Mandatory Explanations for RAG QA System

This document provides the mandatory technical explanations required for the Retrieval-Augmented Generation (RAG) Question Answering system. It focuses on design decisions, observed limitations, and evaluation metrics tracked during development.

---

1. Why a Specific Chunk Size Was Chosen

In this project, the document chunking configuration was: - **Chunk Size:** 500 characters - **Chunk Overlap:** 100 characters

Reasoning Behind This Choice

The chunk size of **500 characters** was selected as a balance between **context richness** and **retrieval accuracy**:

- **Too small chunks** (e.g., 100–200 characters) often break meaningful context. Important explanations or definitions get split across chunks, which reduces answer quality.
- **Too large chunks** (e.g., 1000+ characters) dilute semantic focus. Embeddings become less precise, which negatively affects similarity search results.

A chunk size of 500 characters ensures that: - Each chunk contains a complete logical idea (paragraph-level context) - Embeddings remain semantically focused - Retrieved context fits well within LLM prompt limits

The **100-character overlap** helps preserve continuity between chunks, especially for concepts that span across paragraph boundaries.

---

2. Observed Retrieval Failure Case

Failure Scenario

A retrieval failure was observed when a **user question was vague or overly generic**, such as:

> "Explain the concept in detail"

What Went Wrong

- The query lacked specific keywords present in the documents
- The retriever (FAISS similarity search) returned chunks that were **semantically similar but contextually irrelevant**
- As a result, the LLM generated a **generic answer** instead of a document-grounded response

Example

If the document discussed multiple topics (e.g., cloud computing, virtualization, and security), a vague query caused the retriever to pull unrelated sections.

Mitigation Strategies (Future Work)

- Improve query reformulation using LLM-based query rewriting
- Enforce more structured user questions
- Use hybrid retrieval (keyword + vector search)

---

3. Metric Tracked During the Project

Tracked Metric: Response Latency

Latency was tracked to evaluate system performance and user experience.

Why Latency Matters

- RAG systems involve multiple steps: retrieval + LLM generation
- High latency negatively impacts usability, especially in interactive applications

Observations

- Average response latency ranged between **2–4 seconds**
- Most latency was introduced during:
- Vector similarity search
- LLM API response time

How It Was Monitored

- API response time was logged at the FastAPI endpoint level
- Manual testing using multiple question queries

Future Improvements

- Cache frequently asked questions
- Use faster embedding models
- Optimize FAISS index configuration

Conclusion

These design choices and evaluations helped ensure that the RAG system provides: - Accurate and context-aware answers - Reasonable performance - Clear understanding of system limitations

This document demonstrates thoughtful architectural decisions and real-world evaluation of the RAG-based Question Answering system.