Multi-Modal RAG QA System

Technical Report

By Ritvik, Date:11th January 2026

Repository: https://github.com/retvq/multi-modal-rag

Abstract

This report presents the design and implementation of a production-grade Multi-Modal Retrieval-Augmented Generation (RAG) system capable of processing and answering questions from documents containing text, tables, and figures. The system achieves a 90% benchmark pass rate using Gemini 2.0 Flash for generation, hybrid retrieval with cross-encoder reranking, and modality-aware chunking. Key innovations include real-time evaluation benchmarking, session-based document uploads, and graceful LLM fallback mechanisms.

System Architecture

The system is organized into three primary pipelines: Ingestion, Retrieval, and Generation. Each pipeline is modular and independently testable.

Ingestion Pipeline

The ingestion pipeline extracts multi-modal content from PDF documents:

- **Text Extraction:** PyMuPDF extracts raw text with page boundaries and section headers
- **Table Extraction:** pdfplumber identifies and parses tabular structures into structured markdown
- **Figure Extraction:** Images are extracted with associated captions and alt-text
- **Chunking:** Content is segmented into 512-token chunks with 50-token overlap
- **Embedding:** SentenceTransformers (`all-MiniLM-L6-v2`) generates 384-dimensional embeddings

Retrieval Pipeline

The retrieval pipeline implements a three-stage approach for optimal accuracy:

1. **Vector Search:** ChromaDB performs approximate nearest neighbor search
2. **Keyword Search:** BM25-based keyword matching captures lexical overlaps
3. **Hybrid Fusion:** Reciprocal Rank Fusion (RRF) combines both result sets
4. **Reranking:** Cross-encoder reranker refines the final ranking

Generation Pipeline

The generation pipeline produces grounded answers with citations:

- **LLM Mode:** Gemini 2.0 Flash generates natural language responses with inline citations
- **Rule-Based Mode:** Deterministic extraction for table lookups and numeric queries
- **Fallback:** Automatic degradation to rule-based when LLM is unavailable

Design Choices

Decision	Choice	Rationale
Embedding Model	all-MiniLM-L6-v2	Fast, lightweight (80MB), strong semantic performance
Vector Database	ChromaDB	Zero-config, persistent storage, metadata filtering
LLM	Gemini 2.0 Flash	Free tier available, fast inference, JSON mode
Retrieval	Hybrid + Reranking	Combines lexical precision with semantic recall
Chunking	512 tokens, 50 overlap	Balances context length with retrieval granularity
Fallback	Rule-based generator	Graceful degradation when LLM unavailable

Table 1: Key design decisions with rationale

Key Innovations

1. **Modality-Aware Retrieval:** Chunks are tagged with TEXT/TABLE/FIGURE modalities, enabling intent-based filtering during retrieval
2. **Session-Based Uploads:** Users can upload custom PDFs that are processed into isolated session-specific vector stores
3. **Real-Time Evaluation:** Built-in benchmark suite in the UI allows live performance testing across modalities

Benchmark Results

The evaluation suite consists of 10 queries spanning four modality categories: text, table, figure, and mixed (cross-modal). Results were collected using the integrated evaluation harness.

Overall Performance

Metric	LLM Mode	Rule-Based Mode
Pass Rate	90% (9/10)	20% (2/10)
Average Latency	6,724 ms	150 ms
Keyword Match Score	67.2%	18.5%

Table 2: Overall benchmark performance comparison

Per-Modality Breakdown (LLM Mode)

Key Observations

1. **LLM is Critical for Accuracy:** The 90% vs 20% pass rate differential demonstrates that LLM-based generation is essential for comprehensive answer quality.

Modality	Pass Rate	Avg Latency	Keyword Score
Text	100% (3/3)	3,947 ms	68.3%
Table	100% (3/3)	11,219 ms	72.2%
Figure	50% (1/2)	6,057 ms	37.5%
Mixed	100% (2/2)	4,812 ms	87.5%

Table 3: Performance breakdown by modality

2. **Tables Achieve Perfect Accuracy:** Structured extraction combined with LLM formatting yields 100% pass rate on table queries, with the highest keyword match scores.
3. **Figures Require Vision Capabilities:** The 50% pass rate on figure queries highlights the limitation of text-only LLMs for chart interpretation. Future work should integrate Gemini Vision.
4. **Hybrid Retrieval Improves Recall:** Combining vector similarity with keyword matching successfully captures domain-specific terminology (e.g., "fiscal balance", "hydrocarbon sector").
5. **Latency-Accuracy Trade-off:** LLM mode is approximately 40× slower than rule-based but provides significantly superior answer quality.

Future Improvements

- **Vision Integration:** Incorporate Gemini 1.5 Pro Vision for actual chart and figure interpretation
- **Streaming Responses:** Implement token streaming to reduce perceived latency
- **Query Caching:** Cache frequent queries to minimize API calls and costs
- **Fine-tuned Reranker:** Train a domain-specific cross-encoder on financial documents
- **Multi-document Support:** Enable cross-document retrieval and comparison

Conclusion

This Multi-Modal RAG system demonstrates production-ready capabilities for document question-answering across text, tables, and figures. The modular architecture enables component-level testing and easy upgrades. With a 90% benchmark pass rate and comprehensive evaluation tooling, the system provides a solid foundation for enterprise document intelligence applications.

Repository: `https://github.com/retvq/multi-modal-rag`