Text to SQL Translator

Jake Waffle

March 11, 2023

Group Members

Jake Waffle, jakewaffle@u.boisestate.edu

Goal

The goal of this project will be to build a text to SQL translator that is viable for a common REST server use case. The use case of interest will be an endpoint that fetches pages of a particular resource and filters the results based on the users' inputs. REST servers already have to deal with business logic, database queries, potentially many concurrent users, and a variety of other responsibilities. This project will hopefully help us to to see how much of an impact a text to SQL translator puts on this type of application and also demonstrate that it is able to perform well enough for production uses.

Natural Language

We're going to be working with English.

Datasets

We're looking at using the Spider and WikiSQL datasets. If need be we can also use the ATIS, Geo, and Academic datasets. These datasets are used for comparing and ranking the various text to SQL translators out there.

Machine Learning Techniques and Libraries

Libraries

We're going to be working in Java using OpenNLP and the Deep Java Library framework paired with the PyTorch engine. With OpenNLP we will have access to a lot of basic NLP utilities for things like tokenization, parts of speech and

lemmitization. With Deep Java Library we'll have access to PyTorch as well as other deep learning utilities. The Deep Java Library has a built in BERT tokenizer and an extension that adds support for the Huggingface tokenizers project that is written in Rust. If Java isn't delivering the necessary training features, then it is my hope that we'll be able to create models in Python using PyTorch and export them so that they can be imported into the main Java application.

Techniques

The natural language query will need to be tokenized and normalized of course. However, the plan for the bulk of the translation is to use transformers. A constrained decoder seems to be necessary for making sure the transformer always creates valid SQL. We also know the schema of the table we're going to be dealing with ahead of time, so another goal is to make a schema-aware encoder.

Evaluation

The evaluation of the project will be done using multiple strategies. First, there will be tests that utilize the datasets we have available to verify the accuracy/correctness of the translator. This translator will primarily focus on translating the where clause and thus we can ignore the other aspects of the datasets when determining the accuracy. We also should ensure that the created SQL queries are valid.

Second, a custom database will be put together to showcase the translator creating real queries that are then executed against the database. The translator and database queries will be hidden behind a simple REST server. Then a website will be constructed that allows for data entry as well as querying the database with natural language. This evaluation strategy will basically be a tech demo that tests the real world viability of the translator.

Exploration

I've read about existing text to SQL translators and articles to see what strategies they use. Transformers are widely used to accomplish this task in general. Our use case is simpler and won't require the generation of entire queries, but we can still make use of a lot of the information that exists out there.

Necessary Technical Abilities

We're going to need to know about tokenization and normalization for readying the natural language for the translator. The translator is going to require

knowledge of transformers since we're shooting for a transformer with schema-aware encoding and constrained decoding. The output of the translator being SQL will also require some amount of database skills.

Outside of the translator, there's going to be a REST server, website, and database. A variety of skills will be needed for wiring all of that together for a minimal showcase of the translator. That's going to include web development, API development, and database skills.

6 Week Plan

Week 1

Set up the Java project with the necessary NLP libraries. This week should be used for ensuring that the project is viable with these libraries. A test transformer should be put together. The datasets should also be downloaded and loaded into the project.

Week 2

Focus on preprocessing/postprocessing the inputs and outputs of the translator. Users will be referring to the data using different words than the actual table and column names in the database. The provided natural language will need to be tokenized and normalized to refer to the actual table and column names. The preprocessing will potentially need to include SQL injection safeguards as well. The output of the translator will also potentially need to be postprocessed so that it is ready to be used for a database query.

Week 3

The real translator should start being worked on. We need to train a model for our needs using the datasets we have. A schema-aware encoder strategy needs to be figured out for the transformer because the natural language will have a relationship with the target table's schema. A constrained decoder should also be put together so that we can make sure the transformer model only outputs valid SQL. The result of this week's work should be a trained model.

Week 4

This week should be used for focusing on testing and evaluating the translator. The same datasets should have already been split up into test and training data. We're going to be using the test data to check the accuracy of the translator that's using the trained model. We're also going to test that the output of the translator is a valid SQL query. Alongside all of this we will be making necessary changes to the translator as needed.

Week 5

A website, REST server and database should be put together to demonstrate the translator. The end goal will be to have a website that allows data entry and data querying using a search bar. The search bar will send a request to the backend, the translator will translate the natural language to a query, the query will be used against the database, and then the results will be made viewable in the website. There should also be a script that loads test information into the database. Maybe the demo database can be filled with information about movies and actors from IMDB or something like that.

Week 6

There will be unexpected issues that come up and we are going to keep this week open for addressing those issues. If no issues come up, then this time will be used to make improvements or finish something that needed more time than expected. A load test would be a nice to have to see how well the server holds up under load.