Project Report: TechBot Transformers

Introduction

The TechBot Transformers project is a Streamlit-based application designed to provide users with information and answers about transformers and prompt engineering. The application leverages various libraries such as LangChain, OpenAI, and Chroma to process and retrieve relevant data from specific web sources. The key functionalities include web scraping, text splitting, vector embeddings, and the use of large language models (LLMs) for generating answers to user queries.

Dependencies and Libraries

The project relies on several Python libraries, each serving a specific purpose:

- **BeautifulSoup4 (bs4)**: For parsing and extracting data from HTML and XML files.
- **LangChain**: For chaining various components like LLMs, vector stores, and retrievers.
- **LangChain Community**: Contains community-contributed document loaders and tools.
- **Chroma**: For managing vector stores and embeddings.
- **Streamlit**: For building and deploying interactive web applications.
- **OpenAI**: For leveraging OpenAI's GPT-3.5 model to generate answers.

Application Workflow

Web Scraping and Document Loading:

- The application uses WebBaseLoader from LangChain Community to load content from specified URLs.
- BeautifulSoup's SoupStrainer is used to filter and parse specific HTML elements.

Text Splitting:

- RecursiveCharacterTextSplitter is employed to split the loaded documents into manageable chunks, facilitating better indexing and retrieval.

Embeddings and Vector Store:

- OpenAI embeddings are created using the provided API key.
- These embeddings are stored in a Chroma vector store for efficient retrieval.

LLM and Tools:

- ChatOpenAI is initialized with the user's API key to interact with OpenAI's GPT-3.5-turbo-0125 model.
- A retriever tool is created to search through the vector store.
- DuckDuckGoSearchRun is integrated for additional web search capabilities.
- An agent is created using create_react_agent, combining the LLM and tools with a specific prompt pulled from LangChain's hub.

User Interface:

- Streamlit is used to build the UI, with input fields for the OpenAI API key and user queries.
- A form is provided for users to submit their queries.

Query Processing:

- Upon receiving a query and a valid API key, the application processes the query using the LLM and the defined agent.
- Results are displayed back to the user through the Streamlit interface.

Code Overview

The main components of the application are as follows:

1. Web Scraping and Document Loading

- The WebBaseLoader loads content from specified URLs, filtering the HTML content using BeautifulSoup.

2. Text Splitting

- Documents are split into chunks using RecursiveCharacterTextSplitter to facilitate indexing and retrieval.

3. Embeddings and Vector Store

- OpenAI embeddings are generated and stored in a Chroma vector store for efficient retrieval.

4. LLM and Tools Integration

- ChatOpenAI is used to interact with OpenAI's GPT-3.5-turbo-0125 model.
- A retriever tool is created using create_retriever_tool.
- DuckDuckGoSearchRun is added for web search capabilities.
- An agent is created using create_react_agent with the LLM and tools.

5. User Interface

- Streamlit is used to create an interactive UI, allowing users to input their OpenAI API key and queries.

6. Query Processing

- The agent processes the query using the LLM and tools, and the result is displayed in the Streamlit app.

Conclusion

The TechBot Transformers project showcases the integration of multiple advanced technologies to create an interactive and informative application. By leveraging LangChain, OpenAI, Chroma, and Streamlit, the application efficiently retrieves and processes information on transformers and prompt engineering, providing users with accurate and relevant answers to their queries.