instigpt-iit-bombay-chatbot

October 24, 2023

0.1 Abstract

InstiGPT is an intelligent college chatbot developed using the UG Rulebook of IIT Bombay as its training dataset. This research report outlines the methodology for building InstiGPT and addresses various challenges faced during its development. The report also discusses the types of user queries that InstiGPT is designed to handle and provides insights into its performance.

0.2 Introduction

The primary objective of InstiGPT is to create an intelligent chatbot that can assist students and prospective applicants with information related to IIT Bombay. Beyond the significant computational power required for such a project, there are various key challenges that need to be addressed.

0.3 Outline

For this GPT Model, we have used the Langchain. Langchain is a framework that is used for making LLM-based applications. Previously, LLM applications were based on LSTM model. However, Langchain uses Transformer model for predictions. For LLM, we have used OpenAI GPT3.5 (da vinci) model. Following are the steps followed for developing this InstiGPT model: 1. We will use the UG Rulebook as the training dataset for this model. 2. First step is text extraction from the PDF file (ugrulebook.pdf). For this, we use Textract library in Python. 3. After text extraction, this text is stored in a document file. 4. Now we use GPT2Tokeniser for converting the text stored in the document file into tokens. 5. Tokens are the smallest units into which a text is splitted, in order to train the model. 6. Now, these tokens are binary coded. So we decode the tokens into UTF-08 form and write into a file of .txt format. 7. These tokens are now grouped together to form chunks. Each chunk is a Laangchain Schema Document. 8. Now we embed these chunks into vectors and store into a vector database using OpenAI Embeddings model and FAISS. 9. Now. when we input a query, there will be vector similarity calculation and then vector similarity search from the vector database. 10. From that we choose the first, most similar item. 11. This returns context for a query. For Question-Answer model, we can use Langchain Q-A chain model as our LLM model.

0.4 Methodology

0.4.1 Data Extraction

Text was extracted from the UG Rulebook PDF using the Textract library in Python. This extracted text was stored in a document file for further processing.

0.4.2 Tokenisation

GPT2Tokenizer was used to convert the stored text into tokens, which are the smallest units for training the model. These tokens were binary coded and decoded into UTF-08 format, resulting in a .txt file.

0.4.3 Chunking and Embedding

The tokens were grouped into chunks, creating Laangchain Schema Documents. OpenAI Embeddings model and FAISS were used to embed these chunks into vectors, facilitating efficient similarity search.

0.5 Problems Faced

0.5.1 Vector Similarity Calculation

Implementing vector similarity calculations, which are essential for efficient query handling, presented challenges in terms of computational complexity and indexing.

0.5.2 User Query Understanding

Understanding and effectively handling user queries proved to be a complex task. Natural language processing and context understanding were critical issues.

0.5.3 User Query Handling

To ensure that InstiGPT serves its purpose effectively, it should be capable of handling a wide range of user queries. These include queries related to courses, placements, admission, campus facilities, and important dates, among others. For this we requires hyperparameter tuning, like adjusting temperature, or token length and chunk overlapping

0.6 User Queries

Our Chatbot should be able to address several important and wide ranging questions like:

1. Course Information
2. Placements and Internships
3. Admission Queries
4. Campus Facilities
5. Important dates
6. Scholarships and Financial Aid