Auto-complete NLQ and Convert to SQL

FIGHT ON ! - Sneha Kedia, Supraja Krovvidi, Priyanshi Agrawal, Laxmi Garde, Priya Bannur

Description

A system to predict SQL queries from incomplete NLQs by using auto-completion and BERT based conversion module for RDBMS.

Example

INPUT YOUR QUERY: How many papers

HOW MANY PAPERS ARE WRITTEN BY \$AUTHID\$? HOW MANY PAPERS WERE PUBLISHED IN THE YEAR \$YEAR\$ HOW MANY PAPERS WERE PUBLISHED BEFORE YEAR \$YEAR\$ HOW MANY PAPERS WERE RELEASED IN YEAR \$YEAR\$?

Input Unfinished NLQ Auto-complete module Data Cleaning TF-IDF Matrix Cosine Similarity Processed NLQ

Method

Discussion

School of Engineering

- Baseline NLQ-SQL model:
 SyntaxSQLNet, a syntax tree model for cross-domain tasks.
- Unlike baseline model, we encoded the relational schema (natural numbers mapping) and embedded it as a part of model.
- Flexibility for implementation on databases with varied schemas and can be fine-tuned for accurate results.

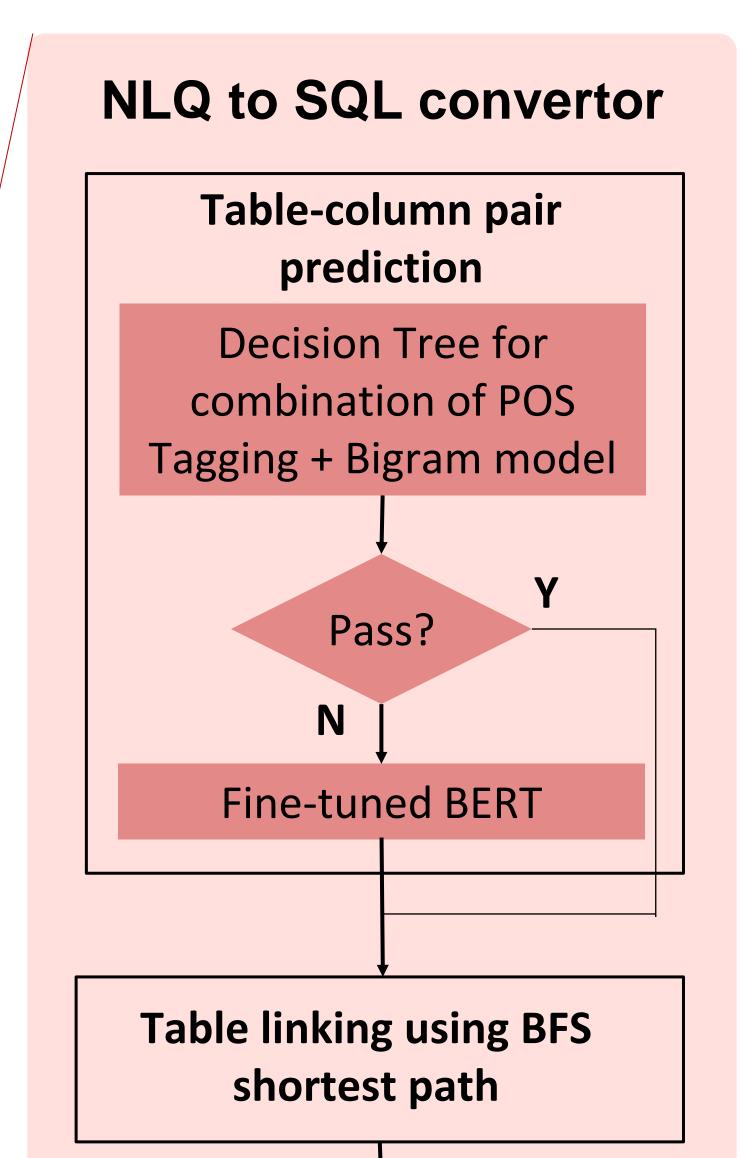
HOW MANY PAPERS ARE WRITTEN BY A-2511? HOW MANY PAPERS WERE PUBLISHED IN THE YEAR 2008 HOW MANY PAPERS WERE PUBLISHED BEFORE YEAR 1978 HOW MANY PAPERS WERE RELEASED IN YEAR 2014?

SELECT DISTINCT COUNT(*) FROM PaperID_AuthID WHERE PaperID_AuthID.AuthID='A-2511'; SELECT DISTINCT COUNT(*) FROM ConfID_PaperID JOIN ConfID_Venue_Year ON ConfID_PaperID.ConfID=ConfID_Venue_Year.ConfID WHERE ConfID_Venue_Year.Year='2008'; SELECT DISTINCT COUNT(*) FROM ConfID_PaperID JOIN ConfID_Venue_Year ON ConfID_PaperID.ConfID=ConfID_Venue_Year.ConfID WHERE ConfID_Venue_Year.Year='1978'; SELECT DISTINCT COUNT(*) FROM ConfID_PaperID JOIN ConfID_Venue_Year ON ConfID_PaperID.ConfID=ConfID_Venue_Year.ConfID WHERE ConfID_Venue_Year.Year='2014';

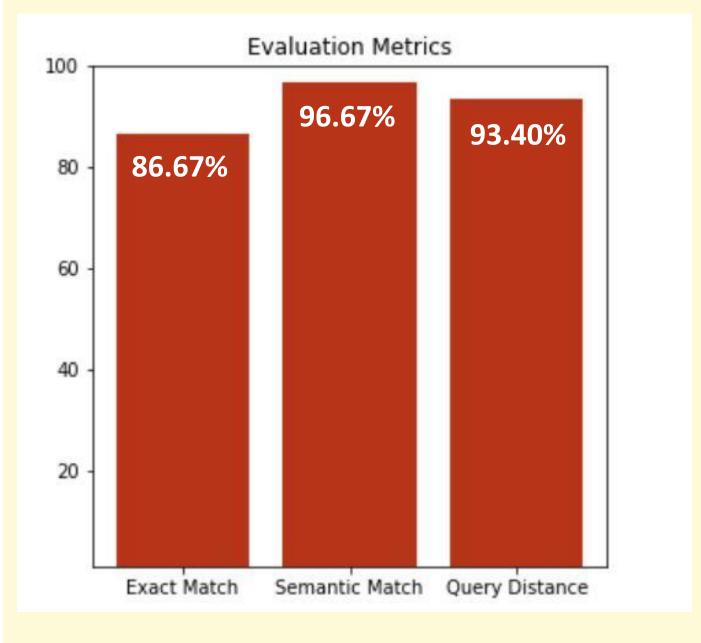
Results

- Exact Match: % of test set in which predicted SQL query is same as actual.
- Semantic Match: % of test set in which output of predicted and actual SQL query is same, though the queries might be different.
- Query Distance: Closeness of match based on no. of edges

N Placeholder? Y Recommender system • Random selection • Based on user history Final NLQ Final NLQ NLQ to SQL convertor SQL query RDBMS



between the given columns in BFS tree. (QD = 1.132)



Zero Result Rate = 0.29%

Dataset

Created domain specific dataset using the **ACL-Anthology dataset** (Singh et al. 2018).

- Generated 300+ unique single-table, multi-table NLQs.
- Paraphrased NLQs to SQL queries.
- Augmented SQL queries (replaced placeholders) to create 3000+ queries.
- Annotated SQLs manually, used a semi-automated script for checking SQL table references.

Named Entity Recognition and Condition Prediction (NER + fine-tuned BERT)

Conclusion

In comparison to existing architectures, our approach is customized for a specific database and hence giving good accuracy. Our model cannot predict complex queries with clauses like AND, HAVING, ORDER By and nested queries beyond 5 table linkages.