**Text2Sql: Generating SQL queries using AI**

Pranav Kushare (Roll- 50), 11910626, Prof Dr Sanika Patankar

***Abstract* - Nowadays, artificial intelligence applications are increasingly integrated into every aspect of our lives. One of the newest applications in artificial intelligence and natural language is text generation, which has received considerable attention in recent years due to the advancements in deep learning and language modeling techniques. In this paper we are going to review the working of the project “Text2Sql: Generating SQL queries using AI”. We will see how AI and NLP can automatically translate a user’s natural textual query into SQL.**

***Keywords - NLP, Language Model, gpt2, Generative pretrained model***

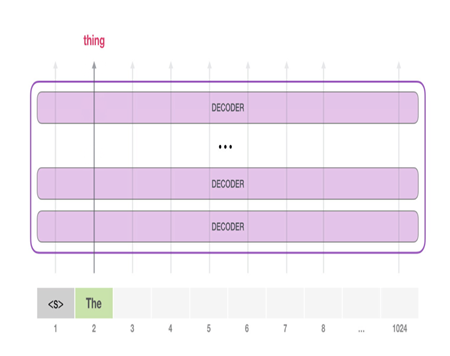
# Introduction

In this era of technology, the abundance of data is increasing exponentially. That's why there is a phrase “Data is a new oil”. Most of this data is relational. It means that there is some relation between the attributes of data. Vast variety of data is stored in a relational database. One might often need to access the data by using some condition/ query. And writing sql queries can be challenging for non-technical users. It is quite time-consuming for everyone to learn SQL. So this project Text2SQL presents a web application which uses a deep learning language model trained to translate users textual questions into SQL query.

# Implementation details

The main backbone of project is gpt-neo which stands for generative pretrained transformer model. Generative Pre-trained Transformer (GPT-2) is an open-source artificial intelligence created by OpenAI in February 2019. GPT-2 translates text, answers questions, summarizes passages,[5] and generates text output on a level that, while sometimes indistinguishable from that of humans, can become repetitive or nonsensical when generating long passages.

It is a general-purpose learner; it was not specifically trained to do any of these tasks, and its ability to perform them is an extension of its general ability to accurately synthesize the next item in an arbitrary sequence.GPT-2 was created as a "direct scale-up" of OpenAI's 2018 GPT model, with a ten-fold increase in both its parameter count and the size of its training dataset.

 Figure 1. Gpt2 architecture

The GPT-2 is built using transformer decoder blocks. BERT, on the other hand, uses transformer encoder blocks. We will examine the difference in the following section. But one key difference between the two is that GPT2, like traditional language models, outputs one token at a time. The way these models actually work is that after each token is produced, that token is added to the sequence of inputs. And that new sequence becomes the input to the model in its next step. This is an idea called “auto-regression”. This is one of the ideas that made RNNs unreasonably effective.

# METHODOLOGY

As shown in Figure 1, the followed methodology to generate SQL query consists of four phases. In this section, these phases are discussed in more detail. Basically the project can be divided into 4 steps . Refer figure3 for the same. It involves Data gathering, data preprocessing, Model building and training and finally model evaluating.

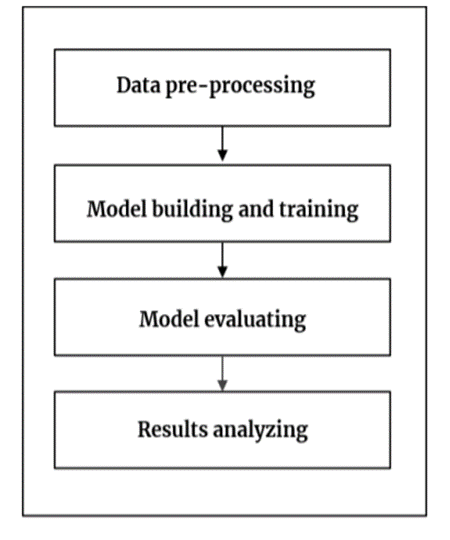
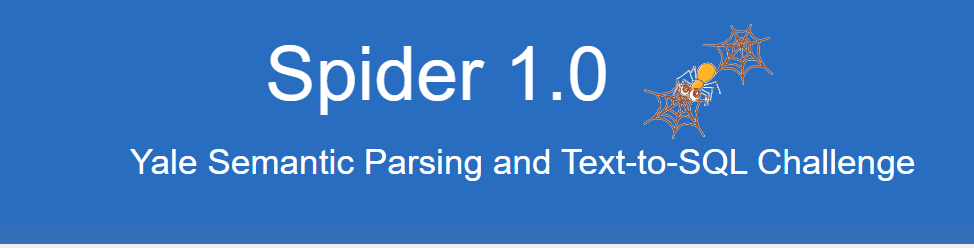


Figure 3. The phases of the methodology

In this phase, I performed dataset pre-processing by removing or modifying data that is incorrect, incomplete, irrelevant, or duplicate. Removed unrelated characters or symbols such as punctuation marks, line space, $, #, brackets, parentheses, and other unrelated characters from the dataset



Spider is a large-scale complex and cross-domain semantic parsing and text-to-SQL dataset annotated by 11 Yale students. The goal of the Spider challenge is to develop natural language interfaces to cross-domain databases. It consists of 10,181 questions and 5,693 unique complex SQL queries on 200 databases with multiple tables covering 138 different domains.

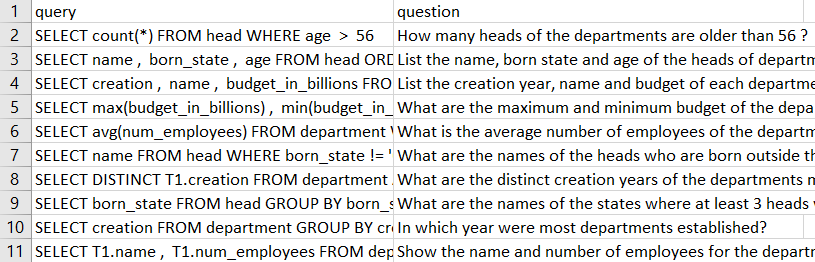


Figure 4. Dataset

The dataset has been taken from open source website by yale university which they made for some projects. It contains almost 6000 pairs of textual questions and their respective sql query.

# Results and discussion

The final output of the project is a webapp which will be useful for generating sql queries . The webapp is designed with the help of python library known as streamlit .

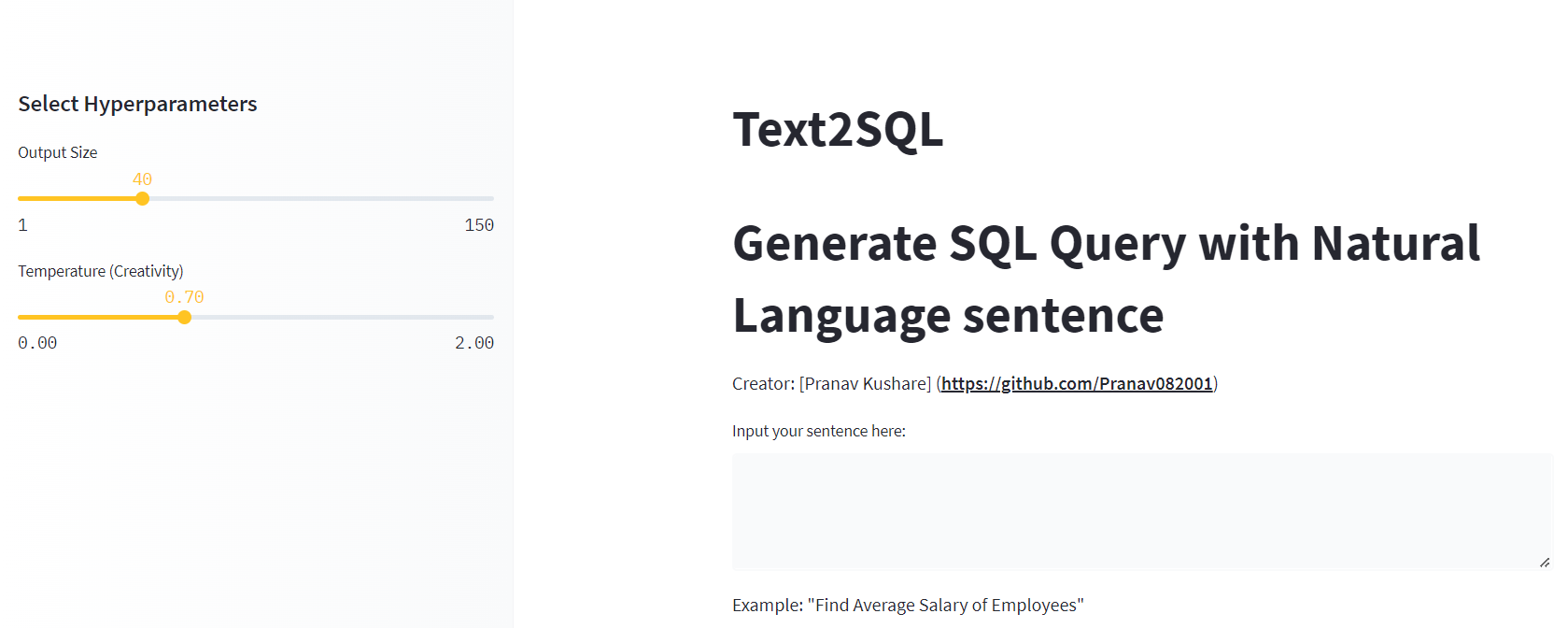
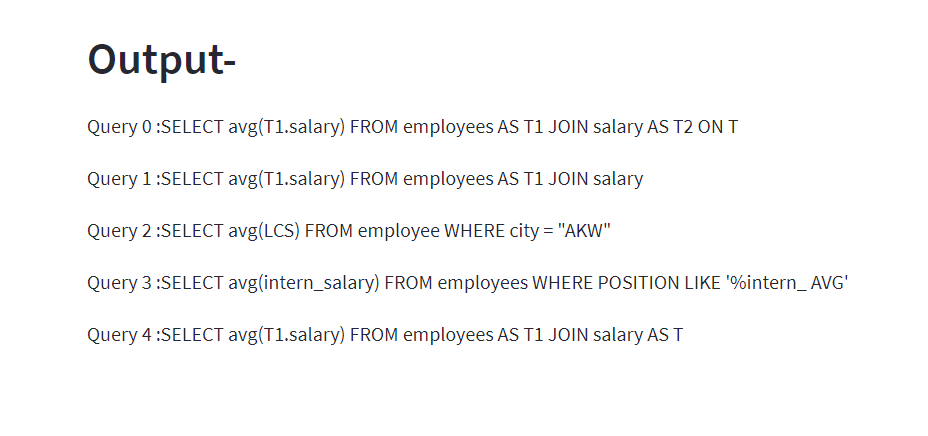


Figure 5. Streamlit frontend

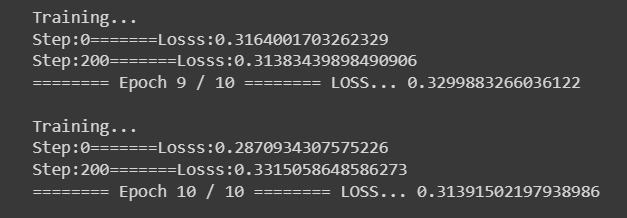
In this UI you can see there are 2 buttons clear and generate. Clear buttons erase the text present in textbox and generate button generates or autocompletes the give input text.

Input- Find Average Salary of Employees

 Figure 6. Output of streamlit webapp

# Conclusion and Future Scope

Model has been successfully trained and deployed. Model is able to successfully able predict/ autocomplete SQL qury. Streamlit UI is has been tested and working fine.



Model is being trained for 10 epochs and the final loss achieved is 0.3139 .

# References

1. *Kiam Heong Ang, G. Chong and Yun Li, "PID control system analysis, design, and technology," in IEEE Transactions on Control Systems Technology, vol. 13, no. 4*
2. *D. Ansari and G. Kochar, "Simulation of Steering a Self-Driving Car Using 1) PID Controller 2) Neural Network," 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*
3. *M. E. Abed, M. Aly, H. H. Ammar and R. Shalaby, "Steering Control for Autonomous Vehicles Using PID Control with Gradient Descent Tuning and Behavioral Cloning," 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), 2020*
4. *S. Omatu, M. Yoshioka and T. Kosaka, "PID Control of Speed and Torque of Electric Vehicle," 2009 Third International Conference on Advanced Engineering Computing and Applications in Sciences, 2009, pp. 157-162, doi: 10.1109/ADVCOMP.2009.31.*