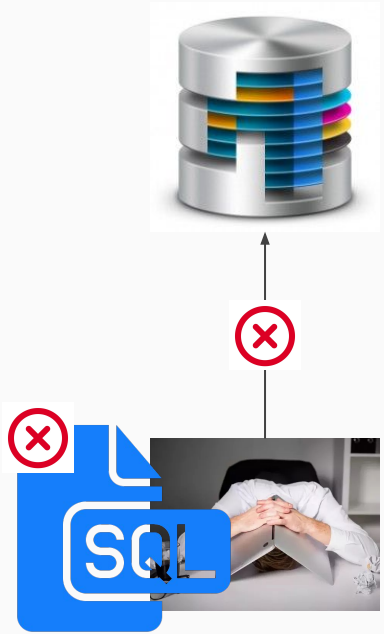
Text to SQL

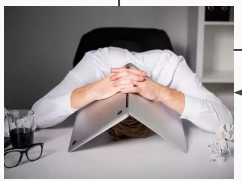
Automated ever-improving ad-hoc SQL query generation

Atakan Okan

Business Problem



Business Problem



What is the SQL query if I want dataset X?

Defining the Problem

JUST ADDRESSING:

SELECT X
FROM Y
WHERE Z > W

SQLNet: GENERATING STRUCTURED QUERIES FROM NATURAL LANGUAGE WITHOUT REINFORCEMENT LEARNING

Xiaojun Xu*
Shanghai Jiao Ton

Robust Text-to-SQL Generation with Execution-Guided Decoding

Chenglong Wang,[1*] Kedar Tatwawadi,[2*] Marc Brockschmidt,[3] Po-Sen Huang,[3†]
Yi Mao,[3] Oleksandr Polozov,[3] Rishabh Singh[4†]
[1]University of Washington [2]Stanford University [3]Microsoft Research [4]Google Brain
clwang@cs.washington.edu kedart@stanford.edu
{mabrocks,pshuang,maoyi,polozov}@microsoft.com rising@google.com

DialSQL: Dialogue Based Structured Query Generation

Izzeddin
Department of
{izzed

INCSQL: TRAINING INCREMENTAL TEXT-TO-SQL PARSERS WITH NON-DETERMINISTIC ORACLES

Tianze Shi *
Kedar Tatwawadi *

Natural Language to Structured Query Generation via Meta-Learning

rd.edu

Po-Sen Huang*, Chenglong Wang†, Rishabh Singh*, Wen-tau Yih‡, Xiaodong He*◇
*Microsoft Research †University of Washington

TypeSQL: Knowledge-based Type-Aware Neural Text-to-SQL Generation

Tao Yu
Yale University
tao.yu@yale.edu

Zifan Li
Yale University
zifan.li@yale.edu

Zilin Zhang
Yale University
zilin.zhang@yale.edu

Achieving 90% accuracy in WikiSQL

ragomir Radev
Yale University
r.radev@yale.edu

Wonseok Hwang
wonseok.hwang@navercorp.com

Jinyeong Yim
jinyeong.yim@navercorp.com

Seunghyun Park
seung.park@navercorp.com

Minjoon Seo
minjoon.seo@navercorp.com

Clova AI Research, NAVER Corp., Seongnam, Korea
January 9, 2019*

Natural Language to Structured Language

STRUCTURED
QUERY
LANGUAGE

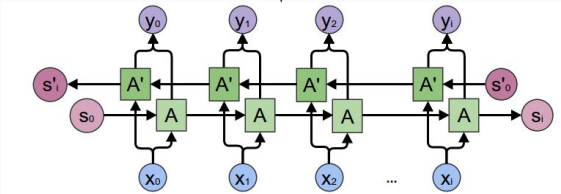
Our SQL generation process should use its structural rules!

SyntaxSQLNet: How it works

QUESTION
What are the maximum and minimum budget of the departments?

SyntaxSQLNet: How it works

QUESTION
What are the maximum and minimum budget of the departments?

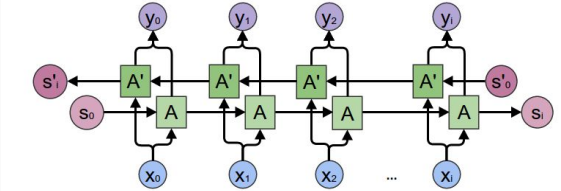


ENCODING
Bidirectional LSTM with Attention

QUESTION EMBEDDING (CONTEXT VECTOR)

SyntaxSQLNet: How it works

QUESTION
What are the maximum and minimum budget of the departments?



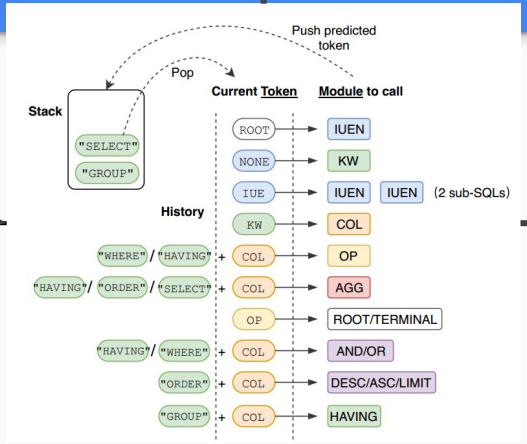
ENCODING
Bidirectional LSTM with Attention

QUESTION EMBEDDING (CONTEXT VECTOR)

STACK

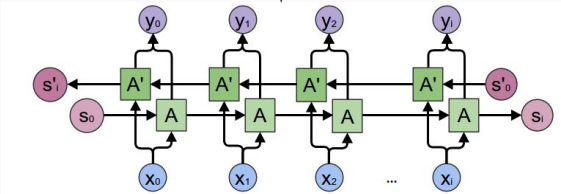
ROOT

RECURSIVE



SyntaxSQLNet: How it works

QUESTION
What are the maximum and minimum budget of the departments?



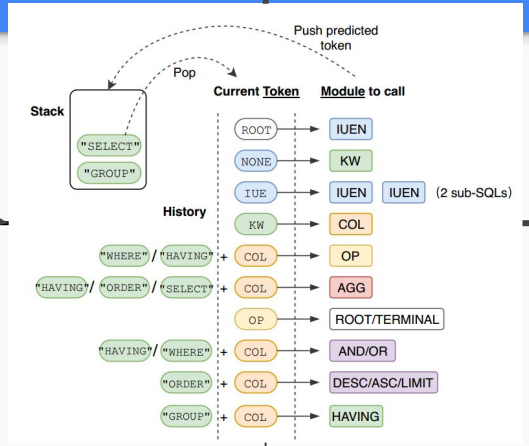
ENCODING
Bidirectional LSTM with Attention

QUESTION EMBEDDING (CONTEXT VECTOR)

STACK

ROOT

RECURSIVE



GENERATED SET
{'sql': {'select': [('department', 'budget in billions', 5), 'min', ('department', 'budget in billions', 5), 'max']}}

PARSED QUERY
select min(Budget_in_Billions),max(Budget_in_Billions) from department

Challenges

USER SUPPLIED QUESTION	CORRECT QUERIES	SYNTAXSQLNET
What are the maximum and minimum budget of the departments?	**select min(Budget_in_Billions), max(Budget_in_Billions) from department**	**select min(Budget_in_Billions), max(Budget_in_Billions) from department**
What is the department ID of the state department?	**select Department_ID from department where Name = 'State'**	**select Department_ID from department where Name = 'terminal'**
What are the names of the departments that were founded after 1800?	**select Name from department where Creation > 1800**	**select T1.Name from department as T1 join management as T2 on T1.Department_ID = T2.department_ID join head as T3 on T2.head_ID = T3.head_ID where T3.age > 'terminal'**

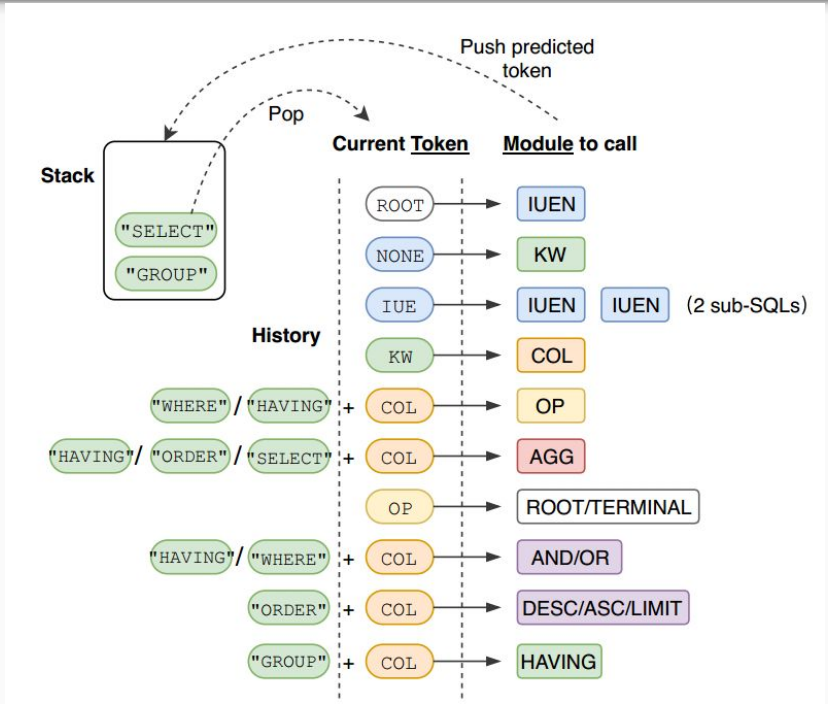
Wrong column selection

Question: Simple query
Model prediction: complex query
Correct answer: simple query

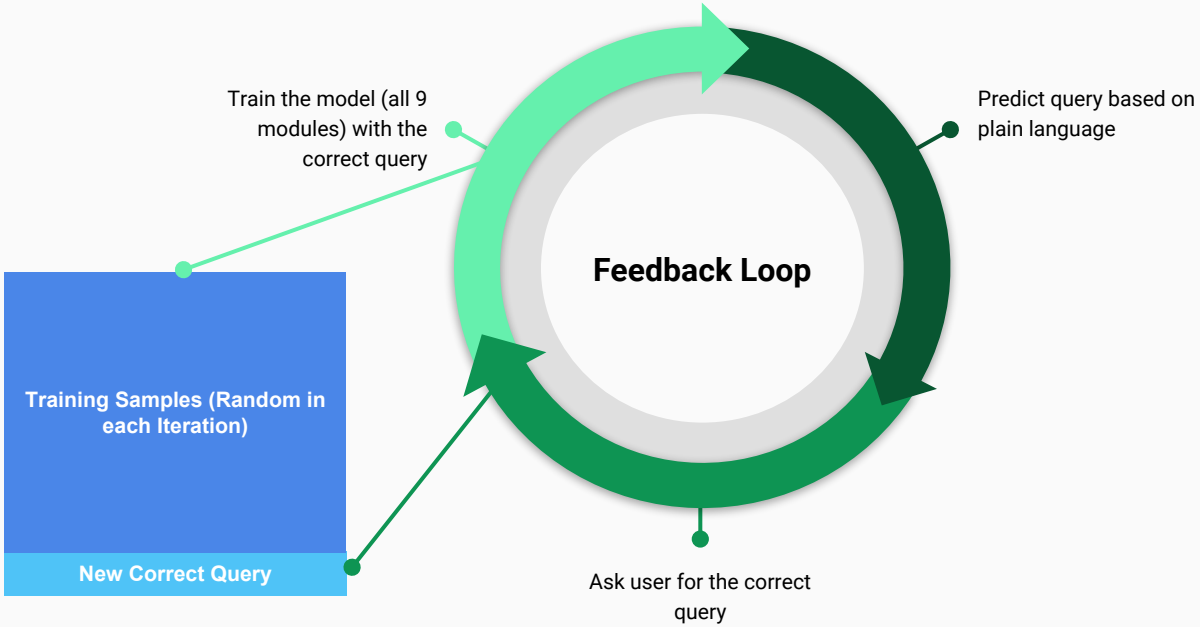
Other Challenges

- Decoder: 9 Modules -> Trained
  - LONG TRAINING TIME

- Each module has different datasets created by different parsers



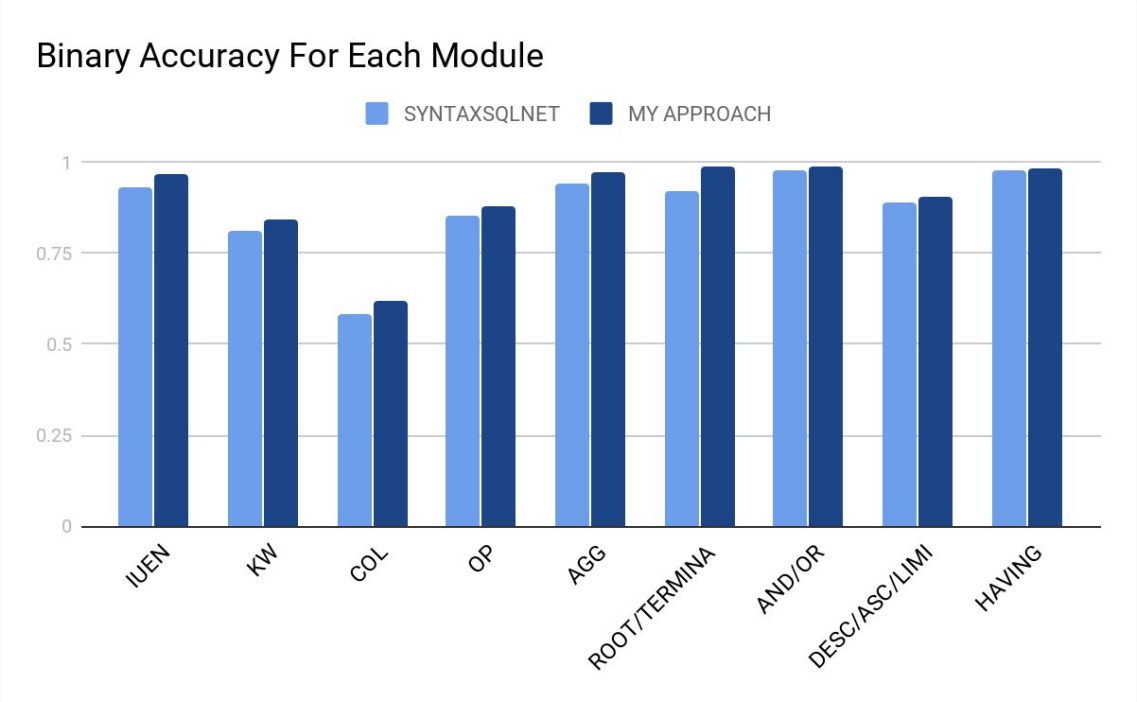
Human-in-the-loop Training / Active Learning



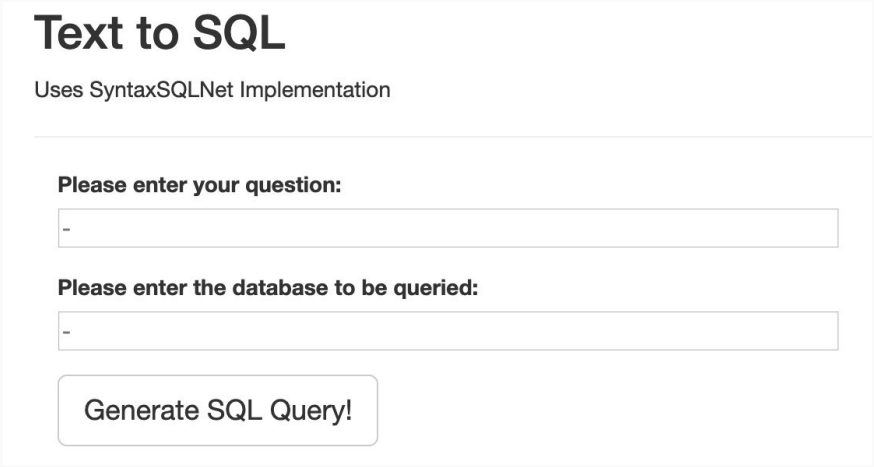
Improvement

USER SUPPLIED QUESTION	CORRECT QUERIES	SYNTAXSQLNET	MY APPROACH
What are the maximum and minimum budget of the departments?	**select min(Budget_in_Billions), max(Budget_in_Billions) from department**	**select min(Budget_in_Billions), max(Budget_in_Billions) from department**	**select min(Budget_in_Billions), max(Budget_in_Billions) from department**
What is the department ID of the state department?	**select Department_ID from department where Name = 'State'**	**select Department_ID from department where Name = 'terminal'**	**select Department_ID from department where Name = 'terminal'**
What are the names of the departments that were founded after 1800?	**select Name from department where Creation > 1800**	**select T1.Name from department as T1 join management as T2 on T1.Department_ID = T2.department_ID join head as T3 on T2.head_ID = T3.head_ID where T3.age > 'terminal'**	**select Name from department where Creation > 1800**

Side by Side Comparison



DEMO


Text to SQL
Uses SyntaxSQLNet Implementation
Please enter your question:
-
Please enter the database to be queried:
-
Generate SQL Query!

DEMO

Text to SQL

Uses SyntaxSQLNet Implementation

Please enter your question:

-

Please enter the database to be queried:

-

Generate SQL Query!

Generated Query:

Question:
What are the maximum and minimum budget of the departments?

DB Name:
department_management

Copy paste this query into your SQL query UI:
select min(Budget_in_Billions),max(Budget_in_Billions) from department

DEMO

Text to SQL

Uses SyntaxSQLNet Implementation

Please enter your question:

-

Please enter the database to be queried:

-

Generate SQL Query!

Generated Query:

Question:
What are the maximum and minimum budget of the departments?

DB Name:
department_management

Copy paste this query into your SQL query UI:
select min(Budget_in_Billions),max(Budget_in_Billions) from department

Query Feedback:

Please enter your question:

-

Please enter the database to be queried:

-

Please enter the correct query:

Enter Correct Query Here

Train the model!

Database Tables:

department

	Department_ID	Name	Creation	Ranking	Budget_in_Billions	Num_Employees
0	1	State	1789	1	9.96	30266.0
1	2	Treasury	1789	2	11.10	115897.0
2	3	Defense	1947	3	439.30	3000000.0

head

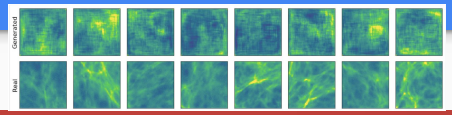
	head_ID	name	born_state	age
0	1	Tiger Woods	Alabama	67.0
1	2	Sergio García	California	68.0
2	3	K. J. Choi	Alabama	69.0

management

	department_ID	head_ID	temporary_acting
0	2	5	Yes
1	15	4	Yes
2	2	6	Yes

Atakan Okan

- Replicating numerical simulations using MMD-GANs at Flatiron Institute CCA
- NYU MS in Data Science
  - Data Science
  - Deep Learning
    - Computer Vision
    - Natural Language Processing
    - Generative Adversarial Networks
- Junior Data Scientist at Blisce Venture Capital
  - Data Infrastructure/Databases
  - Business Intelligence
  - Natural Language Processing



FLATIRON INSTITUTE
Center for Computational Astrophysics

NYU Center for Data Science

blisce/

STADIUM GOODS

BOXED

Q&A

APPENDIX

Mathematical Notations and Explanations

$$\mathbf{H}_{1/2} = \mathbf{softmax}(\mathbf{H}_1 \mathbf{W} \mathbf{H}_2^{\top}) \mathbf{H}_1.$$

$$\mathcal{P}(\mathbf{U}) = \mathbf{softmax}\left(\mathbf{V}\mathbf{tanh}(\mathbf{U})\right)$$

H_Q = LSTM Hidden State on question embedding

H_HS = LSTM Hidden State on path history

H_COL = LSTM Hidden State on column embeddings

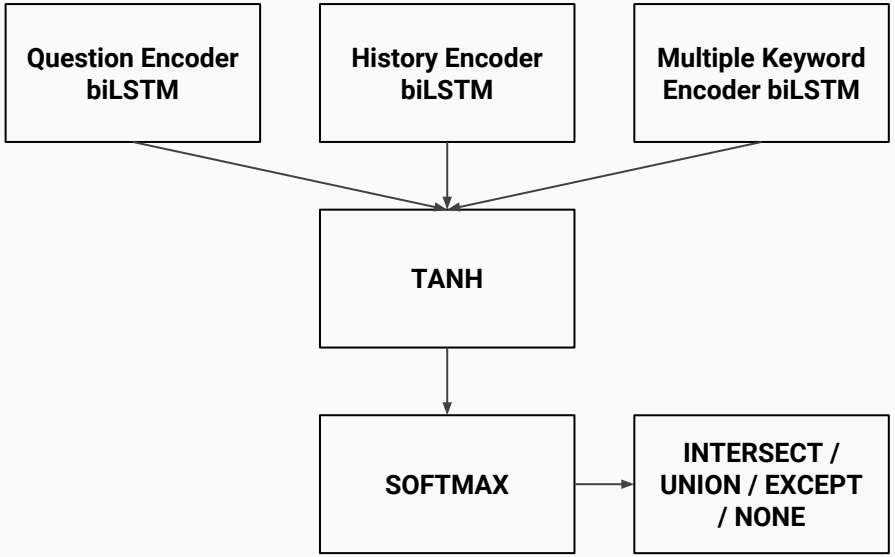
H_MKW = LSTM hidden state on multiple keywords

H_KW = LSTM hidden state on keyword embeddings

IUEN

INTERSECT, UNION, EXCEPT, NONE

Only one will be selected via softmax

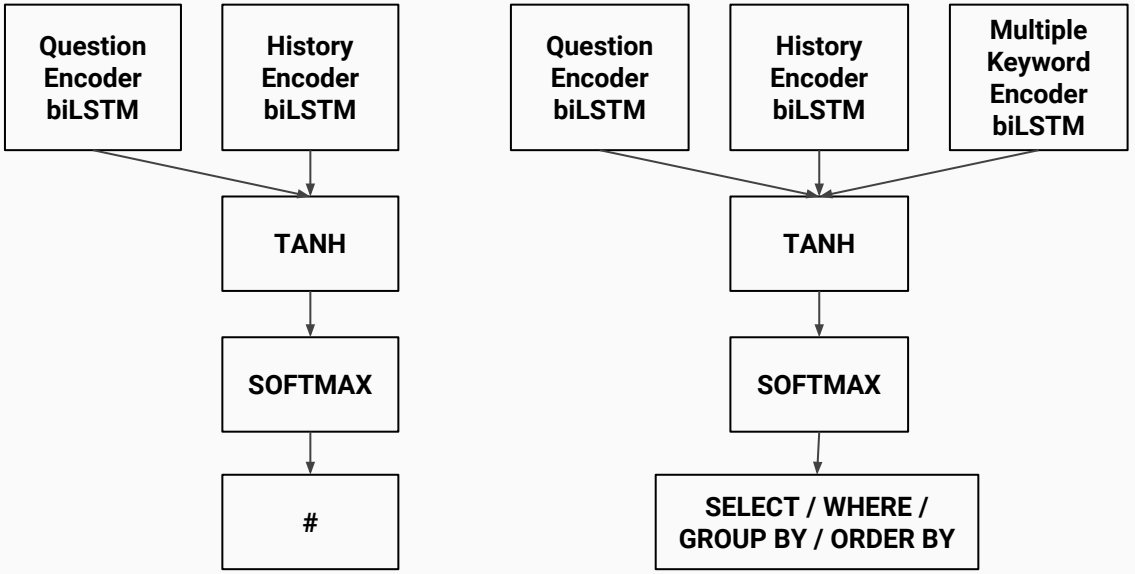


$$P_{\text{IUEN}} = \mathcal{P}\left(\mathbf{W}_1\mathbf{H}_{\text{Q/MKW}}^{\top} + \mathbf{W}_2\mathbf{H}_{\text{HS/MKW}}^{\top} + \mathbf{W}_3\mathbf{H}_{\text{MKW}}^{\top}\right)$$

KW

KEYWORD: SELECT, WHERE, GROUP BY & ORDER BY

1. Predict the number of keywords in SQL
2. Then predict keywords from the list



$$P_{\mathrm{KW}}^{\mathrm{num}} = \mathcal{P}\left(\mathbf{W}_1^{\mathrm{num}} \mathbf{H}_{\mathrm{Q/KW}}^{\mathrm{num}\top} + \mathbf{W}_2^{\mathrm{num}} \mathbf{H}_{\mathrm{HS/KW}}^{\mathrm{num}\top}\right)$$

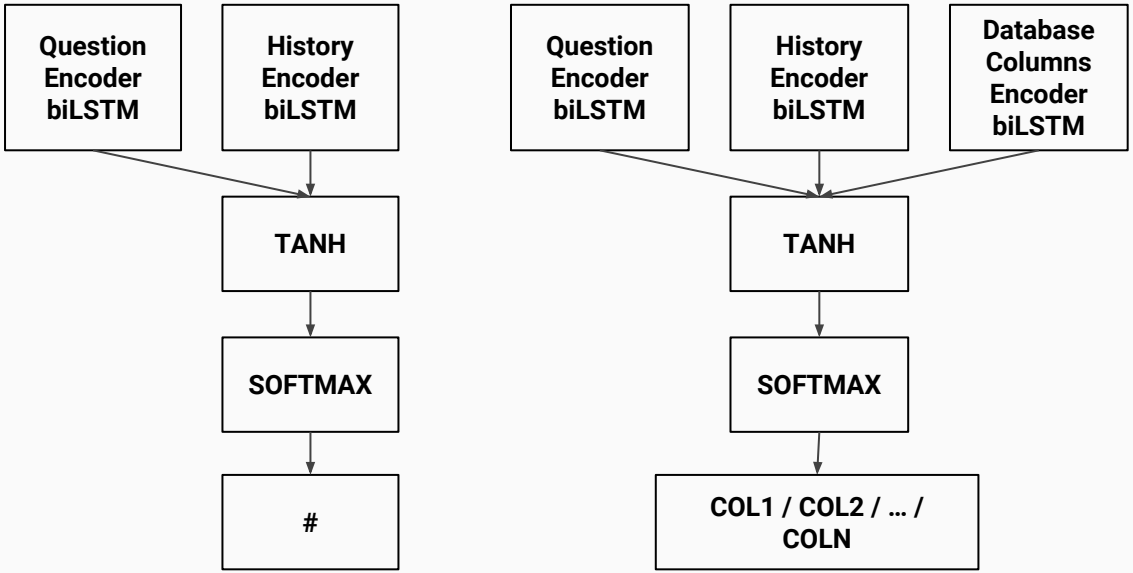
$$P_{\mathrm{KW}}^{\mathrm{val}} = \mathcal{P}\left(\mathbf{W}_1^{\mathrm{val}} \mathbf{H}_{\mathrm{Q/KW}}^{\mathrm{val}\top} + \mathbf{W}_2^{\mathrm{val}} \mathbf{H}_{\mathrm{HS/KW}}^{\mathrm{val}\top} + \mathbf{W}_3^{\mathrm{val}} \mathbf{H}_{\mathrm{KW}}^{\top}\right)$$

COL

COLUMN

PREDICTS THE TABLE COLUMNS

1. Predict the number of columns
2. Predict which ones



$$P_{\mathrm{COL}}^{\mathrm{num}} = \mathcal{P}\left(\mathbf{W}_1^{\mathrm{num}} \mathbf{H}_{\mathrm{Q/COL}}^{\mathrm{num}\top} + \mathbf{W}_2^{\mathrm{num}} \mathbf{H}_{\mathrm{HS/COL}}^{\mathrm{num}\top}\right)$$

$$P_{\mathrm{COL}}^{\mathrm{val}} = \mathcal{P}\left(\mathbf{W}_1^{\mathrm{val}} \mathbf{H}_{\mathrm{Q/COL}}^{\mathrm{val}\top} + \mathbf{W}_2^{\mathrm{val}} \mathbf{H}_{\mathrm{HS/COL}}^{\mathrm{val}\top} + \mathbf{W}_3^{\mathrm{val}} \mathbf{H}_{\mathrm{COL}}^{\top}\right)$$

OP

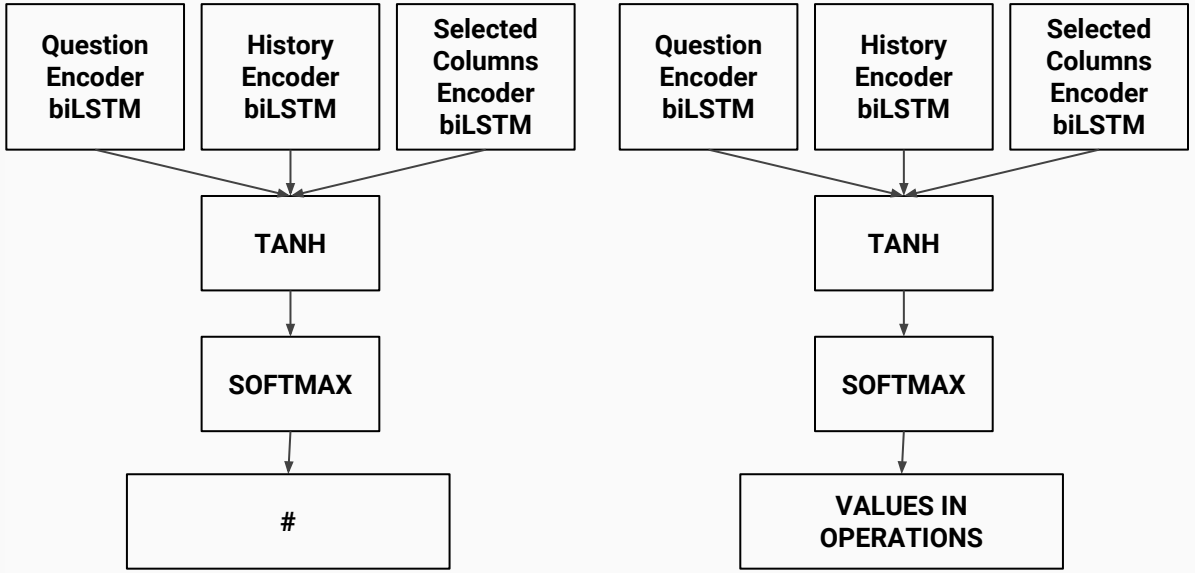
OPERATIONS

= / > / < / >= / <= / != / LIKE / NOT IN / IN / BETWEEN

For each predicted column that is in the WHERE clause:

1. Predict the number of operators on each column
2. Predict which operation(s) from the list

H_CS = Embedding of one of the predicted columns from COL module



$$P_{\mathrm{OP}}^{\mathrm{num}} = \mathcal{P}\left(\mathbf{W}_1^{\mathrm{num}}\mathbf{H}_{\mathrm{Q/CS}}^{\mathrm{num}\top} + \mathbf{W}_2^{\mathrm{num}}\mathbf{H}_{\mathrm{HS/CS}}^{\mathrm{num}\top} + \mathbf{W}_3^{\mathrm{num}}\mathbf{H}_{\mathrm{CS}}^{\top}\right)$$

$$P_{\mathrm{OP}}^{\mathrm{val}} = \mathcal{P}\left(\mathbf{W}_1^{\mathrm{val}}\mathbf{H}_{\mathrm{Q/CS}}^{\mathrm{val}\top} + \mathbf{W}_2^{\mathrm{val}}\mathbf{H}_{\mathrm{HS/CS}}^{\mathrm{val}\top} + \mathbf{W}_3^{\mathrm{val}}\mathbf{H}_{\mathrm{CS}}^{\top}\right)$$

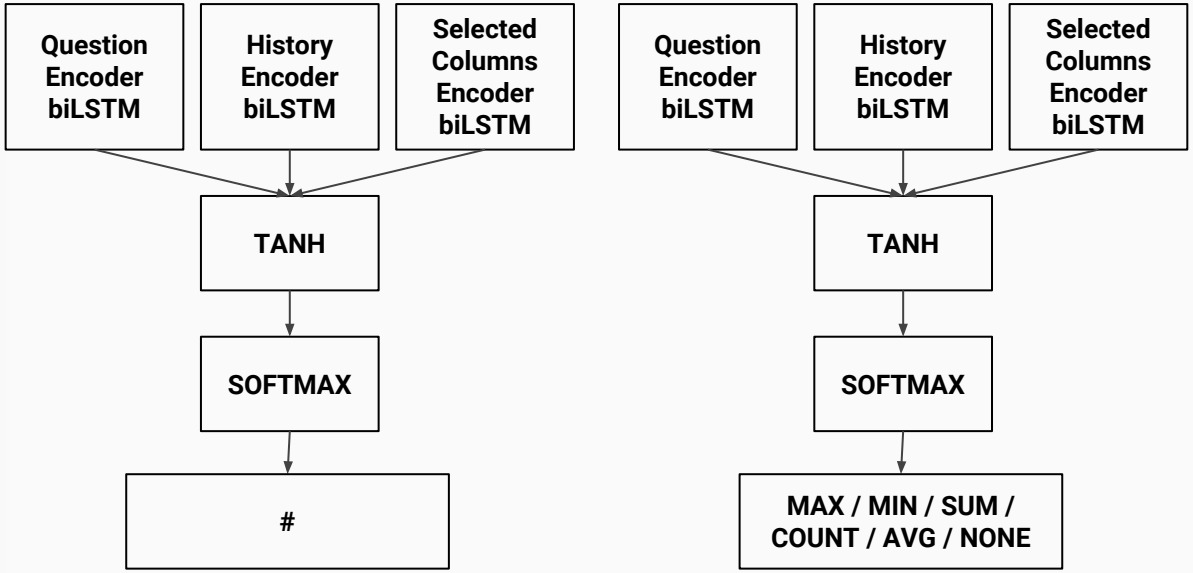
AGG

AGGREGATOR

MAX / MIN / SUM / COUNT / AVG / NONE

For each predicted column in COL module:

1. Predict the number of aggregators on that column
2. Predict which aggregators



$$P_{\text{AGG}}^{\text{num}} = \mathcal{P}\left(\mathbf{W}_1^{\text{num}}\mathbf{H}_{\text{Q/CS}}^{\text{num}\top} + \mathbf{W}_2^{\text{num}}\mathbf{H}_{\text{HS/CS}}^{\text{num}\top} + \mathbf{W}_3^{\text{num}}\mathbf{H}_{\text{CS}}^{\top}\right)$$

$$P_{\text{AGG}}^{\text{val}} = \mathcal{P}\left(\mathbf{W}_1^{\text{val}}\mathbf{H}_{\text{Q/CS}}^{\text{val}\top} + \mathbf{W}_2^{\text{val}}\mathbf{H}_{\text{HS/CS}}^{\text{val}\top} + \mathbf{W}_3^{\text{val}}\mathbf{H}_{\text{CS}}^{\top}\right)$$

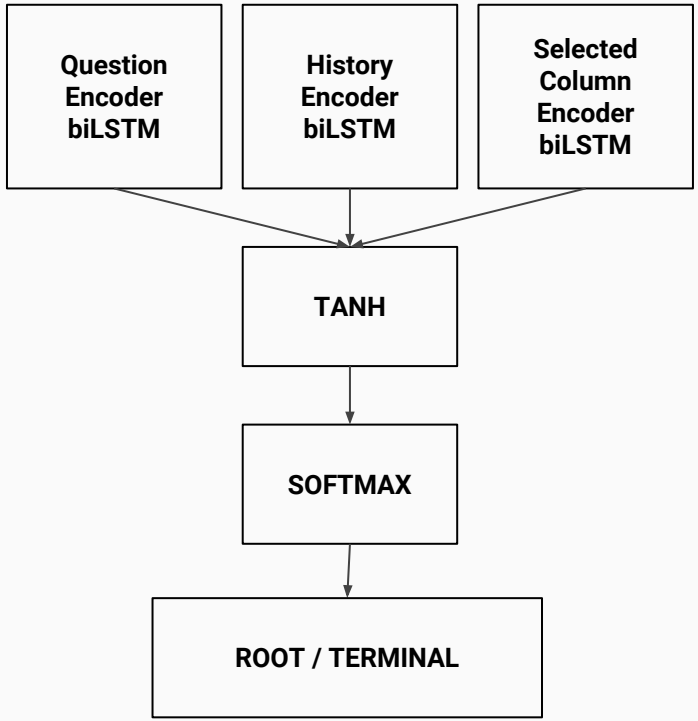
ROOT/TERMINAL

ROOT OF THE SUBQUERY // TERMINAL VALUE

ENABLES NESTED QUERIES

For each of the predicted column from the COL module that is in the WHERE clause:

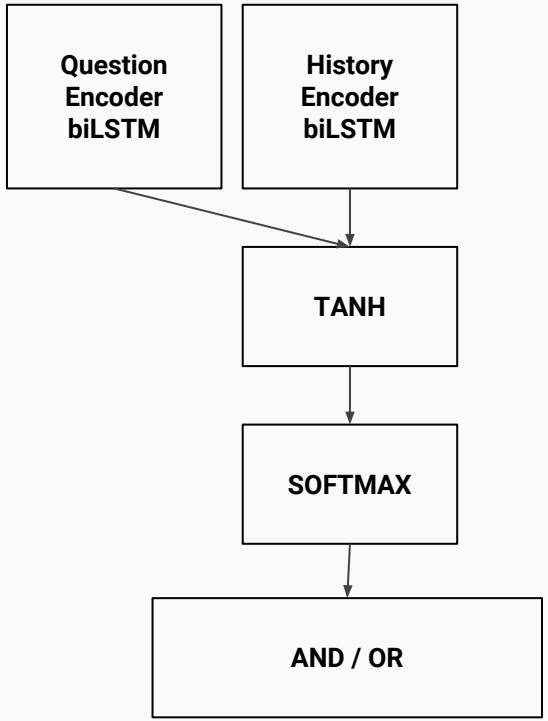
1. Call OP module
2. Predict whether the next decoding step is a “ROOT” node or a value terminal node



$$P_{\mathrm{RT}} = \mathcal{P}\left(\mathbf{W}_1\mathbf{H}_{\mathrm{Q/CS}}^{\top} + \mathbf{W}_2\mathbf{H}_{\mathrm{HS/CS}}^{\top} + \mathbf{W}_3\mathbf{H}_{\mathrm{CS}}^{\top}\right)$$

AND/OR

For each condition column predicted from the COL module with number bigger than 1



$$P_{\mathrm{AO}} = \mathcal{P}\left(\mathbf{W}_1\mathbf{H}_{\mathrm{Q}}^{\top} + \mathbf{W}_2\mathbf{H}_{\mathrm{HS}}^{\top}\right)$$

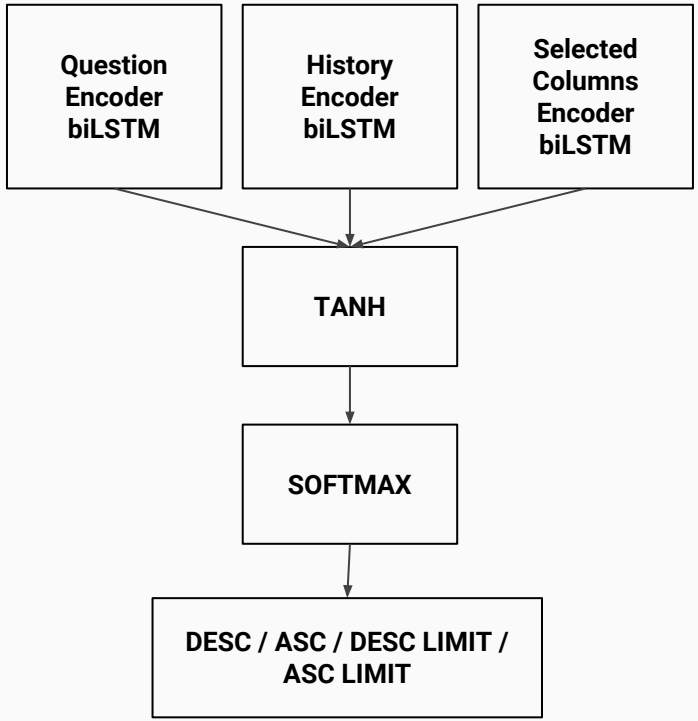
DESC/ASC/LIMIT

KEYWORDS ASSOCIATED WITH ‘ORDER BY’

CALLED ONLY IF ORDER BY IS PREDICTED BEFORE

DESC/ASC/DESC LIMIT/ASC LIMIT

For each predicted column from the COL module that is in the ORDER BY clause

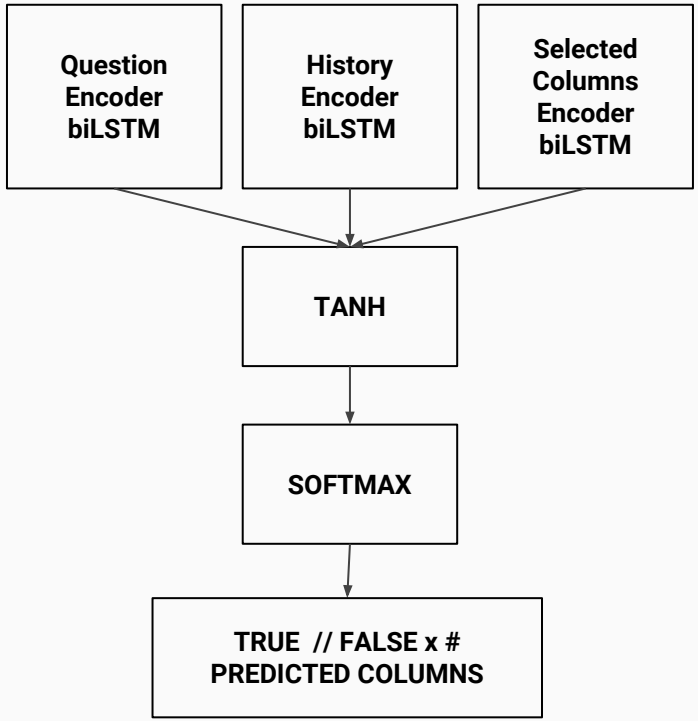


$$P_{\text{DAL}} = \mathcal{P}\left(\mathbf{W}_1\mathbf{H}_{\text{Q/CS}}^{\top} + \mathbf{W}_2\mathbf{H}_{\text{HS/CS}}^{\top} + \mathbf{W}_3\mathbf{H}_{\text{CS}}^{\top}\right)$$

HAVING

PREDICTS WHETHER ‘ORDER BY’ CONTAINS ‘HAVING’ ELEMENT

For each predicted column from the COL module that is in the GROUP BY clause, predict whether it is in the HAVING clause



$$P_{\text{HAVING}} = \mathcal{P}\left(\mathbf{W}_1\mathbf{H}_{\text{Q/CS}}^{\top} + \mathbf{W}_2\mathbf{H}_{\text{HS/CS}}^{\top} + \mathbf{W}_3\mathbf{H}_{\text{CS}}^{\top}\right)$$