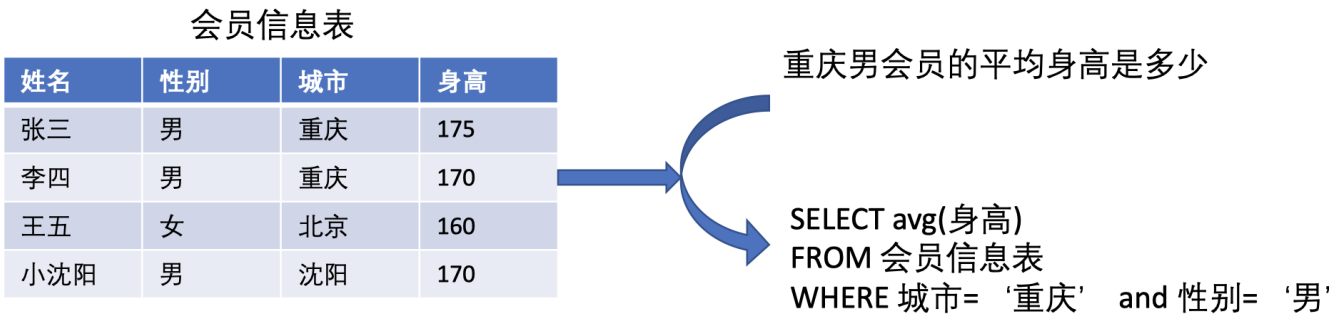
What is the Text2SQL task?

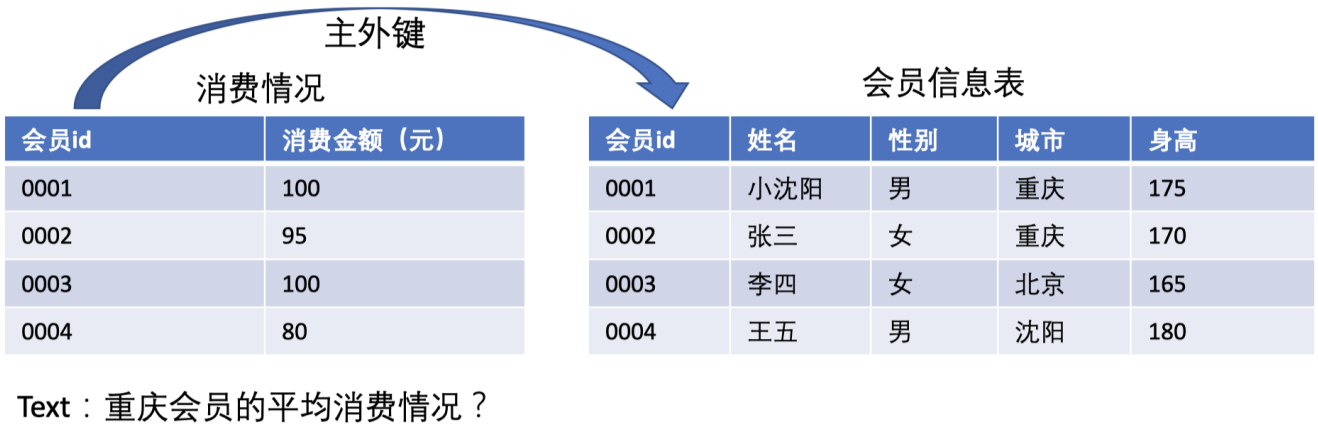
- Text2SQL，Text to SQL, also called NL2SQL
- Convert the user's natural language into executable SQL given a database (Context)
- A subtask of Semantic parsing, similar to machine reading comprehension QA tasks, but with more complex questions.



Text2SQL Classification

Sort by the number of tables in the database (Context):

- There is only one table in the database: single table Text2SQL(TableQA)
- There are multiple tables in the database: Text2SQL across tables, the primary and foreign key connections of the tables should be considered



Text：重庆会员的平均消费情况？

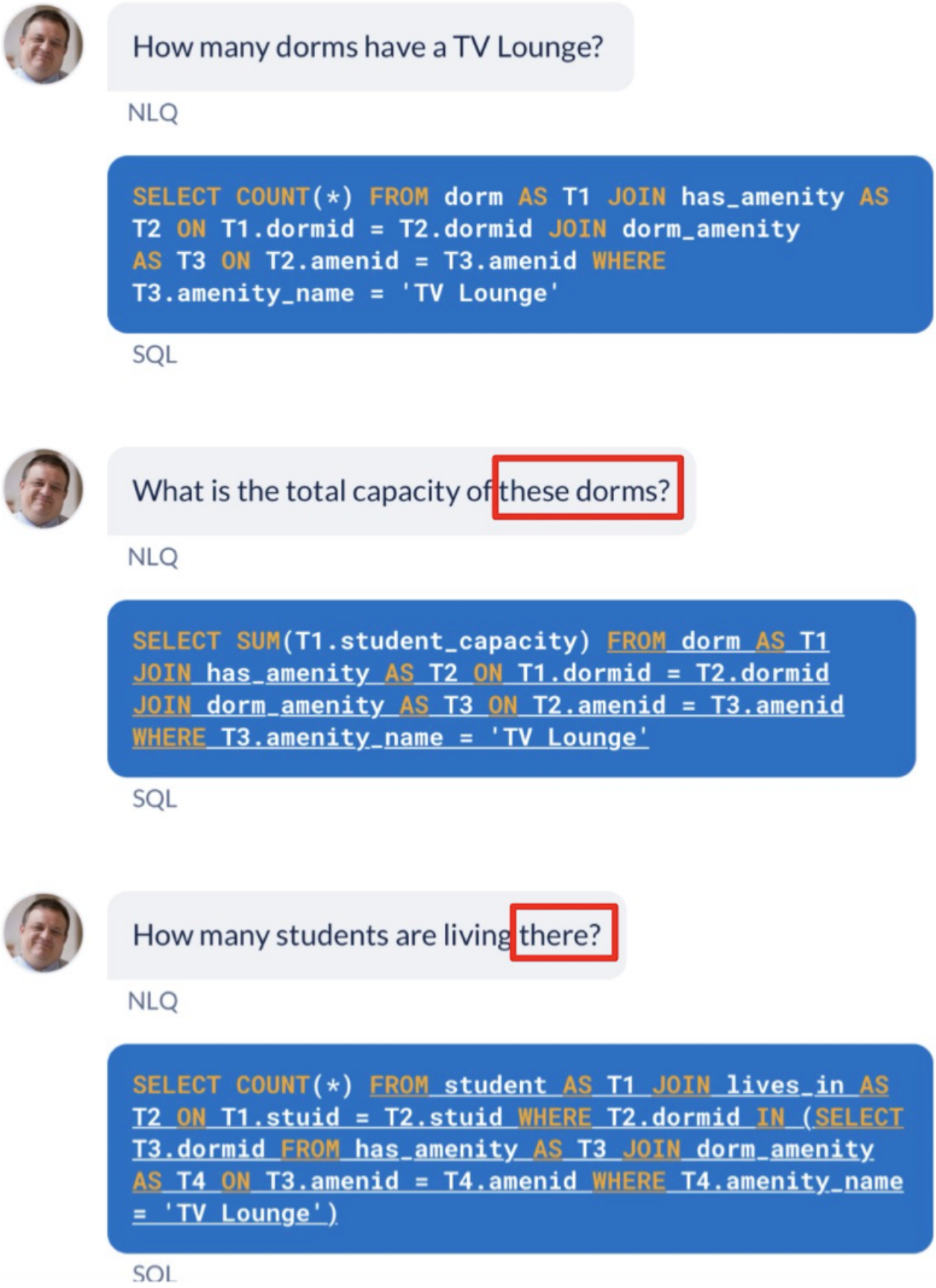
Sort by data field:

- Single Domain Text2SQL: Use on only one domain
- Cross-domain Text2SQL:
- Data from different fields can be used in the Text2SQL system, such as Finance, HR
- The system needs to "understand" the language style in different fields e.g. "Who wrote "A Dream of Red Mansions" (author)" "How much money did customers in Beijing spend (consumption amount)"

Sort by interaction:

- Multiple rounds of Text2SQL: Conversational, the current output SQL is related to all historical Text

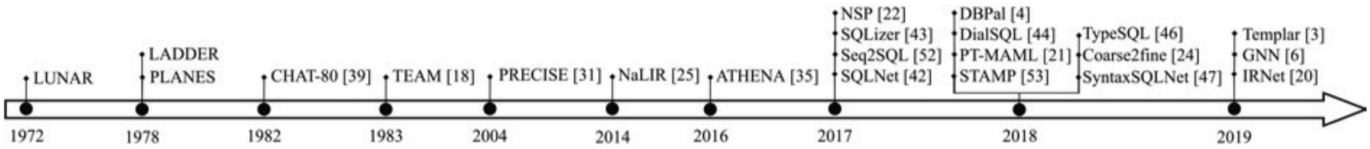
- One round of Text2SQL: one question and one answer



Milestone

Before deep learning: rule-based, there exists a problem of semantic generalization

- 1980s: Manually define rules, generate SQL from the schema of the database
- 2000s: Rule Template + Simple Statistical Parsing Model, then Deep Learning
- 2017-present: Deep learning end-to-end systems, large-scale pre-trained models



Kim, Hyeonji, et al. "Natural language to SQL: Where are we today?." *Proceedings of the VLDB Endowment* 13.10 (2020): 1737-1750.

Common Benchmark

数据集	发布机构	领域	是否跨表	是否多轮
WikiSQL	Salesforce	多领域	否	否
Spider	Salesforce、耶鲁	多领域	是	否
SParC	Salesforce、耶鲁	多领域	是	是
CoSQL	Salesforce、耶鲁	多领域	是	是（包含用户问询，确认等意图）
CSpider**	西湖大学	多领域	是	否
TableQA**	追一科技	多领域	否	否
DuSQL**	百度	多领域	是	否
CHASE**	西安交大、MSRA	多领域	是	是

** 标记表示为中文数据集

Frontier progress of Text2SQL

X-SQL

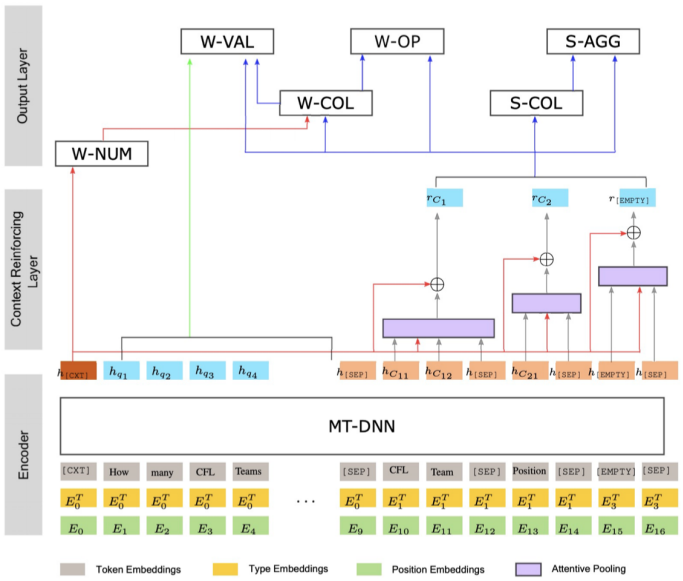
- Microsoft, Text2SQL limited to a single table
- Typical method on WikiSQL
- The SQL statement is relatively simple, and the different components of the SQL can be predicted separately (multi-task)

20到30岁员工的平均身高是多少？⟹

```
{
    "agg":[
        "avg"
    ],
    "cond_conn_op":"and",
    "sel":[
        "身高"
    ],
    "table_name":"人员信息表",
    "conds":[
        [
            "年龄",
            ">=",
            "20"
        ],
        [
            "年龄",
            "<",
            "30"
        ]
    ]
}
```

⟹

Select avg(身高)
From 人员信息表
Where 年龄>=20 and 年龄<30



$$p^{\text{S-COL}}(C_i) = \text{SOFTMAX}\left(W^{\text{S-COL}} r_{C_i}\right)$$

$$p^{\text{S-AGG}}(A_j|C_i) = \text{SOFTMAX}\left(W^{\text{S-AGG}}[j,:] r_{C_i}\right)$$

$$p^{\text{W-COL}}(C_i) = \text{SOFTMAX}\left(W^{\text{W-COL}} r_{C_i}\right)$$

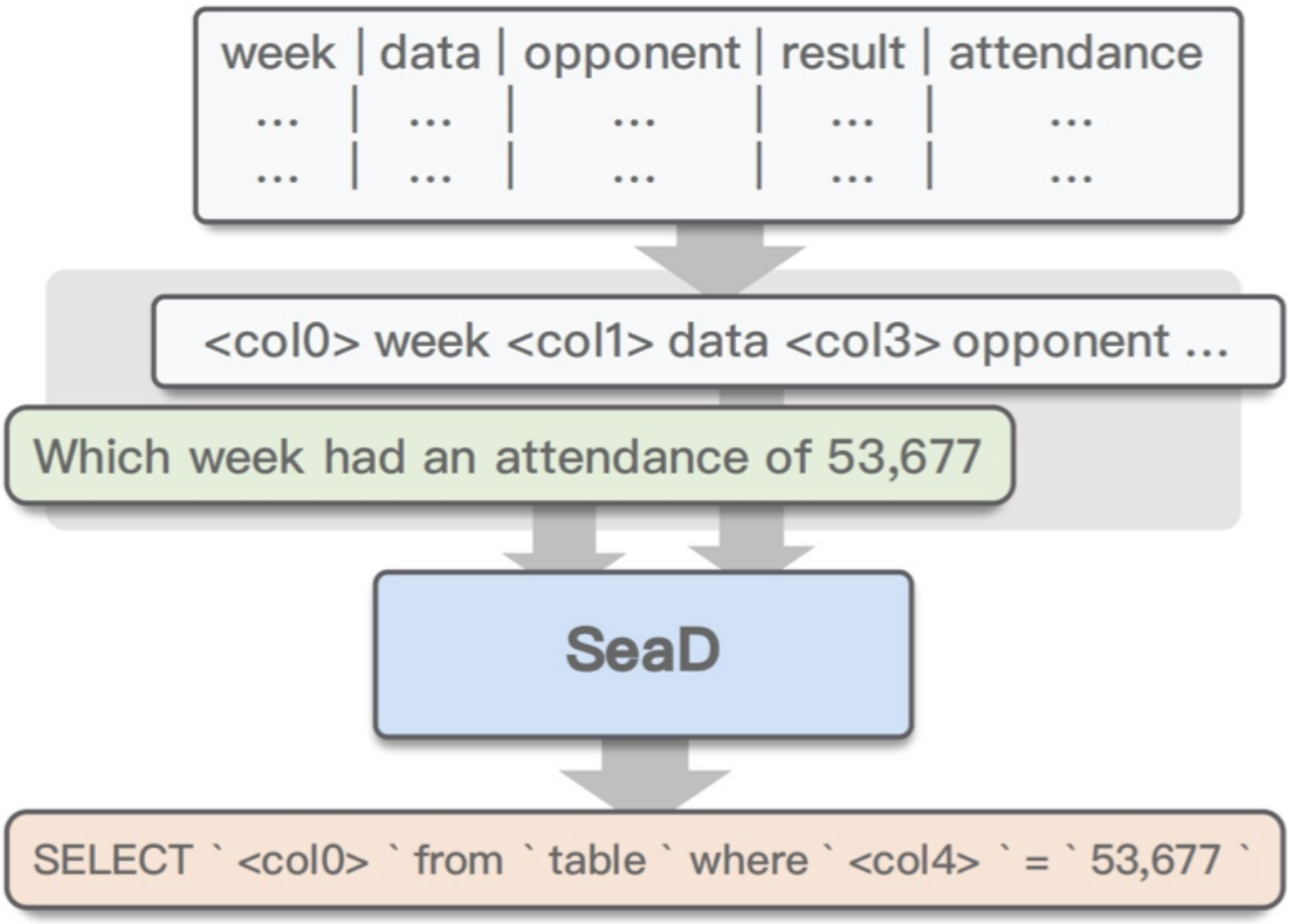
$$p_{\text{start}}^{\text{W-VAL}}(q_j|C_i) = \text{SOFTMAX}\ g\left(U^{\text{start}} h_{q_j} + V^{\text{start}} r_{C_i}\right)$$

$$p_{\text{end}}^{\text{W-VAL}}(q_j|C_i) = \text{SOFTMAX}\ g\left(U^{\text{end}} h_{q_j} + V^{\text{end}} r_{C_i}\right).$$

抽取式value预测

SEAD

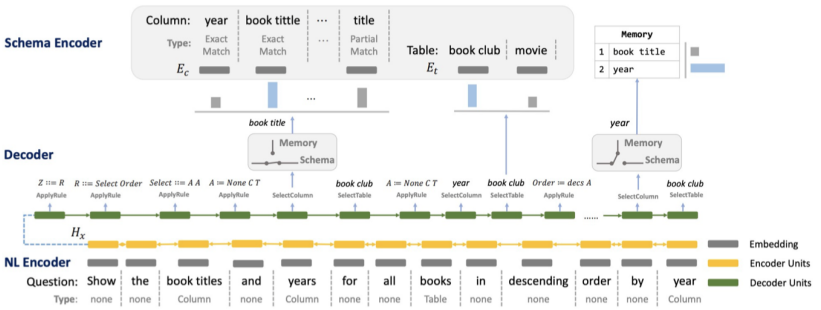
- Ant Financial
- WikiSQL is currently SOTA
- Seq2seq model directly generates SQL
- Data augmentation
- 1. Addition, deletion and modification of column names
- 2. Entity disorder in Text and SQL

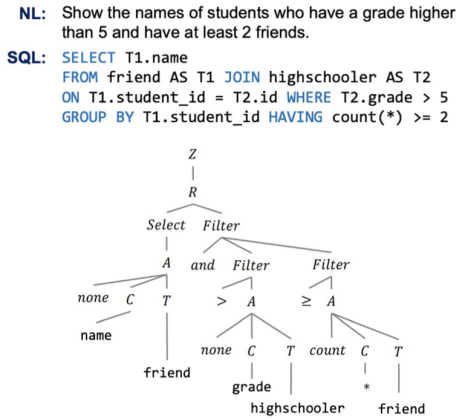


Model	Dev		Test	
	Acc_{lf}	Acc_{ex}	Acc_{lf}	Acc_{ex}
Bart	81.4	87.1	81.2	86.8
Bart_{ptr}	82.8	88.6	82.4	88.3
Bart_{ptr} + infilling	82.8	88.7	82.7	88.6
SeaD (Shuffle-only)	83.5	89.0	83.2	88.8
SeaD (Erosion-only)	84.2	89.6	84.1	89.4
SeaD	84.6	90.2	84.7	90.1

IRNET

- MSRA, the design of the intermediate language SemQL
- Coarse-to-fine prediction: first predict the skeleton of the SQL, and then predict the entities in the SQL





解码步骤

$$p(y|x,s) = \prod_{i=1}^{T} p(a_i|x,s,a_{<i}),$$

门机制控制列的选取

$$p(a_i = \text{SELECTCOLUMN}[c]|x,s,a_{<i}) = p(\text{MEM}|x,s,a_{<i})p(c|x,s,a_{<i},\text{MEM}) + p(\text{S}|x,s,a_{<i})p(c|x,s,a_{<i},\text{S})$$

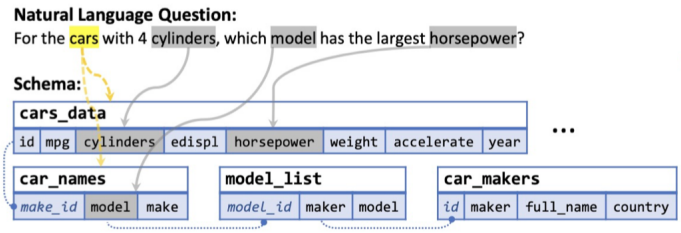
$$p(\text{MEM}|x,s,a_{<i}) = \text{sigmod}(\boldsymbol{w}_m^{\top}\boldsymbol{v}_i)$$

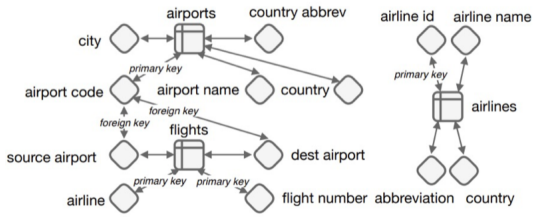
$$p(\text{S}|x,s,a_{<i}) = 1 - p(\text{MEM}|x,s,a_{<i})$$

RATSQL

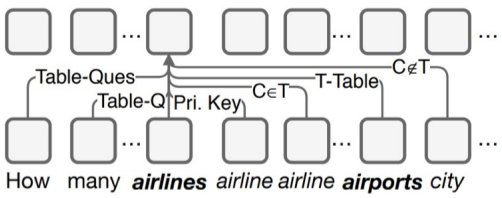
- Microsoft, Complex Text2SQL Classic Program
- Embedding of Schema Linking, Schema Graph information: Relation-Aware-Transformer
- Tree-Structured decoding, tree decoding, generates an abstract syntax tree (AST) corresponding to SQL
- The input of the decoder LSTM, the relevant nodes are introduced in a targeted manner, and the information is enhanced

Relation



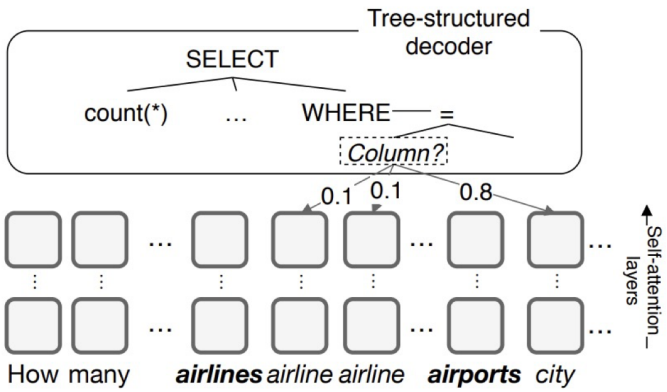


Relation embedding



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

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



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

上一时刻action的embedding

父节点的隐向量

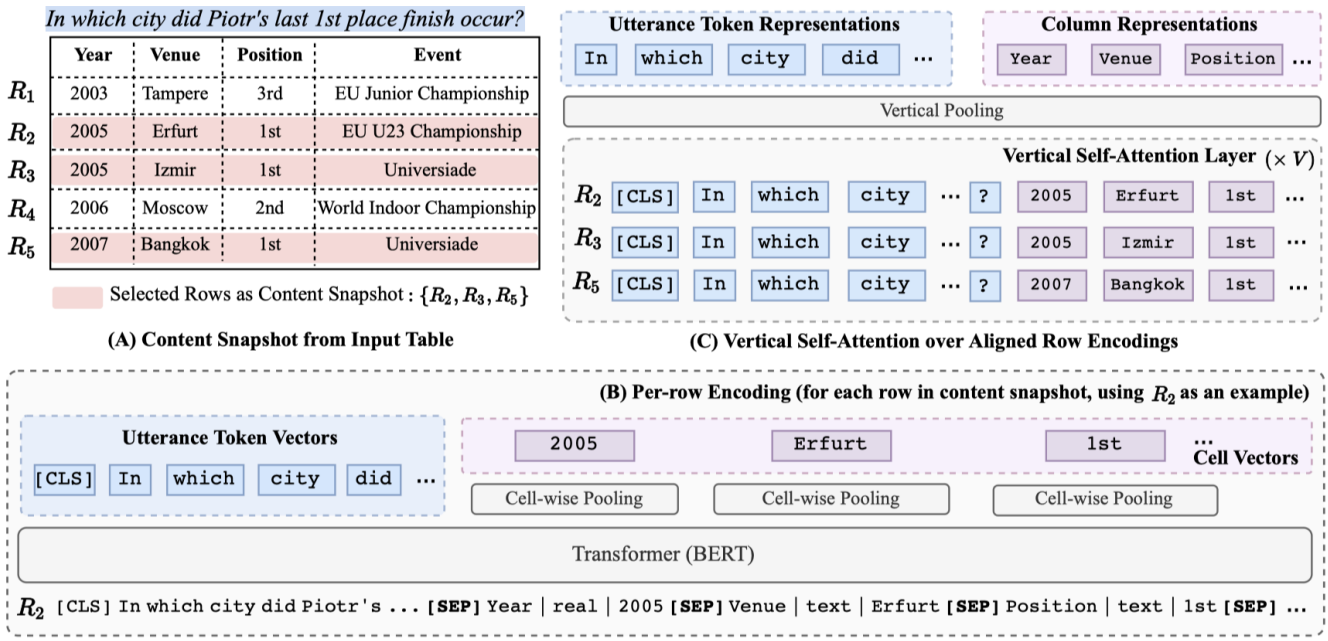
节点类型的embedding

上一时刻h在encoder端的attention向量

父节点action的embedding

Pretraining of Text2SQL

模型名称	数据来源	预训练方法	适用场景
TABERT	爬取English Wikipedia、WDC WebTable Corpus中table、table周围的文字	MLM、 根据列值预测列名、 采样的列值恢复	单轮
Grappa	从已有数据中挖掘SQL模板，替换其中的表、列名称，生成新数据	MLM、 预测每个列对应的操作、 select or where or group...	单轮
GAP	在已有的text2sql、data2text数据上训练模型，用模型再去产生更多样本	MLM、判断列是否在text中出现、替换列名并回复、 text2sql任务本身	单轮
SCORE	挖掘多轮场景下的SQL模板，替换其中的表、列名称，生成新数据	MLM、 预测每个列对应的操作、 预测两轮之间的SQL差异	多轮



Large-scale pretrained models

- T5-3B model performs outstanding on Spider and CoSQL datasets
- Large-scale pre-trained models have strong few-shot capabilities in SQL generation scenarios
- The larger the model, the higher the trend of accuracy

System	Development		Test	
	EM%	EX%	EM%	EX%
BRIDGE v2 + BERT (ensemble)† (Lin et al., 2020)	71.1	70.3	67.5	68.3
SmBoP + GraPPa† (Rubin and Berant, 2021)	74.7	75.0	69.5	71.1
RATSQL + GAP† (Shi et al., 2021)	71.8	-	69.7	-
DT-Fixup SQL-SP + RoBERTa† (Xu et al., 2021)	75.0	-	70.9	-
LGESQL + ELECTRA† (Cao et al., 2021)	75.1	-	**72.0**	-
T5-Base (Shaw et al., 2021)	57.1	-	-	-
T5-3B (Shaw et al., 2021)	70.0	-	-	-
T5-Base (ours)	57.2	57.9	-	-
T5-Base+PICARD	65.8	68.4	-	-
T5-Large	65.3	67.2	-	-
T5-Large+PICARD	69.1	72.9	-	-
T5-3B (ours)	69.9	71.4	-	-
T5-3B+PICARD	74.1	76.3	-	-
T5-3B†	71.5	74.4	68.0	70.1
T5-3B+PICARD†	**75.5**	**79.3**	**71.9**	**75.1**

Some thinking

- There is still a lot to improve in the accuracy of complex SQL
- The accuracy of cross-table SQL is still very low
- The understanding of the semantics of the table itself is still unexplored
- Promising directions: intermediate language design, large-scale language models

The difference between research and implementation

- The tables in the database are organized differently
- Differences in model training data
- The complexity of the data itself:
- Complex relationship of column values
- Column-to-column relationship

Product Perspective Considerations

- In which form to design the product: conversational or search-based (single round, multiple rounds)
- Platform-based products or individual optimization in vertical fields (single-domain or cross-domain)
- How to let the business side access the database at low cost (what kind of data, can the algorithm handle it)
- The user does not know the database schema, how to let the user know how to "ask"
- Only do data query, or integrate a complete set of solutions such as "data analysis, intelligent insight, report display, recommendation system, and early warning"?