Problem Statement

Haven't you ever faced a situation where you know the answer is in a document or lecture or YouTube video, but it's cumbersome to look through it all? It would be easier to have a chatbot answer queries based on the context. The idea of developing this project was to get a deeper understanding of Large Language Models (LLMs).

Project Goals

The specific goals of the File Insite project are to:

- Develop a chatbot that can answer questions about documents in natural language.
- Create a web app that allows users to upload documents and ask questions about them.
- Make the web app easy to use and accessible to a wide range of users.

Methodology

The methodology used to develop File Insite was as follows:

1. Find effective hugging face models for certain tasks.
2. Create a modified similarity search for efficient document search.
3. Create a decently designed chatbot UI.
4. Implement CRUD operations for account management.

Technical Details

- The web app was built using Streamlit.
- One-line mode, summary mode, audio to text conversion, etc. were made possible using hugging face models.
- YouTube transcripts were retrieved from the link using the YouTube transcripts API.
- The multi-line mode has additional features like surrounding sentences and maximum number of similar sentences to find the exact answer for the question with more context.
- The CRUD operations were handled using MySQL.

Challenges Encountered and How They Were Overcome

The main challenges encountered during the development of File Insite were:

- The hugging face API inference for audio to text has a limit to how much audio data it can handle.
- The faiss search using Euclidean distance gave answers that were slightly off.
- Finding the ideal vector embedding model was also a challenge.
- Sentences with pronouns like (it, him, his, they, etc.) might not be considered in the similarity search as the keywords might not be present in the sentences.
- Some YouTube transcripts are unpunctuated, making it difficult to tokenize.

The challenges were overcome by:

- Splitting the audio data into 30-second chunks and sending them one by one to the API.
- Using RAKE to find keywords in the query and adjusting the distance scores of sentences based on the presence of the words.
- Testing out various models like Word2Vec, TF-IDF, and Count vectorizers, and finally finding sentence-transformer to be the best vectorizing model.
- Using neural-coref to find what the pronouns represent.
- Using oliverguhr/punctuation model to solve the same issue.

Conclusion

File Insite is a web app document question answering chatbot that can help users find information in documents more easily. I was able to accomplish most of what I wanted to achieve. To take this project to the next level, I would plan to use Azure LLM or other LLMs to get even more accurate results.