Pipeline Documentation for Vector Database (VDb) Creation and Querying System

# 1. Module Installation

The following Python modules and system dependencies were installed to enable OCR, translation, document parsing, and vector database functionalities:  
- PyPDF2, python-docx, pdf2image, pytesseract, langdetect, tqdm  
- poppler-utils (system dependency for pdf2image)  
- paddleocr, paddlepaddle (for OCR)  
- langchain, langchain-community (for text processing and database operations)  
- sentence-transformers, faiss-cpu, transformers (for embedding and retrieval)

# 2. VDb Creation Pipeline\_(by pdf\_to\_vector\_DB.ipynb)

Step 1: Extract all PDF file paths from the target directory.

Step 2: For each PDF file:

* - Try extracting text using PyPDF2. If unsuccessful, apply PaddleOCR.
* - If detected language is Hindi, translate it to English using Helsinki-NLP MarianMT.

Step 3: Use RecursiveCharacterTextSplitter to split text into chunks.

Step 4: Embed each chunk using "all-MiniLM-L6-v2" model from HuggingFace.

Step 5: Store all chunks in a FAISS vector database and save to disk.

# 3. Chunk Retrieval and Reranking\_(by testing\_phase.ipynb)

Step 1: Load FAISS vector database from disk using HuggingFace embeddings.

Step 2: Perform top-k similarity search using input query.

Step 3: Use CrossEncoder ("ms-marco-MiniLM-L-6-v2") to rerank the top documents.

Step 4: Select and return top 3 reranked chunks for further processing.

# 4. Final Querying using Mistral Model (via Ollama)(by my\_script.py)

- The top 3 reranked chunks and the query are sent to a locally running Mistral model (accessed via Ollama API).  
- The response provides relevant summarized or direct answers.  
- Prompt structure: Combined context + user query.

# 5. Modules Used

- os, glob, gc: File and memory management  
- tqdm: Progress tracking  
- PyPDF2, pdf2image: PDF processing  
- numpy: Image array manipulation for OCR  
- paddleocr: OCR engine  
- langdetect: Language detection  
- transformers (MarianMTModel, MarianTokenizer): Translation  
- langchain (RecursiveCharacterTextSplitter, HuggingFaceEmbeddings, FAISS): Text splitting, embedding, vector DB  
- sentence\_transformers (CrossEncoder): Reranking  
- requests: API communication with Ollama