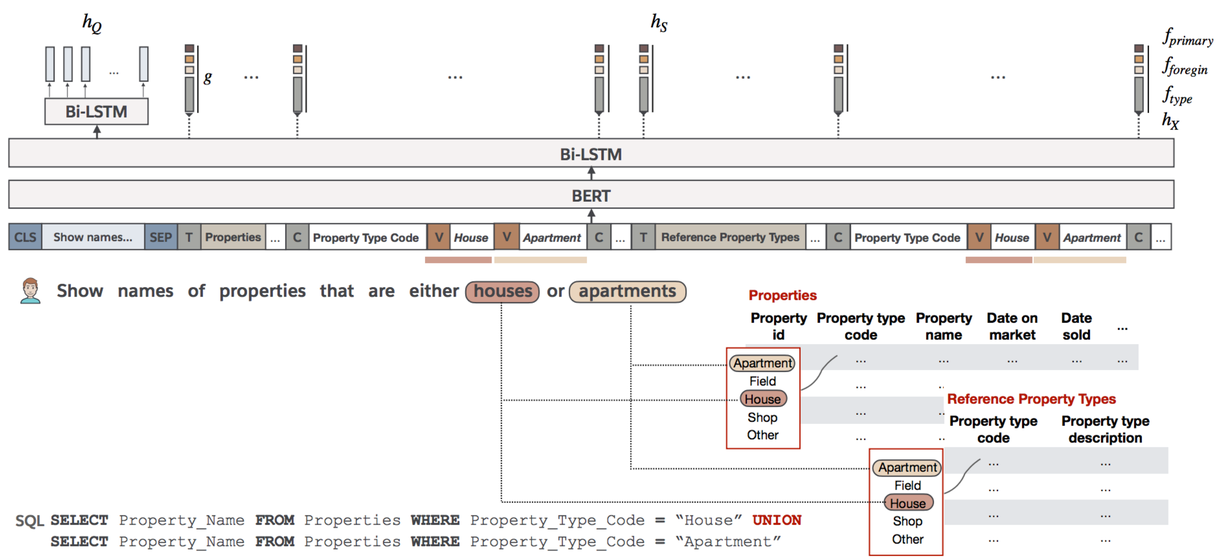
Bridging Textual and Tabular Data for Cross-Domain Text-to-SQL Semantic Parsing

Xi Victoria Lin, Richard Socher and Caiming Xiong

Overview

Querying databases with natural language has great practical value. Recent creation of large-scale datasets have enabled the development of text-to-SQL semantic parsers that generalize to unseen databases. However, state-of-the-art neural models for cross-domain text-to-SQL are getting increasingly *more complex and domain specific.* In this work, we present **a simple sequential encoding** that effectively contextualizes the input question, DB schema and relevant DB cells using **generic sequence modeling architectures** including BERT and LSTM layers.

BRIDGE Encoder



① We represent the contextualization between **text, DB schema** and **the relevant DB cells** using a tagged sequence. ② We use BERT to capture both the lexical features and compositional dependencies in the sequence, which is enhanced with the usage of **"anchor texts"**. ③ The LSTM layers are necessary for performance improvement.

Schema-Consistency Guided Decoding

1. We use a pointer-generator decoder that generates output SQL queries as a sequence. At each decoding step, the decoder output one of the three types of tokens: (1) SQL reserved tokens, (2) a token from the input question and (3) a DB schema component (table/column) treated as an atomic token.
2. Simple sequential decoder does not guarantee the syntactical correctness or the consistency w.r.t. the DB schema in the output.

Observation:

1. In each SQL query or subquery, all fields can only be those in the tables mentioned in its *FROM* clause.

Approach:

1. Rewrite each SQL (sub)query so that the from clause is in the left-most position (execution order).
2. Mask all table fields in the beginning of decoding. Unmask fields of a table after the table appeared in *FROM* clause.

Experiment Results

Task & Dataset	# Examples	Resource	Annotation	Cross-domain
SPIDER (Yu et al., 2018c)	10,181	database	SQL	✓
Fully-sup. WIKISQL (Zhong et al., 2017)	80,654	single table	SQL	✓

Model	Dev	Test
Global-GNN (Bogin et al., 2019b) ♠	52.7	47.4
EditSQL + BERT (Zhang et al., 2019)	57.6	53.4
GNN + Bertrand-DR (Kelkar et al., 2020)	57.9	54.6
IRNet + BERT (Guo et al., 2019)	61.9	54.7
RAT-SQL v2 ♠ (Wang et al., 2019)	62.7	57.2
RYANSQL + BERT$_L$ (Choi et al., 2020)	*66.6*	58.2
RYANSQL v2 + BERT$_L$ ⋄	*70.6*	*60.6*
RAT-SQL v3 + BERT$_L$ ♠ (Wang et al., 2019)	*69.7*	*65.6*
BRIDGE (k = 1) (ours) ♠ ♡	65.3	–
BRIDGE (k = 2) (ours) ♠ ♡	65.5	59.2

Exact set match on the Spider dev and test sets, compared to other top-performing approaches on the leaderboard as of June 1st, 2020.

Model	Dev		Test	
	EM	EX	EM	EX
SQLova (Hwang et al., 2019)	81.6	87.2	80.7	86.2
X-SQL (He et al., 2019b)	83.8	89.5	83.3	88.7
HydraNet (Lyu et al., 2020)	83.6	89.1	83.8	89.2
BRIDGE +B$_L$ (k = 2) ♠	**85.1**	**91.1**	**84.8**	**90.4**
SQLova+EG (Hwang et al., 2019)	84.2	90.2	83.6	89.6
BRIDGE +B$_L$ (k = 2)+EG ♠	86.1	**92.5**	85.8	91.7
X-SQL+EG (He et al., 2019b)	86.2	92.3	86.0	91.8
HydraNet+EG (Lyu et al., 2020)	**86.6**	92.4	**86.5**	**92.2**

Exact match (EM) and Execution accuracy (EX) on the WikiSQL dev and test sets, compared to other top-performing models on the leaderboard as of August 20th, 2020.

Ablation Study

Model	Exact Set Match (%)	
	Mean	Max
BRIDGE (k = 2)	65.8 ± 0.8	66.9
- SC-guided decoding	65.4 ± 0.7	66.3 (**-0.6**)
- static SQL check	64.8 ± 0.9	65.9 (**-1.0**)
- execution order	64.2 ± 0.1	64.3 (**-2.6**)
- table shuffle & drop	63.9 ± 0.3	64.3 (**-2.6**)
- anchor text	63.3 ± 0.6	63.9 (**-3.0**)
- BERT	17.7 ± 0.7	18.3 (**-48.6**)

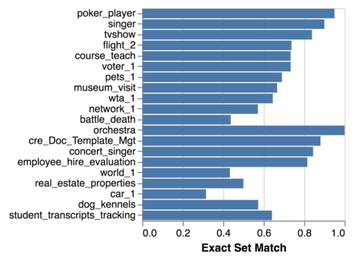
BRIDGE ablation on the Spider dev set measured by exact set match of each model variations averaged over three different random seeds.

Performance by SQL Complexity

Model	Easy	Medium	Hard	Ex-Hard	All
count	250	440	174	170	1034
Dev					
BRIDGE (k = 2) ♠	**88.4**	68	51.7	39.4	65.5
RAT-SQL v3+B$_L$ ♠	86.4	**73.6**	**62.1**	**42.9**	**69.7**
Test					
BRIDGE (k = 2) ♠	80	62	51	35.6	59.2
IRNet+B	77.2	58.7	48.1	25.3	54.7
RAT-SQL v3+B$_L$ ♠	**83.0**	**71.3**	**58.3**	**38.4**	**65.6**

Spider dev set performance broken down by SQL complexity.

Performance by Database



Conclusion

1. BRIDGE is competitive on Spider and WikiSQL.
2. The model can be further improved using a better cell value linker.
3. We may not be able to completely get rid of modeling structure and compositionality explicitly.
4. We have on going work further improving BRIDGE. BRIDGE is integrated in a live demo. http://naturalsql.com

Contact
- **Victoria Lin (corresponding author) xilin@salesforce.com**
- Caiming Xiong, cxiong@salesforce.com
- Richard Socher, rsocher@salesforce.com

Salesforce Reference
- Photon: A Robust Cross-Domain Text-to-SQL System. ACL Demo 2020. Jichuan Zeng*, Xi Victoria Lin*, Steven C.H. Hoi, Richard Socher, Caiming Xiong, Michael Lyu, Irwin King.
- Live Demo: http://naturalsql.com