RAG Pipeline Development and Evaluation Assignment

Assignment Overview

Objective: Design, implement, and evaluate a complete RAG (Retrieval-Augmented Generation) pipeline for processing PDF documents with comprehensive evaluation at each stage.

Background

You are tasked with building a production-ready RAG system that can process PDF documents, store them efficiently in a vector database, and provide accurate responses to user queries. The system should be robust, scalable, and well-evaluated.

Part 1: Document Chunking Strategy

Task 1.1: Chunking Implementation

Implement a comprehensive PDF chunking system with the following requirements:

1. **Multiple Chunking Strategies:**
    a. Fixed-size chunking with overlap
    b. Semantic chunking (sentence/paragraph boundaries)
    c. Hierarchical chunking (sections, subsections)
    d. Custom chunking based on document structure
2. **Parameter Selection and Justification:**
    a. Document your choice for chunk size (e.g., 512, 1024, 2048 tokens)
    b. Justify overlap percentage (e.g., 20%, 50%)
    c. Explain your approach for handling tables, images, and special formatting
    d. Describe metadata preservation strategy

Deliverables:

- Working code with multiple chunking strategies
- Detailed justification document (500-750 words) explaining parameter choices
- Comparative analysis of different chunking approaches

- Quality metrics for each chunking strategy

Part 2: Vector Database Implementation

Task 2.1: Vector Store Setup

Implement vector storage with the following specifications:

1. **Embedding Model Selection:**
    a. Choose and justify your embedding model (e.g., sentence-transformers, OpenAI, Cohere)
    b. Compare at least 2 different embedding models
    c. Document embedding dimensions and computational requirements
2. **Vector Database Implementation:**
    a. Choose vector DB (FAISS, Pinecone, Weaviate, Qdrant, etc.)
    b. Implement efficient indexing strategy
    c. Include metadata filtering capabilities
    d. Implement batch processing for large documents

Deliverables:

- Complete vector store implementation
- Embedding model comparison report
- Performance benchmarks (indexing speed, search latency)

Part 3: Retrieval Evaluation

Task 3.1: Retrieval Metrics Implementation

Develop comprehensive retrieval evaluation:

1. **Evaluation Metrics:**
    a. Precision@K, Recall@K, F1@K
    b. Mean Reciprocal Rank (MRR)
    c. Normalized Discounted Cumulative Gain (NDCG)
    d. Hit Rate
    e. Custom relevance scoring
2. **Test Dataset Creation:**
    a. Create ground truth query-document pairs
    b. Include easy, medium, and hard queries

c. Test edge cases and failure modes

Deliverables:

- Test dataset with ground truth annotations

Part 4: Re-ranking Implementation

Task 4.1: Advanced Re-ranking

Implement and evaluate re-ranking strategies:

1. **Re-ranking Methods:**
    a. Cross-encoder re-ranking
    b. Query-document relevance scoring
    c. Diversity-based re-ranking

Deliverables:

- Working re-ranking implementation
- Before/after comparison analysis
- Performance impact assessment (latency vs. quality trade-off)

Part 5: LLM Integration and Evaluation

Task 5.1: End-to-End Pipeline

Complete the RAG pipeline with LLM integration:

1. **LLM Integration:**
    a. Implement context preparation and prompt engineering
    b. Handle context length limitations
    c. Include citation and source attribution

Deliverables:

- Complete RAG pipeline implementation
- Response quality evaluation framework
- Hallucination detection and analysis

- End-to-end performance metrics