architecture with attention mechanisms.

Encoder-Decoder Framework

The encoder-decoder model is a fundamental architecture used in sequence-to-sequence learning[6], particularly in tasks such as machine translation. This framework is used with attention mechanism [25–28], so we have 3 main components:

Encoder: The encoder is responsible for processing the input sequence and creating a fixed-size context vector that captures the relevant information from the input. Each element of the input sequence is encoded into a hidden representation.

1. Embedding Layer: Converts input tokens (words or sub words) into continuous vector representations. Provides a dense representation of the input words.
2. Recurrent Neural Network (RNN) [29–31] or Transformer Layers [15]: RNN: Captures sequential dependencies in the input. Transformer: Allows for parallel processing of input sequences. Each step processes one token and updates the hidden state.
3. Hidden States: At each time step, the encoder produces a hidden state vector that summarizes the information up to that point in the input sequence.

The final hidden states are used as the context vectors that encode the entire input sequence.

Attention Mechanism [25, 27, 28]:

The attention mechanism allows the decoder to focus on different parts of the input sequence while generating each element of the output sequence. It helps in handling long-range dependencies and improves the model's ability to capture context.

1. Attention Scores: Calculates attention scores for each pair of encoder hidden states and the current hidden state of the decoder. Indicates how much focus should be given to each position in the input sequence when generating the current output element.
2. Context Vector: A weighted sum of the encoder's hidden states based on the attention scores. Provides a dynamic representation of the relevant parts of the input sequence for the current decoding step.

Decoder

The decoder generates the output sequence based on the context vector from the attention mechanism and the previously generated elements of the output sequence.

1. Embedding Layer: Converts the previously generated output tokens into continuous vector representations. Recurrent Neural Network (RNN) or Transformer Layers:
2. RNN: Captures sequential dependencies in the output. Transformer: Allows for parallel processing of output sequences.
3. Hidden States: At each decoding step, the decoder updates its hidden state based on the previously generated tokens and the context vector from the attention mechanism.
4. Output Layer: Generates the probability distribution over the vocabulary for the next token in the output sequence. A SoftMax[32] activation function is commonly used to produce these probabilities.

The model is trained using pairs of input and output sequences. The training objective is to minimize the difference between the predicted output sequence and the target sequence.

The encoder processes the input sequence, the attention mechanism captures relevant information, and the decoder generates the output sequence. The attention mechanism allows the model to focus on different parts of the input sequence during the decoding process, enhancing its ability to handle various types of input-output relationships.

Text-To-SQL Dataset

In this section, we delve into the intricacies of the Text-to-SQL dataset, a foundational component in the realm of Natural Language Interfaces to Databases (NLIDB). The dataset serves as a crucial resource for training and evaluating Text-to-SQL models, providing a diverse range of queries and schemas that challenge the semantic understanding and query generation capabilities of deep learning systems.

We begin by exploring the high-level topology of the Text-to-SQL dataset (Fig2.), shedding light on its structure, composition, and relevance to real-world applications. Through a detailed examination of key datasets such as GeoQuery and Restaurants, we uncover the unique characteristics and challenges posed by each dataset, ranging from geographical information to restaurant details.

Furthermore, we analyze the role of datasets in shaping the performance and generalizability of Text-to-SQL models, emphasizing the importance of dataset quality, size, and complexity in driving advancements in NLIDB technologies. By unraveling the nuances of Text-to-SQL datasets, we aim to provide a comprehensive understanding of the foundational elements that underpin the development and evaluation of state-of-the-art Text-to-SQL systems.

Here below some Text-To-SQL datasets:

GeoQuery [33]

The GeoQuery dataset is focused on U.S. geography and contains approximately 800 Prolog facts asserting relational tables for basic information about U.S. states. This includes information such as population, area, capital city, neighboring states, major rivers, major cities, and the highest and lowest points along with their elevations.

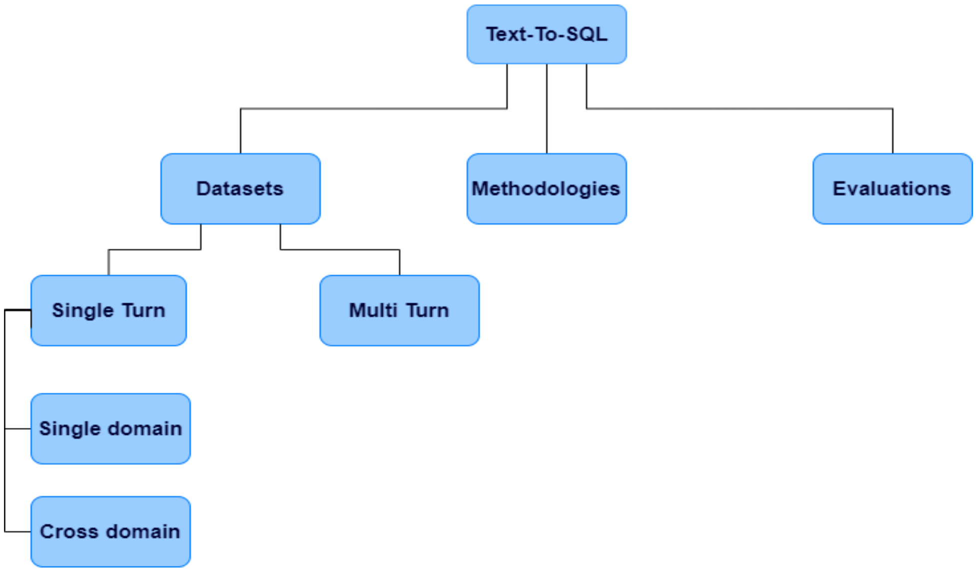

Figure 2. High-level Topology for text-to-SQL with Dataset Details

Restaurants [34]

The Restaurant dataset contains information about thousands of restaurants in Northern California, including the name of the restaurant, its location, its specialty, and a guidebook rating. The dataset was used in the context of a probabilistic framework for semantic shift-reduce parsing, which was applied using a hand-built grammar constructed to reflect typical queries in this domain and translated into a logic form resembling SQL. The algorithm used for the Restaurant dataset was TABULATE, an ILP method motivated by combining the advantages of two ILP approaches in CHILLIN. The dataset was one of three domains used to demonstrate the performance of the new approach, with the other two domains being U.S. Geography and Job query systems. The results of the experiments show that the probabilistic framework achieved near-perfect accuracy in both recall and precision in the Restaurant domain, given only roughly 30% of the training data.

Scholar [35]

The SCHOLAR dataset is a semantic parsing dataset for academic database searches. It comprises natural language utterances labeled with SQL queries, specifically designed for querying an academic database. The dataset includes 816 labeled utterances, divided into a 600/216 train/test split. Additionally, the dataset is accompanied by a database containing academic papers with their authors, citations, journals, keywords, and datasets used.

WikiSQL [36]

The WikiSQL dataset is a crucial component of the Seq2SQL project, providing a large corpus of hand-annotated instances of natural language questions, SQL queries, and SQL tables extracted from Wikipedia. This dataset is an order of magnitude larger than previous semantic parsing datasets, containing 80,654 examples and 24,241 tables. The tables used in WikiSQL are available in both raw JSON format and as an SQL database, making it a comprehensive and valuable resource for training and evaluating natural language interfaces for databases. The collection of WikiSQL involved a paraphrase phase and a verification phase. During the paraphrase phase, tables extracted from Wikipedia were used, and small tables were removed based on specific criteria to ensure the quality and relevance of the data. The dataset was then crowd-sourced on Amazon Mechanical Turk, where workers paraphrased generated questions for the tables. The paraphrases were subsequently verified by other workers to ensure accuracy and variation. This meticulous process resulted in a high-quality dataset suitable for training and evaluating natural language interfaces for databases. Overall, WikiSQL serves as a valuable resource for developing and testing natural language interfaces for databases, providing a diverse and realistic collection of questions, SQL queries, and SQL tables extracted from the web.

ParaphraseBench [37]

ParaphraseBench is a benchmark dataset curated as part of the DBPal project to test the robustness of natural language interfaces for databases (NLIDBs) against different linguistic variations. It consists of 290 pairs of NL-SQL queries that model a medical database with one table containing patient attributes such as name, age, and disease. The queries are grouped into categories based on the linguistic variation used in the NL query, including naive, syntactic paraphrases, morphological paraphrases, lexical paraphrases, and missing information. The benchmark is available online and can be used to evaluate the performance of NLIDBs.

Spider [38]

The Spider dataset is a large-scale, human-labeled dataset designed for complex and cross-domain semantic parsing and text-to-SQL tasks. It consists of over 10,000 questions and 5,600 unique complex SQL queries on 200 databases, covering 138 different domains. The dataset was annotated by 11 college students and is distinct from previous semantic parsing tasks in that it requires models to generalize well to both new SQL queries and new database schemas.

This makes it a challenging and realistic semantic parsing task, presenting ample opportunities for future research and improvement in the field of natural language processing.

The Spider dataset is unique in that it contains databases with multiple tables in different domains and complex SQL queries, testing the ability of a system to generalize not only to new SQL queries and database schemas but also to new domains. It addresses the need for a large and high-quality dataset for a new complex and cross-domain semantic parsing task, providing a valuable resource for researchers and practitioners in the field.

The dataset and task are publicly available, allowing for experimentation and development of state-of-the-art models to tackle the challenges presented by the Spider dataset. It has the potential to benefit both the natural language processing and database communities, offering opportunities for advancements in semantic parsing and text-to-SQL tasks.

CSpider [39] Chinese Version of Spider

Spider-SSP [40]

Spider-SSP refers to a specific split of the SPIDER dataset, which is a non-synthetic text-to-SQL dataset containing 10,181 questions and 5,693 unique SQL queries across 138 domains. The Spider-SSP split consists of 3,282 training examples and 1,094 test examples, and it includes various subsets such as a random split, a split based on source length, a TMCD split, and a template split. The primary evaluation of models on Spider-SSP involves the text-to-SQL task, which presents challenges related to schema linking and modeling complex SQL syntax. The PDF discusses the results of different models, including T5-Base, T5-3B, NQG-T5-Base, and NQG-T5-3B, on the Spider-SSP split, highlighting the performance of NQG-T5 despite the complex nature of SQL syntax. The findings suggest that while the text-to-SQL task is not well modeled by the NQG grammar due to SQL's complex syntax, NQG-T5 still performs well by relying on T5. Overall, Spider-SSP serves as a valuable benchmark for evaluating the performance of semantic parsing models in handling natural language variation and compositional generalization challenges.

Spider-Syn [41]

Spider-Syn is a curated dataset derived from the Spider benchmark, modifying NL questions by substituting schema-related words with selected synonyms. This alteration disrupts the direct correspondence between questions and table schemas, leading to a substantial accuracy drop. Proposed defenses, such as incorporating synonym annotations and adversarial training, show effectiveness, with the former being notably impactful.

In Spider-Syn, a total of 5672 questions have been altered compared to the original Spider dataset. Among these modifications, 5634 cases involve changes to the schema item words, while modifications to the cell value words occur in only 27 cases. The alterations are achieved through the replacement of approximately 492 different words or phrases in the questions, utilizing 273 synonymous words and 189 synonymous phrases.

On average, there is almost one change per question in the Spider-Syn examples, with approximately 7.7 words or phrases modified per domain. Notably, the dataset preserves 2201 original Spider questions in the training set and 161 in the development set.

During the modification process between the training and development sets, 52 words or phrases were consistently modified, representing 35% of the changes observed in the development set. This highlights a degree of overlap in the alterations made between these two sets.

Spider-DK [42]

Spider-DK is a human-curated dataset. It is based on the Spider benchmark, which is commonly used for evaluating text-to-SQL translation models. In Spider-DK, NL (natural language) questions are selected from the original Spider dataset, and some samples are modified by adding domain knowledge that reflects real-world question paraphrases. The goal of Spider-DK is to investigate the robustness of text-to-SQL models when faced with questions requiring rarely observed domain knowledge.

Critera2SQL [43]

This dataset is unique in that it focuses on eligibility criteria from clinical trials related to diseases such as Sepsis, Heart attack, Diabetes, and Alzheimer's. It contains 2003 eligibility criteria along with their corresponding SQL queries, covering 984 concepts.

The dataset includes eligibility criteria with varying levels of complexity, encompassing cases such as Order-sensitive, Counting-based, and Boolean-type criteria. These criteria pose challenges that are specific to the medical domain and are not typically addressed in general natural-language-to-SQL datasets.

To create the dataset, the authors collected eligibility criteria from clinical trials registered in Clinicaltrials.gov, focusing on specific keywords related to the diseases of interest. The criteria were preprocessed to ensure clarity and compatibility with electronic health record tables. A concept set was extracted from the eligibility criteria for generating column names in synthetic patient-record tables and for SQL annotations.

The SQL annotations in the dataset were created by SQL experts, following a standardized structure of SELECT statements with **WHERE** clauses. Annotators filled in the conditions part of the WHERE clause based on the eligibility criteria, with column names matching the terms used in the criteria. The dataset also includes additional very long eligibility criteria to enhance coverage of counting-based cases.

Overall, the Criteria2SQL dataset provides a valuable resource for training and evaluating models that aim to automatically parse eligibility criteria and generate corresponding

SQL queries, facilitating the process of cohort definition for clinical research.

SQUALL [44]

SQUALL is a dataset enriching 11,276 English-language Wiki Table Questions with manually created SQL equivalents and alignments between SQL and question fragments.

XSP [45]

The Cross-Database Semantic Parsing (XSP) dataset is a collection of datasets used to evaluate systems that map natural language utterances to executable SQL queries in databases that were not seen during training. Unlike traditional Semantic Parsing tasks that focus on single-database scenarios, XSP introduces additional challenges such as generalizing to new schema structures, domain-specific phrases, and database conventions. In XSP, the training examples consist of input-output pairs ({x(l), y(l), D(l)i}) and evaluation examples consist of input-output pairs ({x(l), y(l), D(l)j}), where each D represents a database. Importantly, the training and evaluation datasets do not overlap, adding complexity to the generalization process.

The XSP dataset aims to address the limitations of traditional Semantic Parsing tasks by evaluating systems on diverse datasets that were originally designed for single-database semantic parsing. By repurposing well-studied datasets like GeoQuery and ATIS in the XSP context, researchers can uncover new generalization challenges and assess the system's ability to adapt to unseen databases with varying schema structures and language use.

The XSP dataset provides a comprehensive evaluation setup for cross-database semantic parsing systems, highlighting the importance of diverse training and evaluation datasets to improve the generalization capabilities of models in this challenging domain.

SEOSS-Queries [46]

The SEOSS-Queries dataset is a comprehensive resource designed to facilitate text-to-SQL and question-answering tasks in the field of software engineering. Here is a description of the dataset:

Objective: The main objective of the SEOSS-Queries dataset is to address the information needs of stakeholders involved in software development projects. It aims to assist in making informed decisions by providing a structured collection of natural language utterances and corresponding SQL queries.

Compilation: The dataset was compiled by extracting natural language utterances and SQL queries from various sources, including previous studies, software projects, issue-tracking tools, and expert surveys. The data collection process involved analyzing literature, stakeholder questions, and content from 33 software projects to refine and orchestrate the queries.

Contents: Natural Language Utterances: The dataset consists of 1,162 English utterances that translate into 166 SQL queries. Each query is accompanied by four precise utterances and three more general ones. Additionally, there are 393,086 labeled utterances extracted from issue tracker comments.

Data Format: The dataset is structured with raw data in a format suitable for training and evaluating text-to-SQL models.

Accessibility: The data is publicly available through the Figshare repository under a specific identification number.

Value and Applications: The SEOSS-Queries dataset provides a valuable resource for machine learning scientists and researchers to train and evaluate text-to-SQL models in the software engineering domain. Stakeholders, such as developers, can leverage text-to-SQL models trained on this dataset to query database information efficiently for decision-making. The dataset can be utilized in the fields of Machine Learning and Natural Language Processing (NLP) for tasks such as classification, clustering, and analyzing developers' information needs. The SEOSS-Queries dataset offers a rich collection of natural language utterances and SQL queries tailored to address the information needs of stakeholders in software engineering projects, providing a valuable resource for research and practical applications in the field.

FIBEN [47]

The FIBEN dataset is a significant component of the ATHENA++ system, designed to facilitate the handling of complex business intelligence queries through natural language. Here is a detailed description of the FIBEN dataset:

Composition: The FIBEN dataset is constructed by combining two financial datasets, namely the SEC and TPoX datasets.

Complexity: It is noted for its complexity, with a significantly larger number of tables per database schema compared to other benchmarks. This complexity mirrors that of an actual financial data warehouse, providing a realistic environment for query evaluation.

Ontologies: The FIBEN dataset is based on a combination of two financial ontologies, namely the Financial Industry Business Ontology (FIBO) and the Financial Report Ontology (FRO). These ontologies contribute to the domain complexity required to express real-world financial business intelligence queries effectively.

Query Types: The dataset contains 300 natural language queries, each corresponding to 237 distinct SQL queries. These queries cover a wide range of nested query types, ensuring a comprehensive evaluation of the system's capabilities.

Query Generation: The natural language queries in the FIBEN dataset are typical analytical queries crafted by business intelligence experts. These experts were tasked with creating queries that encompass all four nested query types, along with various SQL query constructs, ensuring a diverse and challenging query set.

Availability: The benchmark queries of the FIBEN dataset are accessible at the following link: https://github.com/IBM/fiben-benchmark

The FIBEN dataset serves as a robust benchmark for evaluating the performance of NLIDB systems, particularly in handling complex financial business intelligence queries. Its comprehensive nature and realistic representation of financial data make it a valuable resource for advancing natural language interfaces to databases.

CSS [48]

CSS, a large-scale Cross-Schema Chinese text-to-SQL medical dataset, addresses the challenges of cross-domain and single-domain text-to-SQL tasks by proposing a cross-schema text-to-SQL task. CSS consists of 4,340 question/SQL pairs across 2 databases, expanded to 19 databases with 29,280 examples for generalized model training. The dataset also serves as a significant corpus for single-domain Chinese text-to-SQL studies, and benchmarking baselines showcase its potential and utility.

Aclanthology [49]

The ACL Anthology dataset is a collection of more than 40,000 research articles published in computational linguistic events, including conferences, workshops, and journals. It represents one of the largest collections of natural language processing research papers and provides a comprehensive resource for researchers in the field. The dataset is used to construct the Computational Linguistic Knowledge Graph (CLKG). The CLKG construction methodology involves processing full-text PDFs, extracting metadata and structure information, and constructing a heterogeneous graph consisting of four entities: author, paper, venue, and field. CLKG facilitates high-quality search and exploration of current research progress in the computational linguistics field.

DuSQL [50]

DuSQL is a comprehensive industry-oriented dataset for natural language interface to databases. It contains a huge number of question/SQL pairs covering a wide range of domains such as cities, singers, movies, animals, etc. It is much larger than other complex datasets and covers about 70% of information from Baike. DuSQL conforms to the distribution of SQL queries in real applications and contains enough question/SQL pairs for all common types. It is constructed based on a thorough quality control process and conforms to various evaluation metrics. Overall, DuSQL is a pragmatic and valuable dataset for natural language interaction with databases.

KaggleDBQA [51]

The KaggleDBQA dataset contains a total of 1,687 questions, split across eight databases. Each question is paired with at least one SQL query that corresponds to the correct answer to the question. The questions are constructed to be similar to those that a user might ask of a database, where the user has limited knowledge of the database schema and terminology. The queries are annotated with the table and column names that they use, allowing for more accurate evaluation of model performance. The dataset includes a "few-shot" evaluation setting, where a model is trained on a small number of examples and evaluated on a separate set of questions, to simulate a more realistic scenario where a model is given limited training data before being deployed in a new application.

Dataset Influence on Performance and Generalization

Different types of datasets play a crucial role in influencing the performance and generalization of text-to-SQL models. Table 1. presents an overview of text-to-SQL benchmarks, showcasing various datasets used for training and evaluating text-to-SQL models, highlighting the diversity in dataset characteristics that can influence model performance and generalization.

1. **Diversity of Schemas**: Datasets with diverse database schemas help models generalize better to unseen databases by exposing them to a wide range of structures and query types.
2. **Diversity (Complexity) of Queries**: Datasets containing complex and varied natural language queries challenge models to handle diverse linguistic patterns, enhancing their generalization capabilities.
3. **Data Size and Quality**: Larger datasets with high-quality annotations contribute to improved model performance by providing more training examples and reducing overfitting.
4. **Domain Specificity**: Domain-specific datasets focus on particular industries or topics, allowing models to specialize in specific domains but may limit generalization to other domains.

Table 1. Overview of text-to-SQL Benchmarks

				#Domain	#SQL	#DB	#Tables		
1	GenQuery	Single Turn	en	1	247	1	6	880	Single domain
2	Restaurants	Single Turn	en	1	378	1	3	525	Single domain
3	Scholar	Single Turn	en	1	193	1	7	817	Single domain
4	WikiSQL	Single Turn	en		77840	26521	1	80654	Single domain
5	ParaphraseBench	Single Turn	en	1			1	290	Single domain
6	Spider	Single Turn		138	5693	200	1020	10181	cross-domain
7	CSpider	Single Turn	zh	138	5693	200	1020	10181	cross-domain

8	Critera2SQL	Single Turn	en	1				2003	Single domain
9	SQUALL	Single Turn	en		11276	2108	2108	15620	cross-domain
10	XSP	Single Turn	en						cross-domain
11	Spider-Syn	Single Turn	en		4525	160	876	8034	cross-domain
12	Spider-DK	Single Turn	en		283	10	48	535	cross-domain
13	SEOSS-Queries	Single Turn			166			1162	
14	FIBEN	Single Turn	en	1	237	1		300	Single domain
15	CSS	Single Turn	en			19		29280	cross-domain
16	DuSQL	Single Turn	zh		23797	200	820	23797	cross-domain
17	KaggleDBQA	Single Turn	en		1687	8		1687	

Best Practices for Dataset Curation and Utilization

1. **Diverse Schema Representation**: Curate datasets with a variety of database schemas to expose models to different structures and improve generalization.
2. **Data Augmentation**: Augment datasets by generating variations of existing queries and schemas to increase diversity and robustness in model training.
3. **Quality Annotations**: Ensure high-quality annotations in datasets to provide accurate ground truth for training models effectively.
4. **Cross-Domain Training**: Incorporate datasets from multiple domains to train models on diverse data sources, enhancing their ability to generalize across different domains.
5. **Regular Updates**: Continuously update and expand datasets to reflect evolving database structures and query patterns, ensuring models remain relevant and adaptable to new challenges.

To meet all these requirement, [52] proposes The UNITE benchmark comprises 18 publicly available text-to-SQL datasets covering various domains, including Wikipedia, healthcare, education, geography, transportation, software engineering, and finance.

Evaluation Metrics

Metrics that consider the semantic meaning of the generated SQL query. This includes evaluating whether the generated query correctly captures the user's intent.

Execution Accuracy

1. Definition: Measures the percentage of generated SQL queries that execute successfully on the corresponding database.
2. Calculation: The number of correctly executed queries divided by the total number of queries.

$$\textbf{score}\ (\mathbf{V}, \mathbf{V}^{\mathbf{hat}}) = \begin{cases} 1, V = V^{hat} \\ 0, V \neq V^{hat} \end{cases}$$

$$\textbf{EX} = \frac{\sum_{n=1}^{N} score(V, V^{hat})}{N}$$

Exact Match Accuracy

1. Definition: Measures the percentage of generated SQL queries that exactly match the reference (ground truth) SQL queries.
2. Calculation: The number of queries with an exact match divided by the total number of queries.

$$\textbf{score}(\mathbf{y}, \mathbf{y}^{\mathbf{hat}}) = \begin{cases} 1, y = y^{hat} \\ 0, y \neq y^{hat} \end{cases}$$

$$\textbf{EM} = \frac{\sum_{n=1}^{N} score(y, y^{hat})}{N}$$

Single-Turn Common Approach

The transformation of natural language text into SQL queries, commonly referred to as

Text-to-SQL, is an important task at the intersection of natural language processing (NLP) and databases. This task has significantly benefited from advancements in deep learning technologies. The methodologies can be broadly categorized into **Seq-to-seq models, Transformer-based approaches, pre-trained models, and Prompt Engineering strategies**. Each of these approaches has contributed uniquely to the progress in text-to-SQL research.

Tabel 2. gives cons and pros of each approach.

Seq-to-seq Text to SQL

Pure Seq-to-Seq

Sequence-to-sequence (Seq-to-seq) frameworks are among the pioneering techniques applied to the Text-to-SQL task. These models utilize an encoder-decoder architecture to transform natural language input into SQL queries. The encoder processes the input sentence into a fixed-length context vector, and the decoder generates the corresponding SQL query based on this context. [53] introduced data recombination techniques to improve robustness and versatility in Seq-to-seq models for semantic parsing. They showcased how the models could effectively generalize to a variety of inputs by recombining training data to form novel combinations.

One significant advancement in Seq-to-seq modeling is the Seq2SQL model by [36], which combines sequence-to-sequence neural networks with reinforcement learning to optimize SQL generation. This combination addresses the challenge of ensuring that generated queries are both accurate and efficiently structured. Reinforcement learning enables the model to fine-tune its outputs by maximizing rewards based on the correctness of the SQL query. This approach demonstrated substantial improvements in execution accuracy and logical form accuracy over traditional supervised learning methods.

[54] further explored the feasibility of building an effective semantic parser with minimal annotated data, emphasizing the practicality of such models in real-world applications. Their work demonstrated the potential of creating robust models for translating natural language to SQL, even with limited training data.

Seq-to-tree [55,56,65–70,57–64]

Coarse-to-Fine Decoding

[62] introduce a structure-aware neural architecture with a coarse-to-fine decoding approach for semantic parsing [56,60,64,65,70]. This method involves generating a rough sketch of the meaning representation first, followed by filling in the details. This two-stage process allows the model to handle low-level information more effectively and results in competitive performance across various domains, despite using relatively simple decoders.

Grammar-Based Decoding

Grammar-based decoders [57,61,66,67,69] are particularly adept at capturing the syntactic rules of SQL. [61] propose a Syntactic Neural Model for general-purpose code generation that uses a grammar model to capture the underlying syntax of the target programming language. This grammar-oriented approach has shown effectiveness in scaling the generation of complex programs from natural language, achieving improved results over traditional code generation techniques.

In a similar vein, [66] presents TRANX, a transition-based neural abstract syntax parser that leverages an abstract syntax description language. TRANX demonstrates high accuracy by utilizing the syntax of the target meaning representation to constrain the output space, thereby enhancing the generalizability and effectiveness of the decoder in semantic parsing and code generation tasks.

Table 2. SWAPS Analysis of Different Group

	Strengths	**Weaknesses**	**Accuracy**	**Practical Applications**	**Scalability**
Seq2SQL Model	Combines sequence-to-sequence neural networks with reinforcement learning for optimized SQL generation, leading to improved execution accuracy and logical form accuracy	May struggle with handling complex schemas and long-range dependencies	Demonstrated substantial improvements in accuracy over traditional supervised learning methods	Effective for generating accurate and structured SQL queries, suitable for various real-world applications	May face challenges in scaling to handle complex schemas efficiently
Transformer-based Methods	Revolutionized NLP tasks, including Text-to-SQL, by capturing long-range dependencies effectively	May require significant computational resources and data for training	Demonstrated high accuracy in handling complex schemas and dependencies	Suitable for tasks requiring understanding and generation of SQL queries from natural language inputs	Can be scalable but may require optimization for efficiency in large-scale applications.
Prompt Engineering and Pretrained Models	Enhance model adaptability, contextual understanding, and performance in generating SQL queries	May rely heavily on pre-training data and prompt engineering for optimal performance.	Significantly improves model performance in SQL query generation	Offers practical solutions for real-world Text-to-SQL tasks with minimal training data	Can be scalable with proper engineering and optimization

However, Seq-to-seq or Seq-To-Tree Text-To-SQL models, even with attention mechanism [56] [71], have limitations in handling complex schemas and long-range dependencies in text, which has paved the way for exploring more advanced architectures like transformers.

1. **Transformer-Based Methods**

Transformers have revolutionized natural language processing (NLP) tasks, including text-to-SQL, by enabling the modeling of long-range dependencies and parallelized training. The self-attention mechanisms of transformer models allow them to capture dependencies between tokens in a sentence, making them exceptionally suitable for understanding and generating SQL queries from natural language inputs.

[72] introduced BERT (Bidirectional Encoder Representations from Transformers), which set new benchmarks in NLP by leveraging deep bidirectional context understanding. BERT’s attention mechanisms enable it to model intricate relationships in input data, critical for tasks like Text-to-SQL. BERT’s pre-training on extensive corpora results in rich contextual embeddings that improve the model’s understanding and generation of language.

[73] proposed TypeSQL, which uniquely incorporates type information from the database schema into the transformer model. By resolving ambiguities in natural language queries through type annotations, TypeSQL ensures more accurate and contextually relevant SQL queries. This approach significantly improves the model’s performance in understanding and processing various database schema types. [74]

emphasized the importance of encoding database structures using graph neural networks. [75] introduced BRIDGE, a sequential architecture leveraging BERT to model dependencies between natural language questions and relational databases.

Following BERT and its variants[76–79], [80] developed RAT-SQL (Relation-Aware Transformer for SQL), incorporating relation-aware self-attention mechanisms to encode complex schema information and context. This model extends BERT by integrating relational data understanding with a transformer architecture, enhancing accuracy in generating SQL queries by mapping schema relations alongside the input text. Other variants of RAT-SQL[81] [82] [83, 84] arise after the original RAT-SQL. [85] introduced LGESQL, a line graph-enhanced model that integrates local and non-local relations within the graph iteration.

The advancements in transformer-based methods have demonstrated their capacity to handle complex schemas and long-range dependencies effectively, making them robust solutions for the Text-to-SQL challenge.

2. **Pretrained Text-to-SQL Models**

Pretrained language models have shown immense potential in various NLP tasks, including Text-to-SQL. These models leverage pre-trained embeddings encapsulating rich language semantics and syntax, enabling effective understanding and generation of SQL queries with limited task-specific training. [86] introduced PICARD, a method for constraining large pre-trained language model decoders through incremental parsing, significantly enhancing performance on challenging benchmarks like Spider.

[87] introduced TaPas, a model built on BERT to process and generate queries from structured table data. TaPas employs weak supervision and pre-trained BERT embeddings, enhancing the model's robustness in SQL generation. This approach excels in scenarios involving tables, effectively interpreting structured data to generate SQL queries.

[17] demonstrated the efficacy of pre-trained language models for semantic parsing, showing that fine-tuning these models on task-specific data significantly outperforms traditional methods. Pre-trained embeddings provide rich linguistic understanding, allowing the model to handle diverse and complex queries effectively.

[88] presents a Structure-Grounded pretraining framework (StruG) for text-to-SQ that can effectively learn to capture text-table alignment based on a parallel text-table corpus. Another significant model is StructBERT by [89]**,** which integrates structural information from SQL grammar into the BERT model, enabling the handling of complex SQL queries more effectively. StructBERT's design incorporates both syntactic and semantic aspects, resulting in improved performance on challenging text-to-SQL benchmarks.

Other researchers have focused on novel pre-training and encoding techniques to enhance text-to-SQL models. [90] presented a grammar pre-training method (GP) to decode deep relations between questions and databases, while [91] introduced GraPPa, a grammar-augmented pre-training approach for table semantic parsing.

These pre-trained models exemplify the power of leveraging extensive pre-training to achieve high performance in SQL query generation, even with minimal task-specific supervision.

3. **Prompt Engineering**

Prompt engineering is an innovative technique that leverages pre-trained large language models[16,18,97,98,76,77,79,92–96] for various tasks, including Text-to-SQL. This method involves designing specific input prompts to guide pre-trained models in performing desired tasks with minimal task-specific training data, harnessing the extensive knowledge embedded in these models.

[98] showcased the capabilities of GPT-3 through prompt engineering, demonstrating that GPT-3 can perform various tasks, including Text-to-SQL, using few-shot learning

with well-crafted prompts. This approach allows models to understand and execute complex tasks with minimal additional training, highlighting the power of large-scale pretraining.

[99] explored the use of cloze-style prompts for few-shot learning in text classification and natural language inference, showing their effectiveness in improving model performance. Their approach indirectly benefits SQL query generation by enhancing the model's adaptability and contextual understanding. [100] emphasized the effectiveness of prompt engineering in adapting pre-trained models to new tasks with minimal task-specific data. Their research demonstrated significant improvements in SQL query generation by leveraging prompt engineering to enhance model performance.

[101] proposed DIN-SQL, which decomposes the text-to-SQL task into smaller sub-tasks to improve LLM performance. [102] developed DialSQL, a dialogue-based framework that enhances structured query generation through user interaction and validation.

Prompt engineering represents a flexible and powerful method, enabling high performance in text-to-SQL tasks with minimal training data. This approach maximizes the capabilities of large pre-trained models, offering practical solutions for real-world applications.

Future Directions

Although previous methods have made significant strides, there are still several obstacles in creating high-quality text-to-SQL parsers. Building on the research presented in this manuscript, we identify several avenues for future investigation in the text-to-SQL parsing domain:

1. **Enhanced Generalizability**: Future research could focus on improving model generalizability to unseen databases. Developing models that can effectively adapt to new database schemas with minimal fine-tuning could lead to more versatile and widely applicable text-to-SQL systems.

[103] initially found that existing text-to-SQL datasets are too structured to effectively assess the potential for generalization. Therefore, they developed a framework to generate text-to-SQL data for testing generalizability. The outcomes of their experiments indicate a lack of model generalization. Furthermore, the analysis suggests that overfitting of natural language and database schema patterns is the root cause of this issue. This generalizability can be reached either by adding extra training data to bring more unseen patterns in the evaluation stage[103] or with the Large Language Model for Text-to-SQL as proposed by [104].

2. **Interpretability** [105] **and Robustness**: There is a growing need for text-to-SQL models to be more interpretable and robust. Research efforts could concentrate on developing models that not only generate accurate SQL queries but also provide explanations for their decisions, enhancing transparency and trust in the system's outputs.

3. **Human Interaction Integration**: Leveraging human interaction for error correction and validation in text-to-SQL systems could be a promising direction. Developing interactive frameworks that allow users to provide feedback on generated queries and refine them collaboratively could improve the overall accuracy and user experience of such systems.

4. **Multi-Modal Data Integration**: Integrating multiple modalities of data, such as text, images, and audio, into text-to-SQL models could open up new possibilities for more comprehensive and contextually rich query generation. Research in this area could explore how different data types can be effectively combined to enhance the understanding and generation of SQL queries.

5. **Efficiency and Scalability**: Future advancements in text-to-SQL research could focus on developing more efficient and scalable

models. Improving the computational efficiency of models while maintaining high performance could enable the deployment of text-to-SQL systems in real-time applications and large-scale databases.

Conclusion

In conclusion, this literature review highlights the significance of deep learning in advancing text-to-SQL parsing, emphasizing the need for comprehensive dataset curation, model evaluation, and future research directions. The analysis of strengths and weaknesses, along with a focus on accuracy and scalability, underscores the importance of informed decision-making in developing effective NLIDB technologies. As the field continues to evolve, addressing challenges such as generalizability, interpretability, and human interaction integration will be crucial for enhancing the usability and trustworthiness of text-to-SQL systems in diverse applications.

Acknowledgments

We would like to thank Christian BOPE for his leadership talks and guidance that shaped this review. We also thank anonymous reviewers for their crucial input.

Conflict of Interest Statement

All authors declare that they have no conflicts of interest.

References

[1]. Peng, H., Li, G., Zhao, Y. and Jin, Z., 2022, Rethinking Positional Encoding in Tree Transformer for Code Representation, *Conference on Empirical Methods in Natural Language Processing*, 3204–14.

[2]. He, P., Liu, X., Gao, J. and Chen, W., 2021, Deberta: Decoding-Enhanced Bert with Disentangled Attention.

[3]. Vinyals, O., Fortunato, M. and Jaitly, N., 2015, Pointer Networks, *Advances in Neural Information Processing Systems*, p. 2692–700.

[4]. Kool, W., Van Hoof, H. and Welling, M., 2019, Attention, Learn To Solve Routing Problems! *ICLR*,.

[5]. Gu, J., Lu, Z., Li, H. and Li, V. O. K., 2016, Incorporating Copying Mechanism in Sequence-to-Sequence Learning.

[6]. Sutskever, I., Vinyals, O. and Le, Q. V., 2014, Sequence to Sequence Learning with Neural Networks, *Advances in Neural Information Processing Systems*, p. 3104–12.

[7]. Ba, J. L., Kiros, J. R. and Hinton, G. E., 2016, Layer Normalization.

[8]. Galassi, A., Lippi, M. and Torroni, P., 2021, Attention in Natural Language Processing, *IEEE Transactions on Neural Networks and Learning Systems*, *Institute of Electrical and Electronics Engineers Inc*, **32**, 4291–308, https://doi.org/10.1109/TNNLS.2020.3019893.

[9]. Shaw, P., Uszkoreit, J. and Vaswani, A., 2018, Self-Attention with Relative Position Representations, *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, p. 464–8, https://doi.org/10.18653/v1/n18-2074.

[10]. Jia, W., Dai, D., Xiao, X. and Wu, H., 2020, Arnor: Attention Regularization Based Noise Reduction for Distant Supervision Relation Classification, *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, p. 1399–408, https://doi.org/10.18653/v1/p19-1135.

[11]. Zehui, L., Liu, P., Huang, L., Chen, J., Qiu, X. and Huang, X., 2019, DropAttention: A Regularization Method for Fully-Connected Self-Attention Networks.

[12]. He, Q., Sedoc, J. and Rodu, J., 2021, Trees in Transformers: A Theoretical Analysis of the Transformer's Ability to Represent Trees.

[13]. Shiv, V. L. and Quirk, C., 2019, Novel Positional Encodings to Enable Tree-Based Transformers, *Advances in Neural Information Processing Systems*.

[14]. Kitaev, N., Kaiser, Ł. and Levskaya, A., 2020,

Reformer: The Efficient Transformer, *8th International Conference on Learning Representations, ICLR 2020*.
[15]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N. et al., 2017, Attention is all You Need, *Advances in Neural Information Processing Systems*, p. 5999–6009.
[16]. OpenAI, :, Achiam, J., Adler, S., Agarwal, S., Ahmad, L. et al., 2023, GPT-4 Technical Report, **4**, 1–100.
[17]. Yin, P., Neubig, G., Yih, W. and Riedel, S., 2020, TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data, *Association for Computational Linguistics (ACL)*, 8413–26, https://doi.org/10.48550/arxiv.2005.08314.
[18]. Clark, K., Luong, M. - T., Le, Q. V. and Manning, C. D., 2020, ELECTRA: Pre-training Text Encoders as Discriminators Rather than Generators.
[19]. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury Google, J., Chanan, G. et al., 2019, PyTorch: An Imperative Style, *High-Performance Deep Learning Library*.
[20]. Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J. and Mcclosky, D., 2014, The Stanford CoreNLP Natural Language Processing Toolkit.
[21]. Pennington, J., Socher, R. and Manning, C. D. ,2014, GloVe: Global Vectors for Word Representation [Internet].
[22]. Finegan-Dollak, C., Kummerfeld, J. K., Zhang, L., Ramanathan, K., Sadasivam, S., Zhang, R. et al., 2018, Improving Text-to-SQL Evaluation Methodology. https://doi.org/10.18653/v1/P18-1033
[23]. Li, H., Zhang, J., Li, C. and Chen, H., 2023, RESDSQL: Decoupling Schema Linking and Skeleton Parsing for Text-to-SQL.
[24]. Cai, R., Yuan, J., Xu, B. and Hao, Z., 2021, SADGA: Structure-Aware Dual Graph Aggregation Network for Text-to-SQL.
[25]. Tu, Z., Lu, Z., Yang, L., Liu, X. and Li, H., 2016, Modeling Coverage for Neural Machine Translation, *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, p. 76–85, https://doi.org/10.18653/v1/p16-1008.
[26]. Luong, M.-T., Sutskever, I., Le, Q. V., Vinyals, O. and Zaremba, W., 2014, Addressing the Rare Word Problem in Neural Machine Translation.
[27]. Bahdanau, D., Cho, K. H. and Bengio, Y., 2015, Neural Machine Translation by Jointly Learning to Align and Translate, *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.
[28]. Luong, M. T., Pham, H. and Manning, C. D., 2015, Effective Approaches to Attention-Based Neural Machine Translation, *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, p. 1412–21, https://doi.org/10.18653/v1/d15-1166.
[29]. Schuster, M. and Paliwal, K. K., 1997, Bidirectional Recurrent Neural Networks, *IEEE Transactions on Signal Processing*, **45**, https://doi.org/10.1109/78.650093.
[30]. Hochreiter, S. and Schmidhuber, J., 1997, Long Short-Term Memory. *Neural Computation*, **9**, https://doi.org/10.1162/neco.1997.9.8.1735.
[31]. Cho, K., van Merriënboer, B., Bahdanau, D. and Bengio, Y., 2014, On the Properties of Neural Machine Translation: Encoder–Decoder Approaches, *Proceedings of SSST 2014 - 8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, https://doi.org/10.3115/v1/w14-4012.
[32]. John S., Bridle, 1990, Training Stochastic Model Recognition Algorithms as Networks Can Lead to Maximum Mutual Information Estimation of Parameters, *Advances in Neural Information Processing Systems*, **2**, 211–7.
[33]. Zelle, J. M., Moines, D. and Mooney, J., 1996, Learning to Parse Database Queries Using Inductive Logic Programming.
[34]. Tang, L. R. and Mooney, R. J., 1996, Automated Construction of Database Interfaces : Integrating Statistical and Relational Learning for Semantic Parsing.
[35]. Iyer, S., Konstas, I., Cheung, A., Krishnamurthy, J. and Zettlemoyer, L., 2017, Learning A Neural Semantic Parser From User Feedback, *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics,*

Proceedings of the Conference (Long Papers), **1**, 963–73, https://doi.org/10.18653/v1/P17-1089.
[36]. Zhong, V., Xiong, C. and Socher, R., 2017, Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning.
[37]. Utama, P., Weir, N., Basik, F., Binnig, C., Cetintemel, U., Hättasch, B., et al., 2018, An End-to-end Neural Natural Language Interface for Databases.
[38]. Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z. et al, 2018, Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task.
[39]. Min, Q., Shi, Y. and Zhang, Y., 2019 A Pilot Study for Chinese SQL Semantic Parsing, *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 3652–8, https://doi.org/10.18653/v1/d19-1377.
[40]. Shaw, P., Chang, M. W., Pasupat, P. and Toutanova, K., 2021, Compositional Generalization and Natural Language Variation: Can a Semantic Parsing Approach Handle Both? *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 922–38, https://doi.org/10.18653/v1/2021.acl-long.75.
[41]. Gan, Y., Chen, X., Huang, Q., Purver, M., Woodward, J. R., Xie, J. et al, 2021, Towards Robustness of Text-to-SQL Models Against Synonym Substitution, *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2505–15, https://doi.org/10.18653/v1/2021.acl-long.195.
[42]. Gan, Y., Chen, X. and Purver, M., 2021, Exploring Underexplored Limitations of Cross-Domain Text-to-SQL Generalization, *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, 8926–31, https://doi.org/10.18653/v1/2021.emnlp-main.702.
[43]. Yu, X., Chen, T., Yu, Z., Li, H., Yang, Y., Jiang, X. et al, 2020, Dataset and Enhanced Model for Eligibility Criteria-to-SQL Semantic Parsing, *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings*, 5829–37.
[44]. Shi, T., Zhao, C., Boyd-Graber, J., Daumé, H. and Lee, L., 2020, On the Potential of Lexico-Logical Alignments for Semantic Parsing to SQL Queries, *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, 1849–64, https://doi.org/10.18653/v1/2020.findings-emnlp.167.
[45]. Suhr, A., Chang, M. W., Shaw, P. and Lee, K., 2020, Exploring Unexplored Generalization Challenges for Cross-Database Semantic Parsing, *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 8372–88, https://doi.org/10.18653/v1/2020.acl-main.742.
[46]. Tomova, M. T., Hofmann, M. and Mäder, P., 2022, SEOSS-Queries - a Software Engineering Dataset for Text-to-SQL and Question Answering Tasks, *Data in Brief, Elsevier Inc*, **42**, 108211, https://doi.org/10.1016/j.dib.2022.108211.
[47]. Armin, S., Ghahani, V., Khadirsharbiyani, S., Kotra, J. B. and Kandemir, M. T., 2020, Athena: An Early-Fetch Architecture To Reduce On-Chip Page Walk Latencies, *ProcVLDB Endow*, **22**, 2747–2759, https://doi.org/10.1145/3559009.3569684.
[48]. Zhang, H., Li, J., Chen, L., Cao, R., Zhang, Y., Huang, Y. et al., 2023, CSS: A Large-Scale Cross-Schema Chinese Text-to-SQL Medical Dataset, *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 6970–83, https://doi.org/10.18653/v1/2023.findings-acl.435.
[49]. Singh, M., Dogga, P., Patro, S., Barnwal, D., Dutt, R., Haldar, R. et al, 2018, CL Scholar: The ACL Anthology Knowledge Graph Miner, *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session*, 16–20, https://doi.org/10.18653/v1/n18-5004.

[50]. Wang, L., Zhang, A., Wu, K., Sun, K., Li, Z., Wu, H. et al., 2020, DuSQL: A Large-Scale and Pragmatic Chinese Text-to-SQL Dataset, *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 6923–35, https://doi.org/10.18653/v1/2020.emnlp-main.562.

[51]. Lee, C. H., Polozov, O. and Richardson, M., 2021, KaggleDBQA: Realistic Evaluation of Text-to-SQL parsers, *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2261–73, https://doi.org/10.18653/v1/2021.acl-long.176.

[52]. Lan, W., Wang, Z., Chauhan, A., Zhu, H., Li, A., Guo, J. et al., 2023, UNITE: A Unified Benchmark for Text-to-SQL Evaluation.

[53]. Jia, R. and Liang, P., 2016, Data Recombination for Neural Semantic Parsing, *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers, Association for Computational Linguistics (ACL),* **1**, 12–22. https://doi.org/10.18653/v1/p16-1002.

[54]. Wang, Y., Berant, J. and Liang, P., 2015, Building a Semantic Parser Overnight, *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, **1**, 1332–42, https://doi.org/10.3115/v1/p15-1129.

[55]. Hou, W. and Nie, Y., Seq2seq-Attention Question Answering Model.

[56]. Dong, L. and Lapata, M., 2016, Language to Logical Form with Neural Attention, *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, p. 33–43, https://doi.org/10.18653/v1/p16-1004.

[57]. Xiao, C., Dymetman, M. and Gardent, C., 2016, Sequence-Based Structured Prediction for Semantic Parsing, *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, p. 1341–50, https://doi.org/10.18653/v1/p16-1127.

[58]. Goldman, O., Latcinnik, V., Naveh, U., Globerson, A. and Berant, J., 2018, Weakly Supervised Semantic Parsing with Abstract Examples, *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, p. 1809–19, https://doi.org/10.18653/v1/p18-1168.

[59]. Iyer, S., Konstas, I., Cheung, A., Krishnamurthy, J. and Zettlemoyer, L., 2017, Learning a Neural Semantic Parser from User Feedback, *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, p. 963–73, https://doi.org/10.18653/v1/P17-1089.

[60]. Krishnamurthy, J., Dasigi, P. and Gardner, M., 2017, Neural Semantic Parsing with Type Constraints for Semi-Structured Tables, *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, p. 1516–26, https://doi.org/10.18653/v1/d17-1160.

[61]. Yin, P. and Neubig, G., 2017, A Syntactic Neural Model for General-Purpose Code Generation, *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, p. 440–50, https://doi.org/10.18653/v1/P17-1041.

[62]. Dong, L. and Lapata, M., 2018, Coarse-to-Fine Decoding for Neural Semantic Parsing, *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, p. 731–42, https://doi.org/10.18653/v1/p18-1068.

[63]. Herzig, J. and Berant, J., 2018, Decoupling Structure and Lexicon for Zero-Shot Semantic Parsing, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, p. 1619–29, https://doi.org/10.18653/v1/d18-1190.

[64]. Kamath, A. and Das, R., 2018, A Survey on Semantic Parsing.

[65]. Suhr, A., Iyer, S. and Artzi, Y., 2018, Learning to Map Context-Dependent Sentences to Executable Formal Queries, *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies - Proceedings of the Conference, p. 2238–49, https://doi.org/10.18653/v1/n18-1203.
[66]. Yin, P. and Neubig, G., 2018, Tranx: A Transition-Based Neural Abstract Syntax Parser for Semantic Parsing and Code Generation, *EMNLP 2018 - Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Proceedings*, p. 7–12, https://doi.org/10.18653/v1/d18-2002.
[67]. Yin, P., Zhou, C., He, J. and Neubig, G., 2018, structVae: Tree-Structured Latent Variable Models for Semi-Supervised Semantic Parsing, *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, **1**, 754–65, https://doi.org/10.18653/v1/p18-1070.
[68]. Shi, P., Ng, P., Wang, Z., Zhu, H., Li, A. H., Wang, J. et al., 2021, Learning Contextual Representations for Semantic Parsing with Generation-Augmented Pre-Training, *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, p. 13806–14, https://doi.org/10.1609/aaai.v35i15.17627.
[69]. Baranowski, A. and Hochgeschwender, N., 2021, Grammar-Constrained Neural Semantic Parsing with LR Parsers, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 1275–9, https://doi.org/10.18653/v1/2021.findings-acl.108.
[70]. Huang, S., Li, Z., Qu, L. and Pan, L., 2021, On Robustness of Neural Semantic Parsers, *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*, p. 3333–42, https://doi.org/10.18653/v1/2021.eacl-main.292.
[71]. Xu, X., Liu, C. and Song, D., 2017, SQLNet: Generating Structured Queries from Natural Language without Reinforcement Learning.
[72]. Devlin, J., Chang, M. W., Lee, K., Google, K. T. and Language, A.I. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [Internet].
[73]. Yu, T., Li, Z., Zhang, Z., Zhang, R. and Radev, D., 2018, TypeSQL: Knowledge-Based type-Aware Neural Text-to-SQL Generation.
[74]. Bogin, B., Gardner, M. and Berant, J., 2019, Global Reasoning Over Database Structures for Text-to-SQL Parsing.
[75]. Lin, X. V., Socher, R. and Xiong, C., 2020, Bridging Textual and Tabular Data for Cross-Domain Text-to-SQL Semantic Parsing.
[76]. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O. et al., 2019, BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.
[77]. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D. et al., 2019, ROBERTa: A Robustly Optimized BERT Pretraining Approach.
[78]. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. et al., 2020, Albert: A Lite Bert for Self-Supervised Learning of Language Representations [Internet].
[79]. Sanh, V., Debut, L., Chaumond, J. and Wolf, T., 2019, DistilBERT, A Distilled Version of Bert: Smaller, Faster, Cheaper And Lighter.
[80]. Wang, B., Shin, R., Liu, X., Polozov, O. and Richardson, M., 2020, RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers [Internet].
[81]. [81] Scholak, T., Li, R., Bahdanau, D., de Vries, H. and Pal, C., 2020, DuoRAT: Towards Simpler Text-to-SQL Models, https://doi.org/10.18653/v1/2021.naacl-main.103.
[82]. Hui, B., Geng, R., Wang, L., Qin, B., Li, B., Sun, J. et al., 2022, S^{2}SQL: Injecting Syntax to Question-Schema Interaction Graph Encoder for Text-to-SQL Parsers.
[83]. N. M. Ndongala, 2023, Light RAT-SQL: A RAT-SQL with More Abstraction and Less Embedding of Pre-existing Relations. *Texila International Journal of Academic Research*, **10**, 1–11, https://doi.org/10.21522/tijar.2014.10.02.art001.
[84]. N. M. Ndongala, 2024, Topological Relation Aware Transformer, *Texila International Journal of Academic Research*, **11**, 160–74, https://doi.org/10.21522/tijar.2014.11.01.art015.
[85]. Cao, R., Chen, L., Chen, Z., Zhao, Y., Zhu, S. and Yu, K., 2021, LGESQL: Line Graph Enhanced Text-to-SQL Model with Mixed Local and Non-

Local Relations.
[86]. Scholak, T., Schucher, N. and Bahdanau, D., 2021, PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models.
[87]. Herzig, J., Nowak, P. K., Müller, T., Piccinno, F. and Eisenschlos, J., 2020, TaPas: Weakly Supervised Table Parsing via Pre-training, https://doi.org/10.18653/v1/2020,acl-main.398.
[88]. Deng, X., Awadallah, A. H., Meek, C., Polozov, O., Sun, H. and Richardson, M., 2020, Structure-Grounded Pretraining for Text-to-SQ, https://doi.org/10.18653/v1/2021,naacl-main.105.
[89]. Wang, W., Bi, B., Yan, M., Wu, C., Xia, J., Bao, Z. et al., 2019, StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding, *8th International Conference on Learning Representations, ICLR 2020*, *International Conference on Learning Representations, ICLR*.
[90]. Zhao, L., Cao, H. and Zhao, Y., 2021, GP: Context-free Grammar Pre-training for Text-to-SQL Parsers.
[91]. Yu, T., Wu, C.-S., Lin, X. V., Wang, B., Tan, Y. C., Yang, X. et al., 2020, GraPPa: Grammar-Augmented Pre-Training for Table Semantic Parsing.
[92]. Shoeybi, M., Patwary, M., Puri, R., Legresley, P., Casper, J. and Catanzaro, B., 2020, Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism [Internet].
[93]. Xu, C., Zhou, W., Ge, T., Wei, F. and Zhou, M., 2020, BERT-of-Theseus: Compressing BERT by Progressive Module Replacing [Internet].
[94]. Polu, S., Han, J. M., Zheng, K., Baksys, M., Babuschkin, I. and Sutskever, I., 2022, Formal Mathematics Statement Curriculum Learning.
[95]. Liu, Y., Dmitriev, P., Huang, Y., Brooks, A. and Dong, L., An Evaluation of Transfer Learning for Classifying Sales Engagement Emails at Large Scale [Internet].
[96]. Fedus, W., Zoph, B. and Shazeer, N., 2022, Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity, *Journal of Machine Learning Research*.
[97]. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., 2019, Language Models are Unsupervised Multitask Learners [Internet].
[98]. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P. et al., 2020, Language Models are Few-Shot Learners [Internet].
[99]. Schick, T. and Schütze, H., 2020, Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference, *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*, *Association for Computational Linguistics (ACL),* 255–69, https://doi.org/10.18653/v1/2021, eacl-main.20.
[100]. Gao, T., Fisch, A. and Chen, D., 2020, Making Pre-trained Language Models Better Few-shot Learners, *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, *Association for Computational Linguistics (ACL)*, 3816–30, https://doi.org/10.18653/v1/2021,acl-long.295.
[101]. Pourreza, M. and Rafiei, D., 2023, DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction.
[102]. [Gur, I., Yavuz, S., Su, Y. and Yan, X., DialSQL: Dialogue Based Structured Query Generation.
[103]. Li, J., Chen, L., Cao, R., Zhu, S., Xu, H., Chen, Z. et al., 2023, Exploring Schema Generalizability of Text-to-SQL, *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 1344–60, https://doi.org/10.18653/v1/2023,findings-acl.87.
[104]. Li, H., Zhang, J., Liu, H., Fan, J., Zhang, X., Zhu, J. et al., 2024, CodeS: Towards Building Open-Source Language Models for Text-to-SQL.
[105]. Molnar, C., Interpretable Machine Learning, A Guide for Making Black Box Models Explainable [Internet].