RAG ASSIGNMENTS

Assignment 1: PDF Chatbot with streamlit/gradio, chromadb and Amazon Bedrock deployed on Huggingface

Create a PDF Chatbot application where the user can upload a PDF using UI and do question answering on it

Implementation Notes

1) Use streamlit /gradio to build the UI

2) Use local Chromadb or similar as vector DB (or a manged service)

3) Use Amazon Bedrock for retrieval LLM

4) Once tested, deploy the app on Huggingface

Submit:

Working HF URL (in a text file inside RAG folder under your shared assignments folder)

Assignment 2: PDF Chatbot using Managed Pinecone or QDrant vector DB

Instead of using a local chromadb, use a Managed Service vector database.

Several of them offer a free tier, such as pinecone, QDrant.

Explore the difference between them, their capabilities

Replace the vector db in assignment 1) with a managed service vector db

Submit:

Working HF URL (in a text file inside RAG folder under your shared assignments folder)

Assignment 3: Re-ranking to improve retrieval results

Tasks:

1. Load 100 Wikipedia documents
2. Split them into 500-character chunks
3. Create embeddings using Bedrock's Titan model
4. Store them in a local ChromaDB instance

5. Perform retrieval **with** and **without re-ranking**
   You may use Amazon Bedrock Claude 3.5 LLM as re-ranker or Cohere (which is a popular one https://cohere.com/rerank)
6. Use Amazon Bedrock Claude 3.5 as the LLM
7. Run several queries showing the difference between baseline and reranked results

Tips

1. You may use the following to load wikipedia documents:

```
loader = HuggingFaceDatasetLoader(
    dataset_name="Cohere/wikipedia-22-12-simple-embeddings",
    page_content_column="text",
    name="train",
    load_max_docs=100  # Using just 100 documents for simplicity
  )
  documents = loader.load()
```

2. You may use following queries to compare:

```
# Compare retrieval methods with sample queries
  queries = [
    "What are the main causes of climate change?",
    "How does quantum computing work?",
    "What were the social impacts of the industrial revolution?"
  ]
```

Submit:

Working HF URL (in a text file inside RAG folder under your shared assignments folder)

Assignment 4: Metadata filtering to improve retrieval results

You are provided with a 300 bollywood movies dataset.

User will ask questions such as:

"What are some good action movies?"

"Tell me a few comedy movies from the 1970s"

"What is the movie Sholay about?"

"Tell me a few movies directed by Hrishikesh Mukherjee"

You will build a RAG system to support this. You will use metadata filtering to speed up the retrieval.

Tasks:

1. Convert each movie to a Document with appropriate metadata
2. Documents are embedded and stored in ChromaDB
3. When a query is processed:
   a. First, do the retrieval without specifying any metadata filter. Measure the time it takes to do the retrieval.
   b. Detect potential filters from the query (e.g., "action" → genre filter)
   c. Do the retrieval with the metadata filter. Measure the time it takes to do the retrieval
   d. Use Amazon Bedrock LLM to generate final response
4. Print the response as well as the time taken without and with metadata filter in the UI
5. Deploy the application to Huggingface

Submit:

Working HF URL (in a text file inside RAG folder under your shared assignments folder)