

# Text-to-SQL Semantic Parsing

Group 4  
Nguyen Tan Viet  
Ngo Quoc Bao

March 16, 2022

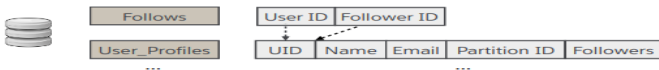
## Definition of problem

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  $\mathcal{C} = \{c_{11}, \dots, c_{1|\mathcal{T}_1|}, \dots, c_{n1}, \dots, c_{N|\mathcal{T}_N|}\}$ . 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,  $\tau \in \{\text{number}, \text{text}, \text{time}, \text{boolean}, \text{etc}\}$

# Problem

## Definition of problem: SQL Cross domain database

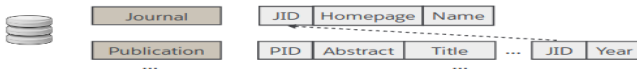
### Domain Twitter



List the name and *number of* followers for each user

SQL `SELECT name, followers FROM User_Profiles`

### Domain Academic



Return me the *number of* papers on PVLDB

SQL `SELECT COUNT(DISTINCT t2.title)  
FROM Publication AS T2 JOIN Journal AS T1  
ON T2.JID = T1.JID WHERE T1.name = "PVLDB"`

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.

## Represent schema as Graph

### Input

$x = \text{"What is the name of the semester with the most students registered?"}$

$\mathcal{T} = \{\text{student, semester, student\_semester, program,...}\}$

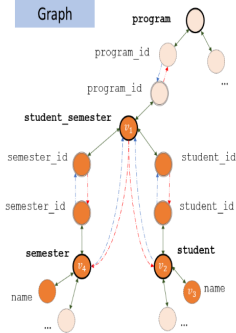
$\mathcal{C}_{\text{student}} = \{\text{name, cell\_number, ...}\}$

$\mathcal{C}_{\text{student\_semester}} = \{\text{semester\_id, student\_id, program\_id}\}$

$\mathcal{C}_{\text{semester}} = \{\text{semester\_id, name, program\_id, details, ...}\}$

$\mathcal{F} = \{(\text{student.student\_id, student\_semester.student\_id}), (\text{semester.semester\_id, student\_semester.semester\_id}), \dots\}$

### Graph



### Decoder

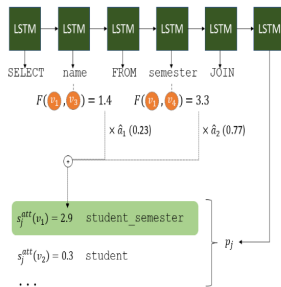


Figure 3: Left: DB schema and question. Middle: A graph representation of the schema. Bold nodes are tables, other nodes are columns. Dashed red (blue) edges are foreign (primary) keys edges, green edges are table-column edges. Right: Use of the schema by the decoder. For clarity, the decoder outputs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

## Represent schema as Graph

1. Use decoder uses graph structure to guide decoding
2. E.g: LSTM generates token by token when decoder generates token schema  $\nu$ , the subsequent generated tokens can be considered as  $\tau(\nu)$ .
3. For example to generate SQL query `SELECT FROM student WHERE student.lastname == "Nguyen"`. Decoder first generates `SELECT FROM` then the next token must be table name according to grammar rules.

## Represent problem as Seq2seq models for text2SQL

1. Encoder encodes natural question, sequential schema and translates to SQL logical form.
2. Shaw et al.(2020) showed T5 model with 3 billion parameters achieves state of the art on spider.

# Literal review

## Use DB content

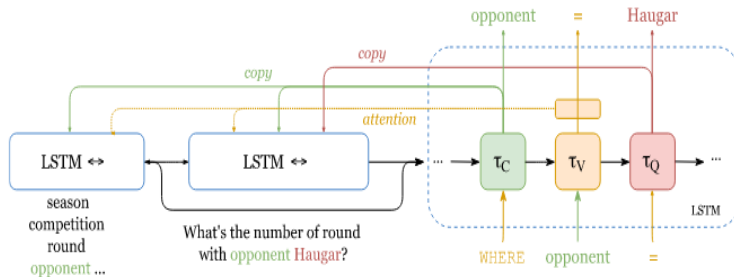
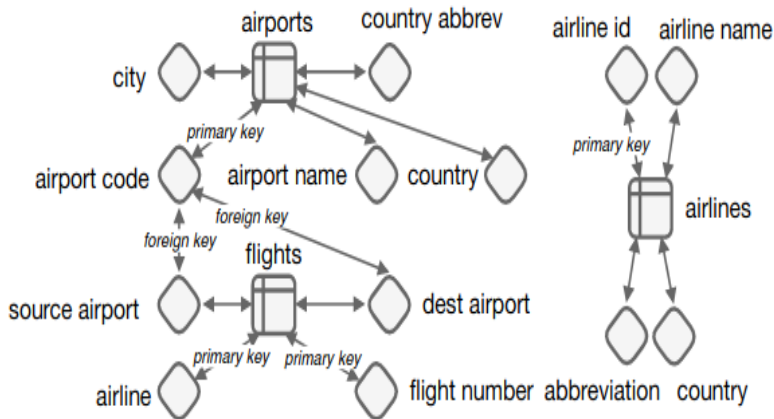


Figure 2: Overview of the base model. The model encodes table columns as well as the user question with a BiLSTM and then decodes the hidden state with a typed LSTM, where the decoding action for each cell is statically determined.

# Literal review

## Use DB content- RAT SQL





## Use DB content- RAT SQL

Type of $x$	Type of $y$	Edge label	Description
Column	Column	SAME-TABLE	$x$ and $y$ belong to the same table.
		FOREIGN-KEY-COL-F	$x$ is a foreign key for $y$ .
		FOREIGN-KEY-COL-R	$y$ is a foreign key for $x$ .
Column	Table	PRIMARY-KEY-F	$x$ is the primary key of $y$ .
		BELONGS-TO-F	$x$ is a column of $y$ (but not the primary key).
Table	Column	PRIMARY-KEY-R	$y$ is the primary key of $x$ .
		BELONGS-TO-R	$y$ is a column of $x$ (but not the primary key).
Table	Table	FOREIGN-KEY-TAB-F	Table $x$ has a foreign key column in $y$ .
		FOREIGN-KEY-TAB-R	Same as above, but $x$ and $y$ are reversed.
		FOREIGN-KEY-TAB-B	$x$ and $y$ have foreign keys in both directions.

## Use DB content- RAT SQL

1. RAT-SQL uses relational transformers to encode relations of tokens.

$$e_{ij}^{(h)} = \frac{x_i W_Q^{(h)} (x_j W_K^h + r_{ij}^K)^T}{\sqrt{d_z/H}}$$

$$z_i^{(h)} = \sum_{j=1}^n \alpha_{ij}^{(h)} (x_j W_V^{(h)} + r_{ij}^V)$$

2. RAT-SQL performs schema-linking, name-based linking, value-based linking based on relation types above.
3. Encode schema and question is independent.

# New point in research

1. Model has access to value of each field called picklists (e.g *Property\_type\_code* can have one of the following values { " *Apartment*" , " *Field*" , " *House*" , " *Shop*" , " *Other*" } )  
→ protect individual data record and sensitive fields such as User IDs or credit numbers.
2. Use LSTM-based pointer generator with multihead-attention as decoder

## Question-schema serialization and encoding

1.  $X = [CLS], Q, [SEP], [T], t_1, [C], c_{11}, \dots, c_{1|T_1|}, [T], t_2, [C], c_{21}, \dots, c_{N|T_N|}, [SEP]$
2.  $X$  is encoded with Bert
3. Followed by a directional LSTM  $\mathbf{h}_X \in \mathbb{R}^{|X| \times n}$
4. The question  $\mathbf{h}_X$  is passed through another bi-LSTM  $\mathbf{h}_Q \in \mathbb{R}^{|Q| \times n}$
5. Dense lookup features to represent meta-data of schema  
 $f_{pri} \in \mathbb{R}^{2 \times n}, f_{for} \in \mathbb{R}^{2 \times n}, f_{type} \in \mathbb{R}^{|\tau| \times n}$

## Question-schema serialization and encoding

Meta-data features and base encoding are fused via a feed-forward layer  $g(\mathbb{R}^{4n} \rightarrow \mathbb{R}^n)$ :

1.  $\mathbf{h}_S^{t_i} = g([\mathbf{h}_X^p; \mathbf{0}; \mathbf{0}; \mathbf{0}])$
2.  $\mathbf{h}_S^{c_{ij}} = g([\mathbf{h}_X^q; f_{pri}^u, f_{for}^v, f_{type}^w])$   
 $= \text{ReLU}(\mathbf{W}_g[\mathbf{h}_X^q; f_{pri}^u, f_{for}^v, f_{type}^w] + \mathbf{b}_g)$
3.  $\mathbf{h}_S = [\mathbf{h}^{t_1}, \dots, \mathbf{h}^{t_{|\tau|}}, \mathbf{h}^{c_{11}}, \dots, \mathbf{h}^{c_{N|T_N|}}] \in \mathbb{R}^{|S| \times n}$

# New point in research

## Bridging

Anchor text, perform fuzzy matching between question  $Q$  and the picklists of each field, if found the matched values in DB, the matched field values are inserted in  $X$ , succeeding corresponding field name and separated by token  $[V]$ .

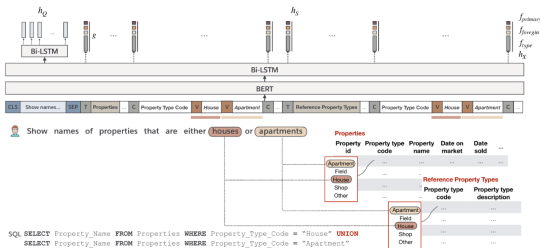


Figure 2: The BRIDGE encoder. The two phrases “houses” and “apartments” in the input question both matched to two DB fields. The matched values are appended to the corresponding field names in the hybrid sequence.

## Decoder

The decoder starts from 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  $S$ .

Mathematically, at each step  $t$ , given the decoder state  $s_t$  and the encoder representation  $[h_Q; h_S] \in \mathbb{R}(|Q|+|S|) \times n$

$$e_{tj}^{(h)} = \frac{s_t W_U^{(h)} (h_j W_V^{(h)})^\top}{\sqrt{n/H}}; \alpha_{tj}^{(h)} = \text{softmax}_j \{e_{tj}^{(h)}\}$$

$$z_t^{(h)} = \sum_{j=1}^{|Q|+|S|} \alpha_{tj}^{(h)} (h_j W_V^{(h)}); z_t = [z_t^{(1)}, \dots, z_t^{(H)}]$$

$$p_{gen}^t = \text{sigmoid}(s_t W_{gen}^s + z_t W_{gen}^z + b_{gen})$$

$$p_{out}^t = p_{gen}^t P_V(y_t) + (1 - p_{gen}^t) \sum_{j: \tilde{X}_j = y_t} \alpha_{tj}^{(H)}$$

## Schema-Consistency Guided Decoding

a simple pruning strategy for sequence decoders, based on the fact that the DB fields appeared 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  $\xi$

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

which initially has entries corresponding to all fields set to 0. Once a table  $t_i$  is decoded, all entries in  $\xi$  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.



# experiment result

# ablation study

# error analysis

