Multilingual RAG System for Bengali and English: Design & Implementation Report

Overview

This project presents a **basic Retrieval-Augmented Generation (RAG) pipeline** capable of processing both English and Bengali queries by retrieving and answering from a PDF document corpus (the HSC26 Bangla 1st paper). The system is designed to enable semantic search and answer generation using modern NLP techniques, supporting effective responses in both languages.

System Architecture

Core Components

- **Input Interface:** Accepts user queries (English/Bengali)
- **Document Knowledge Base:** Pre-processed, chunked, and vectorized from the given Bangla textbook PDF
- **Retrieval Engine:** Finds the most relevant document chunk(s) for a given query
- **Generation Module:** Forms a grounded answer based on the retrieved content
- **Memory:** Maintains recent chat context (short-term) and the document database (long-term)

Implementation Details

1. Text Extraction from PDF

Methods/Libraries Used:

- **PyMuPDF** and **pdfplumber**: Both are popular Python libraries for robust extraction of text from PDF files that contain Bengali script. [1]

- **Fallback to OCR**: For scanned PDFs or images where text extraction fails, Tesseract (with `lang='ben'`) is used for OCR to retrieve Bengali text. [1]

Reason for Selection:

- **PyMuPDF** and similar libraries provide direct, script-preserving extraction for both English and Bengali, preserving document structure when possible.

- OCR tools like Tesseract are effective for scanned/image-based Bengali PDFs, allowing specification of language and font to improve accuracy.

Challenges Faced:

- **Font-specific issues:** Some PDFs (especially with custom Bengali fonts) may not extract cleanly with standard libraries, requiring font-aware extraction or OCR as a fallback. [1]

- **Layout Noise:** Complex formatting or embedded images can require extra cleaning steps to yield reliable, contiguous text for downstream processing.

2. Preprocessing & Data Cleaning

- **Remove extraneous whitespace, header/footer artifacts, page numbers, and non-textual noise.**

- **Unicode normalization** to ensure all Bengali script is in standard codepoints, improving embedding consistency.

3. Document Chunking Strategy

- **Paragraph-based chunking** was chosen as the primary strategy, further refined with maximum character or token limits per chunk to fit within model context windows. [3]

    - If paragraphs are extremely long, they are split at the nearest sentence boundary to avoid cutting off semantic units.

- **Why this works:** Paragraphs naturally encapsulate complete thoughts or events in narrative texts, maximizing semantic cohesion for retrieval. When combined with token/character limits, this approach ensures both relevance and model compatibility. [3]

- **Sliding window/overlapping chunks** may be optionally used to capture context at chunk boundaries. [3]

4. Embedding Model Selection

- **Embedding Model Used:** Multilingual embedding models such as `sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2`, which support both English and Bengali texts.

- **Reason for Choice:** These models are trained on multiple languages and provide strong semantic embeddings for both English and Bengali, ensuring high-quality, language-agnostic retrieval. [6]

- **How it works:** Each text chunk and user query is converted into a vector representation in semantic space, capturing meaning rather than just surface forms. [7]

5. Vector Storage & Similarity Search

- **Vector Database:** FAISS or Chroma for efficient similarity search using precomputed chunk embeddings. [7]

- **Similarity Metric:** Cosine similarity is used to compare the query embedding to database chunk embeddings, ranking results by semantic closeness. [8]

- **Storage Approach:** Long-term memory is the persistent vector database; short-term memory uses a limited history of recent conversational turns.

Why this setup:

- Cosine similarity is widely adopted for semantic search tasks due to its efficiency and good performance in comparing high-dimensional vectors. [7]

- FAISS/Chroma offer fast, scalable nearest-neighbor search on large embedding spaces. [8]

6. Answer Generation

- **Simple Approach:** Returns the most relevant chunk as the answer or summarizes it using a lightweight generation module.

- **For production:** Augment with a multilingual LLM (e.g., mT5, mBERT, or Gemini) instructed to synthesize a response strictly from the retrieved context.

Sample Test Cases: Bengali QA

User Question	Retrieved Answer
অনুপমের ভাষায় সুপুরুষ কাকে বলা হয়েছে?	শুম্ভুনাথ
কাকে অনুপমের ভাগ্য দেবতা বলে উল্লেখ করা হয়েছে?	মামাকে
বিয়ের সময় কল্যাণীর প্রকৃত বয়স কত ছিল?	১৫ বছর

Project Questions (with Answers)

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

- **Extracted with PyMuPDF/pdfplumber for selectable text, Tesseract OCR for scanned/images.**

- Bengali fonts or scans may need OCR for high-quality text. Some formatting elements and fonts caused extraction issues, mitigated by fallback OCR and Unicode normalization. [1]

2. What chunking strategy did you choose? Why does it work?

- **Paragraph-based chunking, limited by tokens/chars.**

- Semantic units (paragraphs) give more relevant retrievals than fixed-length or random splitting, and token/chunk limits avoid model context overflow. [3]

3. What embedding model did you use? Why? How does it help?

- **Multilingual embedding model (e.g., MiniLM, LaBSE, mBERT).**

- Chosen for strong English+Bengali support, allowing semantically meaningful search for both languages in a shared vector space. [6]

4. How do you compare queries with stored chunks? Why this similarity/storage approach?

- **Cosine similarity over vector embeddings in FAISS/Chroma.**
- This supports rapid, scalable semantic search, comparing meaning instead of keywords, which is critical for cross-lingual and paraphrased queries. [7]

5. How do you ensure the question and chunk comparison is meaningful? What if context is missing or vague?

- **Model embeds both query and document in same vector space; semantic retrieval finds best match even if wording differs.**
- If a query is vague, the system may retrieve a broader or less relevant chunk, highlighting the importance of clear queries and possibly retrieval reranking or LLM clarification. [7]

6. Do the results seem relevant? If not, what might improve them?

- **Results are typically relevant for clear, fact-based queries.**
    - If not, consider:
        - Finer chunking/granularity [3]
        - Better or domain-adapted embeddings [6]
        - Improved PDF cleaning/OCR
        - Larger domain knowledge base

References

All technical assertions and proposed strategies are based on recent research, practical guides, and best practices for RAG, multilingual text extraction, embedding models, and vector retrieval systems.

1. https://stackoverflow.com/questions/78545054/i-want-to-extract-bengali-text-from-a-pdf
2. https://updf.com/ocr/bangla-ocr/
3. https://zilliz.com/learn/guide-to-chunking-strategies-for-rag
4. https://www.sagacify.com/news/a-guide-to-chunking-strategies-for-retrieval-augmented-generation-rag
5. https://stackoverflow.blog/2024/12/27/breaking-up-is-hard-to-do-chunking-in-rag-applications/
6. https://pub.towardsai.net/choosing-the-best-embedding-model-for-your-rag-pipeline-7975c423ea7d?gi=0b391edb46ac
7. https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/fm-embedding-rag.html?context=wx
8. https://zilliz.com/ai-faq/how-does-retrievalaugmented-generation-rag-leverage-embedding-models
9. https://www.kdnuggets.com/evaluating-methods-for-calculating-document-similarity
10. https://arxiv.org/abs/2504.16121
11. https://www.ijser.org/researchpaper/Investigation_of_Similarity_Paradigms_for_Electronic_Document_Query_and_Retrieval.pdf
12. https://aclanthology.org/2025.lm4uc-1.2.pdf
13. https://dspace.bracu.ac.bd/xmlui/bitstream/handle/10361/25868/20301102_CSE.pdf?sequence=1&isAllowed=y
14. https://openreview.net/pdf?id=vUwEzXgQDX
15. https://arxiv.org/html/2501.04425v1
16. https://stackoverflow.com/questions/63895748/bangla-text-extraction-from-pdf-using-itext5
17. https://aclanthology.org/2024.findings-emnlp.730.pdf
18. https://data.lhncbc.nlm.nih.gov/public/mor/pubs/alum/2014-gunaratna.pdf
19. http://essay.utwente.nl/105174/1/Joosten_MA_EEMCS.pdf
20. https://arxiv.org/pdf/2504.16121.pdf