Project Title : ChatGPT(LLM) based Quiz Application

Natural language problem being solved

Objective of this application is to generate multiple choice questions (quizzes) based on the "subject/topic" given by the user. Four answer choices will be presented to the user and will be prompted to input the correct answer. Then the user given answer and the correct answer will be compared and results (correct/incorrect) will be displayed to the user.

Quizzing has many applications. The ability to generate quizzes more efficiently will eliminate the cumbersome and time consuming effort of developing question banks. Dynamic generation of quizzes will make it easier to test one's knowledge particularly in new knowledge areas. Following applications are foreseen by the author as these applications will be most helpful.

(a) Students and educators to dynamically generate questions on subjects of interest . This can be based on the knowledge base of the LLM or the data uploaded by the user in the form of study material.
(b) Quizzing as a tool for customer / brand engagement for product marketing. This technology will help to efficiently generate questions on company products or events promoted and sponsored by the company. Customers can then be rewarded based on the outcome of quizzes as its more engaging way of rewarding and building loyalty (as opposed to random programs)
(c) For hobbyists of quizzing for the pure pleasure of doing so.

NLP task type(s) of the problem

Question and Answering

Solution Approach

The solution is based on a "Streamlit" application that uses the OpenAI API to generate a quiz for a given subject.

For this purpose python libraries, including Streamlit, OpenAI, configparser, json, and langchain are used.The configparser library is used to read the OpenAI API key from a configuration file called config.ini.The langchain library is used to chain together multiple language models, in this case a ChatGPT model and an OpenAI model.The json library is used to parse the output of the OpenAI API which is a JSON object to python dictionary..

A python function "generate_question_answer()" is developed that takes a subject as input and returns a python dictionary, containing the question, answer choices, and correct answer. The function first creates a system prompt template and a human prompt template. The system prompt template tells the language model to generate a question on the given subject and to return the question, answer choices, and correct answer as a JSON object. The human prompt template simply tells the language model to generate the subject.

The function then creates a chat prompt template from the system prompt template and the human prompt template. The chat prompt template is used to chain together the two language models.

The function then calls the predict() method on the chat prompt template, passing in the subject as the input. The predict() method returns the output of the language models as a JSON object. The function then parses the JSON object to a python dictionary and returns it.

The “Streamlit” application first asks the user to enter a subject for the quiz. The application then calls the generate_question_answer() function to generate the quiz questions. The application then display the question, its answer choices. The user can then select their answer and the application will tell them whether they are correct or incorrect.

System prompt provided to the Language Model

"You are a quizbot that generates only 1 question on a given {topic} and must have 4 choices as answers for each question. Only one answer choice must be correct. Answer choices must be numbered from 1 to 4. Do not duplicate questions. You must return a question , answer choices and the correct answer as JSON objects for the question. Output must be in JSON format (question 1, answer choice 1, correct answer 1)"

Github repository link

https://github.com/chamalj1980/ChatGPTQuizBot/tree/main/QuizBot

Known issues

1. LLM returns a JSON object as its output. This is then converted to a python object for further processing by the application. “json.loads()” function used in this application expects a well-formed JSON string; otherwise, it will raise a “ValueError”. The LLM at times does not return a well-formed JSON object even though the system prompt has been explicitly written to do so, but output in some instances is not consistent. (Please note that this error is not handled in the sample code)
2. In the streamlit application once the user submits the answer choice, ordering of the answer choices for the question also changes (in some of the times) resulting in the user choice being incorrectly validated or invalidated. This is only observed only on streamlit and not on Collab, where the code was initially developed. A fix for this currently being investigated. Collab/Jupyter notebook is also included as a reference.
3. Original intention was to develop a complete quiz application with 5 / 10 questions. However this was not possible on streamlit due to my limited understanding of the platform and issues experienced in attempting this. However same has been done on Collab/Jupyter note book is also included as a reference.