LGESQL: Line Graph Enhanced Text-to-SQL Model with Mixed Local and Non-Local Relations

—— ACL 2021

Ruisheng Cao, Lu Chen, Zhi Chen, Yanbin Zhao,
Su Zhu and Kai Yu

X-LANCE Lab, Department of Computer Science and Engineering
MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
Shanghai Jiao Tong University, Shanghai, China
State Key Lab of Media Convergence Production Technology and Systems, Beijing, China
AISpeech Co., Ltd., Suzhou, China

Thursday 24th June, 2021

Outline

- Introduction and Motivation
- LGESQL Model
- Experimental Results
- Conclusion

What is Text-to-SQL?

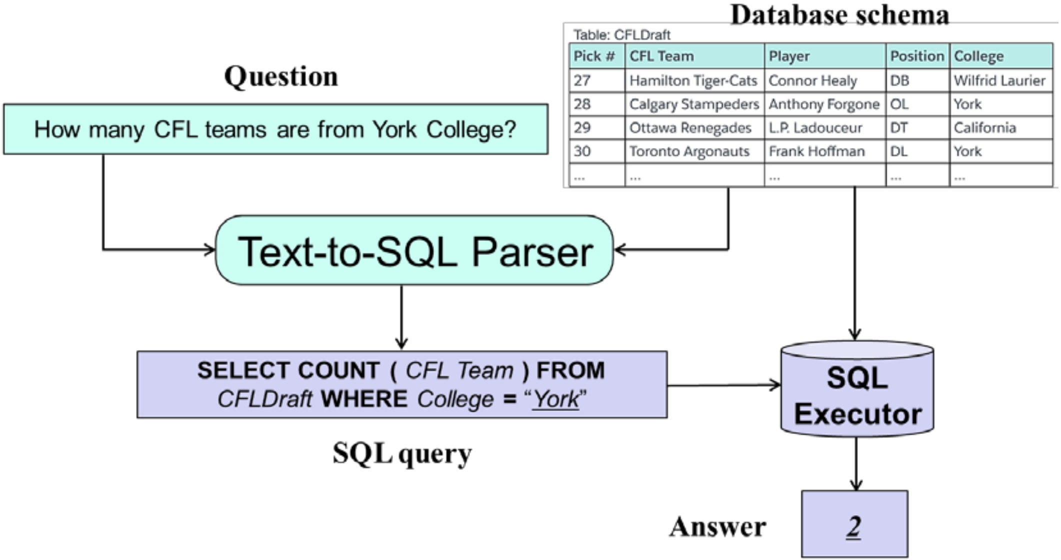
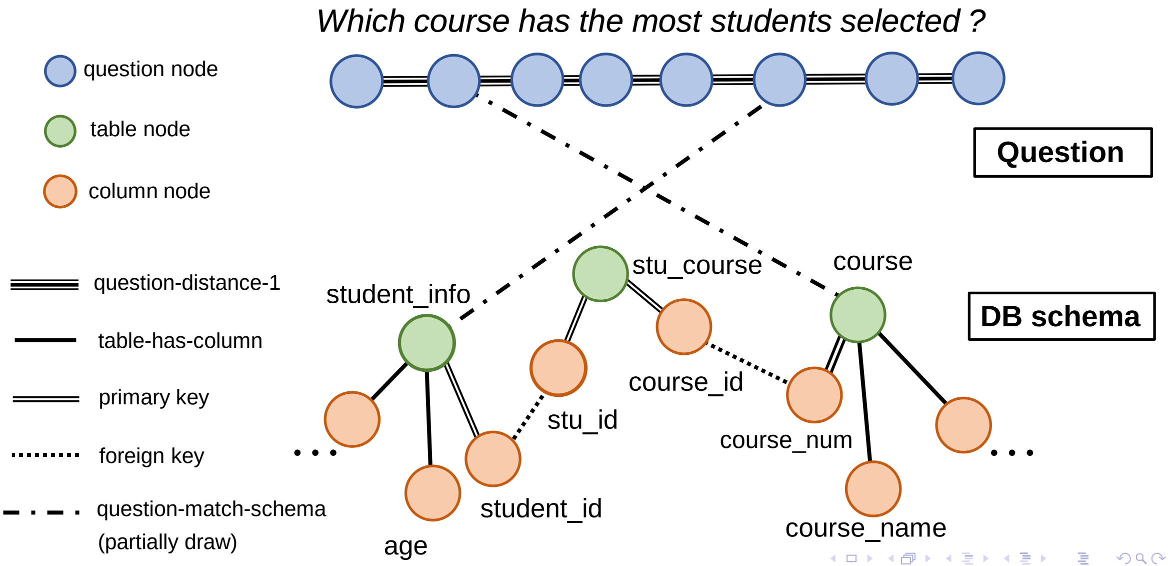


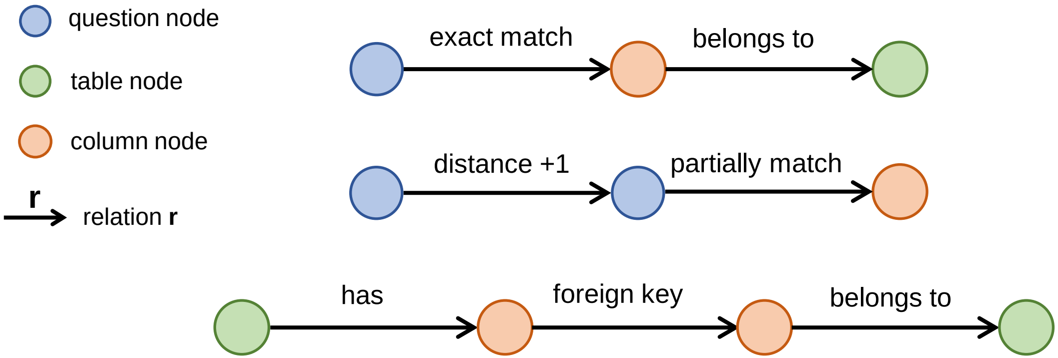
Table: CFLDraft

Pick #	CFL Team	Player	Position	College
27	Hamilton Tiger-Cats	Connor Healy	DB	Wilfrid Laurier
28	Calgary Stampeders	Anthony Forgone	OL	York
29	Ottawa Renegades	L.P. Ladouceur	DT	California
30	Toronto Argonauts	Frank Hoffman	DL	York
...	...	...	...	...

Heterogeneous Graph Encoding Problem



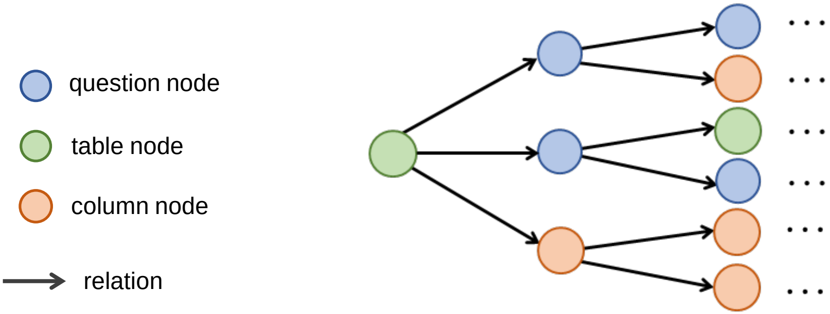
1. Ignore multi-hop relations



some empirically useful meta-paths

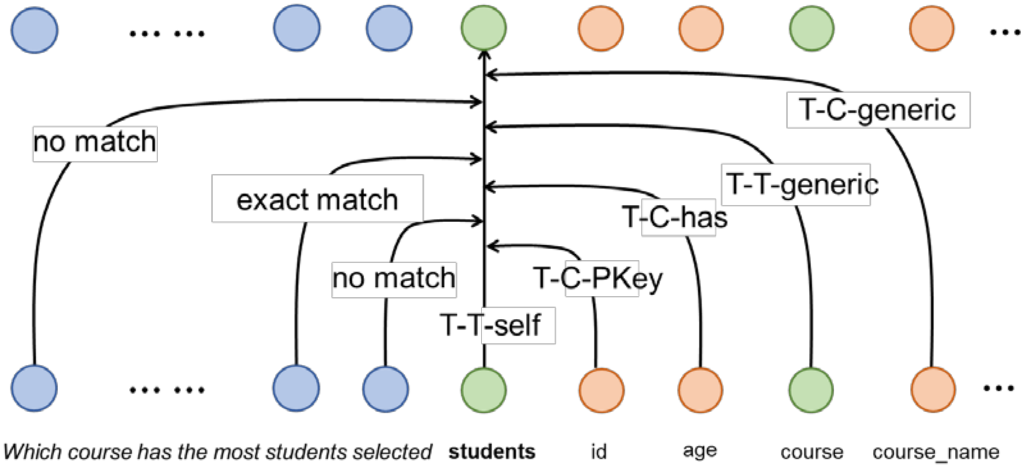
1. Ignore multi-hop relations

- Semantics embedded in the structure of edges
  - Meta-path captures multi-hop connections
  - Path length $\uparrow \implies$ # of meta-paths exponentially $\uparrow$

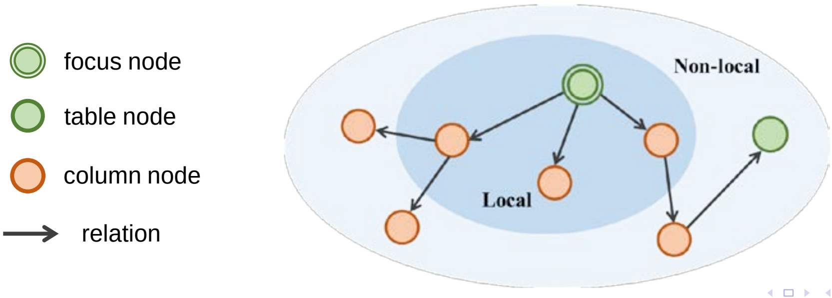


2. Oversmoothing problem

- RATSQL (Wang et al., 2020a)



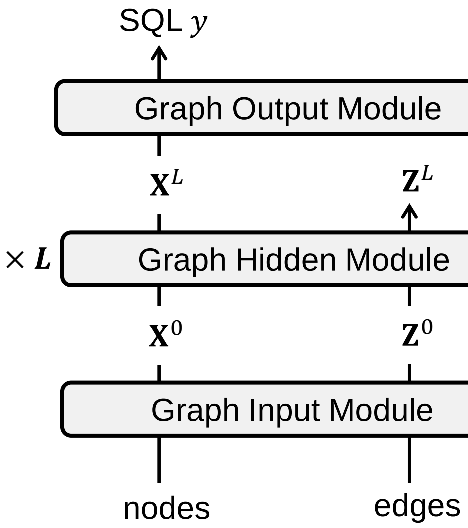
- Distinguish local and non-local relations
  - Local means 1-hop relations
  - Non-local means composite relations with meta-path length > 1
  - Complete graph leads to the ***over-smoothing*** problem



Outline

- Introduction and Motivation
- LGESQL Model
- Experimental Results
- Conclusion

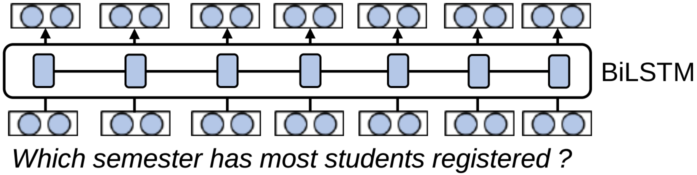
Overall model architecture



- Generate target SQL upon node embeddings
- Incorporate graph structure into node/edge representations
- Initialize node and edge features

Graph Input Module

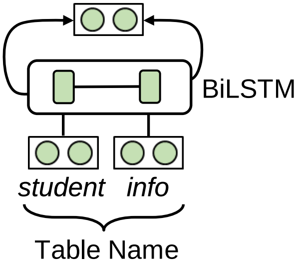
Question nodes: $\mathbf{X}_q^0$

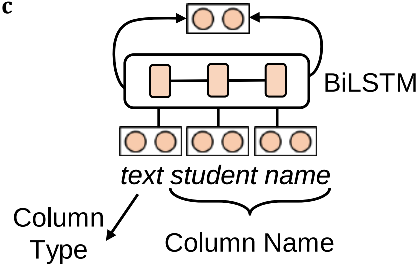


Initial edge embeddings:

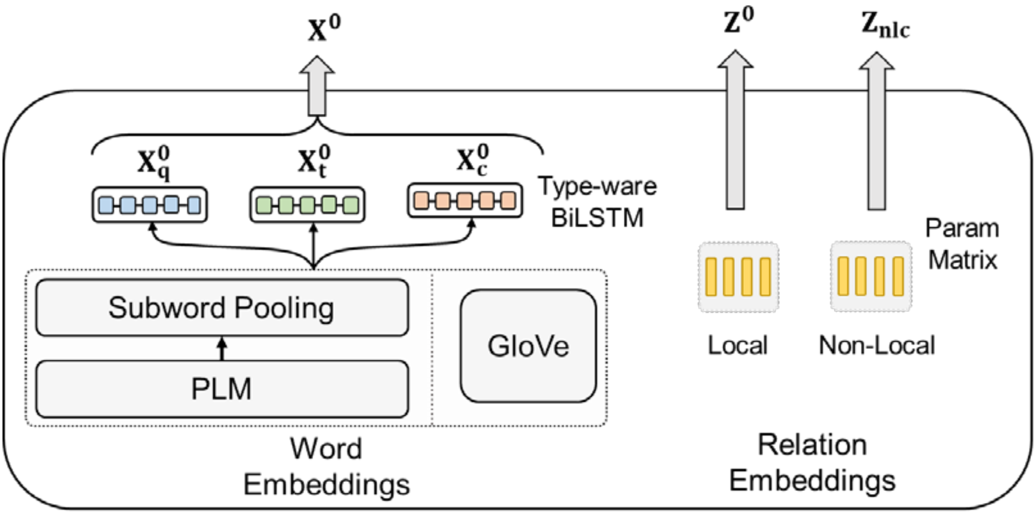
$\mathbf{Z}^0$: local $\qquad$ $\mathbf{Z}_{nlc}$: non-local

Schema nodes: $\mathbf{X}_t^0$ and $\mathbf{X}_c^0$



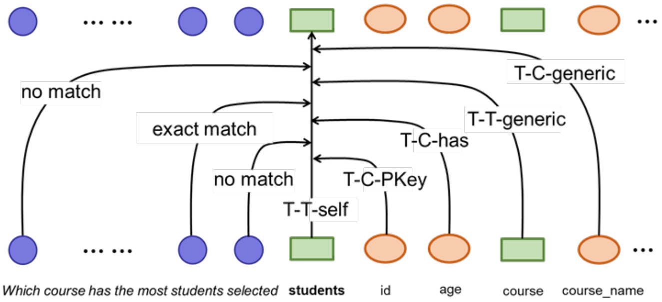


Graph Input Module



Graph Hidden Module (w/o line graph)

- RGAT: relational graph attention network



$$\mathbf{X}^{l+1} = \mathrm{RGAT}^{\mathrm{n}}(\mathbf{X}^{l}, \mathbf{Z}^{l}, G^{n})$$

node features — edge features — graph structure

- Iteration in one RGAT layer

$$e_{ij}^{(h)} = \frac{\boldsymbol{x}_i \mathbf{W}_{\mathrm{Q}}^{(h)} (\boldsymbol{x}_j \mathbf{W}_{\mathrm{K}}^{(h)} + \boldsymbol{z}_{ij})^{\mathrm{T}}}{\sqrt{d_x / H}}$$

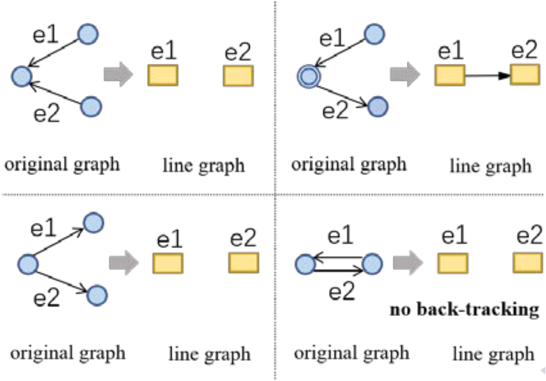
$$\alpha_{ij}^{(h)} = \mathrm{softmax}\{e_{ij}^{(h)}\}_j$$

$$\hat{\boldsymbol{x}}_i^{(h)} = \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(h)} (\boldsymbol{x}_i \mathbf{W}_{\mathrm{V}}^{(h)} + \boldsymbol{z}_{ij})$$

$\boldsymbol{z}_{ij}$: relation-aware edge features

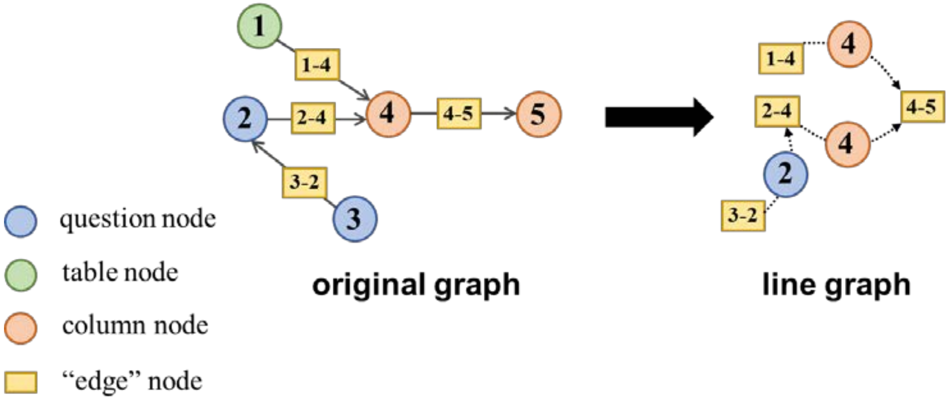
Line Graph Construction

- If $v_1 \overset{e_1}{\rightarrow} v_2 \overset{e_2}{\rightarrow} v_3$ in the original graph, edge $e_1 \overset{v_2}{\rightarrow} e_2$ exists in the line graph
- No back-tracking: if $v_1 = v_3$, remove edge $e_1 \overset{v_2}{\rightarrow} e_2$ in the line graph
- Only the upper right figure has an edge in the line graph (Chen et.al, 2019b)

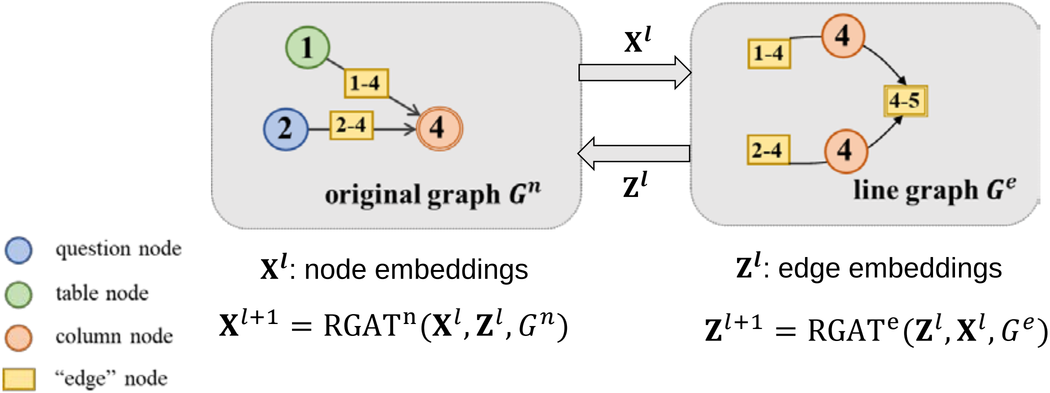


Line Graph Construction

- An illustration of constructing the line graph from the original graph
  - Only local relations are used to construct the line graph



- Node/edge update in one hidden layer
  - Take node 4 and edge 4-5 as example



$\mathbf{X}^l$: node embeddings

$$\mathbf{X}^{l+1} = \mathrm{RGAT}^{\mathrm{n}}(\mathbf{X}^l, \mathbf{Z}^l, G^n)$$

$\mathbf{Z}^l$: edge embeddings

$$\mathbf{Z}^{l+1} = \mathrm{RGAT}^{\mathrm{e}}(\mathbf{Z}^l, \mathbf{X}^l, G^e)$$

Graph Hidden Module

- Integrate both local and non-local relations in the **node-centric** graph
  - Mixed static and dynamic embeddings

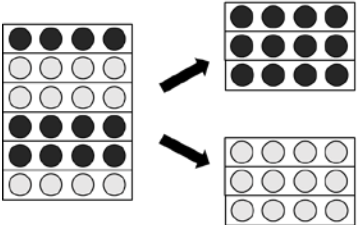
$$\mathbf{X}^{l+1} = \mathrm{RGAT}^{\mathrm{n}}(\mathbf{X}^{l}, \mathbf{Z}^{l}, G^{n})$$

$$\Downarrow$$

$$\mathbf{X}^{l+1} = \mathrm{RGAT}^{\mathrm{n}}(\mathbf{X}^{l}, [\mathbf{Z}^{l}; \mathbf{Z}_{\mathbf{nlc}}], G^{n})$$

$\mathbf{Z}_{\mathbf{nlc}}$ is non-local relation features

G^{n} becomes a complete graph

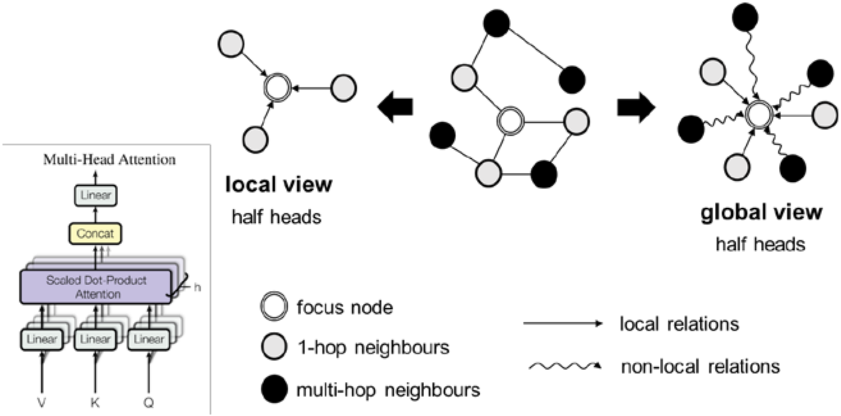


$\mathbf{Z}_{\mathbf{nlc}}$ provided by an embedding matrix

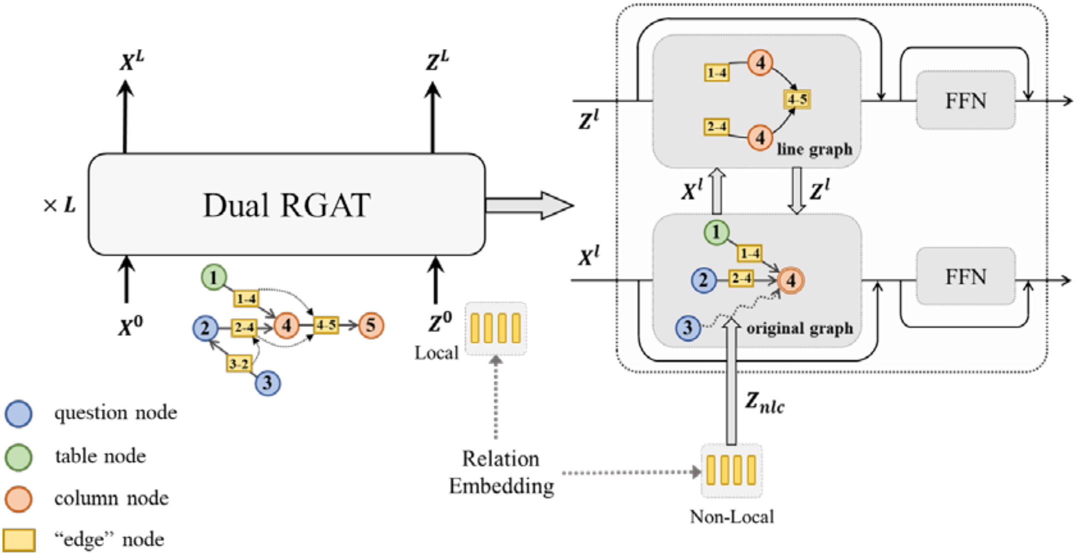
$\mathbf{Z}^{l}$ provided by the line graph

Graph Hidden Module

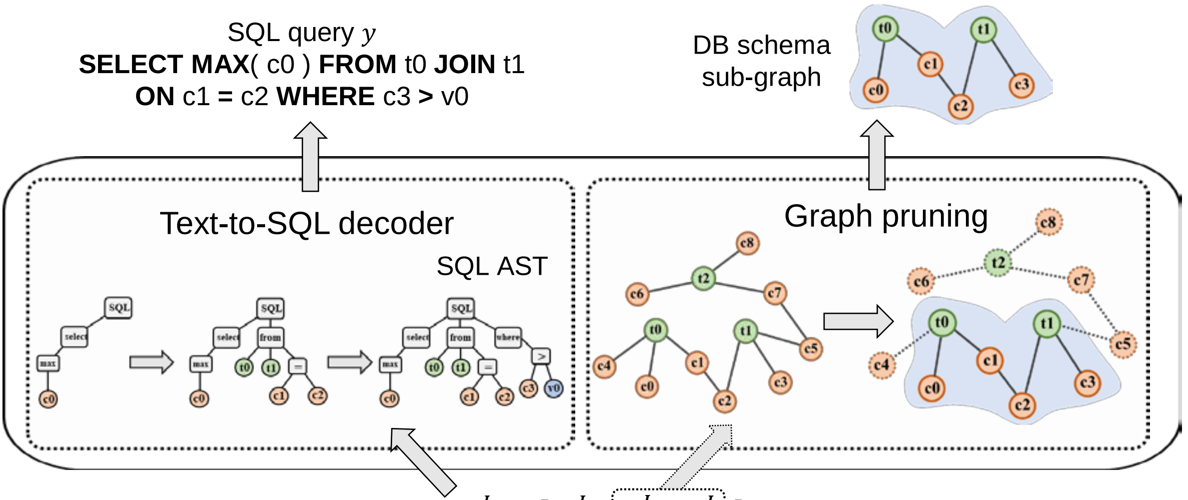
- Integrate both local and non-local relations in the **node-centric** graph
  - Multi-head multi-view concatenation



Graph Hidden Module



Graph Output Module



$$\mathbf{X}^L = [\, \mathbf{X}_q^L \,;\, \mathbf{X}_t^L \,;\, \mathbf{X}_c^L \,]$$

Outline

- Introduction and Motivation
- LGESQL Model
- Experimental Results
- Conclusion

Dataset

- **Spider**: cross-domain zero-shot text-to-SQL benchmark (Yu et al., 2018b)

	Train	Dev
# of samples	8659	1034
# of databases	146	20
Avg # of question nodes	13.4	13.8
Avg # of table nodes	6.6	4.5
Avg # of column nodes	33.1	25.8
Avg # of nodes	53.1	44.1
Avg # of actions	16.3	15.4

- Text-to-SQL decoder is a structured grammar-based transition system (Yin and Neubig, 2017)
- **Exact set match accuracy** on dev and test set

$$Acc_{em} = \frac{\sum_{i=1}^{N} Equal(SQL_i^p, SQL_i^g)}{N}$$

  - compare SQL queries directly
  - $Equal(pred, gold)$ ignores order, e.g. SELECT A, B = SELECT B, A

Main results

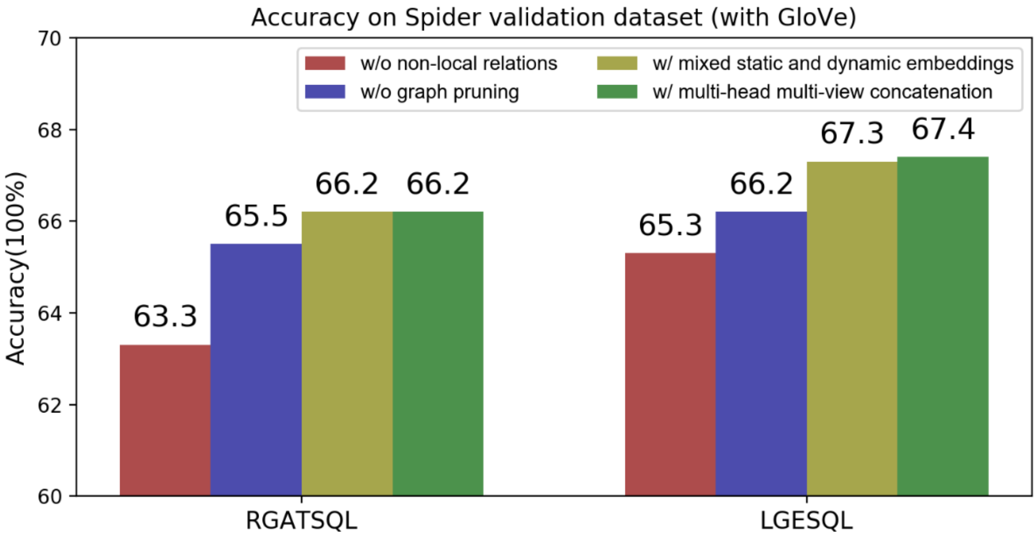
Model	Dev	Test
Without PLM		
GNN (Bogin et al., 2019a)	40.7	39.4
Global-GNN (Bogin et al., 2019b)	52.7	47.4
EditSQL (Zhang et al., 2019b)	36.4	32.9
IRNet (Guo et al., 2019)	53.2	46.7
RATSQL (Wang et al., 2020a)	62.7	57.2
LGESQL	**67.6**	**62.8**
With PLM: BERT		
IRNet (Guo et al., 2019)	53.2	46.7
GAZP (Zhong et al., 2020)	59.1	53.3
EditSQL (Zhang et al., 2019b)	57.6	53.4
BRIDGE (Lin et al., 2020)	70.0	65.0
BRIDGE + Ensemble	71.1	67.5
RATSQL (Wang et al., 2020a)	69.7	65.6
LGESQL	**74.1**	**68.3**
With Task Adaptive PLM		
ShadowGNN (Chen et al., 2021)	72.3	66.1
RATSQL+STRUG (Deng et al., 2021)	72.6	68.4
RATSQL+GRAPPA (Yu et al., 2020)	73.4	69.6
SmBoP (Rubin and Berant, 2021)	74.7	69.5
RATSQL+GAP (Shi et al., 2020)	71.8	69.7
DT-Fixup SQL-SP (Xu et al., 2021)	75.0	70.9
LGESQL+ELECTRA	**75.1**	**72.0**

5.6 With GloVe

2.7 With BERT

2.3 With ELECTRA

Ablation study



Case studies

- LGESQL performs better in predicting **multi-table JOINs** by focusing on local DB connections

HARD: *Count the number of United Airlines flights that arrive in Aberdeen.*

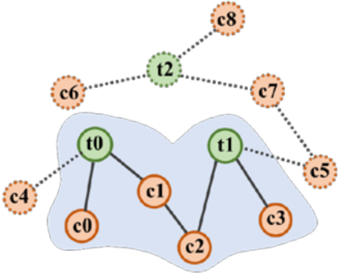
RGATSQL: `SELECT COUNT(*) FROM airlines JOIN airports WHERE airlines.airline = "val" AND airlines.airline = "val"`

LGESQL: `SELECT COUNT(*) FROM airlines JOIN flights JOIN airports WHERE airlines.airline = "val" AND airports.city = "val"`

EXTRA: *Which template type code is used by most number of documents?*

RGATSQL: `SELECT template.template_type_code FROM template GROUP BY template.template_type_code ORDER BY COUNT(*) DESC LIMIT 1`

LGESQL: `SELECT template.template_type_code FROM template JOIN documents GROUP BY template.template_type_code ORDER BY COUNT(*) DESC LIMIT 1`



select max(c0) from t0 **join** t1 **on** c1=c2 **where** c3>“value”

Outline

- Introduction and Motivation
- LGESQL Model
- Experimental Results
- Conclusion

Conclusion

Main contributions

- Utilize a **line graph** to explicitly learn edge features
    - Extend node-centric RGAT into Dual RGAT
    - Local and non-local relations are treated differently
    - Graph pruning improves the discriminative capability of the encoder
- LGESQL achieves state-of-the-art results on the leaderboard of Spider, with GLOVE, BERT and task-adaptive PLM, in the exact set match partition

Thanks & QA

References

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov and Matthew Richardson. 2020a.
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.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang and Dragomir Radev. 2018b.
Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task.
In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Pengcheng Yin and Graham Neubig. 2017.
A syntactic neural model for general-purpose code generation.
In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 440–450, Vancouver, Canada. Association for Computational Linguistics.

Zhengdao Chen, Lisha Li, and Joan Bruna. 2019b.
Supervised community detection with line graph neural networks.
In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net.