Recent Advances in Text-to-SQL:
A Survey of What We Have and What We Expect

Naihao Deng
University of Michigan
dnaihao@umich.edu

Yulong Chen
Westlake University
yulongchen1010@gmail.com

Yue Zhang
Westlake University
yue.zhang@wias.org.cn

Abstract

Text-to-SQL has attracted attention from both the natural language processing and database communities because of its ability to convert the semantics in natural language into SQL queries and its practical application in building natural language interfaces to database systems. The major challenges in text-to-SQL lie in encoding the meaning of natural utterances, decoding to SQL queries, and translating the semantics between these two forms. These challenges have been addressed to different extents by the recent advances. However, there is still a lack of comprehensive surveys for this task. To this end, we review recent progress on text-to-SQL for datasets, methods, and evaluation and provide this systematic survey, addressing the aforementioned challenges and discussing potential future directions. We hope that this survey can serve as quick access to existing work and motivate future research. [1]

1 Introduction

The task of text-to-SQL is to convert natural utterances into SQL queries (Zhong et al., 2017; Yu et al., 2018c). Figure 1 shows an example. Given a user utterance *"What are the major cities in the state of Kansas?"*, the system outputs a corresponding SQL that can be used for retrieving the answer from a database. It builds a natural language interface to the database (NLIDB) to help lay users access information in the database (Popescu et al., 2003; Li and Jagadish, 2014), inspiring research in human-computer interaction (Elgohary et al., 2020). Because the SQL query can be regarded as a semantic representation (Guo et al., 2020), text-to-SQL is also a representative task in semantic parsing, helping downstream applications such as question answering (Wang et al., 2020d). Thus, text-to-SQL has attracted researchers in the natural
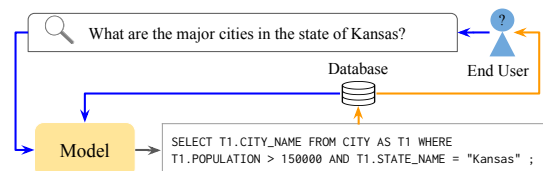

Figure 1: The framework for text-to-SQL systems. Given the database schema and user utterance, the system outputs a corresponding SQL query to query the database system for the result. Appendix B gives more text-to-SQL examples.

language processing (NLP) and the database (DB) community for decades (Codd, 1970; Hemphill et al., 1990; Dahl et al., 1994; Zelle and Mooney, 1996; Popescu et al., 2003; Bertomeu et al., 2006; Wang et al., 2020a; Scholak et al., 2021b).

The challenges in text-to-SQL lie within three aspects: (1) extracting the meaning of natural utterances (encoding); (2) transforming the extracted meaning into another expression which is pragmatically equivalent to the NL meaning (translating) and; (3) producing the corresponding SQL queries (decoding). A wide range of methods has been investigated to address the technical challenges, from representation learning, intermediate structures, decoding, model structures, training objectives, and other perspectives. In addition, much work has been conducted on data resources and evaluation. However, relatively little work has been done in the literature to provide a comprehensive survey of the landscape. The only exceptions are (Katsogiannis-Meimarakis and Koutrika, 2021) and (Kalajdjieski et al., 2020), but they cover a limited scope. To this end, we aim to provide a systematic survey that involves a broader range of text-to-SQL research and addresses the aforementioned challenges.

In this paper, we survey the recent progress on text-to-SQL, from datasets (§ 2), methods (§ 3) to evaluation (§ 4) [2] and highlight potential direc-

---

[1] The Github Link for this survey is: https://github.com/text-to-sql-survey-coling22/text-to-sql-survey-coling22.github.io.

[2] Note that most work discussed in this paper is in English

Datasets	#Size	#DB	#D	#T/DB	Issues addressed	Sources for data
Spider (Yu et al., 2018c)	10,181	200	138	5.1	Domain generalization	College courses, DabaseAnswers, WikiSQL
WikiSQL (Zhong et al., 2017)	80,654	26,521	-	1	Data size	Wikipedia
Squall (Shi et al., 2020b)	11,468	1,679	-	1	Lexicon-level supervision	WikiTableQuestions
KaggleDBQA (Lee et al., 2021)	272	8	8	2.3	Domain generalization	Real web daabases
IMDB (Yaghmazadeh et al., 2017)	131	1	1	16	-	Internet Movie Database
Yelp (Yaghmazadeh et al., 2017)	128	1	1	7	-	Yelp website
Advising (Finegan-Dollak et al., 2018)	3,898	1	1	10	-	University of Michigan course information
MIMICSQL (Wang et al., 2020d)	10,000	1	1	5	-	Healthcare domain
SEDE (Hazoom et al., 2021)	12,023	1	1	29	SQL template diversity	Stack Exchange

Table 1: The statistic for recent text-to-SQL datasets. #Size, #DB, #D, and #T/DB represent the numbers of question-SQL pairs, databases, domains, and the averaged number of tables per domain, respectively. The "-" in the #D column indicates an unknown number of domains, and the "-" in the Issues Addressed indicates no specific issue addressed by the dataset. Datasets above and below the line are cross-domain and single-domain, respectively. The complete statistic is listed in Table 7 in Appendix C.

tions for future work (§ 5). Apendix A shows the topology for the text-to-SQL task.

2 Datasets

As shown in Table 1, existing text-to-SQL datasets can be classified into three categories: single-domain datasets, cross-domain datasets and others.

Single-Domain Datasets Single-domain text-to-SQL datasets typically collect question-SQL pairs for a single database in some real-world tasks, including early ones such as Academic (Li and Jagadish, 2014), Advising (Finegan-Dollak et al., 2018), ATIS (Price, 1990; Dahl et al., 1994), GeoQuery (Zelle and Mooney, 1996), Yelp and IMDB (Yaghmazadeh et al., 2017), Scholar (Iyer et al., 2017) and Restaurants (Tang and Mooney, 2000; Popescu et al., 2003), as well as recent ones such as SEDE (Hazoom et al., 2021), ESQL (Chen et al., 2021a) and MIMICSQL (Wang et al., 2020d).

These single-domain datasets, particularly the early ones, are usually limited in size, containing only a few hundred to a few thousand examples. Because of the limited size and similar SQL patterns in training and testing phases, text-to-SQL models that are trained on these single-domain datasets can achieve decent performance by simply memorizing the SQL patterns and fail to generalize to unseen SQL queries or SQL queries from other

unless otherwise specified.

domains (Finegan-Dollak et al., 2018; Yu et al., 2018c). However, since these datasets are adapted from real-life applications, most of them contain domain knowledge (Gan et al., 2021b) and dataset conventions (Suhr et al., 2020). Thus, they are still valuable to evaluate models' ability to generalize to new domains and explore how to incorporate domain knowledge and dataset convention to model predictions.

Appendix B gives a detailed discussion on domain knowledge and dataset convention, and concrete text-to-SQL examples.

Large Scale Cross-domain Datasets Large cross-domain datasets such as WikiSQL (Zhong et al., 2017) and Spider (Yu et al., 2018c) are proposed to better evaluate deep neural models. WikiSQL uses tables extracted from Wikipedia and lets annotators paraphrase questions generated for the tables. Compared to other datasets, WikiSQL is an order of magnitude larger, containing 80,654 natural utterances in total (Zhong et al., 2017). However, WikiSQL contains only simple SQL queries, and only a single table is queried within each SQL query (Yu et al., 2018c).

Yu et al. (2018c) propose Spider, which contains 200 databases with an average of 5 tables for each database, to test models' performance on complicated unseen SQL queries and their ability to generalize to new domains. Furthermore, researchers expand Spider to study various issues of their inter-

est (Lei et al., 2020; Zeng et al., 2020; Gan et al., 2021b; Taniguchi et al., 2021; Gan et al., 2021a).

Besides, researchers build several large-scale text-to-SQL datasets in different languages such as CSpider (Min et al., 2019a), TableQA (Sun et al., 2020), DuSQL (Wang et al., 2020c) in Chinese, ViText2SQL (Tuan Nguyen et al., 2020) in Vietnamese, and PortugueseSpider (José and Cozman, 2021) in Portuguese. Given that human translation has shown to be more accurate than machine translation (Min et al., 2019a), these datasets are annotated mainly by human experts based on the English Spider dataset. These Spider-based datasets can serve as potential resources for multi-lingual text-to-SQL research.

Other Datasets Several context-dependent text-to-SQL datasets have been proposed, which involve user interactions with the text-to-SQL system in English (Price, 1990; Dahl et al., 1994; Yu et al., 2019a,b) and Chinese (Guo et al., 2021). In addition, researchers collect datasets to study questions in text-to-SQL being answerable or not (Zhang et al., 2020), lexicon-level mapping (Shi et al., 2020b) and cross-domain evaluation for real Web databases (Lee et al., 2021).

Appendix C.1 discusses more details about datasets mentioned in § 2.

3 Methods

Early text-to-SQL systems employ rule-based and template-based methods (Li and Jagadish, 2014; Mahmud et al., 2015), which is suitable for simple user queries and databases. However, with the progress in both DB and NLP communities, recent work focuses on more complex settings (Yu et al., 2018c). In these settings, deep models can be more useful because of their great feature representation ability and generalization ability.

In this survey, we focus on the deep learning methods primarily. We divide these methods employed in text-to-SQL research into Data Augmentation (§ 3.1), Encoding (§ 3.2), Decoding (§ 3.3), Learning Techniques (§ 3.4), and Miscellaneous (§ 3.5).

3.1 Data Augmentation

Data augmentation can help text-to-SQL models handle complex or unseen questions (Zhong et al., 2020b; Wang et al., 2021b), achieve state-of-the-art with less supervised data (Guo et al., 2018),

and attain robustness towards different types of questions (Radhakrishnan et al., 2020) .

Typical data augmentation techniques involve paraphrasing questions and filling pre-defined templates for increasing data diversity. Iyer et al. (2017) use the Paraphrase Database (PPDB) (Ganitkevitch et al., 2013) to generate paraphrases for training questions. Appendix B gives an example of this augmentation method. Iyer et al. (2017) and Yu et al. (2018b) collect question-SQL templates and fill in them with DB schema. Researchers also employ neural models to generate natural utterances for sampled SQL queries to acquire more data. For instance, Li et al. (2020a) fine-tune pre-trained T5 model (Raffel et al., 2019) using SQL query as the input to predict natural utterance on WikiSQL, and then randomly synthesize SQL queries from tables in WikiSQL and use the tuned model to generate the corresponding natural utterance.

The quality of the augmented data is important because low-quality data can hurt the performance of the models (Wu et al., 2021). Various approaches have been exploited to improve the quality of the augmented data. After sampling SQL queries, Zhong et al. (2020b) employ an utterance generator to generate natural utterances and a semantic parser to convert the generated natural utterance to SQL queries. To filter out low-quality augmented data, Zhong et al. (2020b) only keep data that have the same generated SQL queries as the sampled ones. Wu et al. (2021) use a hierarchical SQL-to-question generation process to obtain high-quality data. Observing that there is a strong segment-level mapping between SQL queries and natural utterances, Wu et al. (2021) decompose SQL queries into several clauses, translate each clause into a sub-question, and then combine the sub-questions into a complete question.

To increase the diversity of the augmented data, Guo et al. (2018) incorporate a latent variable in their SQL-to-text model to encourage question diversity. Radhakrishnan et al. (2020) augment the WikiSQL dataset by simplifying and compressing questions to simulate the colloquial query behavior of end-users. Wang et al. (2021b) exploit a probabilistic context-free grammar (PCFG) to explicitly model the composition of SQL queries, encouraging sampling compositional SQL queries.

Methods	Adopted by	Applied datasets
Encode type	TypeSQL (Yu et al., 2018a)	WikiSQL
Graph-based	GNN (Bogin et al., 2019a)	Spider
Self-attention	RAT-SQL (Wang et al., 2020a)	Spider
Adapt PLM	SQLova (Hwang et al., 2019)	WikiSQL
Pre-training	TaBERT (Yin et al., 2020)	Spider

Table 2: Typical methods used for encoding in text-to-SQL. The full table of existing methods and more details are listed in Table 8 in Appendix D.

3.2 Encoding

Various methods have been adopted to address the challenges of representing the meaning of questions, representing the structure for DB schema, and linking the DB content to question. We group them into five categories, as shown in Table 2.

Encode Token Types To better encode keywords such as entities and numbers in questions, Yu et al. (2018a) assign a type to each word in the question, with a word being an entity from the knowledge graph, a column, or a number. Yu et al. (2018c) concatenate word embeddings and the corresponding type embeddings to feed into their model.

Graph-based Methods Since DB schemas contain rich structural information, graph-based methods are used to better encode such structures.

As summarized in § 2, datasets prior to Spider typically involve simple DBs that contain only one table or a single DB in both training and testing. As a result, modeling DB schema receives little attention. Because Spider contains complex and different DB in training and testing, Bogin et al. (2019a) propose to use graphs to represent the structure of the DB schemas. Specifically, Bogin et al. (2019a) use nodes to represent tables and columns, edges to represent relationships between tables and columns, such as tables containing columns, primary key, and foreign key constraints, and then use graph neural networks (GNNs) (Li et al., 2016) to encode the graph structure. In their subsequent work, Bogin et al. (2019b) use a graph convolutional network (GCN) to capture DB structures and a gated GCN to select the relevant DB information for SQL generation. RAT-SQL (Wang et al., 2020a) encodes more relationships for DB schemas such as "both columns are from the same table" in their

graph.

Graphs have also been used to encode questions together with DB schema. Researchers have been using different types of graphs to capture the semantics in NL and facilitate linking between NL and table schema. Cao et al. (2021) adopt line graph (Gross et al., 2018) to capture multi-hop semantics by meta-path (e.g., an exact match for a question token and column, together with the column belonging to a table can form a 2-hop meta-path) and distinguish between local and non-local neighbors so that different tables and columns will be attended differently. SADGA (Cai et al., 2021) adopts the graph structure to provide a unified encoding for both natural utterances and DB schemas to help question-schema linking. Apart from the relations between entities in both questions and DB schema, the structure for DB schemas, S^2SQL (Hui et al., 2022) integrates syntax dependency among question tokens into the graph to improve model performance. To improve the generalization of the graph method for unseen domains, ShawdowGNN (Chen et al., 2021b) ignores names of tables or columns in the database and uses abstract schemas in the graph projection neural network to obtain delexicalized representations of questions and DB schemas.

Finally, graph-based techniques are also exploited in context-dependent text-to-SQL. For instance, IGSQL (Cai and Wan, 2020) uses a graph encoder to utilize historical information of DB schemas in the previous turns.

Self-attention Models using transformer-based encoder (He et al., 2019; Hwang et al., 2019; Xie et al., 2022) incorporate the original self-attention mechanism by default because it is the building block of the transformer structure.

RAT-SQL (Wang et al., 2020a) applies relation-aware self-attention, a modified version of self-attention (Vaswani et al., 2017), to leverage relations of tables and columns. DuoRAT (Scholak et al., 2021a) also adopts such a relation-aware self-attention in their encoder.

Adapt PLM Various methods have been proposed to leverage the knowledge in pre-trained language models (PLMs) and better align PLM with the text-to-SQL task. PLMs such as BERT (Devlin et al., 2019) are used to encode questions and DB schemas. The modus operandi is to input the concatenation of question words and schema words

to the BERT encoder (Hwang et al., 2019; Choi et al., 2021). Other methods adjust the embeddings by PLMs. On WikiSQL, for instance, X-SQL (He et al., 2019) replaces segment embeddings from the pre-trained encoder by column type embeddings. Guo and Gao (2019) encode two additional feature vectors for matching between question tokens and table cells as well as column names and concatenate them with BERT embeddings of questions and DB schemas.

HydraNet (Lyu et al., 2020) uses BERT to encode the question and an individual column, aligning with the tasks BERT is pre-trained on. After obtaining the BERT representations of all columns, Lyu et al. (2020) select top-ranked columns for SQL prediction. Liu et al. (2021b) train an auxiliary concept prediction module to predict which tables and columns correspond to the question. They detect important question tokens by detecting the largest drop in the confidence score caused by erasing that token in the question. Lastly, they train the PLM with a grounding module using the question tokens and the corresponding tables as well as columns. By empirical studies, Liu et al. (2021b) claim that their approach can awaken the latent grounding from PLM via this erase-and-predict technique.

Pre-training There have been various works proposing different pre-training objectives and using different pre-training data to better align the transformer-based encoder with the text-to-SQL task. For instance, TaBERT (Yin et al., 2020) uses tabular data for pre-training with objectives of masked column prediction and cell value recovery to pre-train BERT. Grappa (Yu et al., 2021) synthesizes question-SQL pairs over tables and pre-trains BERT with the objectives of masked language modeling (MLM) and predicting whether a column appears in the SQL query as well as what SQL operations are triggered. GAP (Shi et al., 2020a) pre-trains BART (Lewis et al., 2020) on synthesized text-to-SQL and tabular data with the objectives of MLM, column prediction, column recovery, and SQL generation.

3.3 Decoding

Various methods have been proposed for decoding to achieve a fine-grained and easier process for SQL generation and bridge the gap between natural language and SQL queries. As shown in Table 3, we group these methods into five main categories

Methods	Adopted by	Applied datasets
Tree	SyntaxSQLNet (Yu et al., 2018b)	Spider
Sketch	SQLNet (Xu et al., 2017)	WikiSQL
Bottom-up	SmBop (Rubin and Berant, 2021)	Spider
Attention	Wang et al. (2019)	WikiSQL
Copy	Wang et al. (2018a)	WikiSQL
IR	IRNet (Guo et al., 2019)	Spider
Others	Global-GCN Bogin et al. (2019b)	Spider
	Kelkar et al. (2020)	Spider

Table 3: Typical methods used for decoding in text-to-SQL. The full table and more details are listed in Table 9 in Appendix D. IR: Intermediate Representation.

and other technologies.

Tree-based Seq2Tree (Dong and Lapata, 2016) employs a decoder that generates logical forms in a top-down manner. The components in the sub-tree are generated conditioned on their parents apart from the input question. Note that the syntax of the logical forms is implicitly learned from data for Seq2Tree. Similarly, Seq2AST (Yin and Neubig, 2017) uses an abstract syntax tree (AST) for decoding the target programming language, where the syntax is explicitly integrated with AST. Although both Seq2Tree (Dong and Lapata, 2016) and Seq2AST (Yin and Neubig, 2017) do not study text-to-SQL datasets, their uses of trees inspire tree-based decoding in text-to-SQL. SyntaxSQLNet (Yu et al., 2018b) employs a tree-based decoding method specific to SQL syntax and recursively calls modules to predict different SQL components.

Sketch-based SQLNet (Xu et al., 2017) designs a sketch aligned with the SQL grammar, and SQLNet only needs to fill in the slots in the sketch rather than predict both the output grammar and the content. Besides, the sketch captures the dependency of the predictions. Thus, the prediction of one slot is only conditioned on the slots it depends on, which avoids issues of the same SQL query with varied equivalent serializations. Dong and Lapata (2018) decompose the decoding into two stages, where the first decoder predicts a rough sketch, and the second decoder fills in the low-level details conditioned on the question and the sketch. Such coarse-to-fine decoding has also been adopted in other works such as IRNet (Guo et al., 2019). To address the complex SQL queries with

nested structures, RYANSQL (Choi et al., 2021) recursively yields SELECT statements and uses a sketch-based slot filling for each of the SELECT statements.

Bottom-up Both the tree-based and the sketch-based decoding mechanisms can be viewed as top-down decoding mechanisms. Rubin and Berant (2021) use a bottom-up decoding mechanism. Given K trees of height t, the decoder scores trees with height $t + 1$ constructed by SQL grammar from the current beam, and K trees with the highest scores are kept. Then, a representation of the new K trees is generated and placed in the new beam.

Attention Mechanism To integrate the encoder-side information at decoding, an attention score is computed and multiplied with hidden vectors from the encoder to get the context vector, which is then used to generate an output token (Dong and Lapata, 2016; Zhong et al., 2017).

Variants of the attention mechanism have been used to better propagate the information encoded from questions and DB schemas to the decoder. SQLNet (Xu et al., 2017) designs column attention, where it uses hidden states from columns multiplied by embeddings for the question to calculate attention scores for a column given the question. Guo and Gao (2018) incorporate bi-attention over question and column names for SQL component selection. Wang et al. (2019) adopt a structured attention (Kim et al., 2017) by computing the marginal probabilities to fill in the slots in their generated abstract SQL queries. Duo-RAT (Scholak et al., 2021a) adopts the relation-aware self-attention mechanism in both its encoder and decoder. Other works that use sequence-to-sequence transformer-based models or decoder-only transformer-based models incorporate the self-attention mechanism by default (Scholak et al., 2021b; Xie et al., 2022).

Copy Mechanism Seq2AST (Yin and Neubig, 2017) and Seq2SQL (Zhong et al., 2017) employ the pointer network (Vinyals et al., 2015) to compute the probability of copying words from the input. Wang et al. (2018a) use types (e.g., columns, SQL operators, constant from questions) to explicitly restrict locations in the query to copy from and develop a new training objective to only copy from the first occurrence in the input. In addition, the copy mechanism is also adopted in context-dependent text-to-SQL task (Wang et al., 2020b).

Intermediate Representations Researchers use intermediate representations to bridge the gap between natural language and SQL queries. Inc-SQL (Shi et al., 2018) defines actions for different SQL components and let decoder decode actions instead of SQL queries. IRNet (Guo et al., 2019) introduces SemQL, an intermediate representation for SQL queries that can cover most of the challenging Spider benchmark. Specifically, SemQL removes the JOIN ON, FROM and GROUP BY clauses, merges HAVING and WHERE clause for SQL queries. ValueNet (Brunner and Stockinger, 2021) uses SemQL 2.0, which extends SemQL to include value representation. Based on SemQL, NatSQL (Gan et al., 2021c) removes the set operators [3]. Suhr et al. (2020) implement SemQL as a mapping from SQL to a representation with an under-specified FROM clause, which they call SQL^{UF}. Rubin and Berant (2021) employ a relational algebra augmented with SQL operators as the intermediate representations.

However, the intermediate representations are usually designed for a specific dataset and cannot be easily adapted to others (Suhr et al., 2020). To construct a more generalized intermediate representation, Herzig et al. (2021) propose to omit tokens in the SQL query that do not align to any phrase in the utterance.

Inspired by the success of text-to-SQL task, intermediate representations are also studied for SPARQL, another executable language for database systems (Saparina and Osokin, 2021; Herzig et al., 2021).

Others PICARD (Scholak et al., 2021b) and UniSAr (Dou et al., 2022) set constraints to the decoder to prevent generating invalid tokens. Several methods adopt an execution-guided decoding mechanism to exclude non-executable partial SQL queries from the output candidates (Wang et al., 2018b; Hwang et al., 2019). Global-GNN (Bogin et al., 2019b) employs a separately trained discriminative model to rerank the top-K SQL queries in the decoder's output beam, which is to reason about the complete SQL queries instead of considering each word and DB schemas in isolation. Similarly, Kelkar et al. (2020) train a separate discriminator to better search among candidate SQL

---

[3]The operators that combine the results of two or more SELECT statements, such as INTERSECT

queries. Xu et al. (2017); Yu et al. (2018b); Guo and Gao (2018); Lee (2019) use separate submodules to predict different SQL components, easing the difficulty of generating a complete SQL query. Chen et al. (2020b) employ a gate to select between the output sequence encoded for the question and the output sequence from the previous decoding steps at each step for SQL generation. Inspired by machine translation, Müller and Vlachos (2019) apply byte-pair encoding (BPE) (Sennrich et al., 2016) to compress SQL queries to shorter sequences guided by AST, reducing the difficulties in SQL generation.

3.4 Learning Techniques

Apart from end-to-end supervised learning, different learning techniques have been proposed to help text-to-SQL research. Here we summarize these learning techniques, each addressing a specific issue for the task.

Fully supervised Ni et al. (2020) adopt *active learning* to save human annotation. Yao et al. (2019, 2020); Li et al. (2020b) employ *interactive or imitation learning* to enhance text-to-SQL systems via interactions with end-users. Huang et al. (2018); Wang et al. (2021a); Chen et al. (2021a) adopt *meta-learning* (Finn et al., 2017) for domain generalization. Various *multi-task learning* settings have been proposed to improve text-to-SQL models via enhancing their abilities on some relevant tasks. Chang et al. (2020) set an auxiliary task of mapping between column and condition values. SeaD (Xuan et al., 2021) integrates two denoising objectives to help the model better encode information from the structural data. Hui et al. (2021b) integrate a task of learning the correspondence between questions and DB schemas. Shi et al. (2021) integrate a column classification task to classify which columns appear in the SQL query. McCann et al. (2018) and Xie et al. (2022) train their models with other semantic parsing tasks, which improves models' performance on text-to-SQL task.

Weakly supervised Seq2SQL (Zhong et al., 2017) use *reinforcement learning* to learn `WHERE` clause to allow different orders for components in `WHERE` clause. Liang et al. (2018) leverage memory buffer to reduce the variance of policy gradient estimates when applying reinforcement learning to text-to-SQL. Agarwal et al. (2019) use *meta-learning* and *Bayesian optimization* (Snoek et al.,

2012) to learn an auxiliary reward to discount spurious SQL queries in SQL generation. Min et al. (2019b) model the possible SQL queries as a *discrete latent variable* and adopt a hard-*EM*-style parameter updates, letting their model take advantage of the possible pre-computed solutions.

3.5 Miscellaneous

In DB linking, BRIDGE (Lin et al., 2020) appends a representation for the DB cell values mentioned in the question to corresponding fields in the encoded sequence, which links the DB content to the question. Ma et al. (2020) employ an explicit extractor of slots mentioned in the question and then link them with DB schemas.

Model-wise, Finegan-Dollak et al. (2018) use a template-based model which copies slots from the question. Shaw et al. (2021) use a hybrid model which firstly uses a high precision grammar-based approach (NQG) to generate SQL queries, then uses T5 (Raffel et al., 2019) as a back-up if NQG fails. Yan et al. (2020) formulate submodule slot-filling as machine reading comprehension (MRC) task and apply BERT-based MRC models on it. Besides, DT-Fixup (Xu et al., 2021) designs an optimization approach for a deeper Transformer on small datasets for the text-to-SQL task.

In SQL generation, IncSQL (Shi et al., 2018) allows parsers to explore alternative correct action sequences to generate different SQL queries. Brunner and Stockinger (2021) search values in DB to insert values into SQL query.

For context-dependent text-to-SQL, researchers adopt techniques such as turn-level encoder and copy mechanism (Suhr et al., 2018; Zhang et al., 2019; Wang et al., 2020b), constrained decoding (Wang et al., 2020b), dynamic memory decay mechanism (Hui et al., 2021a), treating questions and SQL queries as two modalities, and using bimodal pre-trained models (Zheng et al., 2022).

4 Evaluation

Metrics Table 4 shows widely used automatic evaluation metrics for the text-to-SQL task. Early works evaluate SQL queries by comparing the database querying results executed from the predicted SQL query and the ground-truth (or gold) SQL query (Zelle and Mooney, 1996; Yaghmazadeh et al., 2017) or use *exact string match* to compare the predicted SQL query with the gold one query (Finegan-Dollak et al., 2018). However,

Metrics	Datasets	Errors
Naiive Execution Accuracy	GeoQuery, IMDB, Yelp, WikiSQL, etc	False positive
Exact String Match	Advising, WikiSQL, etc	False negative
Exact Set Match	Spider	False negative
Test Suite Accuracy (execution accuracy with generated databases)	Spider, GeoQuery, etc	False positive

Table 4: The summary of metrics, datasets that use these metrics, and their potential error cases.

execution accuracy can create false positives for semantically different SQL queries even if they yield the same execution results (Yu et al., 2018c). The exact string match can be too strict as two different strings can still have the same semantics (Zhong et al., 2020a). Aware of these issues, Yu et al. (2018c) adopt *exact set match* (ESM) in Spider, deciding the correctness of SQL queries by comparing the sub-clauses of SQL queries. Zhong et al. (2020a) generate databases that can distinguish the predicted SQL query and gold one. Both methods are used as official metrics on Spider.

Evaluation Setup Early single-domain datasets typically use the standard train/dev/test split (Iyer et al., 2017) by splitting the question-SQL pairs randomly. To evaluate generalization to unseen SQL queries within the current domain, Finegan-Dollak et al. (2018) propose SQL query split, where no SQL query is allowed to appear in more than one set among the train, dev, and test sets. Furthermore, Yu et al. (2018c) propose a database split, where the model does not see the databases in the test set in its training time. Other splitting methods also exist to help different research topics (Shaw et al., 2021; Chang et al., 2020).

5   Discussion and Future Directions

Ever since the LUNAR system (Woods et al., 1972; Woods, 1973), systems for retrieving DB information have witnessed an increasing amount of research interest and an enormous growth, especially in the field of text-to-SQL in the deep learning era. With the ever-increasing model performance on the WikiSQL and Spider leaderboards, one can be optimistic because models are becoming more sophisticated than ever. But there are still several challenges to overcome.

First, these sophisticated models suffer a great performance loss when tested against different text-to-SQL datasets from other domains (Suhr et al., 2020; Lee et al., 2021). It is unclear how to incorporate domain knowledge to the models trained on Spider and deploy these models efficiently on different domains, especially those with similar information stored in DB but slightly different DB schemas. Although large-scale datasets promote the cross-domain settings, question-SQL pairs from Spider are free from domain knowledge, ambiguity, or domain convention. Thus, *cross-domain text-to-SQL* needs to be studied in future research to build a practical cross-domain system that can handle real-world requests.

There are different *use cases in real-world scenarios*, which requires models to be robust to different settings and be smart to handle different user requests. For instance, the model trained with DB schemas can need to handle a corrupted table, or no table is provided in its practical use. Besides, the input from users can vary from the standard question input in Spider or WikiSQL, which poses challenges to models trained on these datasets. More user studies need to be done to study how well the current systems serve the end-users and the input pattern from the end-users. Apart from SQL queries, administrators can want to change DB schemas, where a system that can translate the natural language to such DB commands can be helpful. Also, although there are already works on text-to-SQL beyond English (Min et al., 2019a; Tuan Nguyen et al., 2020; José and Cozman, 2021), but we still lack a comprehensive study on multilingual text-to-SQL, which can be challenging but useful in real-life scenarios. Finally, it is important to build NLIDB for people with disabilities. Song et al. (2022) propose speech-to-SQL that translates voice input to SQL queries, which helps visually impaired end users. More work can be done to address various needs from the perspective of end-users, in particular, the needs from minorities.

Text-to-SQL research can also be integrated into *a larger scope of research*. Application-wise, Xu et al. (2020) develop a question answering system for the database, Chen et al. (2020a) generate task-oriented dialogue by retrieving knowledge from the database using the text-to-SQL model. An example of the possible directions is to employ the text-to-SQL model to query databases for fact-checking. Research-wise, Guo et al. (2020) compare SQL queries to other logical forms in semantic pars-

ing, Xie et al. (2022) include text-to-SQL as one of the tasks to achieve a generalized semantic parsing framework. The inter-relations between various logical forms in semantic parsing can be further studied. A generalized framework or a generalized model can come as the fruit for our semantic parsing community.

In hindsight, the development of text-to-SQL has been pushed by the innovation in the general ML/NLP community, such as LSTM (Hochreiter and Schmidhuber, 1997), self-attention (Vaswani et al., 2017), PLMs (Devlin et al., 2019), etc. Recently, *prompt learning* has achieved decent performance on various tasks, in particular, in the low-resource setting (Liu et al., 2021a). Such characteristics align well with the expectation of having a functional text-to-SQL model with a few training samples. Some recent works already explore applying prompt learning to the text-to-SQL task (Xie et al., 2022). The practical expectation for the text-to-SQL task is to deploy the model in different scenarios, requiring robustness across domains. However, prompt learning struggles with being robust, and the performance can be easily affected by the selected data. This misalignment encourages researchers to study how to employ prompt learning in the real-world text-to-SQL task, which can need further understanding of the cross-domain challenges for text-to-SQL.

Another line of research is to *evaluate these sophisticated text-to-SQL systems*. The typical measure is to evaluate the performance of the system on some existing datasets. As there are operational systems using NL input to perform tasks such as getting answers from database management system or building ontologies or playing some games, the performance of these systems can be measured by the diminution of the (human) time taken to get the searched information (Deng et al., 2021; Zhou et al., 2022). While there are context-dependent text-to-SQL datasets available (Yu et al., 2019a,b), researchers can draw inspirations from other fields of research (Zellers et al., 2021) to design interactive set-ups to evaluate text-to-SQL systems. Appendix E discusses tasks relevant to the task of text-to-SQL.

Acknowledgement

References

Rishabh Agarwal, Chen Liang, Dale Schuurmans, and Mohammad Norouzi. 2019. Learning to generalize from sparse and underspecified rewards. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 130–140. PMLR.

Núria Bertomeu, Hans Uszkoreit, Anette Frank, Hans-Ulrich Krieger, and Brigitte Jörg. 2006. Contextual phenomena and thematic relations in database QA dialogues: results from a Wizard-of-Oz experiment. In *Proceedings of the Interactive Question Answering Workshop at HLT-NAACL 2006*, pages 1–8, New York, NY, USA. Association for Computational Linguistics.

Shikhar Bharadwaj and Shirish Shevade. 2022. Efficient constituency tree based encoding for natural language to bash translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3159–3168.

Ben Bogin, Jonathan Berant, and Matt Gardner. 2019a. 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*, pages 4560–4565, Florence, Italy. Association for Computational Linguistics.

Ben Bogin, Matt Gardner, and Jonathan Berant. 2019b. Global reasoning over database structures for text-to-SQL 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)*, pages 3659–3664, Hong Kong, China. Association for Computational Linguistics.

Sridevi Bonthu, S Rama Sree, and MHM Krishna Prasad. 2021. Text2PyCode: Machine translation of natural language intent to python source code. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 51–60. Springer.

Ursin Brunner and Kurt Stockinger. 2021. Valuenet: A natural language-to-SQL system that learns from database information. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2177–2182. IEEE.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the*

2018 Conference on Empirical Methods in Natural Language Processing, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Ruichu Cai, Jinjie Yuan, Boyan Xu, and Zhifeng Hao. 2021. SADGA: Structure-aware dual graph aggregation network for text-to-SQL. *Advances in Neural Information Processing Systems*, 34.

Yitao Cai and Xiaojun Wan. 2020. IGSQL: Database schema interaction graph based neural model for context-dependent text-to-SQL generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6903–6912, Online. Association for Computational Linguistics.

Ruisheng Cao, Lu Chen, Zhi Chen, Yanbin Zhao, Su Zhu, and Kai Yu. 2021. 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)*, pages 2541–2555, Online. Association for Computational Linguistics.

Shuaichen Chang, Pengfei Liu, Yun Tang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. Zero-shot text-to-SQL learning with auxiliary task. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7488–7495.

Chieh-Yang Chen, Pei-Hsin Wang, Shih-Chieh Chang, Da-Cheng Juan, Wei Wei, and Jia-Yu Pan. 2020a. AirConcierge: Generating task-oriented dialogue via efficient large-scale knowledge retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 884–897, Online. Association for Computational Linguistics.

Sanxing Chen, Aidan San, Xiaodong Liu, and Yangfeng Ji. 2020b. A tale of two linkings: Dynamically gating between schema linking and structural linking for text-to-SQL parsing. In *Proceedings COLING-2020, the 28th International Conference on Computational Linguistics*, pages 2900–2912, Barcelona, Spain (Online). Association for Computational Linguistics.

Yongrui Chen, Xinnan Guo, Chaojie Wang, Jian Qiu, Guilin Qi, Meng Wang, and Huiying Li. 2021a. Leveraging table content for zero-shot text-to-SQL with meta-learning. *ArXiv preprint*, abs/2109.05395.

Zhi Chen, Lu Chen, Yanbin Zhao, Ruisheng Cao, Zihan Xu, Su Zhu, and Kai Yu. 2021b. 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*, pages 5567–5577, Online. Association for Computational Linguistics.

DongHyun Choi, Myeong Cheol Shin, EungGyun Kim, and Dong Ryeol Shin. 2021. RYANSQL: Recursively applying sketch-based slot fillings for complex text-to-SQL in cross-domain databases. *Computational Linguistics*, 47(2):309–332.

E. F. Codd. 1970. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387.

Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the ATIS task: The ATIS-3 corpus. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Naihao Deng, Shuaichen Chang, Peng Shi, Tao Yu, and Rui Zhang. 2021. Prefix-to-SQL: Text-to-SQL generation from incomplete user questions. *arXiv preprint arXiv:2109.13066*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany. Association for Computational Linguistics.

Li Dong and Mirella Lapata. 2018. 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)*, pages 731–742, Melbourne, Australia. Association for Computational Linguistics.

Longxu Dou, Yan Gao, Mingyang Pan, Dingzirui Wang, Jian-Guang Lou, Wanxiang Che, and Dechen Zhan. 2022. UniSAr: A unified structure-aware autoregressive language model for text-to-SQL. *ArXiv preprint*, abs/2203.07781.

Ahmed Elgohary, Saghar Hosseini, and Ahmed Hassan Awadallah. 2020. Speak to your parser: Interactive text-to-SQL with natural language feedback. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2065–2077, Online. Association for Computational Linguistics.

Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. Improving text-to-SQL evaluation methodology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360, Melbourne, Australia. Association for Computational Linguistics.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR.

Yujian Gan, Xinyun Chen, Qiuping Huang, Matthew Purver, John R. Woodward, Jinxia Xie, and Pengsheng Huang. 2021a. Towards robustness of text-to-SQL models against synonym substitution. 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)*, pages 2505–2515, Online. Association for Computational Linguistics.

Yujian Gan, Xinyun Chen, and Matthew Purver. 2021b. Exploring underexplored limitations of cross-domain text-to-SQL generalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8926–8931, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yujian Gan, Xinyun Chen, Jinxia Xie, Matthew Purver, John R. Woodward, John Drake, and Qiaofu Zhang. 2021c. Natural SQL: Making SQL easier to infer from natural language specifications. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2030–2042, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia. Association for Computational Linguistics.

Jonathan L Gross, Jay Yellen, and Mark Anderson. 2018. *Graph theory and its applications*. Chapman and Hall/CRC.

Daya Guo, Yibo Sun, Duyu Tang, Nan Duan, Jian Yin, Hong Chi, James Cao, Peng Chen, and Ming Zhou. 2018. Question generation from SQL queries improves neural semantic parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1597–1607, Brussels, Belgium. Association for Computational Linguistics.

Jiaqi Guo, Qian Liu, Jian-Guang Lou, Zhenwen Li, Xueqing Liu, Tao Xie, and Ting Liu. 2020. Benchmarking meaning representations in neural semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1520–1540, Online. Association for Computational Linguistics.

Jiaqi Guo, Ziliang Si, Yu Wang, Qian Liu, Ming Fan, Jian-Guang Lou, Zijiang Yang, and Ting Liu. 2021. 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)*, pages 2316–2331, Online. Association for Computational Linguistics.

Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. 2019. 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*, pages 4524–4535, Florence, Italy. Association for Computational Linguistics.

Tong Guo and Huilin Gao. 2018. Bidirectional attention for SQL generation. *ArXiv preprint*, abs/1801.00076.

Tong Guo and Huilin Gao. 2019. Content enhanced BERT-based text-to-SQL generation. *ArXiv preprint*, abs/1910.07179.

Moshe Hazoom, Vibhor Malik, and Ben Bogin. 2021. Text-to-SQL in the wild: A naturally-occurring dataset based on stack exchange data. In *Proceedings of the 1st Workshop on Natural Language Processing for Programming (NLP4Prog 2021)*, pages 77–87, Online. Association for Computational Linguistics.

Pengcheng He, Yi Mao, Kaushik Chakrabarti, and Weizhu Chen. 2019. X-SQL: reinforce schema representation with context. *ArXiv preprint*, abs/1908.08113.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Jonathan Herzig, Peter Shaw, Ming-Wei Chang, Kelvin Guu, Panupong Pasupat, and Yuan Zhang. 2021. Unlocking compositional generalization in pre-trained models using intermediate representations. *ArXiv preprint*, abs/2104.07478.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wentau Yih, and Xiaodong He. 2018. 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)*, pages 732–738, New Orleans, Louisiana. Association for Computational Linguistics.

Binyuan Hui, Ruiying Geng, Qiyu Ren, Binhua Li, Yongbin Li, Jian Sun, Fei Huang, Luo Si, Pengfei Zhu, and Xiaodan Zhu. 2021a. Dynamic hybrid relation network for cross-domain context-dependent semantic parsing. *ArXiv preprint*, abs/2101.01686.

Binyuan Hui, Ruiying Geng, Lihan Wang, Bowen Qin, Bowen Li, Jian Sun, and Yongbin Li. 2022. S^{2}SQL: Injecting syntax to question-schema interaction graph encoder for text-to-SQL parsers. *ArXiv preprint*, abs/2203.06958.

Binyuan Hui, Xiang Shi, Ruiying Geng, Binhua Li, Yongbin Li, Jian Sun, and Xiaodan Zhu. 2021b. Improving text-to-SQL with schema dependency learning. *ArXiv preprint*, abs/2103.04399.

Wonseok Hwang, Jinyeong Yim, Seunghyun Park, and Minjoon Seo. 2019. A comprehensive exploration on WikiSQL with table-aware word contextualization. *ArXiv preprint*, abs/1902.01069.

Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. 2017. 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)*, pages 963–973, Vancouver, Canada. Association for Computational Linguistics.

Marcelo Archanjo José and Fabio Gagliardi Cozman. 2021. mRAT-SQL+ GAP: A Portuguese text-to-SQL transformer. In *Brazilian Conference on Intelligent Systems*, pages 511–525. Springer.

Jovan Kalajdjieski, Martina Toshevska, and Frosina Stojanovska. 2020. Recent advances in SQL query generation: A survey. *ArXiv preprint*, abs/2005.07667.

George Katsogiannis-Meimarakis and Georgia Koutrika. 2021. A deep dive into deep learning approaches for text-to-SQL systems. In *Proceedings of the 2021 International Conference on Management of Data*, pages 2846–2851.

Amol Kelkar, Rohan Relan, Vaishali Bhardwaj, Saurabh Vaichal, Chandra Khatri, and Peter Relan. 2020. Bertrand-dr: Improving text-to-SQL using a discriminative re-ranker. *ArXiv preprint*, abs/2002.00557.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. 2017. Structured attention networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Chia-Hsuan Lee, Oleksandr Polozov, and Matthew Richardson. 2021. KaggleDBQA: Realistic evaluation of text-to-SQL parsers. 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)*, pages 2261–2273, Online. Association for Computational Linguistics.

Dongjun Lee. 2019. Clause-wise and recursive decoding for complex and cross-domain text-to-SQL generation. 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)*, pages

6045–6051, Hong Kong, China. Association for Computational Linguistics.

Wenqiang Lei, Weixin Wang, Zhixin Ma, Tian Gan, Wei Lu, Min-Yen Kan, and Tat-Seng Chua. 2020. Re-examining the role of schema linking in text-to-SQL. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6943–6954, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Fei Li and Hosagrahar V Jagadish. 2014. Constructing an interactive natural language interface for relational databases. *Proceedings of the VLDB Endowment*, 8(1):73–84.

Ning Li, Bethany Keller, Mark Butler, and Daniel Cer. 2020a. SeqGenSQL–a robust sequence generation model for structured query language. *ArXiv preprint*, abs/2011.03836.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. 2016. Gated graph sequence neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Yuntao Li, Bei Chen, Qian Liu, Yan Gao, Jian-Guang Lou, Yan Zhang, and Dongmei Zhang. 2020b. "what do you mean by that?" a parser-independent interactive approach for enhancing text-to-SQL. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6913–6922, Online. Association for Computational Linguistics.

Chen Liang, Mohammad Norouzi, Jonathan Berant, Quoc V. Le, and Ni Lao. 2018. Memory augmented policy optimization for program synthesis and semantic parsing. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10015–10027.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2020. Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4870–4888, Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pretrain, prompt, and predict: A systematic survey of

prompting methods in natural language processing. *ArXiv preprint*, abs/2107.13586.

Qian Liu, Dejian Yang, Jiahui Zhang, Jiaqi Guo, Bin Zhou, and Jian-Guang Lou. 2021b. Awakening latent grounding from pretrained language models for semantic parsing. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1174–1189, Online. Association for Computational Linguistics.

Qin Lyu, Kaushik Chakrabarti, Shobhit Hathi, Souvik Kundu, Jianwen Zhang, and Zheng Chen. 2020. Hybrid ranking network for text-to-SQL. *ArXiv preprint*, abs/2008.04759.

Jianqiang Ma, Zeyu Yan, Shuai Pang, Yang Zhang, and Jianping Shen. 2020. Mention extraction and linking for SQL query generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6936–6942, Online. Association for Computational Linguistics.

Tanzim Mahmud, KM Azharul Hasan, Mahtab Ahmed, and Thwoi Hla Ching Chak. 2015. A rule based approach for NLP based query processing. In *2015 2nd International Conference on Electrical Information and Communication Technologies (EICT)*, pages 78–82. IEEE.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *ArXiv preprint*, abs/1806.08730.

Qingkai Min, Yuefeng Shi, and Yue Zhang. 2019a. 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)*, pages 3652–3658, Hong Kong, China. Association for Computational Linguistics.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019b. A discrete hard EM approach for weakly supervised question answering. 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)*, pages 2851–2864, Hong Kong, China. Association for Computational Linguistics.

Samuel Müller and Andreas Vlachos. 2019. Byte-pair encoding for text-to-SQL generation. *ArXiv preprint*, abs/1910.08962.

Ansong Ni, Pengcheng Yin, and Graham Neubig. 2020. Merging weak and active supervision for semantic parsing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8536–8543.

Peter Ochieng. 2020. Parot: Translating natural language to SPARQL. *Expert Systems with Applications: X*, 5:100024.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Ana-Maria Popescu, Oren Etzioni, and Henry Kautz. 2003. Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 149–157.

P. J. Price. 1990. Evaluation of spoken language systems: the ATIS domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Karthik Radhakrishnan, Arvind Srikantan, and Xi Victoria Lin. 2020. ColloQL: Robust cross-domain text-to-SQL over search queries. *ArXiv preprint*, abs/2010.09927.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv preprint*, abs/1910.10683.

Ohad Rubin and Jonathan Berant. 2021. SmBoP: Semi-autoregressive bottom-up semantic parsing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 311–324, Online. Association for Computational Linguistics.

Irina Saparina and Anton Osokin. 2021. SPARQLing database queries from intermediate question decompositions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8984–8998, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Torsten Scholak, Raymond Li, Dzmitry Bahdanau, Harm de Vries, and Chris Pal. 2021a. DuoRAT: Towards simpler text-to-SQL models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1313–1321, Online. Association for Computational Linguistics.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021b. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9895–9901, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2021. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? 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)*, pages 922–938, Online. Association for Computational Linguistics.

Peng Shi, Patrick Ng, Zhiguo Wang, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Cicero Nogueira dos Santos, and Bing Xiang. 2020a. Learning contextual representations for semantic parsing with generation-augmented pre-training. *ArXiv preprint*, abs/2012.10309.

Peng Shi, Tao Yu, Patrick Ng, and Zhiguo Wang. 2021. End-to-end cross-domain text-to-SQL semantic parsing with auxiliary task. *ArXiv preprint*, abs/2106.09588.

Tianze Shi, Kedar Tatwawadi, Kaushik Chakrabarti, Yi Mao, Oleksandr Polozov, and Weizhu Chen. 2018. IncSQL: Training incremental text-to-SQL parsers with non-deterministic oracles. *ArXiv preprint*, abs/1809.05054.

Tianze Shi, Chen Zhao, Jordan Boyd-Graber, Hal Daumé III, and Lillian Lee. 2020b. On the potential of lexico-logical alignments for semantic parsing to SQL queries. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1849–1864, Online. Association for Computational Linguistics.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 2960–2968.

Yuanfeng Song, Raymond Chi-Wing Wong, Xuefang Zhao, and Di Jiang. 2022. Speech-to-SQL: Towards speech-driven SQL query generation from natural language question. *ArXiv preprint*, abs/2201.01209.

Alane Suhr, Ming-Wei Chang, Peter Shaw, and Kenton Lee. 2020. Exploring unexplored generalization challenges for cross-database semantic parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8372–8388, Online. Association for Computational Linguistics.

Alane Suhr, Srinivasan Iyer, and Yoav Artzi. 2018. Learning to map context-dependent sentences to executable formal queries. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2238–2249, New Orleans, Louisiana. Association for Computational Linguistics.

Ningyuan Sun, Xuefeng Yang, and Yunfeng Liu. 2020. Tableqa: a large-scale Chinese text-to-SQL dataset for table-aware SQL generation. *ArXiv preprint*, abs/2006.06434.

Lappoon R. Tang and Raymond J. Mooney. 2000. Automated construction of database interfaces: Intergrating statistical and relational learning for semantic parsing. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 133–141, Hong Kong, China. Association for Computational Linguistics.

Yasufumi Taniguchi, Hiroki Nakayama, Kubo Takahiro, and Jun Suzuki. 2021. An investigation between schema linking and text-to-SQL performance. *ArXiv preprint*, abs/2102.01847.

Anh Tuan Nguyen, Mai Hoang Dao, and Dat Quoc Nguyen. 2020. A pilot study of text-to-SQL semantic parsing for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4079–4085, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2692–2700.

Bailin Wang, Mirella Lapata, and Ivan Titov. 2021a. Meta-learning for domain generalization in semantic parsing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–379, Online. Association for Computational Linguistics.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020a. 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*, pages 7567–7578, Online. Association for Computational Linguistics.

Bailin Wang, Ivan Titov, and Mirella Lapata. 2019. Learning semantic parsers from denotations with latent structured alignments and abstract programs. 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)*, pages 3774–3785, Hong Kong, China. Association for Computational Linguistics.

Bailin Wang, Wenpeng Yin, Xi Victoria Lin, and Caiming Xiong. 2021b. Learning to synthesize data for semantic parsing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2760–2766, Online. Association for Computational Linguistics.

Chenglong Wang, Marc Brockschmidt, and Rishabh Singh. 2018a. Pointing out SQL queries from text.

Chenglong Wang, Kedar Tatwawadi, Marc Brockschmidt, Po-Sen Huang, Yi Mao, Oleksandr Polozov, and Rishabh Singh. 2018b. Robust text-to-SQL generation with execution-guided decoding. *ArXiv preprint*, abs/1807.03100.

Huajie Wang, Mei Li, and Lei Chen. 2020b. PG-GSQL: Pointer-generator network with guide decoding for cross-domain context-dependent text-to-SQL generation. In *Proceedings of COLING-2022, the 28th International Conference on Computational Linguistics*, pages 370–380, Barcelona, Spain (Online). Association for Computational Linguistics.

Lijie Wang, Ao Zhang, Kun Wu, Ke Sun, Zhenghua Li, Hua Wu, Min Zhang, and Haifeng Wang. 2020c. 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)*, pages 6923–6935, Online. Association for Computational Linguistics.

Ping Wang, Tian Shi, and Chandan K. Reddy. 2020d. Text-to-SQL generation for question answering on electronic medical records. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 350–361. ACM / IW3C2.

W. Woods, Ronald Kaplan, and Bonnie Webber. 1972. The lunar sciences natural language information system.

William A Woods. 1973. Progress in natural language understanding: an application to lunar geology. In *Proceedings of the June 4-8, 1973, national computer conference and exposition*, pages 441–450.

Kun Wu, Lijie Wang, Zhenghua Li, Ao Zhang, Xinyan Xiao, Hua Wu, Min Zhang, and Haifeng Wang. 2021. Data augmentation with hierarchical SQL-to-question generation for cross-domain text-to-SQL parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing,*

pages 8974–8983, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I Wang, et al. 2022. UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *ArXiv preprint*, abs/2201.05966.

Peng Xu, Dhruv Kumar, Wei Yang, Wenjie Zi, Keyi Tang, Chenyang Huang, Jackie Chi Kit Cheung, Simon J.D. Prince, and Yanshuai Cao. 2021. Optimizing deeper transformers on small datasets. 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)*, pages 2089–2102, Online. Association for Computational Linguistics.

Silei Xu, Sina Semnani, Giovanni Campagna, and Monica Lam. 2020. AutoQA: From databases to QA semantic parsers with only synthetic training data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 422–434, Online. Association for Computational Linguistics.

Xiaojun Xu, Chang Liu, and Dawn Song. 2017. SQLNet: Generating structured queries from natural language without reinforcement learning. *ArXiv preprint*, abs/1711.04436.

Kuan Xuan, Yongbo Wang, Yongliang Wang, Zujie Wen, and Yang Dong. 2021. Sead: End-to-end text-to-SQL generation with schema-aware denoising. *ArXiv preprint*, abs/2105.07911.

Navid Yaghmazadeh, Yuepeng Wang, Isil Dillig, and Thomas Dillig. 2017. SQLizer: query synthesis from natural language. *Proceedings of the ACM on Programming Languages*, 1(OOPSLA):1–26.

Zeyu Yan, Jianqiang Ma, Yang Zhang, and Jianping Shen. 2020. SQL generation via machine reading comprehension. In *Proceedings of COLING-2022, the 28th International Conference on Computational Linguistics*, pages 350–356, Barcelona, Spain (Online). Association for Computational Linguistics.

Ziyu Yao, Yu Su, Huan Sun, and Wen-tau Yih. 2019. Model-based interactive semantic parsing: A unified framework and a text-to-SQL case study. 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)*, pages 5447–5458, Hong Kong, China. Association for Computational Linguistics.

Ziyu Yao, Yiqi Tang, Wen-tau Yih, Huan Sun, and Yu Su. 2020. An imitation game for learning semantic parsers from user interaction. In *Proceedings of the 2020 Conference on Empirical Methods*

in Natural Language Processing (EMNLP), pages 6883–6902, Online. Association for Computational Linguistics.

Xi Ye, Qiaochu Chen, Xinyu Wang, Isil Dillig, and Greg Durrett. 2020. Sketch-driven regular expression generation from natural language and examples. *Transactions of the Association for Computational Linguistics*, 8:679–694.

Pengcheng Yin and Graham Neubig. 2017. A syntactic neural model for general-purpose code generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450, Vancouver, Canada. Association for Computational Linguistics.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.

Tao Yu, Zifan Li, Zilin Zhang, Rui Zhang, and Dragomir Radev. 2018a. 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)*, pages 588–594, New Orleans, Louisiana. Association for Computational Linguistics.

Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir R. Radev, Richard Socher, and Caiming Xiong. 2021. Grappa: Grammar-augmented pre-training for table semantic parsing. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Tao Yu, Michihiro Yasunaga, Kai Yang, Rui Zhang, Dongxu Wang, Zifan Li, and Dragomir Radev. 2018b. 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*, pages 1653–1663, Brussels, Belgium. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, Youxuan Jiang, Michihiro Yasunaga, Sungrok Shim, Tao Chen, Alexander Fabbri, Zifan Li, Luyao Chen, Yuwen Zhang, Shreya Dixit, Vincent Zhang, Caiming Xiong, Richard Socher, Walter Lasecki, and Dragomir Radev. 2019a. 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)*, pages 1962–1979, Hong Kong, China. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018c. 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*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir Radev. 2019b. SParC: Cross-domain semantic parsing in context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4511–4523, Florence, Italy. Association for Computational Linguistics.

John M Zelle and Raymond J Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pages 1050–1055.

Rowan Zellers, Ari Holtzman, Elizabeth Clark, Lianhui Qin, Ali Farhadi, and Yejin Choi. 2021. TuringAdvice: A generative and dynamic evaluation of language use. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4856–4880, Online. Association for Computational Linguistics.

Jichuan Zeng, Xi Victoria Lin, Steven C.H. Hoi, Richard Socher, Caiming Xiong, Michael Lyu, and Irwin King. 2020. Photon: A robust cross-domain text-to-SQL system. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 204–214, Online. Association for Computational Linguistics.

Rui Zhang, Tao Yu, Heyang Er, Sungrok Shim, Eric Xue, Xi Victoria Lin, Tianze Shi, Caiming Xiong, Richard Socher, and Dragomir Radev. 2019. 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)*, pages 5338–5349, Hong Kong, China. Association for Computational Linguistics.

Yusen Zhang, Xiangyu Dong, Shuaichen Chang, Tao Yu, Peng Shi, and Rui Zhang. 2020. Did you ask a good question? a cross-domain question intention classification benchmark for text-to-SQL. *ArXiv preprint*, abs/2010.12634.

Yanzhao Zheng, Haibin Wang, Baohua Dong, Xingjun Wang, and Changshan Li. 2022. HIE-SQL: History information enhanced network for context-dependent text-to-SQL semantic parsing. *ArXiv preprint*, abs/2203.07376.

Ruiqi Zhong, Tao Yu, and Dan Klein. 2020a. Semantic evaluation for text-to-SQL with distilled test suites. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 396–411, Online. Association for Computational Linguistics.

Victor Zhong, Mike Lewis, Sida I. Wang, and Luke Zettlemoyer. 2020b. Grounded adaptation for zero-shot executable semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6869–6882, Online. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating structured queries from natural language using reinforcement learning. *ArXiv preprint*, abs/1709.00103.

Jiawei Zhou, Jason Eisner, Michael Newman, Emmanouil Antonios Platanios, and Sam Thomson. 2022. Online semantic parsing for latency reduction in task-oriented dialogue. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1576, Dublin, Ireland. Association for Computational Linguistics.

A Topology for Text-to-SQL

Figure 5 shows the topology for the text-to-SQL task.

B Text-to-SQL Examples

B.1 Table and Database

Table 6 shows an example of the table in the database for Restaurants dataset. The domain for this dataset is restaurant information, where questions are typically about food type, restaurant location, etc.

There is a big difference in terms of how many tables a database has. For restaurants, there are 3 tables in the database, while there are 32 tables in ATIS (Suhr et al., 2020).

B.2 Domain Knowledge

Question: Will undergrads be okay to take 581 ?
SQL query:

```
SELECT DISTINCT T1.ADVISORY_REQUIREMENT ,
 T1.ENFORCED_REQUIREMENT , T1.NAME FROM
COURSE AS T1 WHERE T1.DEPARTMENT =
"EECS" AND T1.NUMBER = 581 ;
```

In Advising dataset, Department "EECS" is considered as domain knowledge where "581" in the utterance means a course in "EECS" department with course number "581".

B.3 Dataset Convention

Question: Give me some restaurants in alameda ?
SQL query:

```
SELECT T1.HOUSE_NUMBER ,
T2.NAME FROM LOCATION AS T1 , RESTAURANT
 AS T2 WHERE T1.CITY_NAME = "alameda"
AND T2.ID = T1.RESTAURANT_ID ;
```

In Restaurants dataset, when the user queries "restaurants", by dataset convention, the corresponding SQL query returns the column "HOUSE_NUMBER" and "NAME".

B.4 Text-to-SQL Templates

An example of the template for text-to-SQL pair used by Iyer et al. (2017) is as follows:
Question template: Get all <ENT1>.<NAME> having <ENT2>.<COL1>.<NAME> as <ENT2>.<COL1>.<TYPE>
SQL query template:

```
SELECT <ENT1>.<DEF> FROM JOIN_FROM(
<ENT1>, <ENT2>) WHERE JOIN_WHERE(<ENT1>,
<ENT2>) AND
<ENT2>.<COL1> = <ENT2>.<COL1>.<TYPE> ;
```
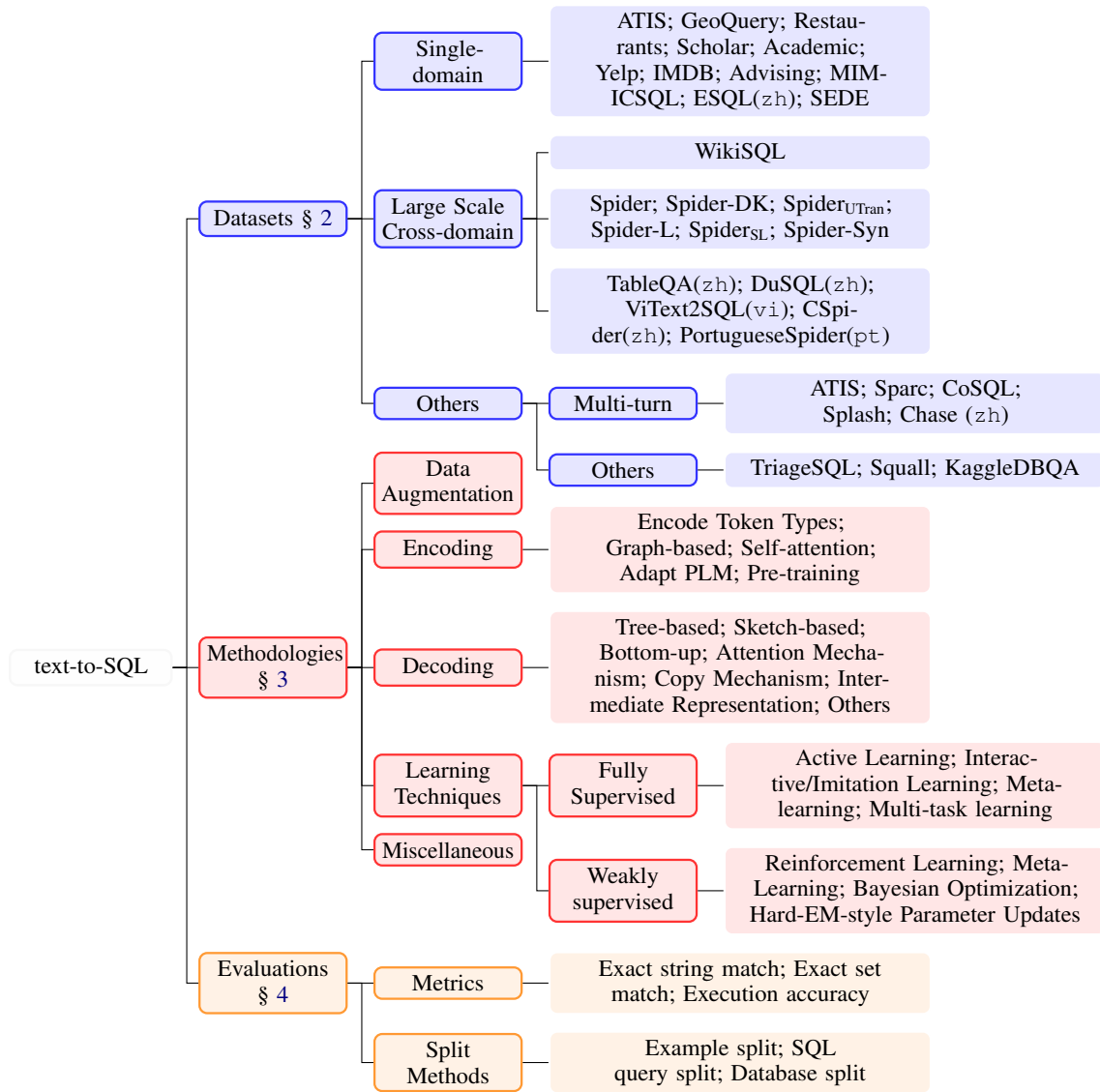
Table 5: Topology for text-to-SQL. Format adapted from Liu et al. (2021a).

CITY_NAME*	COUNTY	REGION
VARCHAR(255)	VARCHAR(255)	VARCHAR(255)
Alameda	Alameda County	Bay Area
Alamo	Contra Costa County	Bay Area
Albany	Alameda County	Bay Area
...	...	...

Table 6: Geography, one of the tables in Restaurants database. * denotes the primary key of this table. We only include 3 rows for demonstration purpose.

Generated question: Get all author having dataset as DATASET_TYPE
Generated SQL query:

```
SELECT author.authorId
FROM author , writes , paper ,
```

paperDataset , dataset WHERE author.
authorId = writes.authorId
AND writes.paperId = paper.paperId
AND paper.paperId = paperDataset.paperId
AND paperDataset.datasetId = dataset.
datasetId AND dataset.datasetName =
DATASET_TYPE ;

, where they populate the slots in the templates with table and column names from the database schema, as well as join the corresponding tables accordingly.

An example of the PPDB (Ganitkevitch et al., 2013) paraphrasing is "thrown into jail" and "imprisoned". The English portion of PPDB contains over 220 million paraphrasing pairs.

B.5 Complexity of Natural Language and SQL Query Pairs

In terms of the complexity for SQL queries, Finegan-Dollak et al. (2018) find that models perform better on shorter SQL queries than longer SQL queries, which indicates that shorter SQL queries are easier in general. Yu et al. (2018c) define the SQL hardness as the number of SQL components. The SQL query is harder when it contains more SQL keywords such as GROUP BY and nested subqueries. Yu et al. (2018c) gives some examples of SQL queries with different difficulty levels:

Easy:
```
SELECT COUNT(*)
FROM cars_data
WHERE cylinders > 4 ;
```

Medium:
```
SELECT T2.name, COUNT(*)
FROM concert AS T1 JOIN stadium AS T2 ON
 T1.stadium_id = T2.stadium_id GROUP
BY T1.stadium_id ;
```

Hard:
```
SELECT T1.country_name
FROM countries AS T1 JOIN continents AS
T2 ON T1.continent = T2.cont_id JOIN
car_makers AS T3 ON T1.country_id = T3.
country
WHERE T2.continent = 'Europe'
GROUP BY T1.country_name
HAVING COUNT(*) >= 3 ;
```

Extra Hard:
```
SELECT AVG(life_expectancy) FROM country
WHERE name NOT IN
  (SELECT T1.name
   FROM country AS T1 JOIN
   country_language AS T2
   ON T1.code = T2.country_code
   WHERE T2.language = "English"
    AND T2.is_official = "T") ;
```

In terms of the complexity of natural utterance, there is no qualitative measure of how hard the utterance is. Intuitively, models' performance can decrease when faced with longer questions from users. However, the information conveyed in longer sentences can be more complete, while there can be ambiguity in shorter sentences. Besides, there can be domain-specific phrases that confuse the model in both short and long utterances (Suhr et al., 2020). Thus, researchers need to consider various perspectives to determine the complexity of natural utterance.

C Text-to-SQL Datasets

Table 7 lists statistics for text-to-SQL datasets.

C.1 More Discussion on Text-to-SQL Datasets

CSpider (Min et al., 2019a), Vi-Text2SQL (Tuan Nguyen et al., 2020) and José and Cozman (2021) translate all the English questions in Spider into Chinese, Vietnamese and Portuguese, respectively. TableQA (Sun et al., 2020) follows the data collection method from WikiSQL, while DuSQL (Wang et al., 2020c) follows Spider. Both TableQA and DuSQL collect Chinese utterance and SQL query pairs across different domains. Chen et al. (2021a) propose a Chinese domain-specific dataset, ESQL.

For multi-turn context-dependent text-to-SQL benchmarks, ATIS (Price, 1990; Dahl et al., 1994) includes user interactions with a SQL flight database in multiple turns. Sparc (Yu et al., 2019b) takes a further step to collect multi-turn interactions across 200 databases and 138 domains. However, both ATIS and Sparc assume all user questions can be mapped into SQL queries and do not include system responses. Later, inspired by task-oriented dialogue system (Budzianowski et al., 2018), Yu et al. (2019a) propose CoSQL. In CoSQL, the dialogue state is tracked by SQL. CoSQL includes three tasks of SQL-grounded dialogue state tracking to generate SQL queries from user's utterance, system response generation from query results, and user dialogue act prediction to detect and resolve ambiguous and unanswerable questions.

Besides, TriageSQL (Zhang et al., 2020) collects unanswerable questions other than natural utterance and SQL query pairs from Spider and WikiSQL, bringing up the challenge of distinguishing answerable questions from unanswerable ones in text-to-SQL systems.

D Encoding and Decoding Method

Table 8 and Table 9 show the encoding and decoding methods that have been discussed in § 3.2 and § 3.3, respectively.

E Other Related Tasks

Other tasks that are related to text-to-SQL include text-to-python (Bonthu et al., 2021), text-to-shell script/bash script (Bharadwaj and Shevade, 2022), text-to-regex (Ye et al., 2020), text-to-SPARQL (Ochieng, 2020), etc. They all take natural language queries as input and output different logical forms. Among these tasks, text-to-SPARQL is closest to text-to-SQL as both SPARQL and SQL can execute on database systems. Therefore, some

Datasets	#Size	#DB	#D	#T/DB	Issues addressed	Sources for data
Spider (Yu et al., 2018c)	10,181	200	138	5.1	Domain generalization	College courses, DabaseAnswers, WikiSQL
Spider-DK (Gan et al., 2021b)	535	10	-	4.8	Domain knowledge	Spider dev set
Spider$_{Utran}$ (Zeng et al., 2020)	15,023	200	138	5.1	Untranslatable questions	Spider + 5,330 untranslatable questions
Spider-L (Lei et al., 2020)	8,034	160	-	5.1	Schema linking	Spider train/dev
Spider$_{SL}$ (Taniguchi et al., 2021)	1,034	10	-	4.8	Schema linking	Spider dev set
Spider-Syn (Gan et al., 2021a)	8,034	160	-	5.1	Robustness	Spider train/dev
WikiSQL (Zhong et al., 2017)	80,654	26,521	-	1	Data size	Wikipedia
Squall (Shi et al., 2020b)	11,468	1,679	-	1	Lexicon-level supervision	WikiTableQuestions (Pasupat and Liang, 2015)
KaggleDBQA (Lee et al., 2021)	272	8	8	2.3	Domain generalization	Real web daabases
ATIS (Price, 1990; Dahl et al., 1994)	5,280	1	1	32	-	Flight-booking
GeoQuery (Zelle and Mooney, 1996)	877	1	1	6	-	US geography
Scholar (Iyer et al., 2017)	817	1	1	7	-	Academic publications
Academic (Li and Jagadish, 2014)	196	1	1	15	-	Microsoft Academic Search (MAS) database
IMDB (Yaghmazadeh et al., 2017)	131	1	1	16	-	Internet Movie Database
Yelp (Yaghmazadeh et al., 2017)	128	1	1	7	-	Yelp website
Advising (Finegan-Dollak et al., 2018)	3,898	1	1	10	-	University of Michigan course information
Restaurants (Tang and Mooney, 2000) (Popescu et al., 2003)	378	1	1	3	-	Restaurants
MIMICSQL (Wang et al., 2020d)	10,000	1	1	5	-	Healthcare domain
SEDE (Hazoom et al., 2021)	12,023	1	1	29	SQL template diversity	Stack Exchange

Table 7: Summarization for text-to-SQL datasets. #Size, #DB, #D, and #T/DB represent the number of question-SQL pairs, databases, domains, and tables per domain, respectively. We put "-" in the #D column because we do not know how many domains are in the Spider dev set and "-" in the Issues Addressed column because there is no specific issue addressed for the dataset. Datasets above and below the line are cross-domain and single-domain, respectively.

end-to-end models that take user queries as the input and output a sequence of logical forms can be applied to both tasks (Raffel et al., 2019). In contrast, methods (Xu et al., 2017) designed to take care of SQL natures cannot be directly applied to SPARQL, which requires carefully modification instead.

Methods	Adopted by	Applied datasets	Addressed challenges
Encode token type	TypeSQL (Yu et al., 2018a)	WikiSQL	Representing question meaning
Graph-based	GNN (Bogin et al., 2019a) Global-GCN (Bogin et al., 2019b) IGSQL (Cai and Wan, 2020) RAT-SQL (Wang et al., 2020a) LEGSQL (Cao et al., 2021) SADGA (Cai et al., 2021) ShawdowGNN (Chen et al., 2021b) S^{2}SQL (Hui et al., 2022)	Spider Spider Sparc, CoSQL Spider Spider Spider Spider Spider, Spider-Syn	(1) Representing question and DB schemas in a structured way (2) Schema linking
Self-attention	X-SQL (He et al., 2019) SQLova (Hwang et al., 2019) RAT-SQL (Wang et al., 2020a) DuoRAT (Scholak et al., 2021a) UnifiedSKG (Xie et al., 2022)	WikiSQL WikiSQL Spider Spider WikiSQL, Spider	
Adapt PLM	X-SQL (He et al., 2019) SQLova (Hwang et al., 2019) Guo and Gao (2019) HydraNet (Lyu et al., 2020) Liu et al. (2021b), etc	WikiSQL WikiSQL WikiSQL WikiSQL Spider-L, SQUALL	Leveraging external data to represent question and DB schemas
Pre-training	TaBERT (Yin et al., 2020) GraPPA (Yu et al., 2021) GAP (Shi et al., 2020a)	Spider Spider Spider	

Table 8: Methods used for encoding in text-to-SQL.

Methods		Adopted by	Applied datasets	Addressed challenges
Tree-based		Seq2Tree (Dong and Lapata, 2016)	-	Hierarchical decoding
		Seq2AST (Yin and Neubig, 2017)	-	
		SyntaxSQLNet (Yu et al., 2018b)	Spider	
Sketch-based		SQLNet (Xu et al., 2017)	WikiSQL	
		Dong and Lapata (2018)	WikiSQL	
		IRNet (Guo et al., 2019)	Spider	
		RYANSQL (Choi et al., 2021)	Spider	
Bottom-up		SmBop (Rubin and Berant, 2021)	Spider	
Attention Mechanism	Attention	Seq2Tree (Dong and Lapata, 2016)	-	Synthesizing information for decoding
	Bi-attention	Seq2SQL (Zhong et al., 2017)	WikiSQL	
	Structured attention	Guo and Gao (2018)	WikiSQL	
	Relation-aware	Wang et al. (2019)	WikiSQL	
	Self-attention	DuoRAT (Scholak et al., 2021a)	Spider	
Copy Mechanism		Seq2AST (Yin and Neubig, 2017)	-	
		Seq2SQL (Zhong et al., 2017)	WikiSQL	
		Wang et al. (2018a)	WikiSQL	
		SeqGenSQL (Li et al., 2020a)	WikiSQL	
Intermediate Representation		IncSQL (Shi et al., 2018)	WikiSQL	Bridging the gap between natural language and SQL query
		IRNet (Guo et al., 2019)	Spider	
		Suhr et al. (2020)	Spider and others♠	
		Herzig et al. (2021)	GeoQuery, ATIS, Scholar	
		Gan et al. (2021c)	Spider	
		Brunner and Stockinger (2021)	Spider	
Others	Constrained decoding	UniSAr (Dou et al., 2022)	WikiSQL, Spide and others♡	Fine-grained decoding
		PICARD (Scholak et al., 2021b)	Spider, CoSQL	
	Execution-guided	SQLova (Hwang et al., 2019)	WikiSQL	
		Wang et al. (2018b)	WikiSQL	
	Discriminative re-ranking	Global-GCN (Bogin et al., 2019b)	Spider	SQL Ranking
		Kelkar et al. (2020)	Spider	
	Separate submodule	SQLNet (Xu et al., 2017)	WikiSQL	Easier decoding
		Guo and Gao (2018)	WikiSQL	
		Lee (2019)	Spider	
	BPE	Müller and Vlachos (2019)	Advising, ATIS, GeoQuery	
	Link gating	Chen et al. (2020b)	Spider	Synthesizing information for decoding

Table 9: Methods used for decoding in text-to-SQL. ♠: Academic, Advising, ATIS, GeoQuery, Yelp, IMDB, Scholar, Restaurants; ♡: TableQA DuSQL, CoSQL, Sparc, Chase.