

打通人与结构化数据间壁垒

# 首届中文NL2SQL挑战赛

队名：Model S

# 术语

**cond\_conn\_op**

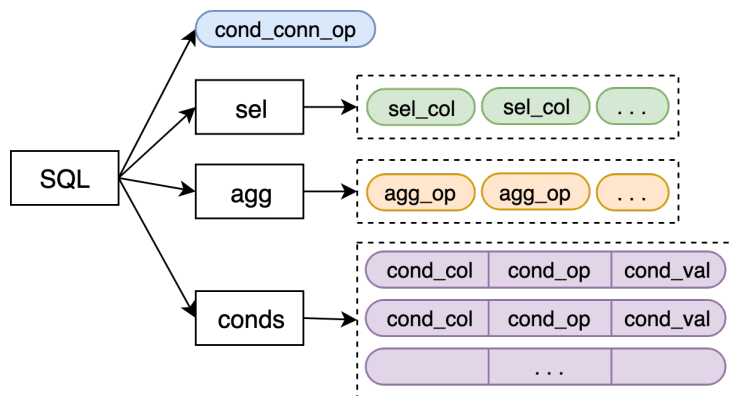
''	0
and	1
or	2

**cond\_op**

>	0
<	1
==	2
!=	3

**agg**

''	0
AVG	1
MAX	2
MIN	3
COUNT	4
SUM	5



sel: [ 1 ]

agg: [ 4 ]

cond\_conn\_op: 1

conds: [ [ 6, 2, '2016' ], [ 7, 2, '融资收购其他资产' ] ]

# 术语

## cond\_conn\_op

''	0
and	1
or	2

## agg

''	0
AVG	1
MAX	2
MIN	3
COUNT	4
SUM	5
NO_OP	6

## cond\_op

>	0
<	1
==	2
!=	3
NO_OP	4

证券代码	证券简称	最新收盘价	定增价除权后至今价格	增发价格	倒挂率	定增年度	增发目的
300148.SZ	天舟文化	4.69	12.48	16.34	37.58	2016.0	配套融资
300148.SZ	天舟文化	4.69	11.29	14.78	41.54	2016.0	融资收购其他资产

```

sel: [ 1 ]
agg: [ 4 ]
cond_conn_op: 1
conds: [ [ 6, 2, '2016' ], [ 7, 2, '融资收购其他资产' ] ]
    
```

Raw label



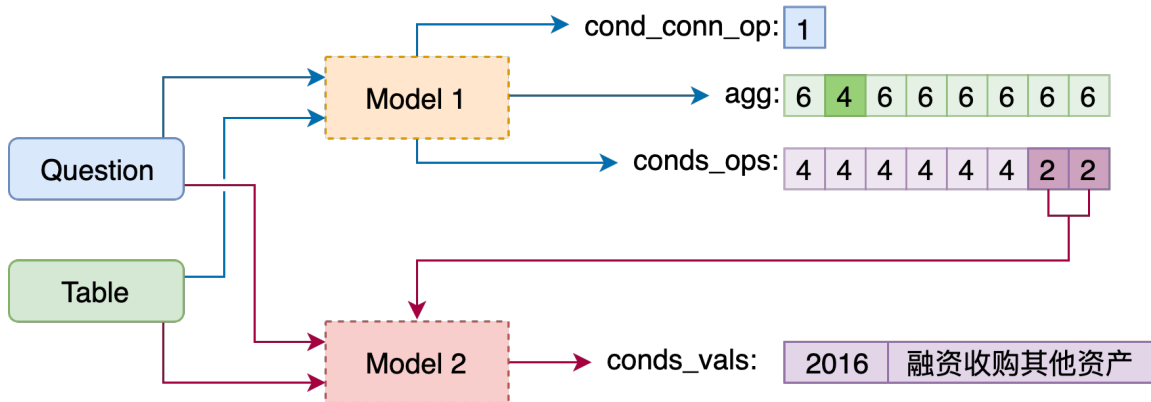
```

agg: [ 6, 4, 6, 6, 6, 6, 6, 6 ]
cond_conn_op: 1
conds_ops: [ 4, 4, 4, 4, 4, 4, 2, 2 ]
conds_vals: [ null, null, null, null, null, null, '2016', '融资收购其他资产' ]
    
```

New label

# 整体架构

将 SQL 拆解成 2 个部分，独立建模



\*Model 2 训练时用标签中真实的 conds\_ops，预测时接受 Model 1 输出的 conds\_ops

# Model 1

Question Tokenization:

CLS 2 0 1 5 年 哪 些 股 票 的 目 增 的 发 是 融 资 收 购 其 他 资 产 SEP

Header Tokenization:

TEXT 证 券 代 码 SEP

TEXT 证 券 简 称 SEP

REAL 最 新 收 盘 价 SEP

REAL 增 发 价 格 SEP



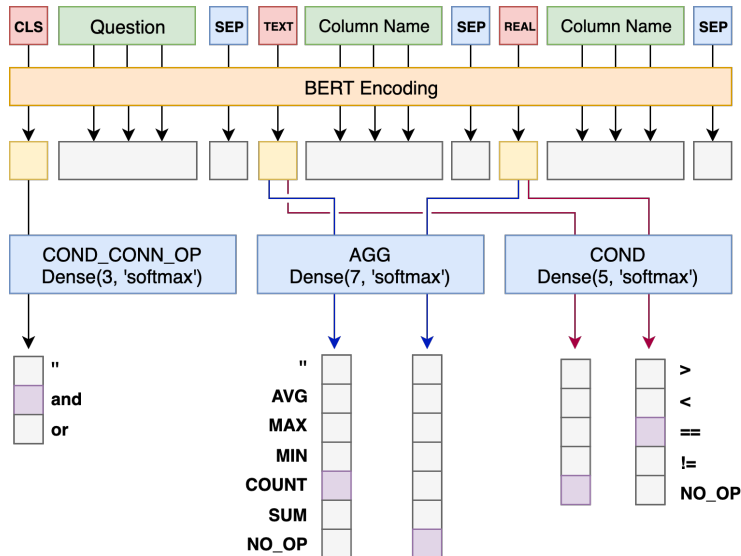
证券代码	证券简称	最新收盘价	定增价除权后至今价格	增发价格	倒挂率	定增年度	增发目的
300148.SZ	天舟文化	4.69	12.48	16.34	37.58	2016.0	配套融资
300148.SZ	天舟文化	4.69	11.29	14.78	41.54	2016.0	融资收购其他资产

Input Concatenation:

CLS Question SEP TEXT Column Name SEP REAL Column Name SEP ... ...

\* 不同类型的 Column 用不同的 token 标记

# Model 1



[CLS] 标记以及每个 column 对应的

[TEXT | REAL] 标记所在位置的向量表征，分别通过对应的 Dense Layer，输出相应的类别。

BERT Encoding 使用了哈工大讯飞联合实验室发布的 BERT-wwm, Chinese

三个 tasks 均使用 cross entropy error，整体的 loss 为各 task loss 之和

## Model 2

Question Tokenization:

CLS 2 0 1 5 年 哪 些 股 票 的 目 增 的 发 是 融 资 收 购 其 他 资 产 SEP

Condition Tokenization:

根据 Model 1 选择的 cond\_col, 枚举 cond\_op 与 cond\_val, 生成候选 ( cond\_col, cond\_op, cond\_val ) 组合

TEXT 类型的 cond\_val 生成自 Table:

增 发 目 的 = 融 资 收 购 其 他 资 产 SEP

增 发 目 的 = 配 套 融 资 SEP

REAL 类型的 cond\_val 生成自 Question:

定 增 年 度 > 2 0 1 5 SEP

定 增 年 度 < 2 0 1 5 SEP

定 增 年 度 = 2 0 1 5 SEP

Input Concatenation:

CLS Question SEP Candidate 1 SEP

CLS Question SEP Candidate 2 SEP

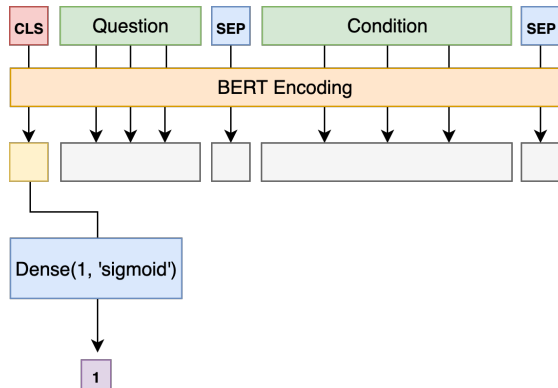
...

CLS Question SEP Candidate K SEP

\*原始数据中每个 query 会拆分为多条样本, 转换成多个二分类问题

\*REAL类型的 cond\_val 利用正则表达式写规则抽取

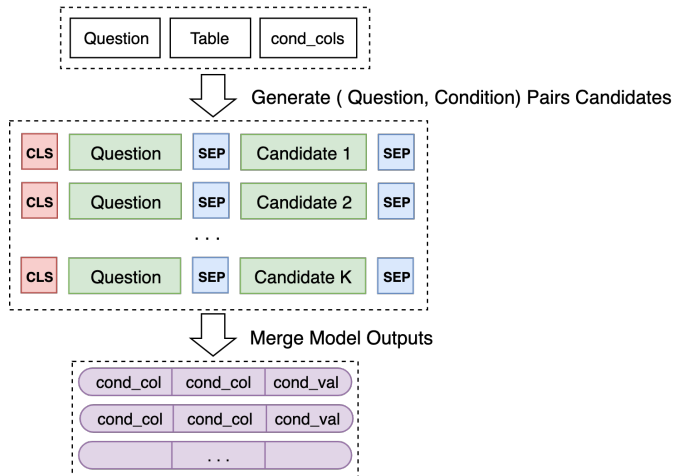
## Model 2



[CLS] 标记所在位置的向量表征，通过一层

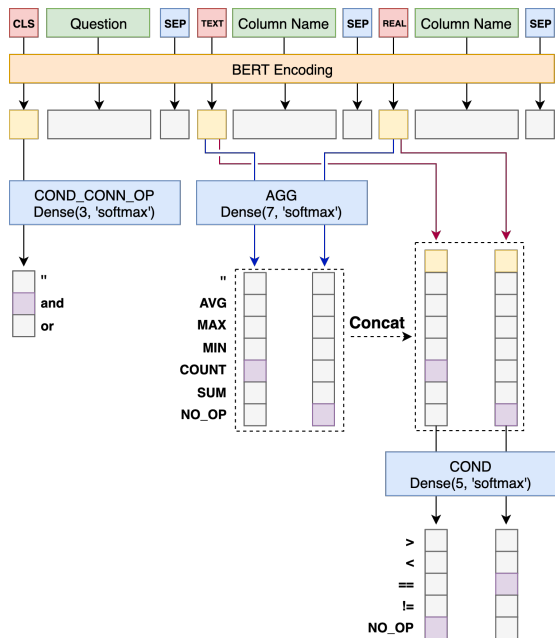
Dense Layer，输出 0 或者 1

BERT Encoding 使用了哈工大讯飞联合实验室发布的 BERT-wwm, Chinese





# 模型训练中的一些探索尝试



Model 1 `conds_ops` 的预测融入 `agg` 的信息

想法:

将模型对 `agg` 的输出拼接到 `conds_ops` 的输入中, 让模型输出 `conds_ops` 时能结合 `agg` 的信息。

Model 1 Total Accuracy:

0.855 -> 0.865

\*Model 1 Total Accuracy 指的是, 除了 `cond_val` 外所有预测都正确

# 模型训练中的一些探索尝试

调整 Model 1 Loss Weight

	loss_cond_conn_op	loss_agg	loss_conds_ops	Best Model 1 Total Accuracy
Epoch 1 - 25	1	1	1	$\approx 0.865$
Epoch 26 - 30	0.2	0.3	1.5	$\approx 0.870$

\* Model 1 Total Accuracy 指的是，除了 cond\_val 外所有预测都正确

# 模型训练中的一些探索尝试

Model 1: 随机打乱 Header 中 Column 的顺序

CLS Question SEP TEXT Column 1 SEP REAL Column 2 SEP TEXT Column 3 SEP

CLS Question SEP TEXT Column 1 SEP TEXT Column 3 SEP REAL Column 2 SEP

...

CLS Question SEP REAL Column 2 SEP TEXT Column 1 SEP TEXT Column 3 SEP

CLS Question SEP REAL Column 2 SEP TEXT Column 3 SEP TEXT Column 1 SEP

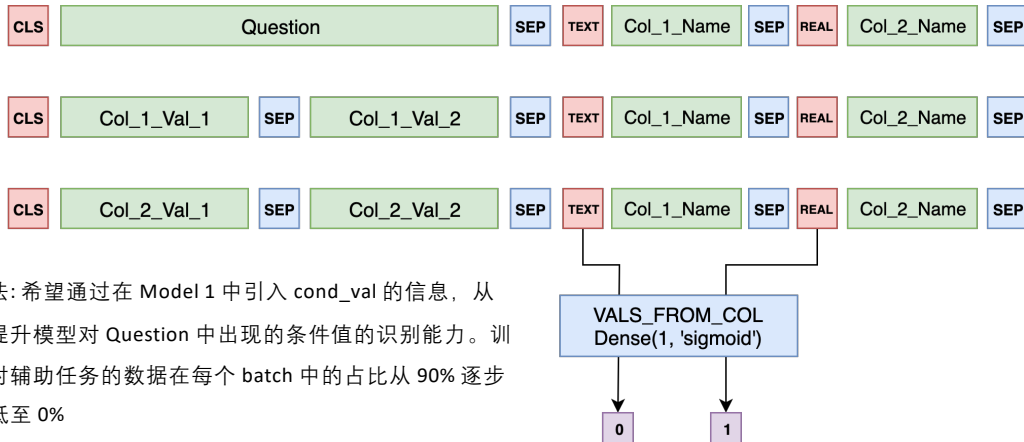
\*对结果没有明显的影响

# 模型训练中的一些探索尝试

## Model 1: 增加辅助训练任务

训练时随机将一部分样本的 Question 部分替换为某一个 Column 的若干个 Value,

对于这部分样本让模型去预测这些取值来自于哪个 Column



想法: 希望通过在 Model 1 中引入 cond\_val 的信息, 从而提升模型对 Question 中出现的条件值的识别能力。训练时辅助任务的数据在每个 batch 中的占比从 90% 逐步降低至 0%

\*线下 Model 1 Best Total Accuracy: 0.8723  
但线上无提升

# 模型训练中的一些探索尝试

Model 2 将 cond\_op 替换为自然语言



线下 conds accuracy: 0.8894

线下 conds accuracy: 0.8912

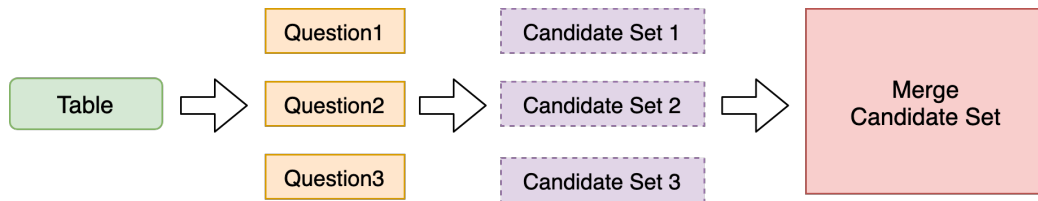
想法: 用自然语言来替换符号, 更接近 BERT 预训练时的语境, 可以提高收敛速度和准确率。

实验结果无显著差异

\*conds accuracy 指的是预测值与真实值在 conds 这个字段上完全一致

# 模型训练中的一些探索尝试

Model 2 来自同一表格的 question 共享 cond\_val candidates



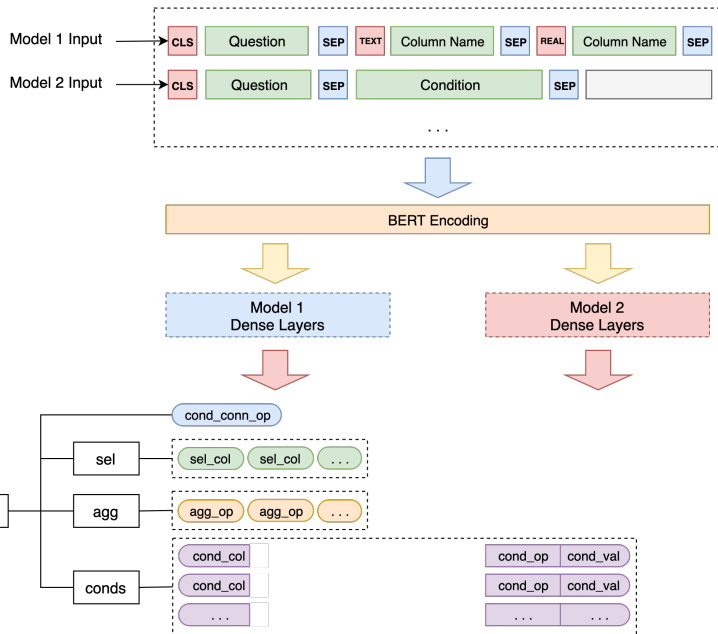
想法: 通过共享同 table 不同 question 提取出来的 cond\_val candidates , 可以融入表格级别的信息, 譬如哪些 column 的数据更有价值。同时这种做法也能提高从 question 文本中提取出来的候选 cond\_val 的覆盖率。

线下 total accuracy 0.8405 -> 0.8537, 线上 0.895 -> 0.906

\*线下 total accuracy 指的是预测SQL与真实SQL完全匹配率

# 模型训练中的一些探索尝试

共享 BERT 编码层的 Multi-Task 模型



\*这种思路可能需要较多精调技巧，由于时间关系未能展开更多的实验。

Thanks