Smart Natural Language Querying and Forecasting

Udacity Capstone Project

Gbadamosi Farouk

8th of October 2021

UDACITY

Introduction

Natural Language Processing (NLP) is one of the most powerful fields in Machine Learning/Artificial Intelligence that helps to further bridge the communication gap between humans and the machine. NLP leverages advanced mathematical concepts that can easily map natural languages like English, Spanish, French, etc. to machine-understandable language. This innovation currently is at the heart of some of the most advanced products in the market which include, Apple's Siri, Google-translate, Interactive Voice Receiver, and so on.

The ability to completely remove the barriers between human and computer interactions and creating a universally accepted language is a fairly complex problem and still an area of continuous research. This area is popularly referred to as Natural Language Interfaces (NLI).

Problem & Solution Statements

It is no longer news that data is the greatest asset of any organization. Organizations are realizing the need to have a resilient data strategy that provides a 360-degree view of their business. Today, C-Level executives, decision-makers are constantly demanding insights and forecasts from their data increasing the pressures on their data teams (data engineers, data scientists, and analysts). Analysts are spending more time than ever developing business reports leaving little room for innovative and critical thinking.

One of the biggest challenges to automating business reporting and empowering C-Level executives has been querying the datasets according to the user's discretion. To interact with most databases, special skillsets such SQL is necessary to dice and slice the data that most end-users lack. In addition, most dashboards or charts today are developed using pre-defined queries based on user's demand. The challenge is that user's demands are constantly shifting as there is always a new question that needs to answered. This makes the process of constantly updating dashboards impracticable to accommodate all requests from users.

The **Smart Natural Language Query and Forecasting Solution** is being proposed to democratize access to insights using natural language. Users will now be empowered to ask questions about their data directly using free form text and will be given real-time answers in the form of charts and visuals. With this solution, data analysts can focus on automating ETL pipelines and other innovative data modeling tasks rather than responding to every new question from the end-user. Also, this solution has a plethora of use cases but most notably, this solution can be used to replace the Frequently Asked Questions (FAQ) sections of most apps today.

Existing Alternatives

As earlier stated, NLI is an active area of research and has received significant attention – academia, and industry – in recent years. Several approaches have been employed to convert text to SQL some of which include: keyword-based systems, pattern-based matching, Parsing-based systems, and grammar-based systems. In academia and as of this writing, the leading benchmark models on popular datasets like SParC, Spider, and WikiSQL are *RAT-SQL + SCoRe, WaveSQL+BERT, T5-3B+PICARD (DB content used)*, *SeaD +Execution-Guided Decoding (Xu 2021)* which use different deep learning techniques to achieve high matching accuracies. In the industry, popular business intelligence tools like Microsoft PowerBI have successfully implemented NLQ systems called Natural Language Q&A. This feature is very powerful and enables users to interact with their data using strictly guided natural language sentences. One of the major

drawbacks of the PowerBI feature is that it doesn't allow free-form sentences and the user must use exact column names as they appear in the database although this is a common problem in most implementations today.

This project will leverage the models from this EditSQL paper to develop a text-to-SQL dashboard application for users. The performance of this model will be benchmarked against an average of **54%** Question match on the test set (results from the paper). This project will also go the extra mile of enabling users to generate forecasts on their data which would be the key differentiator from other implementations seen in the industry and academia.

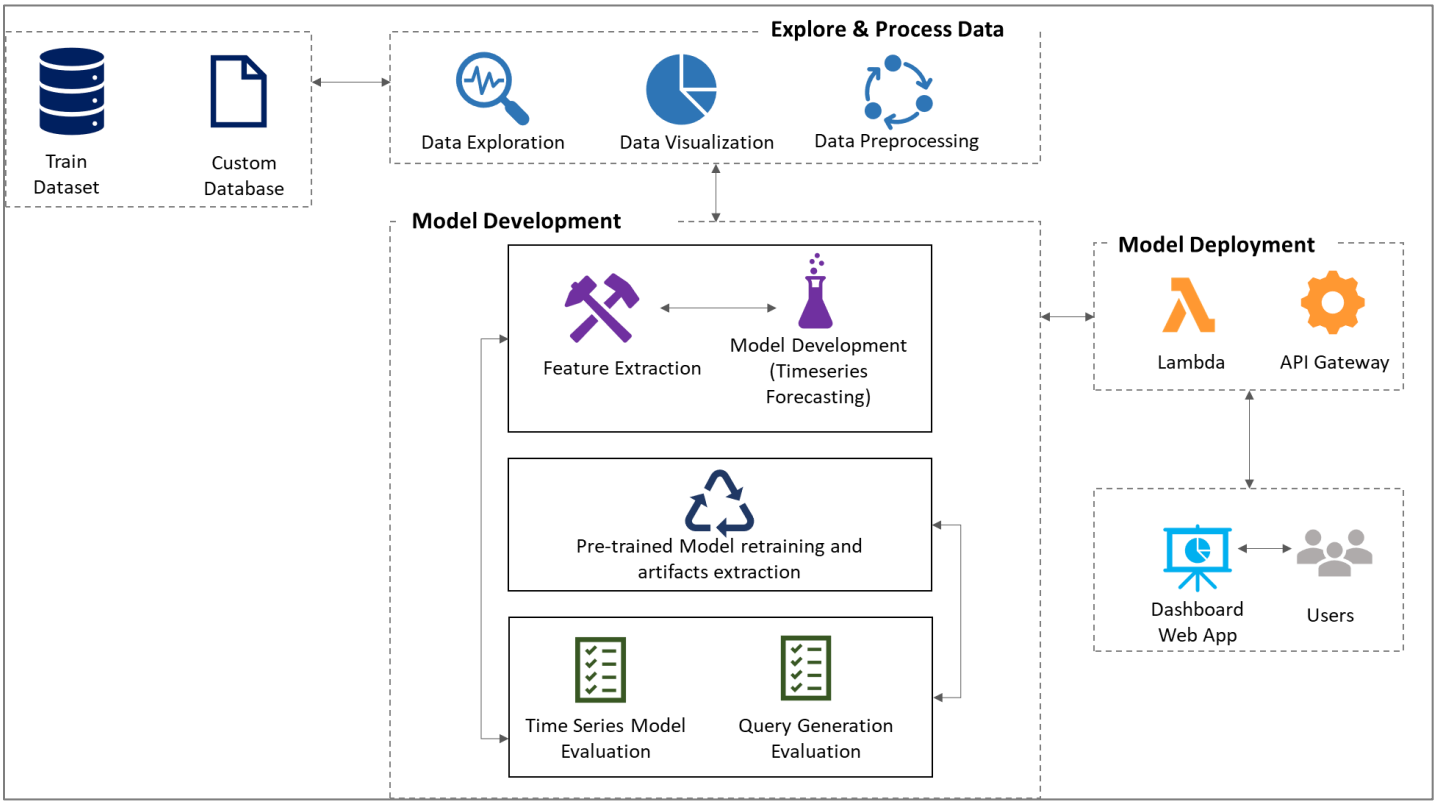
Datasets and Inputs

This is a supervised learning task and each dataset comes with target labels. Two types of datasets will be employed in this project. One is the training dataset and the other is a custom dataset. Both datasets are described below:

1. **Training Dataset**: The **SparC** dataset that we would be used to train our model. The SParC dataset is a multi-domain or multi-context dataset that contains over 200 databases from over 100 different domains. The dataset was developed by Yale & Salesforce from about 12K unique coherent questions from user interactions in those various domains. The dataset is one of the most used and widely cited for the text to SQL challenge and currently maintains a leaderboard of the most "state-of-the-art" models built on its dataset. The dataset can be downloaded from the official page of the challenge
2. **Custom Dataset**: This is the dataset that will be used to test and tune our model. This database to the SParC dataset to test our model's text-to-SQL conversion performance on the new schema. In addition, an interactive web application dashboard would be developed where users can query to generate insights. This dataset can be accessed at: Cryptocurrency time series – which is a minute by minute volume and value data of popular cryptocurrencies like Bitcoin, Ethereum and Litecoin.

UDACITY

Methodology



High Level Solution Architecture

Some important things to highlight in the high-level architecture:

1. **The Model**: Two models will be employed in this project – the forecast model and the text-to-SQL model. The text-to-SQL model would be pre-trained. The pre-trained model would be a deep learning framework from any of the benchmark models or a combination of benchmark models. The choice of the pre-trained model would be determined by **model simplicity, compatibility with AWS Sagemaker and computational requirement** while the choice of the pre-trained model would be determined by the **performance on validation set**.

2. **The Evaluation Metric**: The time-series prediction is a regression task; therefore, the evaluation metric would be **mean-absolute-error** while for the pre-trained model, a combination of **predicted and answer query comparison and accuracy of query result.**

3. **The Lambda function**: The lambda function would be first used to determine the user's intent – whether a forecast or query – before the data is preprocessed and passed to the right model for results.