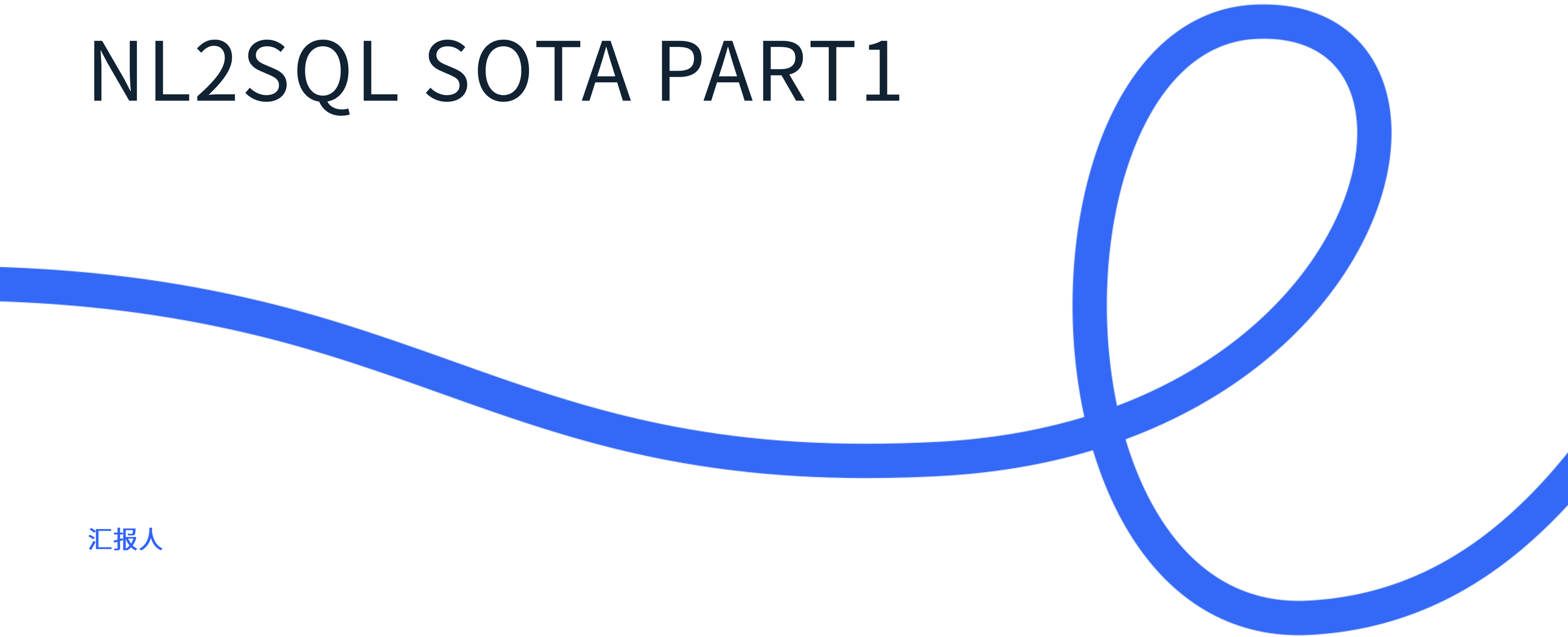
NL2SQL SOTA PART1



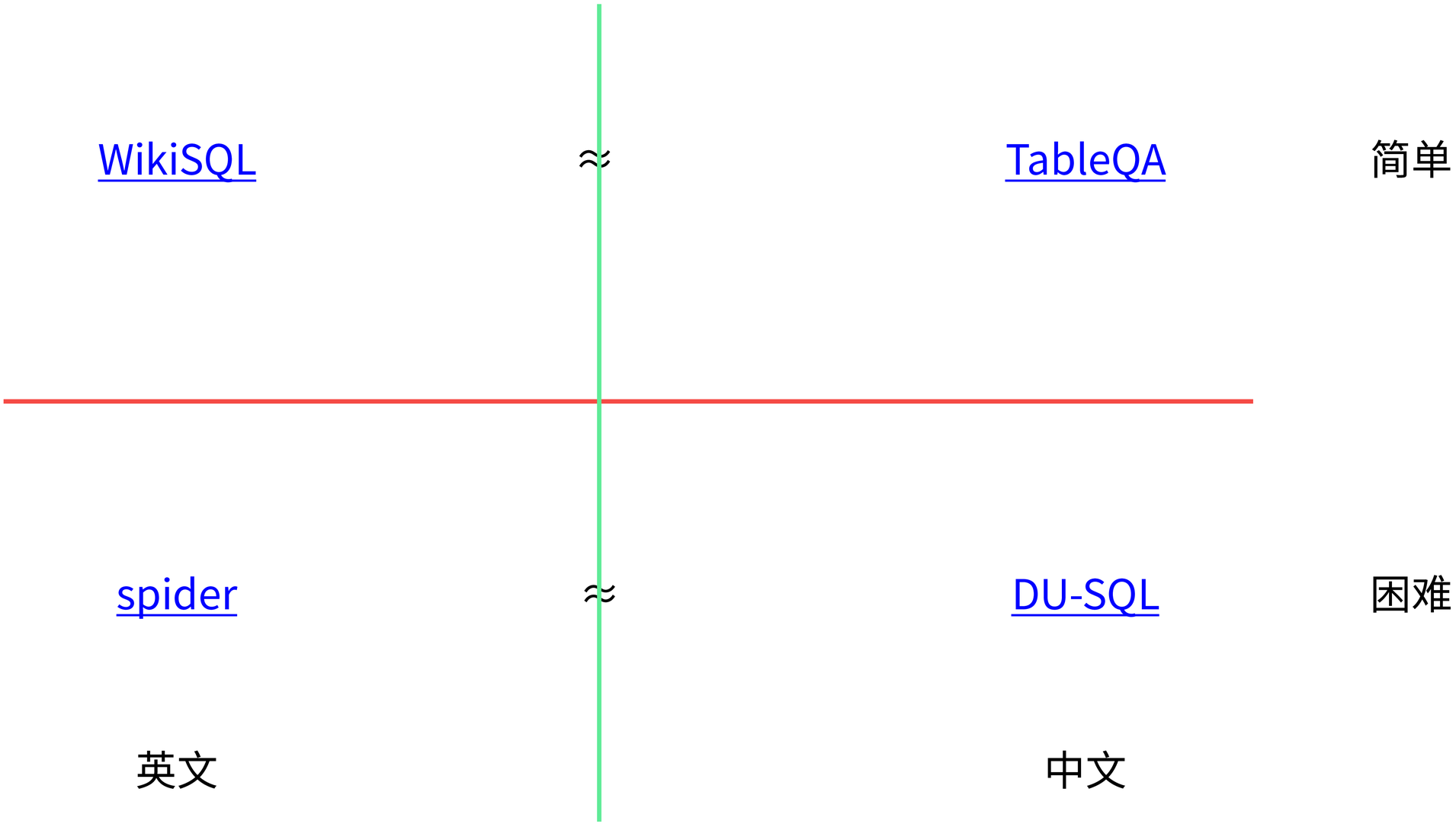
汇报人

目录

01

数据集

数据集	问题和SQL对	数据库数量	单/多领域	表格数/每数据库	单/多轮	语言
ATIS	5280	1	单领域	32	单轮	英文
GeoQuery	877	1	单领域	6	单轮	英文
Scholar	817	1	单领域	7	单轮	英文
Academic	196	1	单领域	15	单轮	英文
IMDB	131	1	单领域	16	单轮	英文
Yelp	128	1	单领域	7	单轮	英文
Advising	3898	1	单领域	10	单轮	英文
Restaurants	378	1	单领域	3	单轮	英文
WikiSQL	80657	26521	多领域	1	单轮	英文
NL2SQL	49974	5291	多领域	1	单轮	中文
Spider	10181	200	多领域	5.1	单轮	英文
CSpider	9691	166	多领域	5.3	单轮	中文
SParC	4298	200	多领域	5.1	多轮	英文
CoSQL	3007	200	多领域	5.1	多轮	英文


WikiSQL
≈
TableQA
简单
spider
≈
DU-SQL
困难
英文
中文

WikiSQL VS TableQA

	WikiSQL	TableQA
语言	英文	中文
领域	多领域	多领域
轮次	单轮	单轮
每数据库/表格数	单表	单表
query数	80654	64891
sql数	77840	20311
table数	26531	6029
select字段数	1	1 or 2
where条件数	以单个为主	以多个为主
where条件操作	AND	AND 或 OR
value值是否存在	全部存在	大部分存在

WikiSQL&TableQA主要挑战

1. 怎么融合nl question和table schema

1. 怎么保证生成sql能准确执行

1. 怎么使用好pre-trained语言模型

02

方法

Key idea

重新发明M-SQL

直观上看

Chinese version

商户类型	地区	区域	商户名称	地址
百货	广西	防城港	防城港港口区家惠超市	兴港大道 95-1 号
百货	广西	南宁	青秀南城百货	民族大道 64 号
百货	广西	南宁	白沙南城百货公司	南宁市白沙大道 20 号
……	……	……	……	……

QUERY：青秀南城百货有限公司在南宁的哪个位置？

SQL：SELECT 地址 WHERE 商户名称=青秀南城百货 AND 区域=Nanning

Answer：民族大道 64 号

one sample

```
{
     "table_id": "a1b2c3d4", # 相应表格的id
     "question": "世茂茂悦府新盘容积率大于1，请问它的套均面积是多少？", # 自然语言问句
    "sql":{ # 真实SQL
        "sel": [7], # SQL选择的列
        "agg": [0], # 选择的列相应的聚合函数，'0'代表无
        "cond_conn_op": 0, # 条件之间的关系
        "conds": [
            [1,2,"世茂茂悦府"], # 条件列，条件类型，条件值，col_1 == "世茂茂悦府"
            [6,0,1]
        ]
    }
}
```

其中，SQL的表达字典说明如下：

```
op_sql_dict = {0:">", 1:"<", 2:"==", 3:"!="}
agg_sql_dict = {0:"", 1:"AVG", 2:"MAX", 3:"MIN", 4:"COUNT", 5:"SUM"}
conn_sql_dict = {0:"", 1:"and", 2:"or"}
```



```
{
    "id":"a1b2c3d4", # 表格id
    "name":"Table_a1b2c3d4", # 表格名称
    "title":"表1：2019年新开工预测 ", # 表格标题
    "header":[ # 表格所包含的列名
        "300城市土地出让",
        "规划建筑面积(万㎡)",
        ……
    ],
    "types":[ # 表格列所相应的类型
        "text",
        "real",
        ……
    ],
    "rows":[ # 表格每一行所存储的值
        [
            "2009年7月-2010年6月",
            168212.4,
            ……
        ]
    ]
}
```

schema

input

NL question: e.g. 哪些楼盘平均售价低于2万或者容积率小于2

{"rows": [["尚东国际名园", 41175.0, 1.29, "7000", "持平"], ["中瑞公寓", 18547.0, 1.2, "毛坯", "高于"], ["宝华盛世花园", 11272.0, 1.23, "毛坯", "高于"]], "name": "Table_c9896f30332111e98b80542696d6e445", "title": "04表5：上海2012年5月新盘周边楼盘去化及价格比较情况（单位：元/平）", "header": ["楼盘", "楼盘均价", "楼盘容积率", "楼盘装修标准", "价格比较"], "common": "资料来源：中投证券研究所，易居，网上房地产", "id": "c9896f30332111e98b80542696d6e445", "types": ["text", "real", "real", "text", "text"]}

output

"sql": {"agg": [0], "cond_conn_op": 2, "sel": [0], "conds": [[1, 1, "20000"], [2, 1, "2"]]}}

idea 1 规则填槽方法（冷启动）

1. 提取NL中的实体列表
2. 通过比较schema header，筛选实体列表
3. 规则+依存分析：归属实体到语义块（填槽）

idea 2 翻译方法（seq2sql）

1. 借助encoder-decoder结构
2. 将sql划分为agg/select/where
3. 存在order-issue问题

idea 3 填槽方法（sqlNet）

1. 借助encoder-decoder结构，转化为seq2set
2. 转化select/where填槽问题
3. 引入type
4. 解码目标变为树结构

idea 4 填槽问题 分类方法

1. 为了更好使用PTMs，放弃encoder-decoder结构，只用encoder
2. 将填槽问题转化为分类问题
3. 使用预训练语言模型
4. 通过多任务联合训练，提高效果

input

NL question: e.g. 哪些楼盘平均售价低于2万或者容积率小于2

{"rows": [["尚东国际名园", 41175.0, 1.29, "7000", "持平"], ["中瑞公寓", 18547.0, 1.2, "毛坯", "高于"], ["宝华盛世花园", 11272.0, 1.23, "毛坯", "高于
"]], "name": "Table_c9896f30332111e98b80542696d6e445", "title": "04表5：上海2012年5月新盘周边楼盘去化及价格比较情况（单位：元/平）"
"header": ["楼盘", "楼盘均价", "楼盘容积率", "楼盘装修标准", "价格比较"], "common": "资料来源：中投证券研究所，易居，网上房地产", "id":
"c9896f30332111e98b80542696d6e445", "types": ["text", "real", "real", "text", "text"]}

output

"sql": {"agg": [0], "cond_conn_op": 2, "sel": [0], "conds": [[1, 1, "20000"], [2, 1, "2"]]}}

NL分类问题（sel任务举例）

inputs → tensor → softmax

NL question

Table schema

多个二分类

p("楼盘"),
p("楼盘均价"),
p("楼盘容积率"),
p("楼盘装修标准"),
p("价格比较")

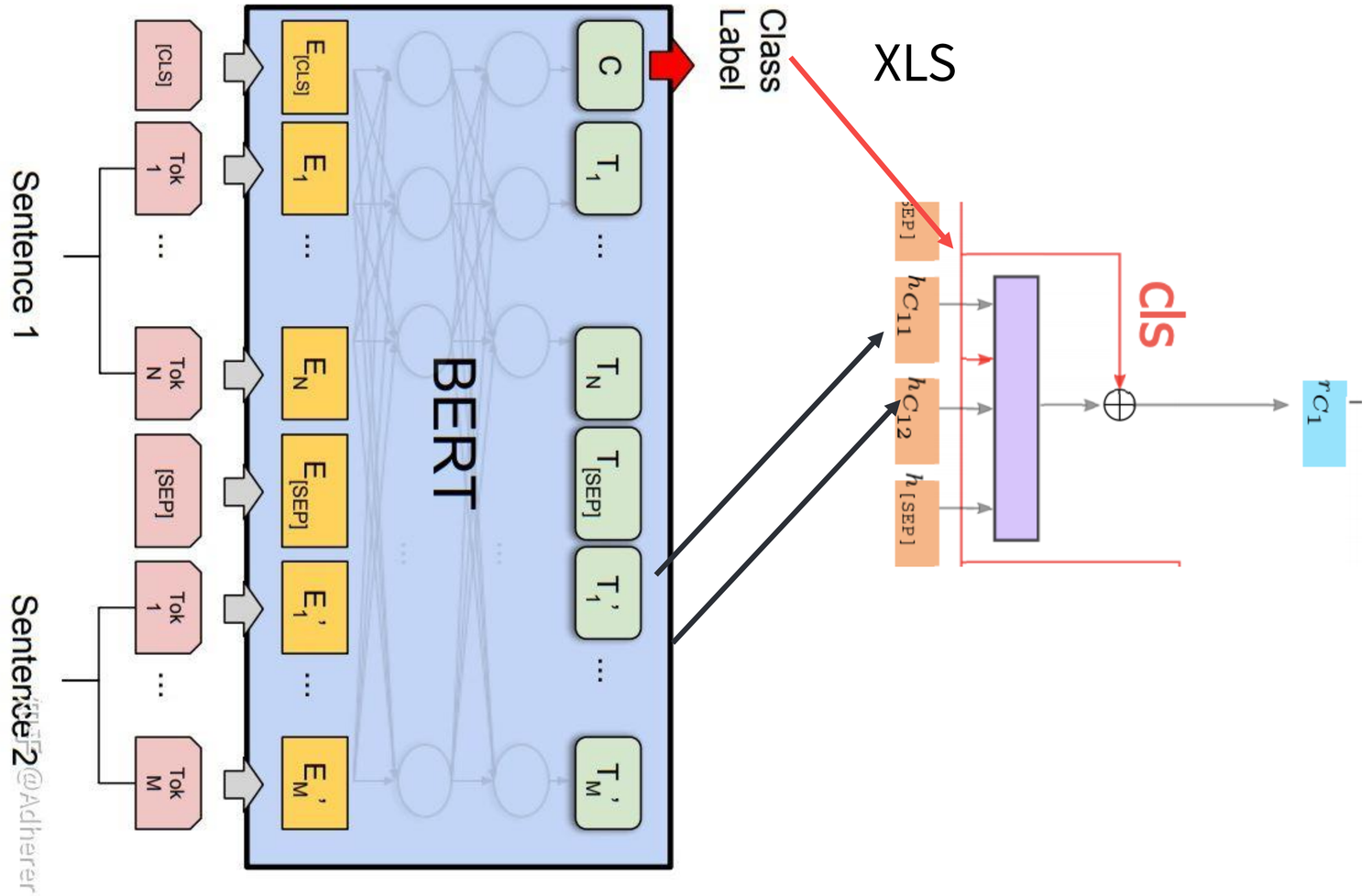
$\approx$ p(s-col|question,schema)

NL分类问题（sel任务举例）

NL question

inputs

Table schema



tensor

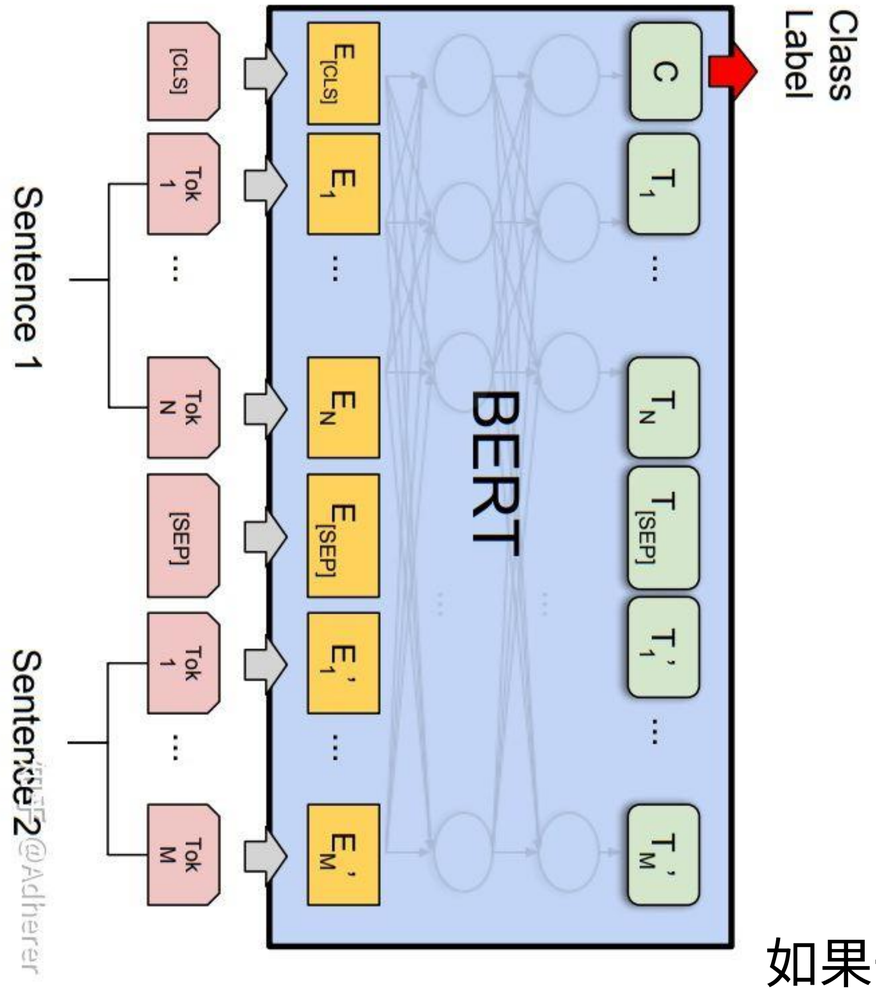
p(s-col|question,schema)

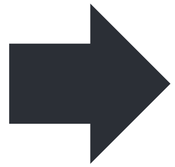
NL分类问题（sel任务举例）

NL question

inputs

Table schema

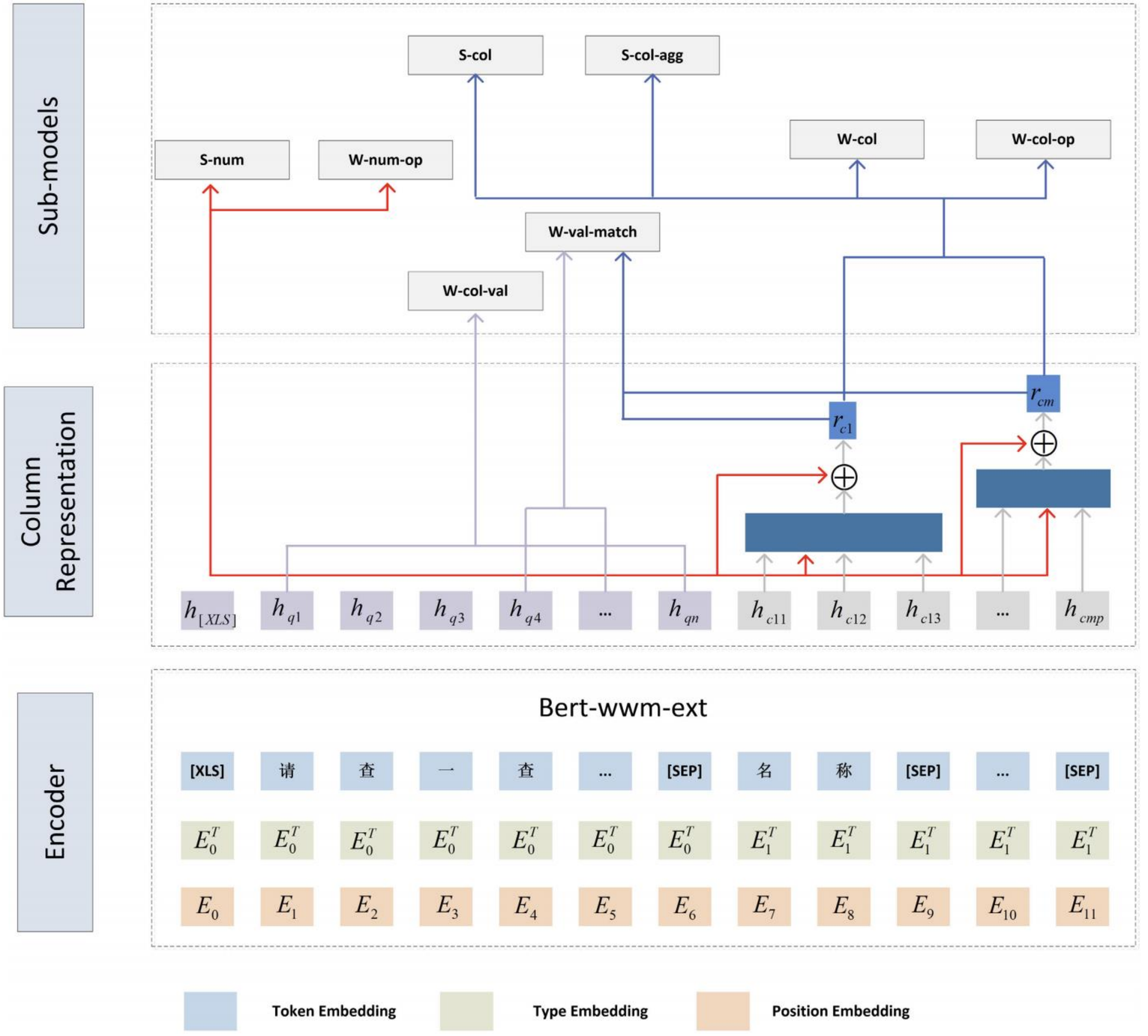




tensor

如果一次输入一个列，直接用cls即可，HydraNet即是此思路

1. column representation层和Encoder层参数共享
2. 将任务转化为8个子任务
3. 输入是token embedding、type embedding、position embedding
4. type embedding表示输入的类型（是query还是headers）



8个子任务（s-num、w-num-op)

子任务	输入	预测	输出
s-num	cls或xls	select的列的数目	1和2的概率值
w-num-op	cls或xls	where中条件数	

Subtask S-num predicts the number of the selected columns. The prediction set is [1, 2]. Subtask W-num-op predicts the relationship between the conditional columns and the number of the conditional columns. The prediction set is [“null-1”, “OR-1”, “AND-1”, “OR-2”, “AND-2”, “OR-3”, “AND-3”]. “null”, “OR” and “AND” represent the relationship between the conditional columns. “1”, “2” and “3” represent the number of the conditional columns. Two subtasks use the global information $h_{[XLS]}$ as the input. S-num is a two-class classification problem, and W-num-op is a seven-class classification problem. S-num and W-num-op are modelled as follows.

$$p_1 = sigmoid(W_1 h_{[XLS]}) \quad (6)$$

$$p_2 = softmax(W_2 h_{[XLS]}) \quad (7)$$

8个子任务（s-col、s-col-agg、w-col、w-col-op）

子任务	输入	预测	输出
s-col	column representation	select的列	被选中的概率值
s-col-agg	column representation	select的列的agg运算	6分类概率值
w-col	column representation	where中的列	被选中的概率值
w-col-op	column representation	where中的列的操作符	4分类概率值

8个子任务（w-col-val、w-col-val-match）

子任务	输入	预测	输出	方法
w-col-val	NL question token embeddings	where条件中的值	每个token被选中的概率	0-1 labeling或BERT+CRF或bert+Bi-LSTM+CRF
w-col-val-match	column representation和NL question token embeddings	匹配where中列与值之间的对应关系	第i个列与val之间的关系	

the representation of the i-th conditional column. We assume that the start index and end index of the extracted value in the query are s and e. The match score about the extracted value and the i-th conditional column is as follows.

$$h_v = \frac{\sum_{i=s}^{e} h_{qi}}{l} \tag{13}$$

$$match_i = sigmoid(u \cdot tanh(W_8 h_v + W_9 rc_i)) \tag{14}$$

where h_v is the value representation. $match_i$ is the match score about the extracted value and the i-th conditional column. W_8, W_9 and u are learnable parameters. W_8 and $W_9 \in R^{d \times d}$. $u \in R^{1 \times d}$. l represents the length of the extracted value span.

评测指标

Logical-form accuracy(LX)：compare the generated SQL statement with the ground truth, and check whether they match each other

Execution accuracy(X)：execute the generated SQL statement and the ground truth to get the SQL query results, and check whether their results match each other.

评测结果

TABLE 1. The performance of various models on TableQA.

Model	Dev LX(%)	Dev X(%)	Dev MX(%)	Test LX(%)	Test X(%)	Test MX(%)
SQLNet [6]	61.28	66.20	63.74	61.42	67.24	64.33
Coarse2Fine [8]	72.98	76.89	74.94	72.61	76.71	74.66
MQAN [9]	75.66	79.21	77.44	74.84	78.75	76.80
SQLova [10]	81.39	85.26	83.33	81.71	85.76	83.74
X-SQL [11]	82.85	86.99	84.92	83.30	87.58	85.44
M-SQL(ours)	**89.13**	**91.86**	**90.50**	**89.31**	**92.13**	**90.72**
M-SQL-Ens(ours)	**90.54**	**93.40**	**91.97**	**90.49**	**93.31**	**91.90**

TABLE 2. The performance of sub-tasks on TableQA test data.

	S-num(%)	S-col(%)	S-col-agg(%)	W-num-op(%)	W-col(%)	W-col-op(%)	W-col-value(%)	Test LX(%)
M-SQL(ours)	99.50	97.82	98.91	97.45	98.50	99.10	96.95	89.31
M-SQL-Ens(ours)	99.55	98.36	98.91	97.68	99.09	99.27	97.00	90.49

一些细节（输入）

细节点	作用
输入：token ids、token_types、header ids	header ids用于区分哪些token是第一列、哪些token是第二列（论文中未说）
token_types:分为0、1而不是像x-sql中的3种类型	原因是3种类型不能收敛
关于padding	其实NL question和column都有，作用是对齐尺寸，方便后续子任务（论文中未说）
作者把列数据类型编码进了token ids	论文中未细说

M-SQL是嵌套式的网络结构，代码实现起来有点麻烦

Ablation

TABLE 3. The results of ablation study.

Model	Test LX(%)	Test X(%)	Test MX(%)
M-SQL	89.31	92.13	90.72
– BERT-wwm-ext + BERT-base	88.90	91.45	90.18
– [XLS] + [CLS]	88.90	91.63	90.27
– 2-type	89.13	91.81	90.47
– 2-type + 3-type	\	\	\
– enhance	88.81	91.63	90.22
– BERT-0/1 + BERT-CRF	87.85	90.63	89.24
– BERT-0/1 + BERT-BILSTM-CRF	87.99	90.63	89.31
– BERT-0/1 + BERT-pointer	88.90	91.81	90.36
– lr + rouge-L	85.90	88.49	87.20
– lr + svr	74.75	77.02	75.89
– lr + bayes	86.35	88.67	87.51

03

总结

&takeaway

总结

TableQA数据集的难度略低于我们当前HRBI数据集

SOTA方法的部分子任务准确度达到了工业级可用

当前askdata基于规则的方法在HRBI数据集上欠拟合

它山之石可以攻玉

https://bytedance.feishu.cn/docs/doccniRnDrYI2npYpPiIVmYRZ5g#wYnP58

Takeaway

m-sql网络结构代码(仅网络结构)：
https://code.byted.org/dp/qabot_mrc/blob/dev_nl2sql/nl2sql/m_sql_base.py

数据集(TableQA)&预训练模型（bert-wwm-ext）&相关的论文（x-sql、m-sql、tableqa、tapas、hydranet、rat-sql）都在hdfs地址：
/home/byte_dp_data_streaming/warehouse/data_intelligence/m-sql/

rat-sql代码：
https://github.com/microsoft/rat-sql　　百度NLP解决NL2sql的一篇分享文章

tapas代码:
https://github.com/google-research/tapas　　TaBERT

TaBERT的思路是预训练一个语言模型和table schema编码模型

TaBERT的思路是预训练一个语言模型和table schema编码模型

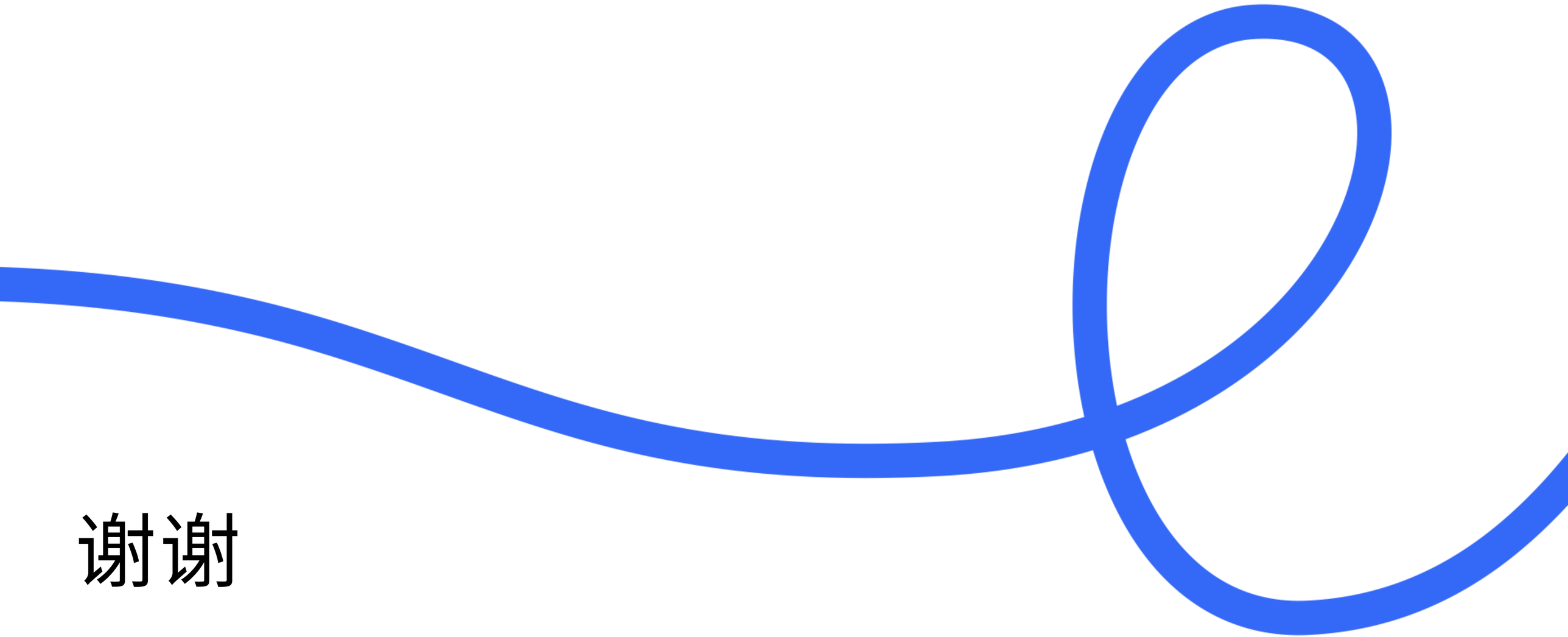
```
from table_bert import Table, Column

table = Table(
    id='List of countries by GDP (PPP)',
    header=[
        Column('Nation', 'text', sample_value='United States'),
        Column('Gross Domestic Product', 'real', sample_value='21,439,453')
    ],
    data=[
        ['United States', '21,439,453'],
        ['China', '27,308,857'],
        ['European Union', '22,774,165'],
    ]
).tokenize(model.tokenizer)

# To visualize table in an IPython notebook:
# display(table.to_data_frame(), detokenize=True)

context = 'show me countries ranked by GDP'

# model takes batched, tokenized inputs
context_encoding, column_encoding, info_dict = model.encode(
    contexts=[model.tokenizer.tokenize(context)],
    tables=[table]
)
```



谢谢