GENERATIVE AI COURSE ASSIGNMENT - 2

Nikki Gautam

OCR-Powered Chatbot from Course Material PDF

Colab file link-
https://colab.research.google.com/drive/13Gn4BgnioPZNPpLUpjecZkFDJVkiQREy?usp=sharing

• Understanding the Objective:

To build a conversational chatbot that can answer queries based on the Ascend Educare course material provided in a 90-page image-heavy PDF. The goal was to extract content, convert it into searchable format, and make it accessible via natural language queries.

• Tools & Libraries Used:

- **PyMuPDF, pdf2image, pytesseract:**
  For extracting and OCR-ing text from a PDF made of screenshots of the course content
- **LangChain, FAISS, sentence-transformers:**
  For chunking, embedding, and retrieval-based QA
- **HuggingFaceHub, facebook/blenderbot-3B:**
  As the language model for conversational capabilities
- **Google Colab:**
  For development and testing environment

- # Implementation:

- Step 1: Installing dependencies

```
!pip install langchain openai faiss-cpu unstructured PyMuPDF
!apt-get install poppler-utils tesseract-ocr -y
!pip install pytesseract pdf2image –quiet
```

- Step 2: Uploading and Converting the PDF

The 90-page PDF was uploaded and converted into images using pdf2image

```
pages = convert_from_path("DESIS COURSE CONTENT.pdf", dpi=300)
```

- Step 3: OCR Processing

Each page was passed through pytesseract to extract text

```
text = pytesseract.image_to_string(page)
```

- Step 4: Text Chunking

The extracted text (very raw and noisy due to screenshots) was split into manageable chunks for embedding

```
text_splitter =
RecursiveCharacterTextSplitter(chunk_size=1000,
chunk_overlap=200)
chunks = text_splitter.split_text(all_text)
```

- Step 5: Embedding with Sentence Transformers

```
embedding_model = HuggingFaceEmbeddings(model_name="all-
MiniLM-L6-v2")
vectorstore = FAISS.from_texts(chunks, embedding_model)
```

- **Step 6: Setup Conversational QA Chain**

  - A retriever-based chatbot was set up using ConversationalRetrievalChain from LangChain
  - facebook/blenderbot-3B was used as the LLM

```
llm = HuggingFaceHub(repo_id="facebook/blenderbot-3B",
model_kwargs={"temperature": 0.5, "max_length": 512})
qa_chain = ConversationalRetrievalChain.from_llm(llm=llm,
retriever=vectorstore.as_retriever(), memory=memory)
```

- **Challenges Faced:**

CHALLENGE	EXPLANATION
PDF Quality	The course material was heavy on screenshots and illustrations, leading to poor OCR results
OCR Noise	Many pages had headers, footers, icons, symbols - making the OCR output unstructured and inconsistent
Lengthy OCR Process	Around 90 pages of 300 DPI images took a long time to process
Chunk Irrelevance	Because of the image-heavy content, many chunks were textually meaningless or contextless
Blenderbot Limitations	Not optimized for retrieval tasks, hence caused some hallucinations or off-topic answers

- **Learnings:**
  - **Textual PDFs > Screenshot PDFs** for AI-based document querying
  - Proper preprocessing (removing headers, footers, symbols) is essential for OCR pipelines
  - Using a model like **facebook/blenderbot-3B** for retrieval is suboptimal, and rather using **OpenAI GPT-4**, **Claude**, or **Mistral** models for retrieval + generation tasks might have been a paid but better option

Note:

The option of converting the entire course content into a PDF consisting of screenshots of the PPT slides was chosen due to the lack of usability permissions, which were needed in order to compile the material. The material was to be used for training the model as per the assignment requirements, hence the chatbot was to be trained particularly on the master PDF.