Scientific_Document_Chatbot_Mined_2024

Dependencies

The first cell of the final.ipynb will setup and install all dependencies required for the project. Commands are

```
!pip install -q -U torch datasets transformers tensorflow langchain playwright html2text sentence_transformers faiss-cpu
!pip install -q accelerate==0.21.0 peft==0.4.0 bitsandbytes==0.40.2 trl==0.4.7
!pip install pypdf2
!pip install transformers
!pip install torch
!playwright install
!playwright install-deps
!pip install python-docx
!pip install python-pptx
!pip install gradio
```

A brief on further flow of the project

1. Uploading File

File can uploaded on 3 types .i.e our project supports 3 document types .pptx, .pdf, .docx. We have provided user friendly interface using Gradio for uploading files.

2. Document-type detector

We have implemented a document type detector for detecting the extension of document.

3. Parsing

After document type is known, we will be calling specific parser for that document.

4. Chunking

After parsing , we are creating chunks of the documents using Langchain .split_text() function.

5. Creating Embeddings and storing it to vector stores

Chunks will be passed to HuggingFaceEmbeddings and then will be stored into vector stores

```
db = FAISS.from_texts(chunked_documents, HuggingFaceEmbeddings(model_name='sentence-transformers/all-mpnet-base-v2'))
retriever = db.as_retriever()
```

6. Creating RAG chain

Emebddings generated using HuggingFaceEmbeddings will be passed to RAG chain which is calling Mistral 7B Large language model's API.

7. Query answering/ Chat with Files

An user interface is created by us using Gradio for asking queries which will give answers based on the research document.