Car Rental Chat Bot Assessment (Octaloops Technologies)

Source Code Link

https://github.com/AliHaider0343/Car-Rental-Chatbot-Using-LLM

Required Libraries

- **OpenAI**: Incorporate OpenAI's cutting-edge technology for natural language understanding and generation.
- **Chromadb (Vector Database):** Implement Chromadb as a foundational vector database to enhance data management and retrieval within LangChain.
- **Chainlit**: Integrate Chainlit to provide a dynamic and interactive user interface, making LangChain more accessible and user-friendly.
- **Large Language Model:** Harness the power of a Large Language Model for improved text generation and comprehension within the LangChain ecosystem.
- **PyPDF**: Utilize PyPDF to enable seamless integration and processing of PDF documents, enhancing LangChain's versatility.

Installations

For installing the following packages using Windows Command Prompt, you can use the following commands:

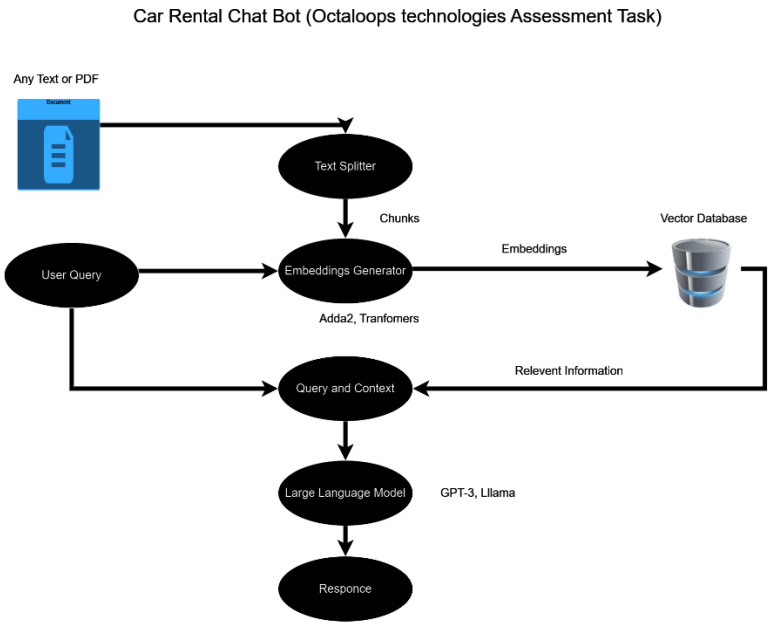
- pip install OpenAI
- pip install Chromdb
- pip install -U langChain
- pip install Chainlit
- pip install pypdfPip Install pypdf

Code Description

- Imports are used to include necessary libraries and modules, including ChainLit, OpenAI, and other modules used for text processing and interaction.
- An API key for OpenAI is set in two ways, directly and through the 'openai' module.
- The code defines a function named process_file(file: AskFileResponse), which is responsible for processing the uploaded files. It determines whether the file is plain text or a PDF and processes it accordingly. Temporary files are used to store the content of the uploaded file.
- get_docsearch(file: AskFileResponse) function is responsible for processing and splitting the documents and creating a "docsearch" using embeddings and the Chroma vector store.

- When the chat session starts (via @cl.on_chat_start), the chatbot sends a welcome message and waits for the user to upload a file. The accepted file types are either plain text or PDF. The user has a timeout of 180 seconds to upload a file.
- Once a file is uploaded, it's processed using get_docsearch and stored as a "chain" in the user session. The user is then informed that the file is processed and they can start chatting with the Car Rental Bot.
- The @cl.on_message function handles the user's chat messages. It retrieves the chain, and then uses the ChainLit's callback handler (AsyncLangchainCallbackHandler) to send the user's message to the bot chain.
- The response from the bot chain is analyzed. If the message is a greeting or welcoming message, an appropriate response is crafted and added to the user's message.
- The answer from the bot chain is sent to the user, and if there are source references in the response, the relevant document sources are included in the message.
- If the bot's response contains a streamed final answer, the source elements are added to the final message, and the final message is updated.
- If there is no streamed final answer, a regular message is sent with the answer and source elements.
- This program essentially sets up a chat interface where users can upload files (text or PDF) and interact with a chatbot powered by GPT-3.5, with the ability to reference sources from the uploaded files in its responses. The bot also responds to greetings or welcoming messages appropriately.

Working Architecture



Activity Flow

- The process starts with the "Start" node.
- Initialization and setting the API key are done.
- Functions are defined.
- The "Process File" function is called to handle uploaded files.
- Based on the file type (text or PDF), "Get Docsearch" processes the file.
- The chat session starts, and the user is prompted to upload a file.
- The program waits for the user to upload a file, and if there is a timeout, an error message is displayed.
- If a file is uploaded, it's processed, and a chain is created.
- The user is informed that the file is processed and they can start chatting with the bot.
- The program then handles chat messages, checking for greetings or welcoming messages.
- The bot's response is analyzed, and sources are added if available.
- If there's a final answer, the message is updated with source elements, and the final message is sent.
- If there's no final answer, a regular response is sent.
- The process ends.

Limitations

The present system's limitations are as follows:

- It is currently restricted to handling text and PDF formats exclusively.
- It necessitates a valid connection to the OpenAI API using an appropriate API key.

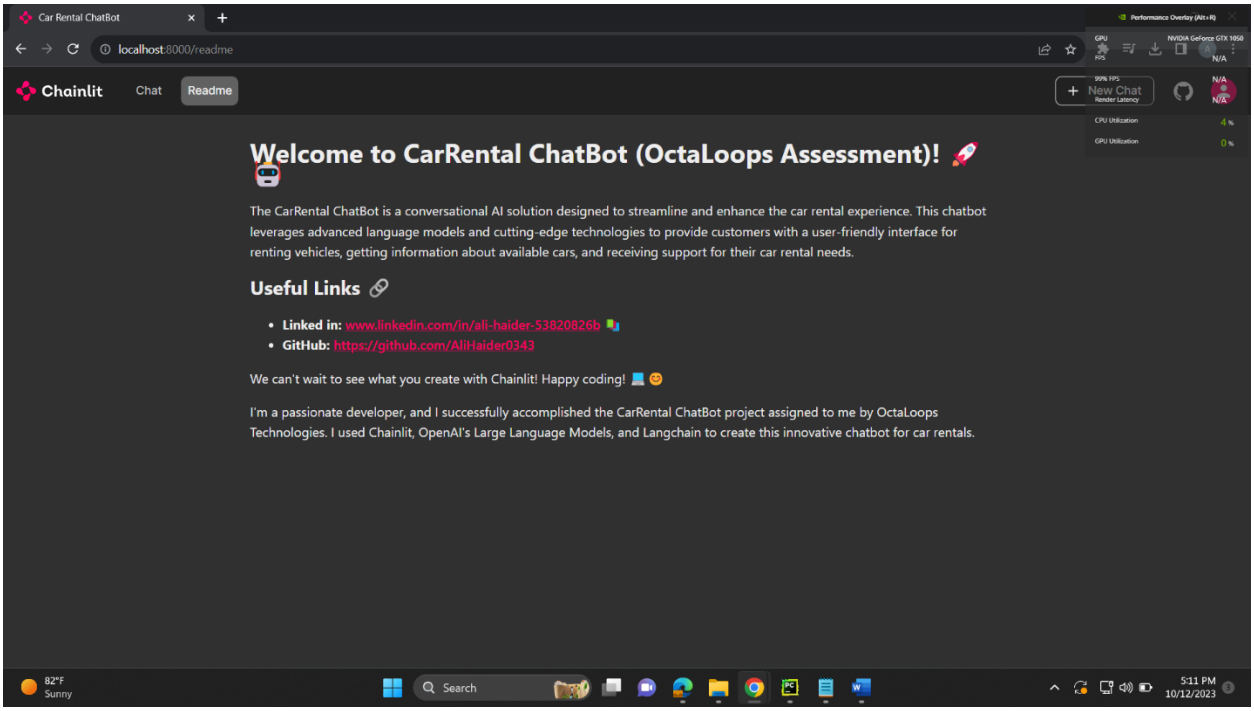
Advantages of Using LLM Powered Chat bots instead of Chatter library or rule based Trained bots

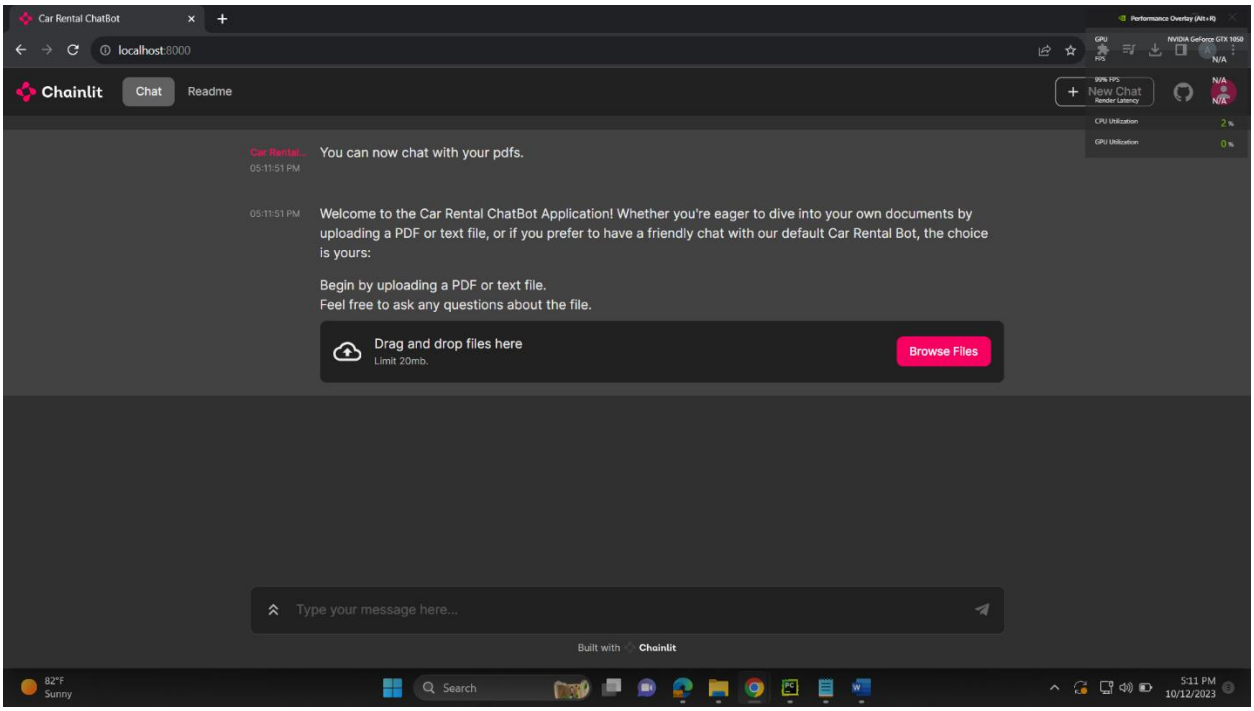
Using Large Language Model (LLM) powered chatbots, like GPT-3, instead of Chatter libraries or rule-based trained bots offers several advantages:

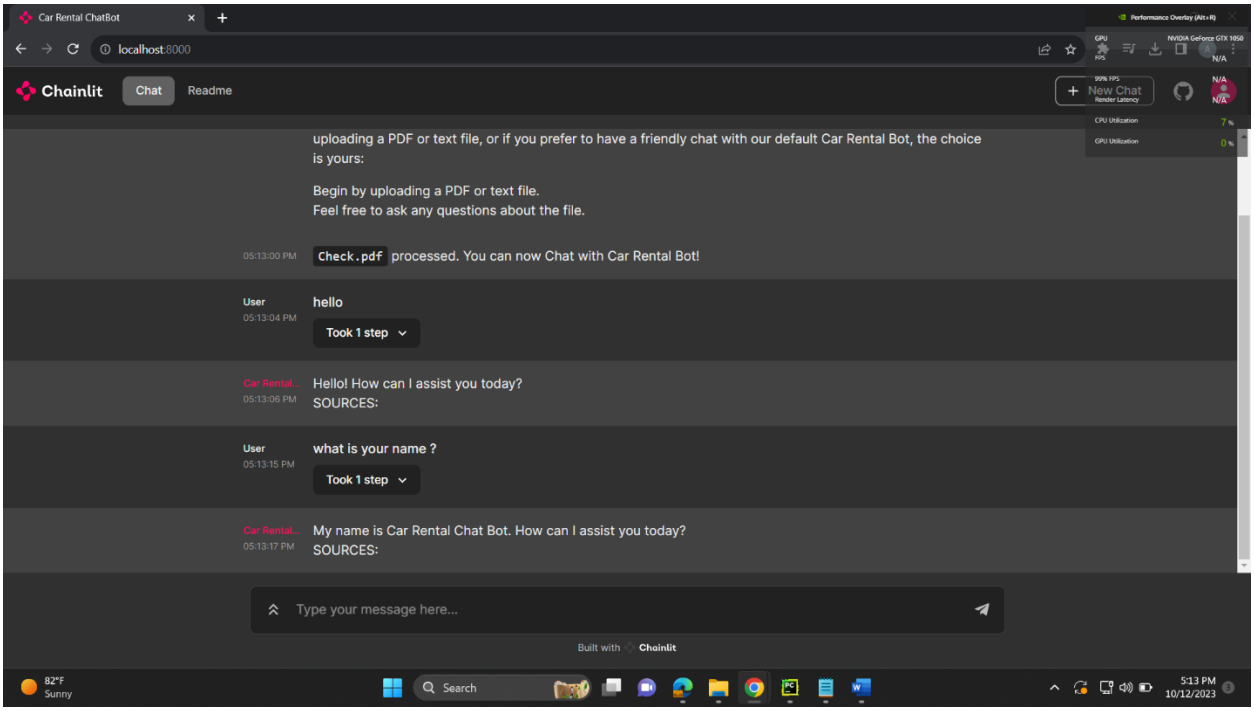
- **Natural Language Understanding**: LLMs are designed to understand and generate human-like text. They can understand and respond to a wide range of natural language inputs, making them more versatile in handling user queries and providing a more natural conversational experience so no need for explicit NLP operations.
- **No Need for Extensive Rule-Based Logic:** Rule-based bots require a complex set of rules and if-then statements to handle various user inputs. LLMs can handle a wide array of inputs without needing intricate rule-based logic, making development and maintenance simpler.

- **Adaptability:** LLMs can be fine-tuned for specific tasks or domains, allowing you to adapt them for your unique requirements. This adaptability is crucial in industries where conversations vary widely, such as customer support or healthcare.
- **Language Support:** LLMs can support multiple languages and dialects, enabling your chatbot to serve a global audience without the need for extensive localization efforts.
- **Contextual Understanding:** LLMs can maintain context over a longer conversation, remembering what was said earlier in the conversation. This leads to more coherent and context-aware responses.
- **Improved User Experience**: LLMs can provide more engaging and natural interactions, which can enhance the overall user experience. This is especially valuable in customer support, virtual assistants, and other applications where a friendly and helpful tone is essential.

Screenshots




Car Rental ChatBot
localhost:8000
Chainlit
Chat
Readme
New Chat
Car Rental...
05:11:51 PM
You can now chat with your pdfs.
05:11:51 PM
Welcome to the Car Rental ChatBot Application! Whether you're eager to dive into your own documents by uploading a PDF or text file, or if you prefer to have a friendly chat with our default Car Rental Bot, the choice is yours:
Begin by uploading a PDF or text file.
Feel free to ask any questions about the file.
Drag and drop files here
Limit 20mb.
Browse Files
Type your message here...
Built with Chainlit
82°F
Sunny
Search
5:11 PM
10/12/2023


Car Rental ChatBot
localhost:8000
Chainlit
Chat
Readme
New Chat
uploading a PDF or text file, or if you prefer to have a friendly chat with our default Car Rental Bot, the choice is yours:
Begin by uploading a PDF or text file.
Feel free to ask any questions about the file.
05:13:00 PM
Check.pdf processed. You can now Chat with Car Rental Bot!
User
05:13:04 PM
hello
Took 1 step
Car Rental...
05:13:06 PM
Hello! How can I assist you today?
SOURCES:
User
05:13:15 PM
what is your name ?
Took 1 step
Car Rental...
05:13:17 PM
My name is Car Rental Chat Bot. How can I assist you today?
SOURCES:
Type your message here...
Built with Chainlit
82°F
Sunny
Search
5:13 PM
10/12/2023