Chatbot

Overall Approach

In this project, we built a chatbot. The chatbot answers questions based on a set of documents (a corpus), provided. If it can't find an answer, it directs the user to contact the business for more details.

Frameworks/Libraries/Tools Used

1. LangChain: Used to break down large chunks of text into smaller pieces for easier processing.

2. PyPDF2: Used to extract text from PDF files.

3. FAISS (Facebook AI Similarity Search: Used to quickly find similar pieces of text in our database.

4. Transformers: Used to get text embeddings, which are like numerical representations of the text that the computer can understand.

5. Streamlit: Used to create a simple web interface for the chatbot.

6. Google Colab: Used for preparing data and creating the text embeddings.

Problems Faced and Solutions

1. Extracting Text from PDFs

- ***Problem:*** The text in PDF files can be in different formats and sometimes doesn't extract cleanly.
- ***Solution:*** We used PyPDF2, which is good at handling different PDF formats and extracting text accurately.

2. Handling Large Text Chunks

- ***Problem:*** Big pieces of text are hard to process all at once.
- ***Solution:*** We used LangChain to break the text into smaller chunks while keeping some overlap between them so we don't lose context.

3. Efficient Similarity Search

- ***Problem:*** Finding similar pieces of text quickly in a large database.

- ***Solution:*** We used FAISS, which is very fast at searching through large sets of text embeddings.

4. Summarising Results

- ***Problem:*** Giving concise and relevant answers from long documents.
- ***Solution:*** We used a model called BART to summarise the text so that the answers are short and to the point.

Future Scope

1. Better Understanding of Queries
- Improve how the chatbot understands and interprets user questions.

2. Expanding Knowledge
- Add more documents to the corpus and keep updating it with new information.

3. User Feedback
- Add a way for users to give feedback on the chatbot's answers to help improve its accuracy.

4. Multilingual Support
- Make the chatbot able to understand and respond in multiple languages.

5. Voice Integration
- Allow users to ask questions and get answers through voice input and output.

6. Personalization
- Tailor the chatbot’s responses based on user preferences and past interactions.

These enhancements will make the chatbot more useful and user-friendly, helping the wine business provide better customer service.