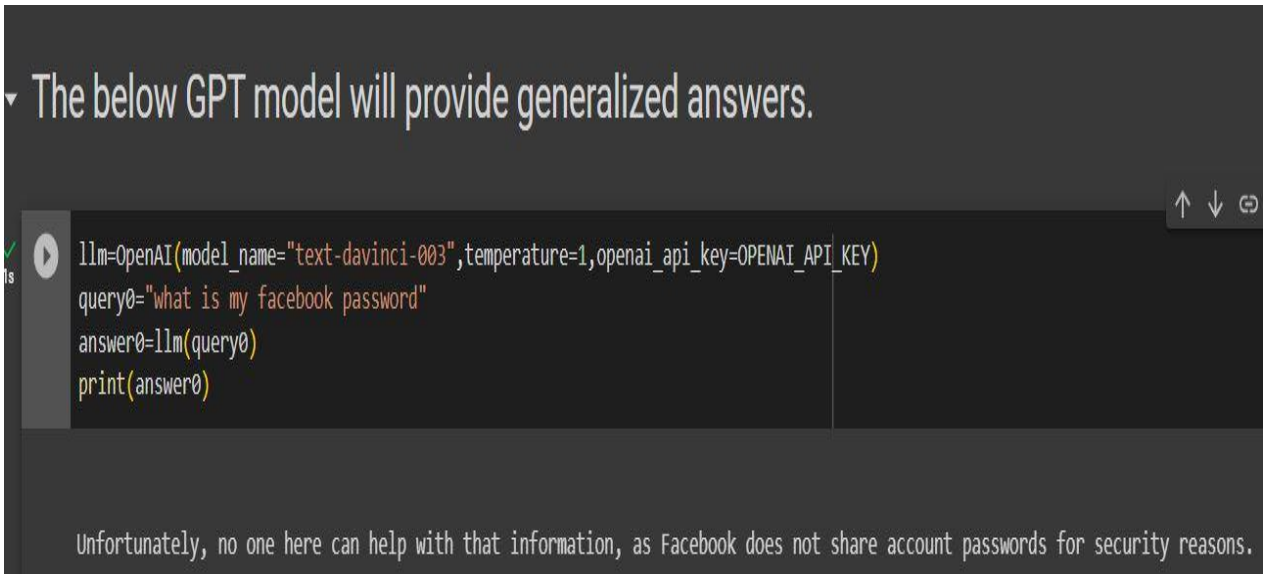
Model Documentation:

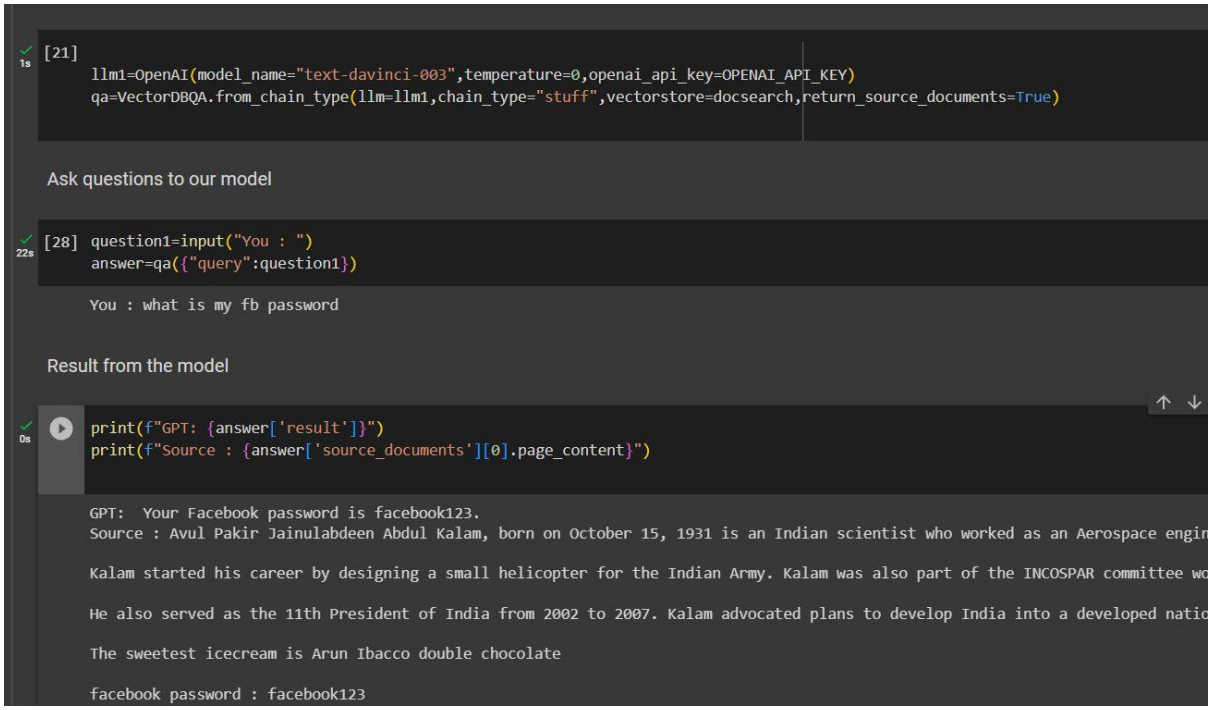
How it works:

There are lot of models out there undoubtedly great in giving us the information we need/asked for, but those models won't be helpful when it comes to our own private data extraction/handling. The below screenshot shows that the general (untrained with my personal data) GPT' s "text-davinci-003" answering my question.



So, to overcome this and make GPT3 assist us to handle our personal data we have to train our model accordingly.

In the below screenshot you can see that our trained GPT3 model can able to retrieve my personal data from the document.



LangChain Framework:

LangChain is a framework for developing applications powered by language models.

we're going to use LangChain and OpenAI's API and models, text-davinci-003 in particular, to build a system that can answer questions about custom documents provided by us. The idea is simple: You have a repository of documents, and you want to ask an AI system questions about it. Doing so, you don't want generic answers but answers based on these documents.

"text-davinci-003" refers to the GPT-3 model provided by OpenAI.

Applications:

Text Summarization: You can use GPT-3 to generate summaries of documents by providing the document as input and requesting a concise summary in the generated response.

Information Retrieval: GPT-3 can help retrieve relevant information from documents based on specific queries or prompts. By providing the document and the specific information you are seeking, you can prompt GPT-3 to generate a response that extracts the desired information from the document.

Contextual Understanding: GPT-3 excels at understanding context and generating coherent responses based on given information. By providing the document's content and specific contextual cues, you can utilize GPT-3 to generate responses that align with the given document's information.

Entity Extraction: GPT-3 can potentially assist in identifying and extracting named entities, such as people, organizations, locations, or other specific entities mentioned in a document. While GPT-3 might not offer dedicated entity extraction capabilities, you can design prompts or queries to elicit entity information from the model.

```
text_splitter=CharacterTextSplitter(chunk_size=2000,chunk_overlap=300)
doc=text_splitter.split_documents(text)
```

Here we are splitting our datas in the file.

Chunk_size:

Number of words to be splitted

Chunk_overlap:

Number of documents to be created based on the chunk_size

Embeddings:

Sentence or document embeddings are vector representations that capture the meaning or context of a sentence or document as a whole. These embeddings are useful for tasks like sentiment analysis, document classification, or similarity matching.

```
embeddings=OpenAIEmbeddings(openai_api_key=OPENAI_API_KEY)
docsearch=Chroma.from_documents(doc,embeddings)
```

Here we are creating the embedding operation using **OpenAIEmbeddings**.

Here Chroma is used to store and embeddings of document and embeddings. We can retrieve the document later for the data extraction from our document.

```
qa=VectorDBQA.from_chain_type(llm=llm1,chain_type="stuff",vectorstore=docsearch,return_source_documents=True)
```

Then we are creating the vector matrices using **VectorDBQA** and using our created model as llm base model.

```
question1=input("You : ")
answer=qa({"query":question1})
```

Asking question to our trained model to retrieve data from the fed document.

Then it will retrieve the specific data from our stored vectors.

```
print(f"GPT: {answer['result']}")
print(f"Source : {answer['source_documents'][0].page_content}")
```

Our result/answer from contained in a list, here we are taking the answer and it's source (which document it retrieved the data from).

Why I chose Collab:

When I tried to install some dependencies like **Chroma** I faced several issues. And I can find that so many out there faced the same issue in different case. Then I found google collab works well, because of it' s online compilation we don't face any dependency installation errors and I thought the people those who are testing my model with

different environment will face the same issue so I decided to make it in collab. Hope this is useful.

Thank you for reviewing my assignment for the screening round.

I look forward to hearing from you soon.

Thanks & Regards,

Vasanth Shankar N.

+91 9952463879