RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers

Bailin Wang, Richard Shin**, Xiaodong Liu, Alex Polozov, Matthew Richardson

Edinburgh
University of Edinburgh
Natural Language Processing
NLP

Microsoft Research

Berkeley
UNIVERSITY OF CALIFORNIA

Background

Text-to-SQL Parsing

database: concert singer

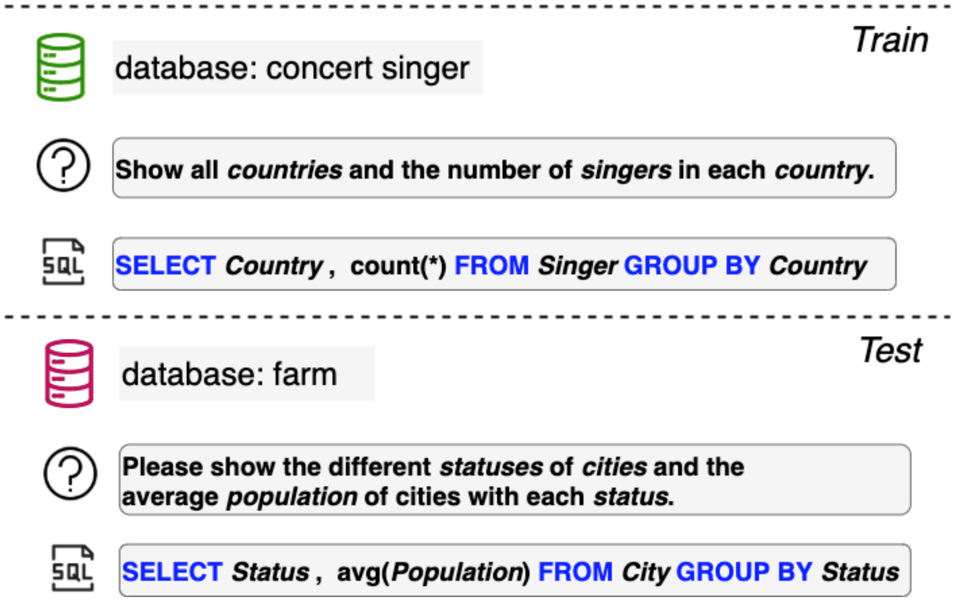
Show all *countries* and the number of *singers* in each *country*.

SELECT *Country* , count(*) FROM *Singer* GROUP BY *Country*

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

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

Cross-Domain Text-to-SQL Parsing



Generalization Challenge: *a parser needs to generalize to unseen domains.*

Application: *reduce annotation effort for multi-domain interfaces*

Capturing Domain Generalization

Motivating Example

Natural Language Question:
For the cars with 4 cylinders, which model has the largest horsepower?

Desired SQL:

```
SELECT T1.model
FROM car_names AS T1 JOIN cars_data AS T2
ON T1.make_id = T2.id
WHERE T2.cylinders = 4
ORDER BY T2.horsepower DESC LIMIT 1
```

Schema:

cars_data							
id	mpg	cylinders	edispl	horsepower	weight	accelerate	year

...

car_names		
make_id	model	make

model_list		
model_id	maker	model

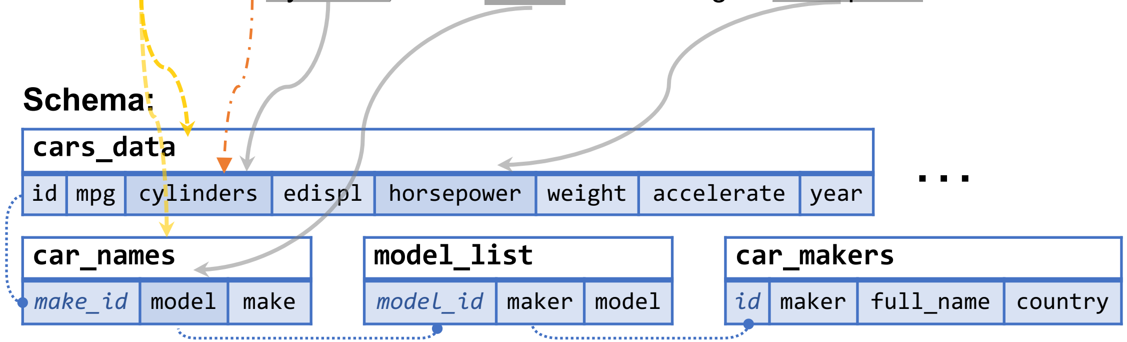
car_makers			
id	maker	full_name	country

Column → Column foreign keys (known)

Schema Encoding: *encode graph-structured database schemas*

Motivating Example

Natural Language Question:
For the cars with 4 cylinders, which model has the largest horsepower?



Desired SQL:

```
SELECT T1.model
FROM car_names AS T1 JOIN cars_data AS T2
ON T1.make_id = T2.id
WHERE T2.cylinders = 4
ORDER BY T2.horsepower DESC LIMIT 1
```

Column → Column foreign keys (known)

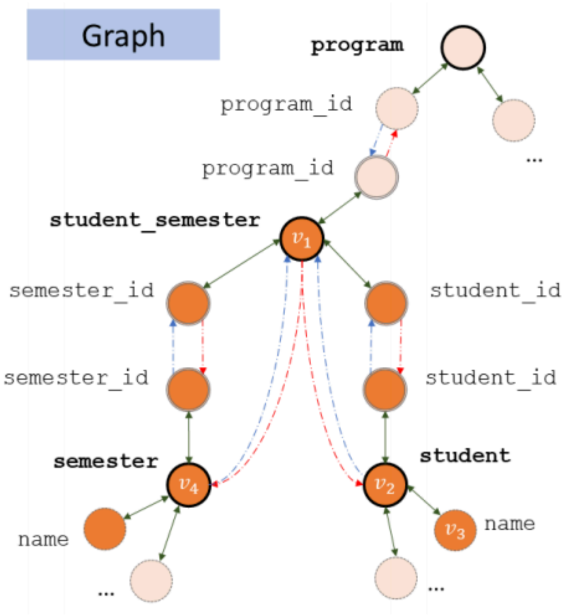
Question → Column linking (latent)

Question → Table linking (latent)

Value → Column linking (latent)

Schema Linking: *capturing latent linking between question and tables/columns*

Previous Work: GNN

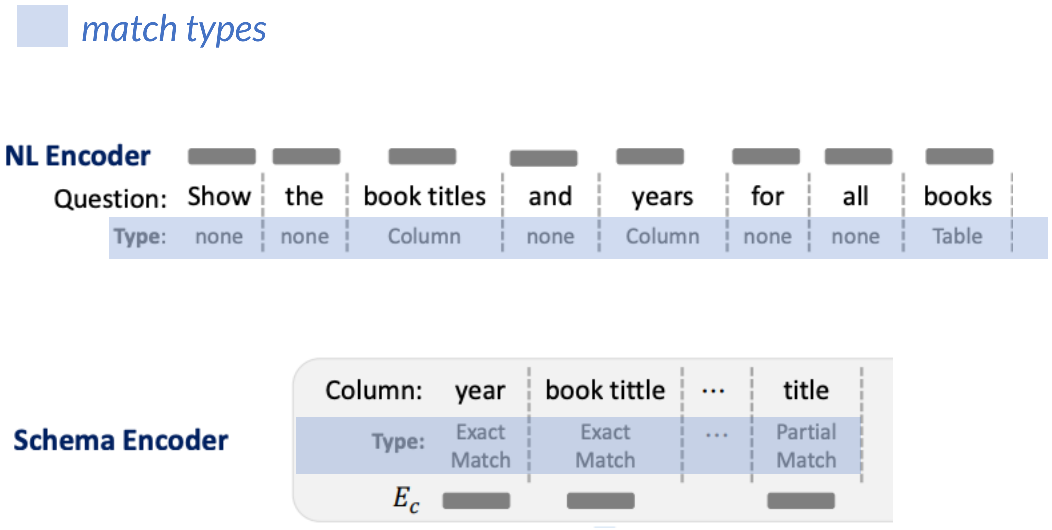


Bogin et al. (2019a) used GNN to encode database schema

Shortcomings:

- message propagation is limited to the schema edges such as foreign key relations
- question and schema are not jointly encoded, thus making it hard for schema linking

Previous Work: IRNet



IRNet (Guo et al., 2019) used string-match based types (highlighted) to facilitate schema linking.

Shortcomings:

- Schema Encoder of IRNet does not fully exploit the schema relations
- IRNet captures unary relations instead of binary relations

Example

Unary relation:

word '**books**' is matched (*to some column*)

Binary relation:

word '**books**' is matched to '**book title**'

A Unified Framework With Relation-Aware Transformer

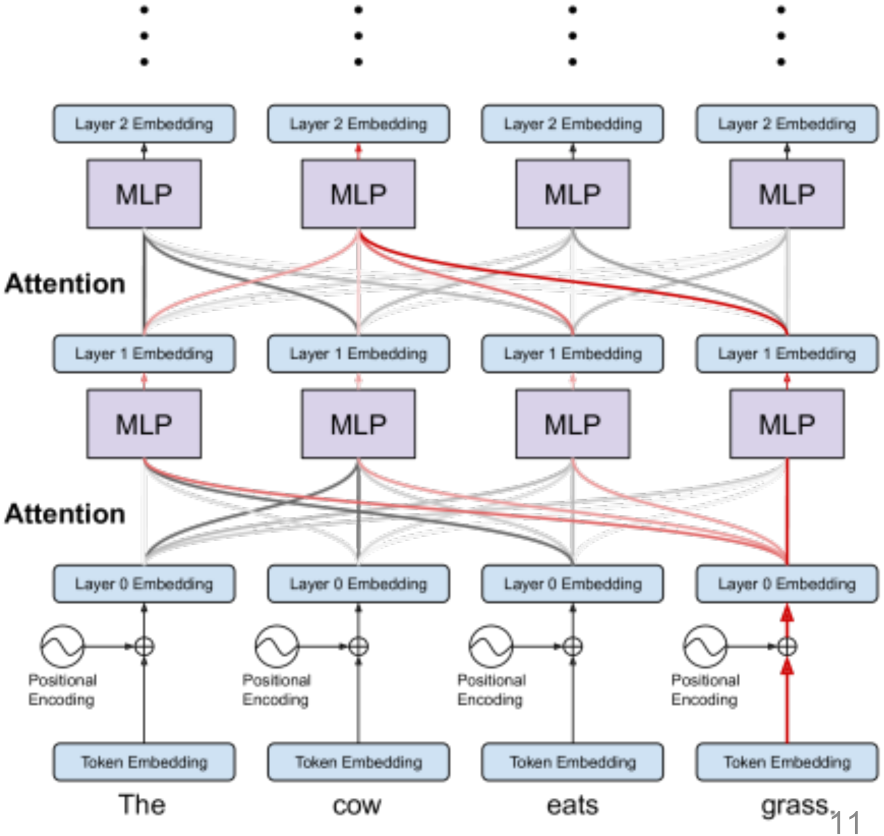
Transformer

[Vaswani et al., 2017]

$$x_i \rightsquigarrow q_i, k_i, v_i$$

$$\alpha_{ij} = \mathrm{softmax}_j \frac{q_i\, k_j^\top}{\sqrt{\dim}}$$

$$y_i = \sum_j \alpha_{ij}\, v_j$$



Relation-Aware Transformer (RAT)

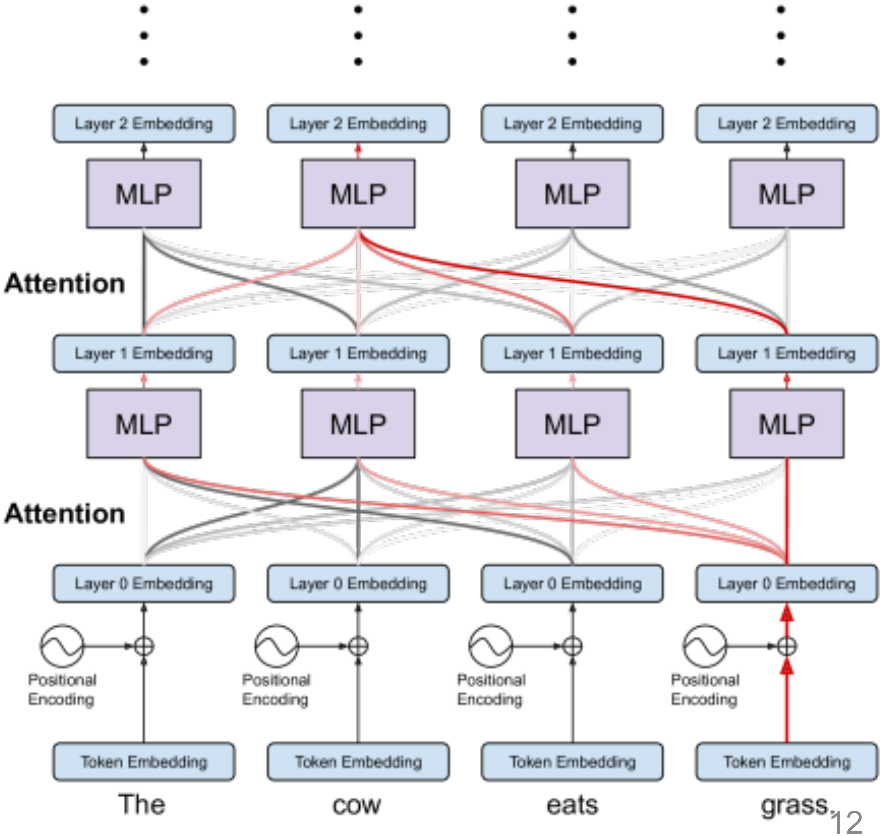
[Vaswani et al., 2017] [Shaw et al., 2018]

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

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

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

Relative positional embeddings



Relation-Aware Transformer (RAT)

[Vaswani et al., 2017] [Shaw et al., 2018]

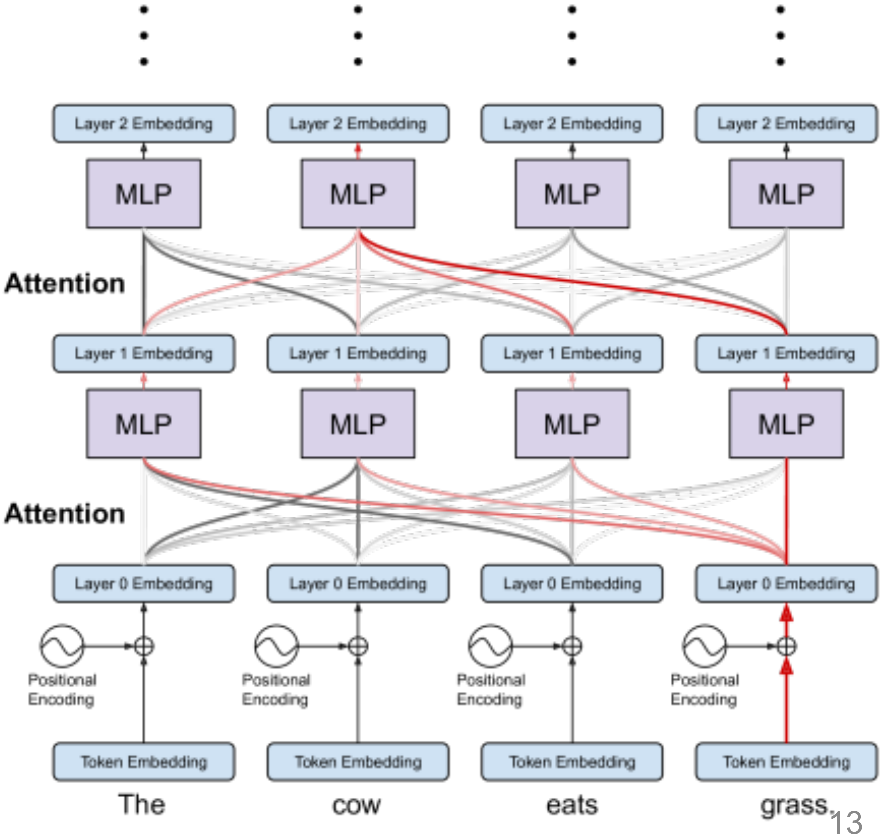
$$\boldsymbol{x}_i \rightsquigarrow \boldsymbol{q}_i, \boldsymbol{k}_i, \boldsymbol{v}_i$$

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

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

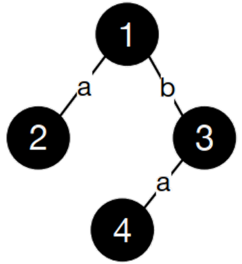
~~Relative positional embeddings~~

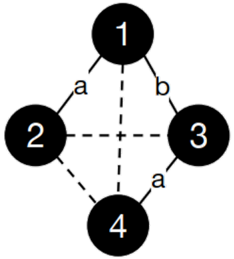
Arbitrary edge features

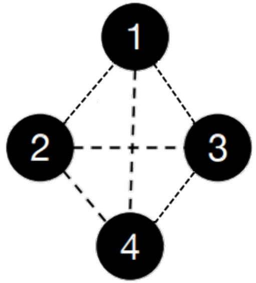


GNNs *vs.* Transformers

	GCN/GGNN	GAT	RAT	Transformer
Message Passing	Neighborhood	Neighborhood	All (w/ diff. edge funcs)	All
Edge features	Yes, neighbors	Yes, neighbors	Yes, all nodes	Induced
Aggregation	Conv/Gating	Self-Attention	Self-Attention	Self-Attention







(Binary) Relations for Schema Encoding

- Example relations
  - ❖ Foreign key relations.
  - ❖ Column/table correspondence.
  - ❖ Table/primary key correspondence.

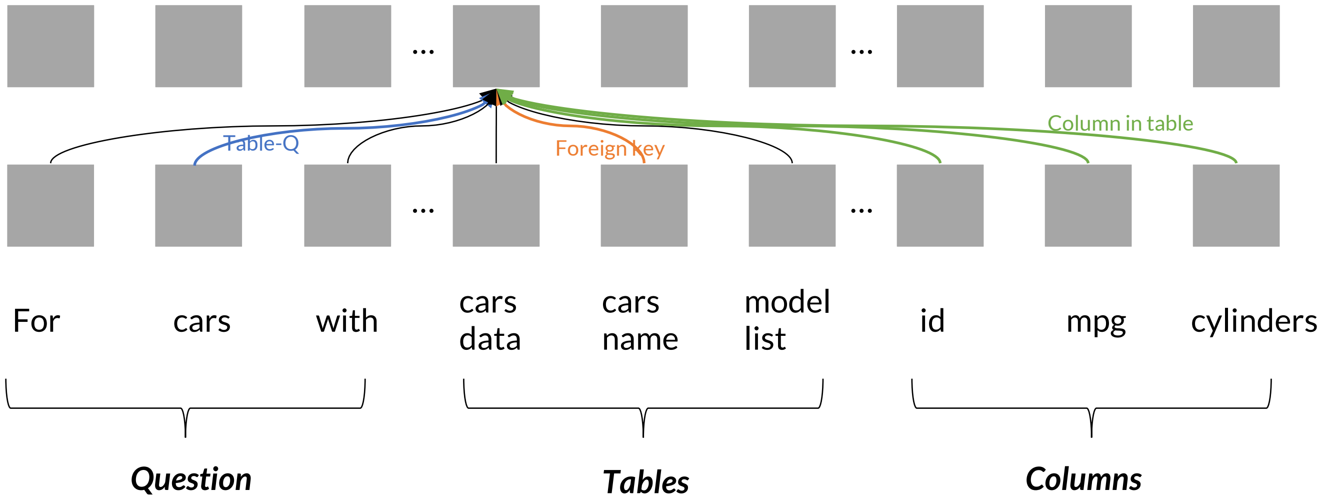
See our paper for a full list of schema relations.

(Binary) Relations for Schema Linking

- Name-based linking relation
  - E.g., link 'cars' to column 'car_names'
  - Exact occurrences of tables/columns
  - Partial occurrences of tables/columns

- Value-based linking relation
  - E.g., link value '*Edinburgh*' to column '*city*'
  - Values as evidence to generate corresponding columns
  - retrieved quickly via DB indices & textual search

Example of RAT Layer



Experiments

Spider Dataset

	Multiple schemas	Complex questions
ATIS, GeoQuery		
WikiSQL		
Spider		

Much more challenging task!

Results on Spider

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

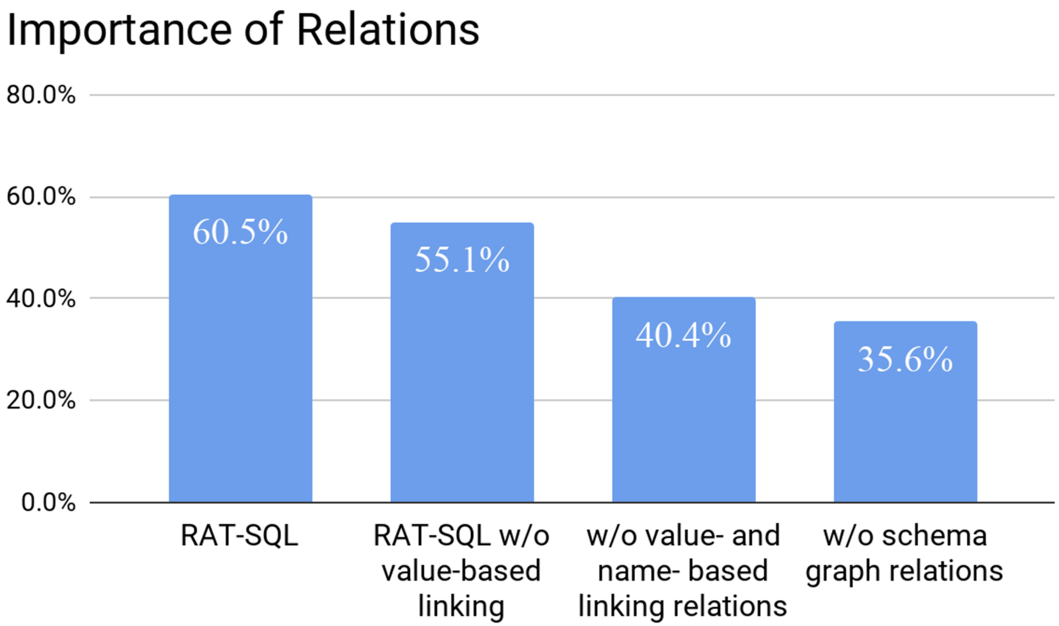
- RAT-SQL achieves the new state-of-the-art performance
- Our non-BERT version is better than most BERT-augmented models

1 May 02, 2020	RATSQL v3 + BERT (DB content used) *Microsoft Research* (Wang and Shin et al., ACL '20) code	69.7	**65.6**
2 May 31, 2020	AuxNet + BART (DB content used) *Anonymous*	70.0	61.9
2 Dec 13, 2019	RATSQL v2 + BERT (DB content used) *Microsoft Research* (Wang and Shin et al., ACL '20) code	65.8	61.9
3 May 31, 2020	AuxNet + BART *Anonymous*	68.0	61.3
4 Feb 18, 2020	RYANSQL v2 + BERT *Kakao Enterprise* (Choi et al., '20)	70.6	60.6
5 Dec 18, 2019	IRNet++ + XLNet (DB content used) *Anonymous*	65.5	60.1

Current Leaderboard

(June 15)

Ablations



Schema linking features and schema graph relations are crucial.

Summary

- We propose RAT-SQL, a unified framework to address *schema encoding* and *schema linking* based on the relation-aware transformer.

- RAT-SQL achieves the new state-of-the-art performance on Spider.

- Code: https://github.com/microsoft/rat-sql

Demo