Report DLP

Bao, Viet

1 Text2SQL

1.1 problem

This task is to produce sql queries based on inputs consits 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 $\mathcal{T} = \{t_1, .., t_N\}$ and fields $\mathcal{C} = \{c_{11}, ..., c_{1|T_1|}, ..., c_{n1}, ..., c_{N|T_N|}\}$.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 abitrary question, the job of the model is to generate stuible query based on the database it's given.

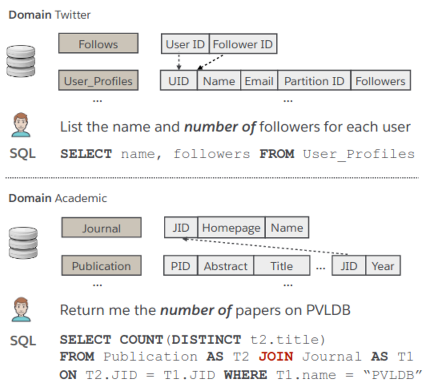

Figure 1: Two questions from the Spider dataset with similar intent resulted in completely different SQL logical forms on two DBs. In cross-DB text-to-SQL semantic parsing, the interpretation of a natural language question is strictly grounded in the underlying relational DB schema.

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 strcutures 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 decode.. Wang et al. (2019) proposes RAT-SQL, a unified graph encoding mechanism which effec-tively 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.

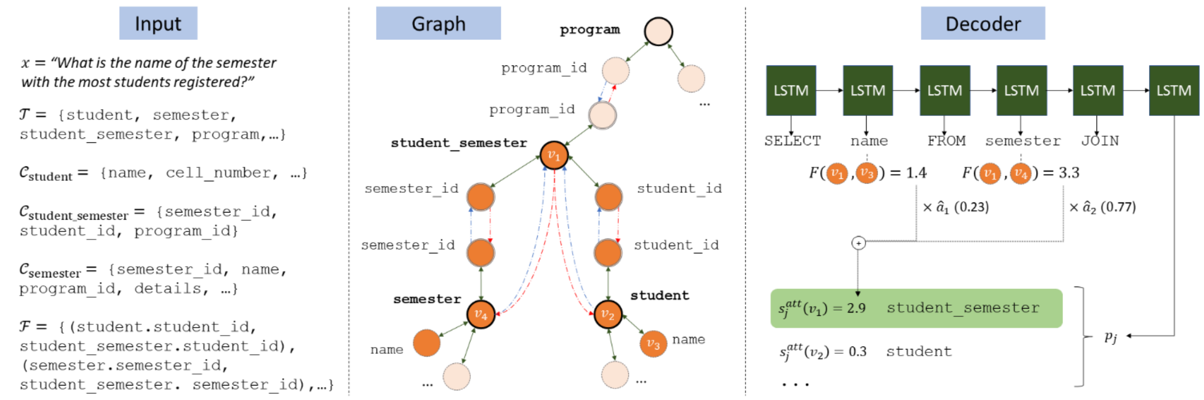

Figure 3: Left: DB schema and question. Middle: A graph representation of the schema. Bold nodes are tables, other nodes are columns. Dashed red (blue) edges are foreign (primary) keys edges, green edges are table-column edges. Right: Use of the schema by the decoder. For clarity, the decoder outputs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

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

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

corresponding to its special token $[T]/[C]$.
Primary keys, foreign keys and data types are also trained by using dense lookup features. These feature then are fused with output X to form final encoding of schema.
$\mathbf{h}_S^{t_i} = g([\mathbf{h}_X^p; \mathbf{0}; \mathbf{0}; \mathbf{0}])$
$\mathbf{h}_S^{c_{ij}} = g([\mathbf{h}_X^q; f_{pri}^u, f_{for}^v, f_{type}^w]) = ReLU(\mathbf{W}_g[\mathbf{h}_X^q; f_{pri}^u, f_{for}^v, f_{type}^w] + \mathbf{b}_g)$
$\mathbf{h}_S = [\mathbf{h}^{t_1}, ..., \mathbf{h}^{t_{|\tau|}}, \mathbf{h}^{c_{11}}, ..., \mathbf{h}^{c_{N|T_N|}}]$

Use of anchor text to link value mentions in the question with the cor- responding DB fields by performing fuzzy string match between Q and the picklist of each field in the DB. The matched field values (anchor texts) are inserted into the question-schema representa- tion X, succeeding the corresponding field names and separated by the special token $[V]$. If multiple values were matched for one field, we concatenate all of them in matching order. If a ques- tion mention is matched with values in multiple fields. We add all matches and let the model learn to resolve ambiguity.
The decoder is initiated with the final state of the question encoder. At each step, the decoder performs one of the following actions: generating a token from the vocabulary V, copying a token from the question Q or copying a schema component from S.

$$e_{tj}^{(h)} = \frac{s_t W_U^{(h)} (h_j W_V^{(h)})^\top}{\sqrt{n/H}}; \alpha_{tj}^{(h)} = \underset{j}{softmax}\{e_{tj}^{(h)}\}$$
$$z_t^{(h)} = \sum_{j=1}^{|Q|+|S|} \alpha_{tj}^{(h)} (h_j W_V^{(h)}); z_t = [z_t^{(1)}, ..., z_t^{(H)}]$$
$$p_{gen}^t = sigmoid(s_t W_{gen}^s + z_t W_{gen}^z + b_{gen})$$
$$p_{out}^t = p_{gen}^t P_{\mathcal{V}}(y_t) + (1 - p_{gen}^t) \sum_{j:\tilde{X}_j = y_t} \alpha_{tj}^{(H)})$$

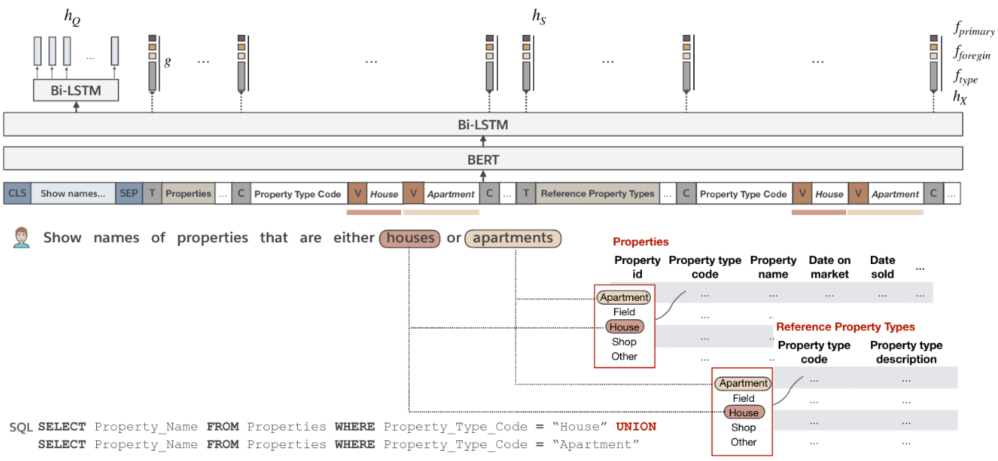

Figure 2: The BRIDGE encoder. The two phrases "houses" and "apartments" in the input question both matched to two DB fields. The matched values are appended to the corresponding field names in the hybrid sequence.

The fact that the DB fields ap- peared in each SQL clause must only come from the tables in the FROM clause.

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
$\overset{\sim}{\alpha}_t^{(H)} = \alpha_t^{(H)}.\xi$
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 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 48704888, 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 75677578, Online. Association for Computational Linguistics.
5. https://www.microsoft.com/en-us/research/uploads/prod/2018/07/Execution-Guided-Neural-Program-Decoding.pdf