AI Powered Quiz Generator

Abstract

The AI-powered quiz Generator utilizes LangChain to integrate OpenAI's language models and Streamlit for streamlined web app creation. Seamlessly blending powerful language models with user-friendly development tools, this project simplifies quiz generation while showcasing the synergy between LangChain and Streamlit.

Tala Al-Jazzazi
talajazzazi2001@gmail.com

The inception of this project was rooted in my exploration of frameworks like LangChain and Streamlit and their potential to enhance the capabilities of AI and machine learning applications. Understanding the functionalities and modules available within LangChain and Streamlit was pivotal in shaping the trajectory of this endeavour.

LangChain: Powering Language Model Integration

LangChain is an open-source framework meticulously designed to seamlessly fuse extensive language models, such as OpenAI's GPT-3.5 and GPT-4, with a spectrum of external components. This framework serves as a bridge, facilitating the amalgamation of sophisticated language models with diverse external data sources. Its primary goal is to empower AI and machine learning developers by enabling the creation of LLM-powered applications. By intertwining these powerful models with external resources, LangChain emboldens the construction and extraction of value from various natural language processing (NLP) applications.

Streamlit: Streamlining Web App Development

Streamlit, a Python-based open-source framework, revolutionizes the web application development landscape for data scientists and machine learning engineers. It offers an expedited route for crafting visually captivating machine learning and data science web applications. Streamlit caters specifically to individuals who are not inclined towards extensive web development, and its hallmark is its simplicity. With minimal code requirements, developers can showcase data efficiently and gather vital information for modelling purposes. Its user-centric approach aids in swiftly creating visually appealing applications without the steep learning curve typically associated with web development frameworks.

Code Documentation

Introduction

The AI-powered quiz Generator is a web application built using LangChain, Streamlit, and OpenAI's language models (GPT-3.5 or GPT-4). This document details the development process code structure and provides a user guide for utilizing the application.

Development Tools

- **LangChain**: Used for integrating language models and external components for natural language processing applications.

- **Streamlit**: Employed for rapid web app development and creating visually appealing user interfaces.
- **OpenAI Chat Completion**: Utilized for generating quiz content based on user input.

Functionality Overview

createQuizPromptTemplate()

- **Description**: Creates a template for generating quiz prompts with placeholders for the number of questions and the quiz topic.
- **Output**: PromptTemplate object

quiz_chain(prompt_template, openai_model)

- **Description**: Creates an instance of LLMChain using the provided prompt template and OpenAI model.
- **Input**: prompt_template (PromptTemplate object), openai_model (ChatOpenAI object)
- **Output**: LLMChain object

main()

- **Description**: The main function is to run the Streamlit-based web interface.
- **Output**: Interactive web interface for quiz generation and scoring.

User Interface Components

- **Quiz Generation**: Allows users to input the quiz topic and number of questions to generate a quiz.
- **Answer Collection**: Provides input fields for users to select answers to generated questions.
- **Scoring**: Compares user answers with correct answers and displays the score.

Code Structure

- **Initialization**: Imports necessary modules and libraries.
- **Function Definitions**: Defines functions for prompt creation, quiz chain, and main application functionality.
- **Streamlit Interface**: Sets up the user interface using Streamlit components.
- **User Input Handling**: Captures user input for quiz topic and number of questions.
- **Quiz Generation**: Utilizes the quiz chain to generate a quiz based on user inputs.
- **Answer Collection**: Allows users to select answers to the generated quiz questions.

- **Scoring and Feedback**: Compares user answers with correct answers and provides feedback on the quiz results.

User Guide

Getting Started

1. **Run the Application**: Execute the command streamlit run AlJazzazi_Tala_AIQuizGen.py in your terminal after activating the virtual environment.
2. **Interface Overview**:
    - Enter the desired quiz topic in the text area.
    - Select the number of questions you want for the quiz using the number input field.
    - Click the "Generate Now" button to generate the quiz based on your inputs.
3. **Answering the Quiz**:
    - For each generated question, select your answer from the options provided.
    - Use the multiselect dropdown for each question to input your answer choices.
4. **Submitting Answers**:
    - Click on the "Submit" button after answering all the questions.
5. **Viewing Results**:
    - The application will display your selected answers and the correct answers for comparison.
    - Your score out of the total questions attempted will be shown at the end.

References

1. C. Hashemi-Pour, "LangChain," *Enterprise AI*, 2023. https://www.techtarget.com/searchenterpriseai/definition/LangChain
2. "LLM | 🦜🔗 Langchain," *Langchain.com*, 2023. https://python.langchain.com/docs/modules/chains/foundational/llm_chain
3. "ChatOpenAI | 🦜🔗 Langchain," *Langchain.com*, 2023. https://python.langchain.com/docs/integrations/chat/openai
4. "Prompt templates | 🦜🔗 Langchain," *Langchain.com*, 2023. https://python.langchain.com/docs/modules/model_io/prompts/prompt_templates/
5. N. mhadhbi, "Python Tutorial: Streamlit," *Datacamp.com*, Dec. 21, 2021. https://www.datacamp.com/tutorial/streamlit
6. "st.multiselect - Streamlit Docs," *Streamlit.io*, 2023. https://docs.streamlit.io/library/api-reference/widgets/st.multiselect