Multilingual RAG System - Technical Assessment Answers

AI Engineer (Level-1) Assessment

July 26, 2025

1 System Overview

This document answers the technical questions for the multilingual RAG system that processes PDF documents and responds to queries in both English and Bengali. The system is deployed at `https://huggingface.co/spaces/sairika/multilingual-rag-system` and uses Together AI's Qwen model for generating contextually relevant answers.

2 Technical Questions & Answers

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

I used PyPDF2 for text extraction because it's reliable, lightweight, and handles Unicode characters well (essential for Bengali text). The library provides good error handling and works consistently across different PDF formats. The main formatting challenges were inconsistent spacing between words, Bengali Unicode rendering issues, and text fragmentation across page breaks. Additionally, scanned or image-based PDFs couldn't be processed since PyPDF2 only extracts text layers. I handled these issues with comprehensive error checking and user feedback when extraction fails.

Q2: What chunking strategy did you choose? Why do you think it works well for semantic retrieval?

I implemented character-limit based chunking with a maximum of 10,000 characters. When documents exceed this limit, the system takes the first 5,000 and last 5,000 characters with a clear truncation indicator. This strategy works well for semantic retrieval because document beginnings typically contain key concepts and introductions, while endings have conclusions and summaries. It preserves the most semantically important parts while staying within the model's context window limitations. Although sentence-based chunking would be more sophisticated, this approach balances simplicity with effectiveness for the assessment scope.

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

I used Qwen/Qwen3-Coder-480B-A35B-Instruct-FP8 via Together AI. This model was chosen for its excellent multilingual capabilities (particularly English and Bengali), large 480 billion parameter architecture, and strong instruction-following abilities. The model captures text meaning through transformer self-attention mechanisms that understand relationships between all words in context. Instead of generating separate embeddings, the model processes the entire PDF content within the system prompt, enabling holistic semantic understanding during response generation. This approach leverages the model's contextual embeddings that adapt based on surrounding text rather than using static word representations.

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

I use system prompt-based contextual processing rather than traditional vector similarity search. The PDF content is embedded directly in the system prompt, and the model performs implicit similarity comparison through its attention mechanisms. Storage is handled in-memory using Gradio session state without external vector databases. I chose this approach for its simplicity (no infrastructure setup required), natural multilingual understanding, and ability to leverage the full 480B parameter model for semantic matching. While less sophisticated than vector databases, this method is appropriate for the assessment scope and enables rapid development with good performance.

Q5: How do you ensure meaningful query-document comparison? What would happen if the query is vague or missing context?

Meaningful comparison is ensured through explicit system prompt instructions that constrain the model to answer only based on PDF content. The model is instructed to state clearly when information isn't available in the document. For vague queries like "What is this about?", the model provides comprehensive summaries using its attention across the entire document. For out-of-scope questions unrelated to the PDF, the system responds that the information isn't available in the document. Missing context is handled by leveraging conversation history and the model's ability to maintain context throughout the session, while partial relevance results in clear statements about what information is or isn't present.

Q6: Do the results seem relevant? If not, what might improve them?

The results are highly relevant based on testing with the live demo. The system accurately answers complex Bengali literary questions, handles cross-language queries effectively, and maintains proper document grounding. Current strengths include high factual accuracy, excellent multilingual understanding, and appropriate handling of out-of-scope queries. However, several improvements could enhance performance: implementing sentence-boundary aware chunking instead of character-based truncation, integrating vector databases like Pinecone for true semantic search, using specialized multilingual embedding models, adding support for multiple documents, and implementing OCR for scanned PDFs. The most impactful improvements would be vector database integration for scalability and better chunking strategies for improved context preservation.

3 Conclusion

The implemented system successfully meets the assessment requirements by providing functional multilingual capabilities, accurate document-grounded responses, and an intuitive user interface. While built with a simplified architecture focused on rapid development, the system demonstrates practical AI engineering skills and provides a solid foundation for future enhancements. The live deployment at `https://huggingface.co/spaces/sairika/multilingual-rag-system` proves the system's real-world functionality.