RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers (Wang et al, 2020)

Agenda

- Overview (text-to-SQL)
- Motivation
- Literature
- *RAT-SQL framework*
- *Relation Aware Self-Attention mechanism*
- *Experiments and error analysis*

What is text-to-SQL?

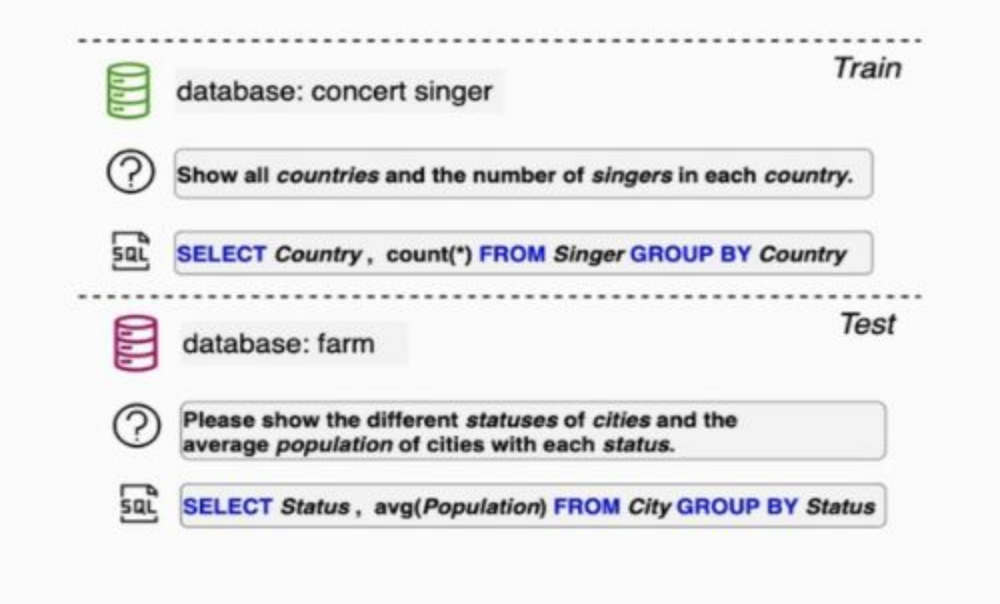
Task: *translating natural language utterance to SQL queries.*

Application: *give people access to vast amounts of databases*

http://slideslive.com/38929348

Why text-to-SQL is a hard problem?

- Generalization to unseen databases and domains.



http://slideslive.com/38929348

Why text-to-SQL is a hard problem?

- Schema encoding and linking

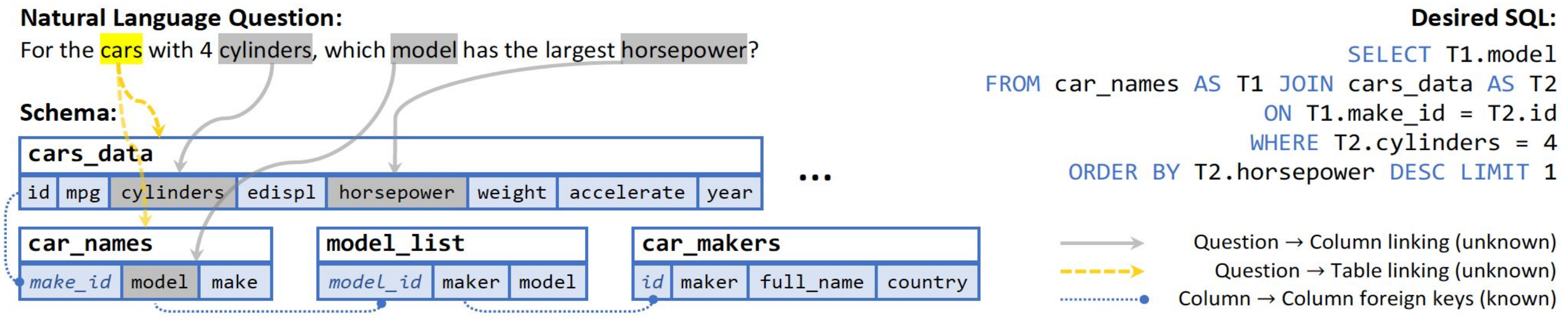


Figure 1: A challenging text-to-SQL task from the Spider dataset.

Motivation

- Address schema encoding and linking problem in text-to-SQL in "RAT-SQL" via *Relation-Aware Self-Attention mechanism*.

- Achieves SOTA performance on ***Spider*** dataset (~8% improvement) for exact match.

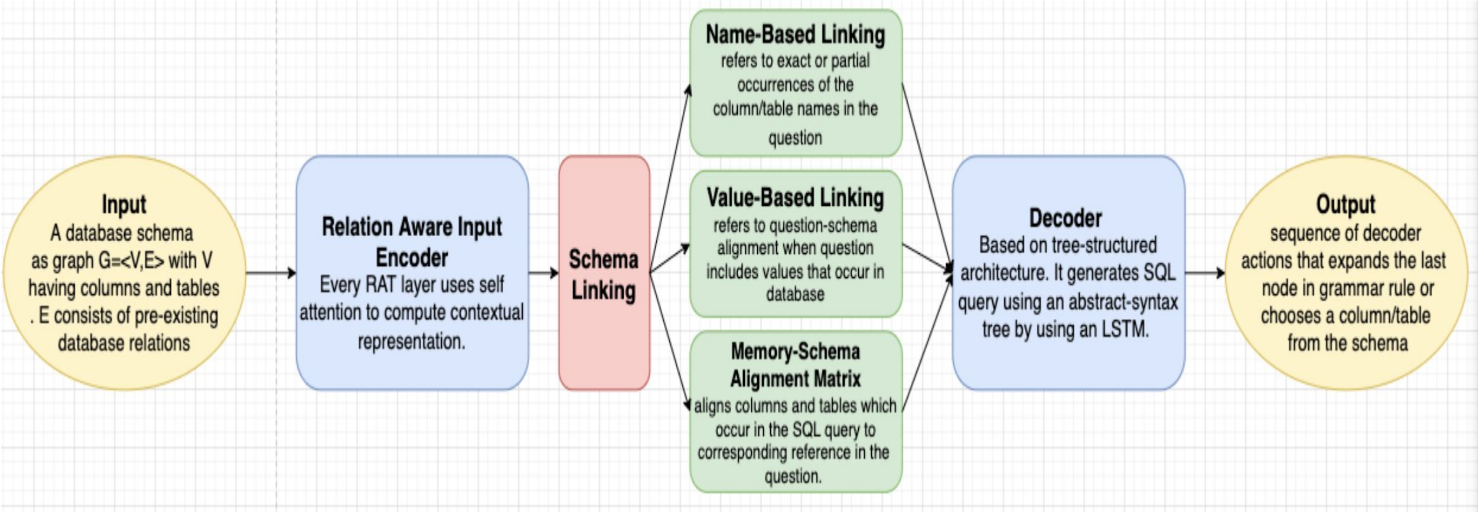
Dataset	# Q	# SQL	# DB	# Domain	# Table /DB	ORDER BY	GROUP BY	NESTED	HAVING
ATIS	5,280	947	1	1	32	0	5	315	0
GeoQuery	877	247	1	1	6	20	46	167	9
Scholar	817	193	1	1	7	75	100	7	20
Academic	196	185	1	1	15	23	40	7	18
IMDB	131	89	1	1	16	10	6	1	0
Yelp	128	110	1	1	7	18	21	0	4
Advising	3,898	208	1	1	10	15	9	22	0
Restaurants	378	378	1	1	3	0	0	4	0
WikiSQL	80,654	77,840	26,521	-	1	0	0	0	0
Spider	10,181	5,693	200	138	5.1	1335	1491	844	388

Table 1: Comparisons of text-to-SQL datasets. **Spider** is the *only one* text-to-SQL dataset that contains both databases with multiple tables in different domains and complex SQL queries. It was designed to test the ability of a system to generalize to not only new SQL queries and database schemas but also new domains.

Literature

- IRNet (Guo et al, 2019)
  - Does not capture binary relations, considers only unary
  - Schema encoder does not exploit schema relations fully.
- GNN (Bogin et al 2019)
  - Does not model context representation of question with schema in encoder.
  - Limits information propagation only to connected nodes defined in predefined graph of foreign keys.

RAT SQL framework



https://medium.com/visionwizard/text2sql-part-4-state-of-the-art-models-cf81a377d4d2

Problem Formulation

Given: natural language question *Q* and schema *S=<C, T>*

Goal: Generate SQL program P represented as abstract syntax tree in the context-free grammar of SQL

Relation-Aware Self-Attention

Goal is to represent:

- pre-existing relational structure in the input (see later)
- soft relations between sequence elements in the same embedding (self-attention)

Relation-Aware Self-Attention

Self-attention in Transformers (Vaswani et al)

$$\boldsymbol{x}_i \rightsquigarrow \boldsymbol{q}_i, \boldsymbol{k}_i, \boldsymbol{v}_i$$

$$\alpha_{ij} = \text{softmax}_j \frac{\boldsymbol{q}_i \boldsymbol{k}_j^\top}{\sqrt{\text{dim}}}$$

$$\boldsymbol{y}_i = \sum_j \alpha_{ij} \boldsymbol{v}_j$$

Relation-Aware Self-attention in RAT-SQL (schema encoding)

$$\boldsymbol{x}_i \rightsquigarrow \boldsymbol{q}_i, \boldsymbol{k}_i, \boldsymbol{v}_i$$

$$\alpha_{ij} = \text{softmax}_j \frac{\boldsymbol{q}_i \left(\boldsymbol{k}_j + \boldsymbol{\beta}_{ij}\right)^\top}{\sqrt{\text{dim}}}$$

$$\boldsymbol{y}_i = \sum_j \alpha_{ij} \left(\boldsymbol{v}_j + \boldsymbol{\varepsilon}_{ij}\right)$$

~~Relative positional embeddings~~

Arbitrary edge features

Modified from: http://slideslive.com/38929348

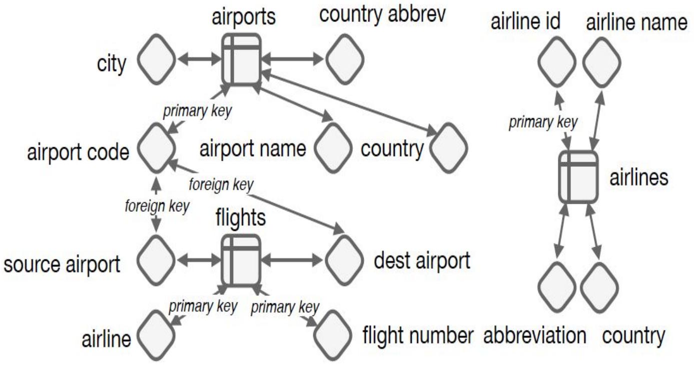
Pre-existing relations in schema

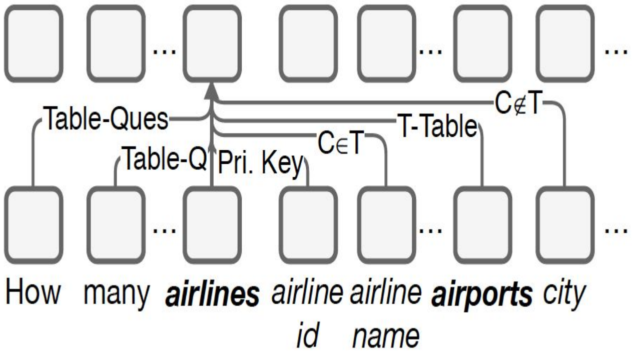
Type of x	Type of y	Edge label	Description
Column	Column	SAME-TABLE	x and y belong to the same table.
		FOREIGN-KEY-COL-F	x is a foreign key for y.
		FOREIGN-KEY-COL-R	y is a foreign key for x.
Column	Table	PRIMARY-KEY-F	x is the primary key of y.
		BELONGS-TO-F	x is a column of y (but not the primary 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.

Input preprocessing

$$\mathcal{G}_Q = \langle \mathcal{V}_Q, \mathcal{E}_Q \rangle$$

$$\mathcal{V}_Q = \mathcal{V} \cup Q = \mathcal{C} \cup \mathcal{T} \cup Q$$





Schema Encoding

- Representation of every node in G: $X = (\boldsymbol{c}_1^{\text{init}}, \cdots, \boldsymbol{c}_{|\mathcal{C}|}^{\text{init}}, \boldsymbol{t}_1^{\text{init}}, \cdots, \boldsymbol{t}_{|\mathcal{T}|}^{\text{init}}, \boldsymbol{q}_1^{\text{init}}, \cdots, \boldsymbol{q}_{|Q|}^{\text{init}}).$
    - Glove processed through BiLSTM
    - Bert pre-trained embedding
- Initial representations are independent of relational information
- Encoder applies stack of self-attention

Schema Linking

- Name-based linking
  - Exact match
  - Partial match
- Value-based linking
  - Value based column name retrieval

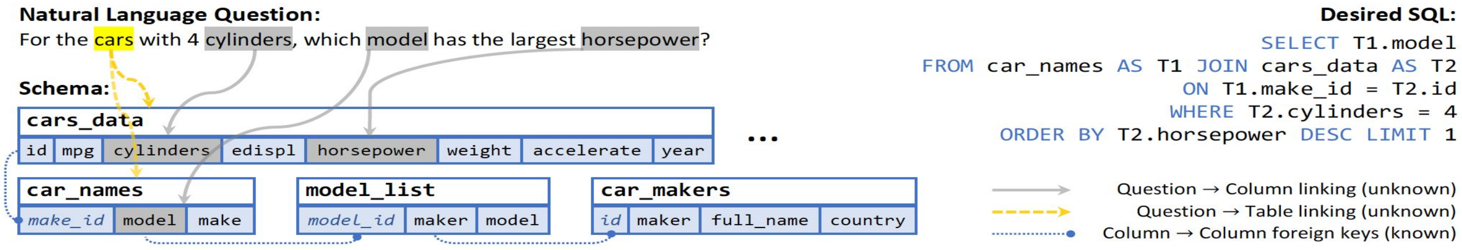


Figure 1: A challenging text-to-SQL task from the Spider dataset.

Memory-schema alignment matrix

Intuition: Tables and column names that appear in program P will appear in question Q

$$\tilde{L}^{\text{col}}_{i,j} = \frac{y_i W^{\text{col}}_Q (\boldsymbol{c}^{\text{final}}_j W^{\text{col}}_K + \boldsymbol{r}^K_{ij})^\top}{\sqrt{d_x}} \tag{3}$$

$$\tilde{L}^{\text{tab}}_{i,j} = \frac{y_i W^{\text{tab}}_Q (\boldsymbol{t}^{\text{final}}_j W^{\text{tab}}_K + \boldsymbol{r}^K_{ij})^\top}{\sqrt{d_x}}$$

$$L^{\text{col}}_{i,j} = \operatorname{softmax}_j\{\tilde{L}^{\text{col}}_{i,j}\} \qquad L^{\text{tab}}_{i,j} = \operatorname{softmax}_j\{\tilde{L}^{\text{tab}}_{i,j}\}$$

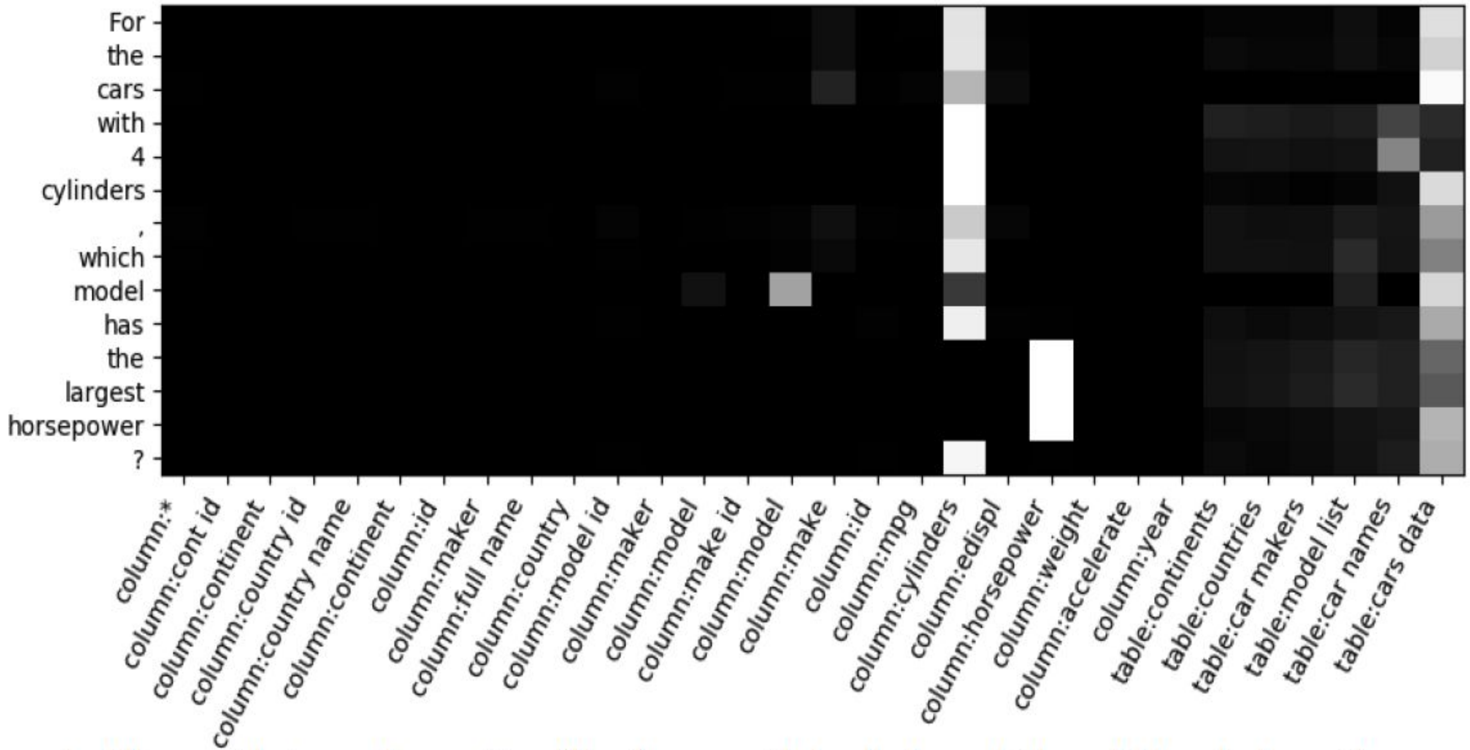


Figure 5: Alignment between the question *"For the cars with 4 cylinders, which model has the largest horsepower"* and the database `car_1` schema (columns and tables) depicted in Figure 1.

Decoder/generation of SQL

- Follows the tree structured architecture of Yin and Neubig (2017)
  - expand into a grammar rule : APPLYRULE

$$\textsc{ApplyRule}[R] \mid a_{<t}, y) = \mathrm{softmax}_R(g(\boldsymbol{h}_t))$$

$$f_{\mathrm{LSTM}}([\boldsymbol{a}_{t-1} \| \boldsymbol{z}_t \| \boldsymbol{h}_{p_t} \| \boldsymbol{a}_{p_t} \| \boldsymbol{n}_{f_t}], \boldsymbol{m}_{t-1}, \boldsymbol{h}_{t-1})$$

Decoder/generation of SQL

- choose a column/table from the schema (terminal): SELECTCOLUMN and SELECTTABLE.

$$\tilde{\lambda}_i = \frac{\boldsymbol{h}_t W_Q^{\text{sc}} (y_i W_K^{\text{sc}})^T}{\sqrt{d_x}} \qquad \lambda_i = \operatorname*{softmax}_i \{\tilde{\lambda}_i\}$$

$$\Pr(a_t = \textsc{SelectColumn}[i] \mid a_{<t}, y) = \sum_{j=1}^{|y|} \lambda_j L_{j,i}^{\text{col}}$$

Results on Spider dataset

Model	Dev	Test
IRNet (Guo et al., 2019)	53.2	46.7
Global-GNN (Bogin et al., 2019b)	52.7	47.4
IRNet V2 (Guo et al., 2019)	55.4	48.5
RAT-SQL (ours)	**62.7**	**57.2**
With BERT:		
EditSQL + BERT (Zhang et al., 2019)	57.6	53.4
GNN + Bertrand-DR (Kelkar et al., 2020)	57.9	54.6
IRNet V2 + BERT (Guo et al., 2019)	63.9	55.0
RYANSQL V2 + BERT (Choi et al., 2020)	**70.6**	60.6
RAT-SQL + BERT (ours)	69.7	**65.6**

Split	Easy	Medium	Hard	Extra Hard	All
RAT-SQL					
Dev	80.4	63.9	55.7	40.6	62.7
Test	74.8	60.7	53.6	31.5	57.2
RAT-SQL + BERT					
Dev	86.4	73.6	62.1	42.9	69.7
Test	83.0	71.3	58.3	38.4	65.6

Easy

What is the number of cars with more than 4 cylinders?

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

Meidum

For each stadium, how many concerts are there?

```
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

Which countries in Europe have at least 3 car manufacturers?

```
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

What is the average life expectancy in the countries where English is not the official language?

```
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")
```

Figure 3: SQL query examples in 4 hardness levels.

https://arxiv.org/pdf/1809.08887.pdf

Results on WikiSQL

	Dev		Test	
Model	LF Acc%	Ex. Acc%	LF Acc%	Ex. Acc%
IncSQL (Shi et al., 2018)	49.9	84.0	49.9	83.7
MQAN (McCann et al., 2018)	76.1	82.0	75.4	81.4
RAT-SQL (ours)	73.6	79.5	73.3	78.8
Coarse2Fine (Dong and Lapata, 2018)	72.5	79.0	71.7	78.5
PT-MAML (Huang et al., 2018)	63.1	68.3	62.8	68.0

Ablation Study

Model	Accuracy (%)
RAT-SQL + value-based linking	**60.54 ± 0.80**
RAT-SQL	55.13 ± 0.84
w/o schema linking relations	40.37 ± 2.32
w/o schema graph relations	35.59 ± 0.85

Error Analysis

- 39% of errors -> a limitation of schema linking
- 29% of errors -> Need of in-domain fine-tuning
  - ‘Older than 21’ -> Age > 21 or age < 21
- 18% of errors -> equivalent implementations of NL but a different SQL syntax

Model	Exact Match	Correctness
RAT-SQL	0.59	0.81
RAT-SQL + BERT	0.67	0.86

Table 7: Consistency of the two RAT-SQL models.

Key takeaways

- RAT-SQL presented a unified framework to address schema representation and schema linking challenges.
- Contextual representation of question with schema in encoder helps.
- Combining predefined hard schema relations with soft alignment on sequence elements (different from GNN) in encoder added value!