

Report DLP

Bao, Viet

1 Text2SQL

1.1 problem

This task is to produce sql queries based on inputs consists of natural language question and database. The task can be formalized as:

Given a natural language Q and the schema $S = \langle \mathcal{T}, \mathcal{C} \rangle$ for a relational database, the parser needs to generate the corresponding SQL query Y . The schema consists of tables $\mathcal{T} = \{t_1, \dots, t_N\}$ and fields $\mathcal{C} = \{c_{11}, \dots, c_{1|T_1|}, \dots, c_{n1}, \dots, c_{N|T_N|}\}$. Each table t_i and each field c_{ij} has a textual name. Some fields are primary keys, used for uniquely indexing each data record, and some are foreign keys, used to reference a primary key in a different table. In addition, each field has a data type, $\tau \in \{number, text, time, boolean, etc\}$.

In the paper Bridging Textual and Tabular Data for Cross-Domain Text-to-SQL Semantic Parsing, there are many databases, and users can input arbitrary question, the job of the model is to generate suitable query based on the database it's given.

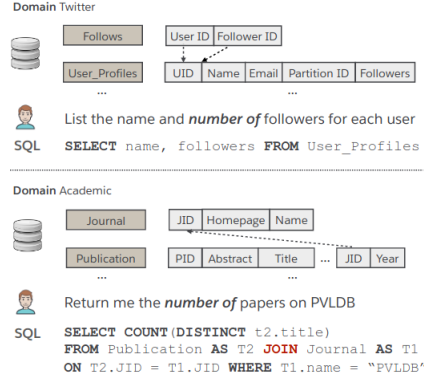


Figure 1: Two questions from the Spider dataset with similar intent resulted in completely different SQL logical forms on two DBs. In cross-DB text-to-SQL semantic parsing, the interpretation of a natural language question is strictly grounded in the underlying relational DB schema.

2 Literature review

Text2SQL recently the field has witnessed a re-surge of interest for text-to-SQL semantic parsing, by virtue of newly released large-scale datasets and matured neural network modeling tools. The task uses the model consists of 2 parts: encoder and decoder. Some works like (Guo et al., 2019; Wang et al., 2019; Choi et al., 2020; Furrer et al., 2020). Bogin et al. (2019a,b) encode schemas as graphs and use graph structures to guide decoding. Guo et al. (2019) proposes schema-linking and SemQL, an intermediate SQL representation customized for

questions in the Spider dataset which was synthesized via a tree-based decoder.. Wang et al. (2019) proposes RAT-SQL, a unified graph encoding mechanism which effectively covers relations in the schema graph and its linking with the question. The overall architecture of RAT-SQL is deep, consisting of 8 relational self-attention layers (Shaw et al., 2018) on top of BERT-large.

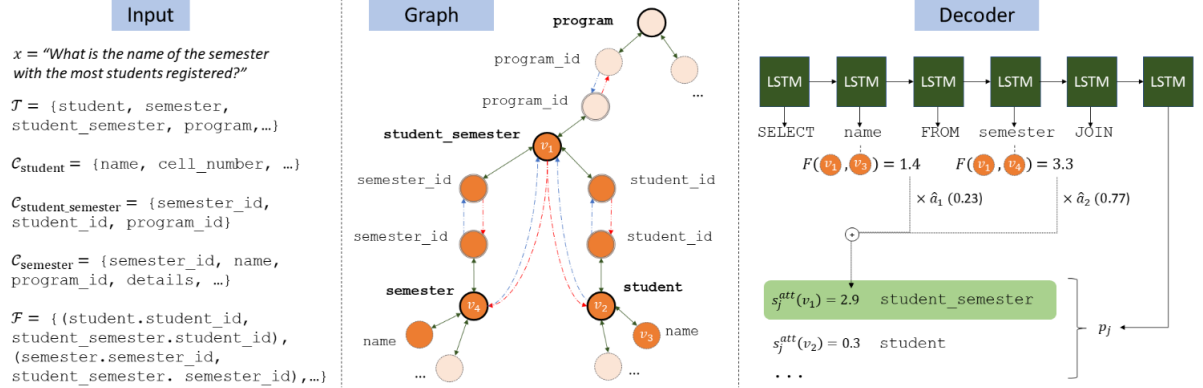


Figure 3: Left: DB schema and question. Middle: A graph representation of the schema. Bold nodes are tables, other nodes are columns. Dashed red (blue) edges are foreign (primary) keys edges, green edges are table-column edges. Right: Use of the schema by the decoder. For clarity, the decoder outputs tokens rather than grammar rules.

Text2SQL can be formulized as seq2seq problem by concatenating natural question and sequential database, then decoder generates sql form. Shaw et al. (2020) shows that the T5 model (Raffel et al., 2020) with 3 billion parameters achieves the state-of-the-art performance on Spider.

Text2SQL can be enchaned by using DB content. Shaw et al. (2019) shows that value information is critical to the cross-DB semantic parsing tasks.

BRIDGE is a general framework for jointly representing question, DB schema and the relevant DB cells. BRIDGE serialized the relational DB schema and uses BERT to model cross- table dependencies. It uses anchor texts which provide more focused signals that link the text and the DB schema.

3 Methods

BRIDGE present each table with its table name followed by its fields. Each table name is preceded by the special token $[T]$ and each field name is preceded by $[C]$. The representations of multiple tables are concatenated to form a serialization of the schema, which is surrounded by two $[SEP]$ tokens and concatenated to the question. Finally, following the input format of BERT, the question is preceded by $[CLS]$ to form the hybrid question-schema serialization.

$$X = [CLS], Q, [SEP], [T], t_1, [C], c_{11}, \dots, c_{1|T_1|}, [T], t_2, [C], c_{21}, \dots, c_{N|T_N|}, [SEP]$$

X is encoded by a Bi-LSTM layer, the question segments output is then passed to another Bi-LSTM. Each table/field is represented using the slice of output X

Lemma 1 Let Y_{exec} be a SQL query with clauses arranged in execution order, then any table field in Y_{exec} must appear after the table.

As a result, we adopt a binary attention mask ξ

$$\tilde{\alpha}_t^{(H)} = \alpha_t^{(H)} \cdot \xi$$

which initially has entries corresponding to all fields set to 0. Once a table t_i is decoded, all entries in ξ corresponding to that table to 1, allows the decoder to only search in the space specified by the condition in Lemma 1 with little overhead in decoding speed.

4 Experiment result

Train/dev/test databases from spider dataset not overlap, test set is hidden from public. In WikiSQL, 49.6% of its dev tables and 45.1% of its test tables are not found in the train set. Both databases have the ability of models to generalize to unseen schema in train set.

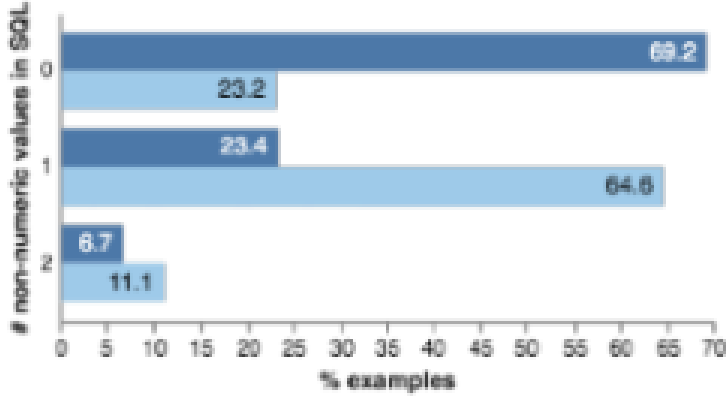
Exact Match (EM): Checks if the predicted SQL exactly matches the ground truth SQL.

Exact Set Match (E-SM): Structural correctness of the predicted SQL. Check orderless set match.

Execution Accuracy (EA): Execution results have the same result.

Anchor Text selection: Fuzzy matching k from DB field picklist and exclude number due to not effectively discriminate between different fields.

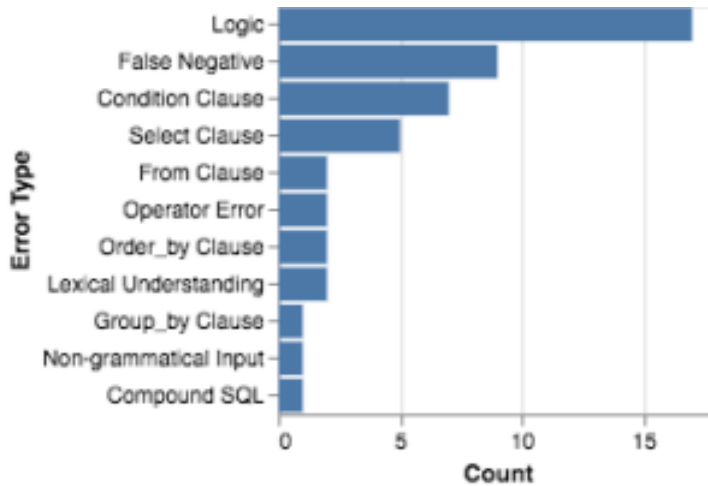
Training use cross-entropy Loss and Adam-SGD optimizer with mini-batch size of 32 and for model, it uses BERT-large + 1-layer LSTMs + 8-head attention





Model	Dev	Test
Global-GNN (Bogin et al., 2019b)	52.7	47.4
EditSQL + BERT (Zhang et al., 2019)	57.6	53.4
GNN + Bertrand-DR (Kelkar et al., 2020)	57.9	54.6
IRNet + BERT (Guo et al., 2019)	61.9	54.7
RAT-SQL v2 (Wang et al., 2019)	62.7	57.2
RYANSQL + BERTL (Choi et al., 2020)	66.6	58.2
SmBoP + BART (Rubin and Berant, 2020)	66.0	60.5
RYANSQL v2 + BERTL	70.6	60.6
RAT-SQL v3 + BERTL (Wang et al., 2019)	69.7	65.6
BRIDGE v1 (Lin et al., 2020)	65.5	59.2
BRIDGE L (ours)	70.0	65.0
BRIDGE L (ours, ensemble)	71.1	67.5


5 Error analysis


50 random Spider dev set failed samples. There are 4 common errors: Logical, Lexical Understanding, Commonsense, Robustness.





 *Show the names of all of the high schooler Kyle's friends.* network_1


 `SELECT Highschooler.name FROM Friend JOIN Highschooler ON Friend.friend_id = Highschooler.ID WHERE Highschooler.name = "Kyle"`


 `SELECT T3.name FROM Friend AS T1 JOIN Highschooler AS T2 ON T1.student_id = T2.id JOIN Highschooler AS T3 ON T1.friend_id = T3.id WHERE T2.name = "Kyle"`


 *What are the full names of all left handed players, in order of birth date?* WTA_1


 `SELECT first_name, last_name FROM players ORDER BY birth_date`


 `SELECT first_name, last_name FROM players WHERE hand = 'L' ORDER BY birth_date`


 *Which address holds the most number of students currently? List the address id and all lines.* student transcripts tracking

 `SELECT Addresses.line_1, Students.current_address_id FROM Addresses JOIN Students ON Addresses.address_id = Students.current_address_id GROUP BY Students.current_address_id ORDER BY COUNT(*) DESC LIMIT 1`

 `SELECT Addresses.address_id , Addresses.line_1 , Addresses.line_2 FROM Addresses JOIN Students ON Addresses.address_id = Students.current_address_id GROUP BY Addresses.address_id ORDER BY count(*) DESC LIMIT 1`

 *What is the model of the car with the smallest amount of horsepower?* car_1

 `SELECT cars_data.Horsepower FROM cars_data ORDER BY cars_data.Horsepower LIMIT 1`

 `SELECT T1.Model FROM CAR_NAMES AS T1 JOIN CARS_DATA AS T2 ON T1.MakeId = T2.Id ORDER BY T2.horsepower ASC LIMIT 1`

6 References

1. <https://github.com/salesforce/TabularSemanticParsing>
2. arXiv:1706.03762
3. 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.
4. Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020. RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7567–7578, Online. Association for Computational Linguistics.
5. <https://www.microsoft.com/en-us/research/uploads/prod/2018/07/Execution-Guided-Neural-Program-Decoding.pdf>