Generating SQL queries from Natural Language (Multi-Table Databases)

1. Problem Statement
2. Dataset Generation
3. Own Algorithm
4. Random Forest
5. Spider 1.0 Dataset
6. Baseline implementation
7. Future Work

Guided by Prof. Mayank Singh | Naman Jain |

Ronak Kaoshik
17110068

Prakash R
17110109

Rohit Patil
17110126

Shaurya Agarawal
17110145

Problem Statement:

An appreciable amount of data in the world exists in the form of a relational database. Most of the data are in the form of multi-table database. We propose a model to convert a natural language query to SQL query for a complex and cross domain database. We would like to produce a systems which generalize well to not only new SQL queries but also new database schemas.

Spider 1.0

MIMIC SQL

Own Dataset

- Baseline model selected: A Translate-Edit Model for Natural Language Question to SQL Query Generation on Multi-relational Healthcare Data
- Dataset used in their implementation: MIMIC SQL (not yet released)
- We converted our dataset to required Schema but couldn't figure out vocabulary encoding used in the model.
- Requested database schema and other details to the authors.

[1] Ping Wang, Tian Shi, and Chandan K. Reddy. 2019. A Translate-Edit Model for Natural Language Question to SQL Query Generation on Multi-relational Healthcare Data.

Own Dataset:

- Creation of our own multi-table dataset from ACL anthology database (**13 tables**) :
    - **75 Unique questions** & SQL-queries pairs with placeholders (each query manually tested over database) and all single table queries.
    - Created an augmented dataset containing **1673 entries**.
    - **11 hours** were invested in total for dataset creation till now

Count all the papers associated with FieldID $FieldID$.	SELECT COUNT(PaperID) FROM PaperID_FieldID WHERE FieldID = '$FieldID$'
List all the Field IDs.	SELECT DISTINCT FieldID FROM PaperID_FieldID ''
List all the Field IDs associated with paperID $PaperID$.	SELECT FieldID FROM PaperID_FieldID WHERE PaperID = '$PaperID$'

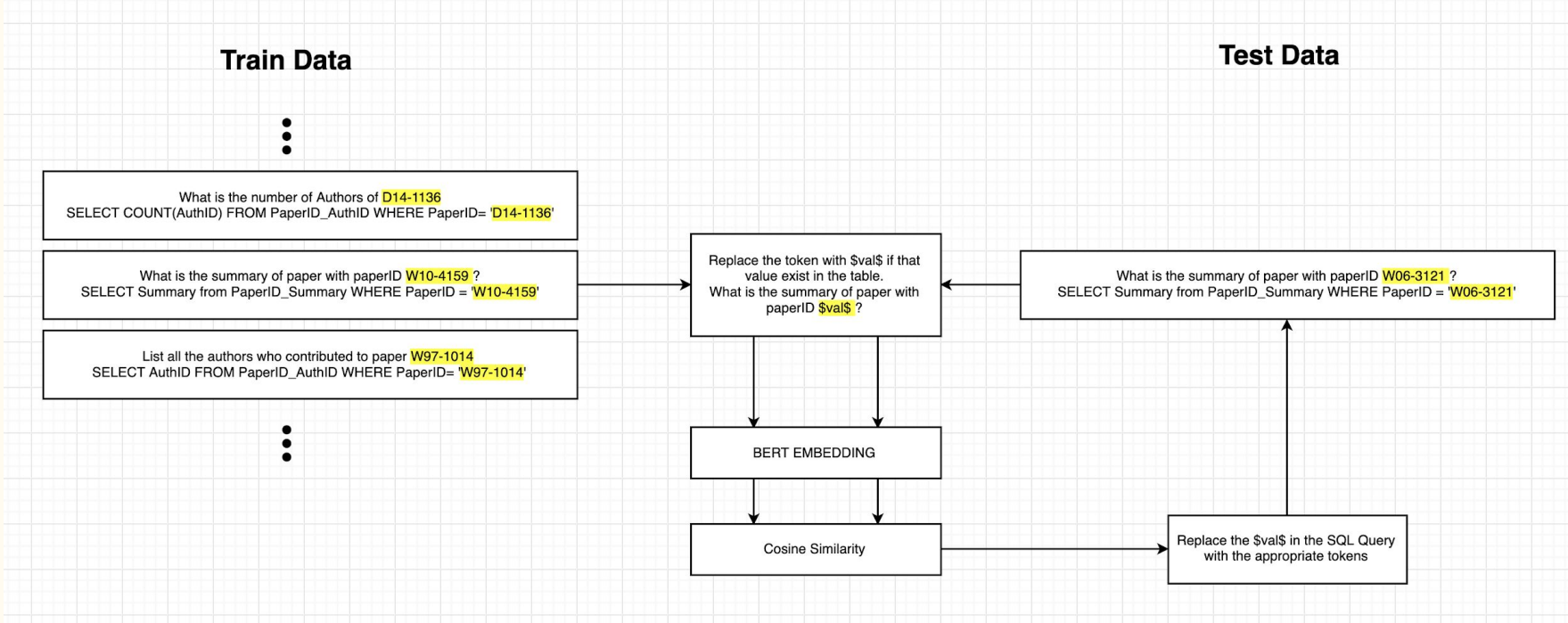
Count all the papers associated with FieldID F-6.	SELECT COUNT(PaperID) FROM PaperID_FieldID WHERE FieldID = 'F-6'
Count all the papers associated with FieldID F-9.	SELECT COUNT(PaperID) FROM PaperID_FieldID WHERE FieldID = 'F-9'
Count all the papers associated with FieldID F-5.	SELECT COUNT(PaperID) FROM PaperID_FieldID WHERE FieldID = 'F-5'
Count all the papers associated with FieldID F-0.	SELECT COUNT(PaperID) FROM PaperID_FieldID WHERE FieldID = 'F-0'
Count all the papers associated with FieldID F-10.	SELECT COUNT(PaperID) FROM PaperID_FieldID WHERE FieldID = 'F-10'
Count all the papers associated with FieldID F-12.	SELECT COUNT(PaperID) FROM PaperID_FieldID WHERE FieldID = 'F-12'

[2] Mayank Singh, Pradeep Dogga, Sohan Patro, Dhiraj Barnwal, Ritam Dutt, Rajarshi Haldar, Pawan Goyal, and Animesh Mukherjee. 2018. CL Scholar: The ACL Anthology Knowledge Graph Miner. CoRR. abs/1804.05514 (2018).

Own Implementation

Exact Match Accuracy: 64.47

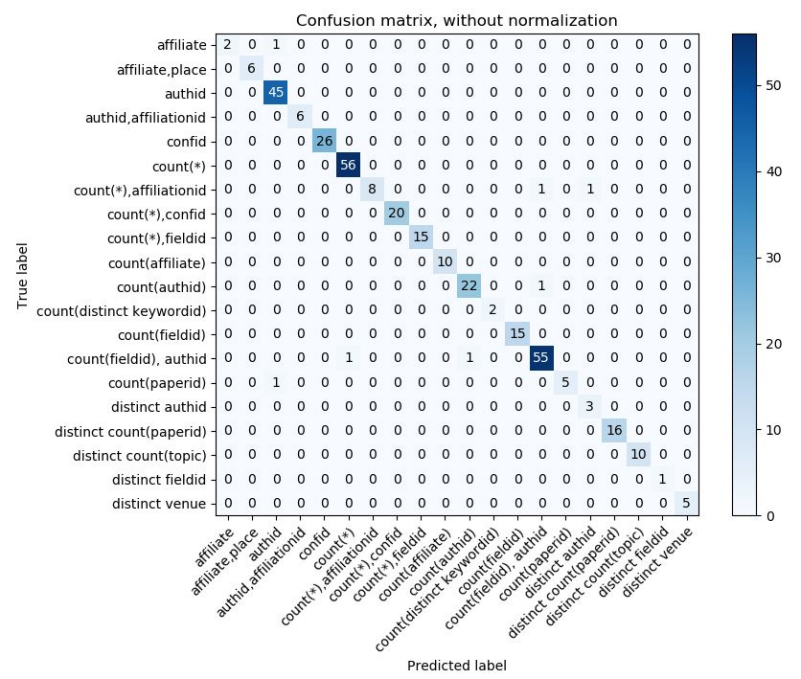
Falls short with LIKE Queries



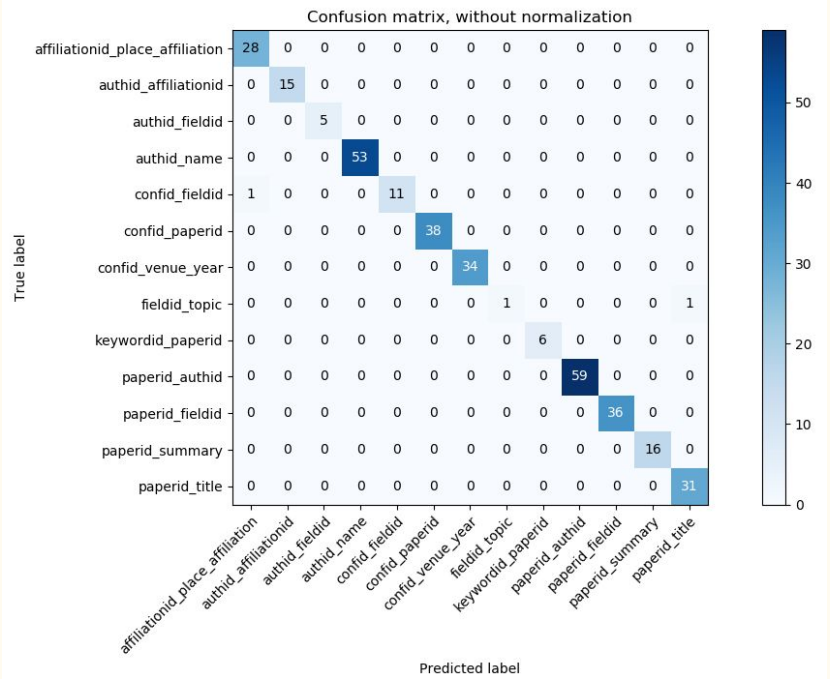
Random Forest Classifier

SELECT CONSTRAINT(COLUMN_NAME) FROM TABLE_NAME CONDITION;

Macro-Avg: 0.95, f_1-score: 0.92



Macro-Avg: 0.99, f_1-score: 0.99



Spider 1.0

Yale Semantic Parsing and Text-to-SQL Challenge

Types	Natural Language Query	SQL Query
Easy	What is the number of cars with more than 4 cylinders?	SELECT COUNT(*) FROM cars_data WHERE cylinders > 4
Medium	For each stadium, how many concerts are there?	SELECT T2.name, COUNT(*) FROM concert AS T1 JOIN stadium AS T2 ON T1.stadium_id = T2.stadium_id GROUP BY T1.stadium_id
Hard	Which countries in Europe have at least 3 car manufacturers?	SELECT T1.country_name FROM countries AS T1 JOIN continents AS T2 ON T1.continent = T2.cont_id JOIN car_makers AS T3 ON T1.country_id = T3.country WHERE T2.continent = 'Europe' GROUP BY T1.country_name HAVING COUNT(*) >= 3
Very Hard	What is the average life expectancy in the countries where English is not the official language?	SELECT AVG(life_expectancy) FROM country WHERE name NOT IN (SELECT T1.name FROM country AS T1 JOIN country_language AS T2 ON T1.code = T2.country_code WHERE T2.language = "English" AND T2.is_official = "T")

	# of Queries	# of unique SQL	# of Databases	# of Domains	# of Tables/DB
Spider	10,181	5,693	200	138	5.1
MIMIC SQL	10,000	-	1	1	5

[3] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018b. Spider: A large scale human-labeled dataset for complex and crossdomain semantic parsing and text-to-sql task. In EMNLP.

Baseline Implementations:

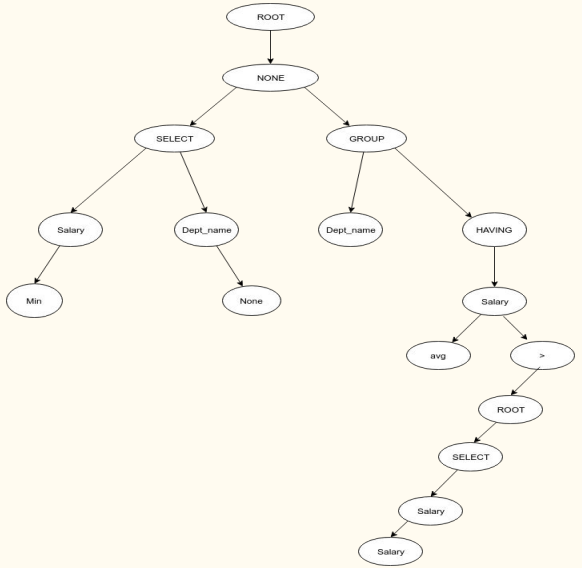
Exact Match Accuracy : 0.248

We implemented the following existing model :

- SyntaxSQLNet : To address the complex text-to-SQL generation task, SyntaxSQLNet employs a tree-based SQL generator

	easy	medium	hard	extra	all
count	252	469	153	160	1034
==========	EXECUTION ACCURACY	==========			
execution	0.000	0.000	0.000	0.000	0.000
==========	EXACT MATCHING ACCURACY	==========			
exact match	0.440	0.226	0.235	0.019	0.248
----------	PARTIAL MATCHING ACCURACY	----------			
select	0.772	0.567	0.804	0.591	0.656
select(no AGG)	0.804	0.578	0.810	0.597	0.670
where	0.546	0.348	0.169	0.141	0.333
where(no OP)	0.556	0.436	0.468	0.296	0.448
group(no Having)	0.536	0.507	0.667	0.703	0.586
group	0.500	0.449	0.667	0.689	0.550
order	0.536	0.500	0.673	0.800	0.648
and/or	1.000	0.911	0.921	0.898	0.932
IUEN	0.000	0.000	0.077	0.000	0.043
keywords	0.825	0.781	0.593	0.652	0.733
----------	PARTIAL MATCHING RECALL	----------			
select	0.766	0.567	0.804	0.588	0.654
select(no AGG)	0.798	0.578	0.810	0.594	0.668
where	0.546	0.348	0.171	0.114	0.321
where(no OP)	0.556	0.436	0.474	0.239	0.433
group(no Having)	0.750	0.515	0.737	0.675	0.610
group	0.700	0.455	0.737	0.662	0.572
order	0.682	0.455	0.649	0.790	0.637
and/or	0.956	0.935	0.946	0.979	0.949
IUEN	0.000	0.000	0.071	0.000	0.038
keywords	0.880	0.772	0.582	0.644	0.733
----------	PARTIAL MATCHING F1	----------			
select	0.769	0.567	0.804	0.589	0.655
select(no AGG)	0.801	0.578	0.810	0.596	0.669
where	0.546	0.348	0.170	0.126	0.327
where(no OP)	0.556	0.436	0.471	0.264	0.440
group(no Having)	0.625	0.511	0.700	0.689	0.597
group	0.583	0.452	0.700	0.675	0.561
order	0.600	0.476	0.661	0.795	0.643
and/or	0.978	0.923	0.933	0.937	0.940
IUEN	1.000	1.000	0.074	1.000	0.041
keywords	0.852	0.777	0.587	0.648	0.733

Evaluation on Spider dataset



Tree based SQL generation

[4] Tao Yu and Michihiro Yasunaga and Kai Yang and Rui Zhang and Dongxu Wang and Zifan Li and Dragomir Radev. SyntaxSQLNet: Syntax Tree Networks for Complex and Cross-Domain Text-to-SQL Task. In EMNLP 2018.

Future Work

- BERT based model has been released for the Spider challenge and the code was made available a few days back. We will be implementing this model in the upcoming days.
- The State-of-the-Art model research paper has been released but code hasn't been released yet. We will be studying their implementation and architecture.

Rank	Model	Dev	Test
1 June 24, 2019	TPNet + BERT *Anonymous*	**63.9**	**55.0**
5 Sep 1, 2019	EditSQL + BERT *Yale University* (Zhang et al., EMNLP '19) code	57.6	53.4
16 Sep 20, 2018	*SyntaxSQLNet + augment* *Yale University* (Yu et al., EMNLP '18) code	24.8	27.2

State-of-the-Art **(To be studied)**

BERT based **(To be implemented)**

SQLNet based **(Implemented)**

[5] Rui Zhang, Tao Yu, He Yang Er, Sungrok Shim, Eric Xue, Xi Victoria Lin, Tianze Shi, Caiming Xiong, Richard Socher, Dragomir Radev. Editing-Based SQL Query Generation for Cross-Domain Context-Dependent Questions. In EMNLP 2019.