System Design Q&A : Bangla-English PDF Chatbot System

Q1. What method or library did you use to extract the text, and why? Did you face any formatting challenges with the PDF content?

Answer: We used a hybrid PDF parsing method: primarily `PyPDF2` for extracting native text and `EasyOCR` for image-based OCR fallback. Each page was first attempted with PyPDF2; if the output was empty or under 20 characters, the image of the page was processed using `OpenCV` (for deskewing, binarization, and sharpening) and then passed to EasyOCR with Bangla and English language models. OCR was essential due to mixed-script PDFs or scanned documents lacking machine-readable text.

Q2. What chunking strategy did you choose (e.g. paragraph-based, sentence-based, character limit)? Why do you think it works well for semantic retrieval?

Answer: We implemented a **structure-aware hybrid chunking strategy**. The text was first split by section markers (e.g., অধ্যায়", "Chapter", "Example"). Longer segments exceeding 512 characters were further tokenized into sentences using regex-based Bangla or English splitters. Sentence groups were formed up to the 512-character threshold. This approach ensures contextually meaningful yet model-compliant input for embedding and retrieval, minimizing semantic drift across chunks.

Q3. What embedding model did you use? Why did you choose it? How does it capture the meaning of the text?

Answer: We used `paraphrase-multilingual-MiniLM-L12-v2` from Sentence Transformers. It supports both Bangla and English and provides dense, semantic embeddings ideal for multilingual tasks. This model captures paraphrastic similarity and sentence-level meaning effectively, making it suitable for both factual and abstract semantic search across mixed-language chunks.

Q4. How are you comparing the query with your stored chunks? Why did you choose this similarity method and storage setup?

Answer: We first retrieve candidates using FAISS-based `IndexHNSWFlat` on normalized embeddings (cosine similarity), allowing fast approximate nearest neighbour (ANN) search. Top results are then reranked using `CrossEncoder (ms-marco-MiniLM-L-6-v2)`, which evaluates pairwise relevance between the user query and retrieved chunks. This two-step (dense + reranker) approach balances speed and precision.

Q5. How do you ensure that the question and the document chunks are compared meaningfully? What would happen if the query is vague or missing context?

Answer: I normalize embeddings before similarity search and use reranking with a cross-encoder for accurate scoring. We also apply Maximal Marginal Relevance (MMR) to balance diversity and relevance in retrieved chunks. If the user query is vague, the semantic similarity scores will be low across the board. In such cases, we return fallback messages like "আমি জানি না।" or "Please clarify your question," minimising misleading outputs.

Q6. Do the results seem relevant? If not, what might improve them (e.g. better chunking, better embedding model, larger document)?

Answer: In most cases, the results are highly relevant due to the structure-aware chunking and reranking pipeline. Recent advancements in large language models (LLMs) have significantly enhanced retrieval quality, especially for multilingual and low-resource languages such as Bangla. These improvements step from:

- **Larger pretraining corpora:** Newer models such as LLaMA, mT5, and XGLM are trained on vastly diverse and massive multilingual datasets, enabling them to better understand underrepresented languages.
- **Improved tokenization:** Subword-based and sentencepiece tokenizers reduce fragmentation in morphologically rich languages like Bangla, allowing more accurate embeddings and alignment across languages.
- **Instruction tuning and alignment:** Fine-tuning on human preferences (e.g., with Reinforcement Learning from Human Feedback – RLHF) has made LLMs better at interpreting vague or implicit queries and returning contextually appropriate responses.
- **Multimodal and cross-lingual alignment:** Research shows that jointly trained models on multilingual data and parallel corpora (e.g., mBERT, mBART50) exhibit enhanced performance on zero-shot and code-switched tasks

Although I considered fine-tuning BNLTK and related Bangla NLP libraries, such integration was deferred due to dependency and compatibility issues. Future iterations may leverage open-source LLMs specifically aligned for Bangla-English contexts, alongside document-level metadata, section-aware indexing, and retrieval-augmented generation (RAG) to enhance relevance and factual grounding further.