考虑一个EncoderLayer batchsize = 1

x 29 \* 768 Relation 29\* 29 经过embedding 29\*29\*96(768/8)

|  |  |  |  |
| --- | --- | --- | --- |
| 768 | …… | …… | 768 |

|  |  |  |  |
| --- | --- | --- | --- |
| 96 | …… | …… | 96 |
| …… | …… | …… | …… |
| …… | …… | …… | …… |
| 96 | …… | …… | 96 |

x 经过 不同的线性层（768->768）以及维度变换生成Q、K、V 1 \* 29 \* 8 \* 96

每行代表一个词的表示将768 分成 8份

|  |  |  |  |
| --- | --- | --- | --- |
| 96 | …… | …… | 96 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Attention过程 首先计算qk

Query:8\*29\*96 key.transpose(-2, -1)8\*96\*29 8\*29\*29

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

->

对于一个head 计算 score 第一行代表第一个词在所有词上的得分

query.permute(0, 2, 1, 3) *# 1\*29\*8\*96* relation.transpose(-2, -1) *# 1\*29\*96\*29*

*每个词每个头上的向量 每个词和每个词的关系向量*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

*每个词每个 头上的向量*

*-> 1\*29\*8\*29 -> 1\*8\*29\*29 ->softmax得到*

*词每个头上的权重*

*计算qv*

*各头 词在每个词上的权重 value*

*1\*8\*29\*29 1\*8\*29\*96 ->1 \* 8 \*29 \*96 各词在各词上的value*

weight.permute(0, 2, 1, 3) *# 1\*29\*8\*29 relation 1\*29\*8\*96*

*各词 各头在所有词的注意权重 -> 1\*29\*8\*96 -> 1\*8\*29\*96 各词在relation上的value*

*相加 再 ->1\*29\*8\*96 -> 1\*29\*768*

sum\_types

得到

Queue.Pop

Queue.Pop

product\_types

如何填充子节点

CHILDREN\_INQUIRE

REF\_APPLY

E

REF\_INQUIRE

GENERAL\_POINTER\_APPLY

GENERAL\_POINTER\_INQUIRE

CHILDREN\_APPLY

SUM\_TYPE\_APPLY

SUM\_TYPE\_INQUIRE

encoder

decoder

X

relations

memory

align martix

qdmr

translator

SPARQL

API

result

**Encoder**

输入：x = question + table + column + value

len(x) = 29

取消位置编码，使用relation-aware attention即可以捕捉question位置关系也可以捕捉question与schema的关系；

relation-aware attention :

预定义多种relations如：(qq\_dist,2)表示两个question token下标差为2；（qcCEM）表示question token和column token 文本完全匹配，共预设了63种。Relations = [ 29\* 29]

对于encoder的一层，将relations embedding得到 R\_K = [29\*29\*96] 、R\_V = [29\*29\*96]

Concat

Linear

Scaled Dot-Product ATTENTION With Relation-Aware

Linear

X

h

R\_V

R\_K

Linear

X

Linear

X

Q

K

V

MatMul

MatMul

K

Q

R\_K

Add & Scale

Mask & Softmax

V

R\_V

MatMul

MatMul

Add & Scale

考虑一条数据

Input: x = question token + table + column + value [1,29,768]

R\_K R\_V = [29,29,96（768 / 8）]

x经过不同的线性层并截成h份：Q、K、V =[1 , 29 , 8 , 96] -> [1 , 8 , 29 ,96]

**首先计算score**：

QKT= [1,8,29,96] \* [1,8,96,29] = [1 ,8 ,29 ,29]

计算Q R\_K = [1 ,29 ,8 ,96] \* [ 1,29,96,29] = [1 , 29 , 8 ,29] ->1\*8\*29\*29 # x的每个头和relations之间的score，

Score = Softmax(mask((QKT + Q R\_K ) /96))

**计算新的表示**：

Score V =[1,8,29,29] \* [1,8,29,96] = [1,8,29,96]

Score R\_V = [1,29,8,29] \*[ 1,29,29,96] = [1,29,8,96] -> [1,8,29,96]

拼接（Score V + Score R\_V）/96 最终得到x新的表示x\_new

最终生成式中刚出现的table、column、value会在question中引用到，因此计算x\_new和x\_new中的table、column、value的relation-aware attention（单头）得到

align\_martix Q[1,29,768] K[1,18,768] R\_K[1,18,768] 得到[1,29,18]表示各单词在除ques部分的得分

MatMul

MatMul

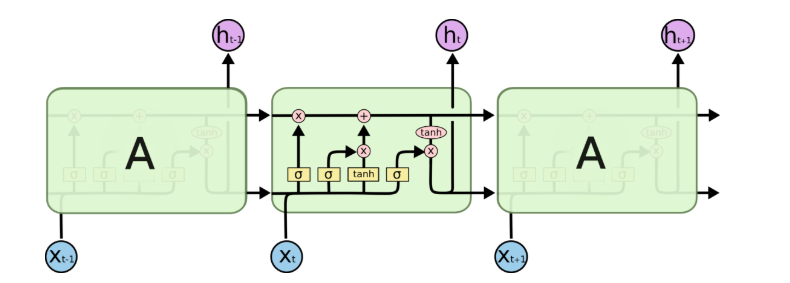
K

Q

R\_K

Add & Scale

Mask & Softmax



**Decoder**

d\_h = d\_c = 512

d\_x = 128 + 768 + 512 + 128 + 64 = 1600

ht 由当前node决定接下来rules （show）or ref （max = 20）or ground （经过softmax后得到对ht 在question、table、column、value上的得分，再与align martix相乘得到h\_t对于table、column、value新的表示，进而选择ground）

xt+1=cat(actiont,contextt+1,statep\_t+1,actionp\_t+1,node\_type\_embt+1)

LSTM

LSTM

ct-1

ht-1

xt

ht

ct

Linear

Tanh

Linear

log\_softmax

& rules\_mask

memory

ScaledDotProductPointer

softmax

clamp & log & log\_softmax

MatMul

step\_mask &

log\_softmax

Linear

Tanh

Linear

choice

Multi\_head attention

align

martix

**update**

update

rules

ref

ground

linear

linear

linear

linear

Q

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

K、V

encoder

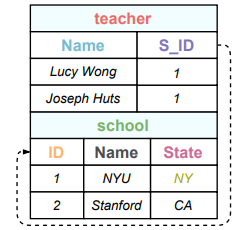
|  |
| --- |
|  |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
|  |
|  |

beam\_search

转化为图数据库

Lucy Wong

Joseph Huts

2

Stanford

CA

arc:teacher:S\_ID

arc:teacher:S\_ID:school:ID

arc:school:State

arc:school:Name

arc:school:State

arc:teacher:Name

arc:teacher:Name

arc:school:ID

arc:school:ID

arc:teacher:S\_ID

1

NYU

arc:school:Name

NY

1

X

h

R\_V

R\_K

X

X

Q

K

V

linear

cat

ScaledDotProduct

三线图

bert\_no\_aug 8 8 768

|  |  |  |
| --- | --- | --- |
| **step** | **dev** | **test** |
| 2000 | 0.5 | 0.4 |
| 4000 | 0.62 | 0.48 |
| 10000 | 0.63 | 0.52 |
| 40000 | 0.67 | 0.54 |

|  |  |  |
| --- | --- | --- |
| **step** | **dev** | **test** |
| 6000 | 0.640 | 0.532 |
| 12000 | 0.692 | 0.543 |
| 20000 | 0.690 | 0.553 |
| 50000 | 0.717 | 0.539 |
| 60000 | 0.719 | 0.566 |
| 81000 | 0.712 | 0.561 |

|  |  |  |
| --- | --- | --- |
| **Dataset** | **dev** | **test** |
| 6000 | 0.6404494382022472 | 0.5316698656429942 |
| 12000 | 0.6921348314606741 | 0.5431861804222649 |
| 20000 | 0.6898876404494382 | 0.5527831094049904 |
| 50000 | 0.7168539325842697 | 0.5393474088291746 |

SELECT[tbl:​Cartoon]

PROJECT[col:​Cartoon:​Title, #1]  
SORT[#2, #2, sortdir:​ascending]

#2:?Cartoon arc:Cartoon:Title ?Title

SELECT ?Title

WHERE

{ ?Cartoon arc:Cartoon:Title ?Title.

}

ORDER BY ASC(?Title)

NextStepSelect

root

step

select

is\_distinct

aggregate

step

grounding

NextStepAgg

agg\_type

agg\_ops

step

FinalStep

ref

Count

1

False

1

服务器账号：zfk@36.cist.cc，登陆命令：ssh zfk@36.cist.cc -p 20000，密码：cistzfk

python run\_text2qdmr.py preprocess text2qdmr/configs/experiments/bert\_qdmr\_train\_aug.jsonnet

python -m torch.distributed.launch --nproc\_per\_node=$NGPUS --use\_env run\_text2qdmr.py train text2qdmr/configs/experiments/bert\_qdmr\_train.jsonnet

fuser -v /dev/nvidia\*

conda activate env-qdmr

cd anaconda3/envs/env-qdmr

export NGPUS=2

python -m torch.distributed.launch --nproc\_per\_node=$NGPUS --use\_env --master\_port `./utils/get\_free\_port.sh` run\_text2qdmr.py train text2qdmr/configs/experiments/bert\_qdmr\_train\_aug.jsonnet

python -m torch.distributed.launch --nproc\_per\_node=$NGPUS --use\_env run\_text2qdmr.py train text2qdmr/configs/experiments/grappa\_qdmr\_train\_aug.jsonnet

bert-large-uncased-whole-word-masking

改维度要改text2qdmr-base.jsonnet、qdmr\_enc self.base\_enc\_hidden\_size 目前是8头8层768

python run\_text2qdmr.py eval text2qdmr/configs/experiments/bert\_qdmr\_train.jsonnet

python run\_text2qdmr.py eval text2qdmr/configs/experiments/bert\_qdmr\_train\_aug.jsonnet

python test\_for\_one.py eval-wo-infer bert\_qdmr\_train\_aug.jsonnet

修改step时改train \_step和bert.jsonnet的max

eval改 run\_text2qdmr的step

cd virtuoso/virtuoso-opensource/database

screen -r vir

virtuoso-t -fd