

## Retrieval-Augmented Generation (RAG)

100% COMPLETE

**Sample Interaction Flow: RAG System for IFC Annual Report 2024**

**Sample 1** +

User Input 1 (Text-based, potentially hitting Phase 1-2 capabilities): "What is IFC's mission and how many member countries does it have?"

System Response 1: "IFC's mission is to end extreme poverty and boost shared prosperity on a livable planet. IFC is owned by 186 member countries."  
[Retrieved Context Snippets]

  

**Sample 2** +

User Input 2 (Table-based query, potentially hitting Phase 3 capabilities): "What was the Net Income for FY24 and FY23?"

System Response 2: "For the fiscal year ended June 30, 2024 (FY24), IFC's Net Income was \$1,485 million. For the fiscal year ended June 30, 2023 (FY23), the Net Income was \$672 million."  
[Retrieved Context Snippets]

  

**Sample 3** +

User Input 3 (Image/Chart-based query, potentially hitting Phase 4 capabilities): "Show me the trend of IFC's Net Income from FY22 to FY24."

System Response 3: "Based on Figure 1 (Income Measures), IFC's Net Income showed an upward trend from FY22 to FY24. In FY22, there was a Net Loss of \$464 million. In FY23, there was a Net Income of \$672 million."  
[Retrieved Context Snippets, image description or plot]

  

**Sample 4** +

User Input 4 (Complex query, potentially requiring multi-hop or advanced table reasoning - Phase 5): "What was the percentage change in 'Total Long-Term Finance Own Account Commitments' from FY23 to FY24, and which region had the highest disbursed investment portfolio in FY24?"

System Response 4 (Illustrative, assuming multi-hop/synthesis): "Let's break this down:

- Percentage change in 'Total Long-Term Finance Own Account Commitments' (FY23 to FY24):
  - In FY24, Total Long-Term Finance Own Account Commitments were \$16,677 million.
  - In FY23, they were \$21,458 million.
  - This represents a decrease of approximately 22.27%.
- Region with the highest disbursed investment portfolio in FY24:
  - The Asia and Pacific region had the highest disbursed investment portfolio in FY24 with \$18.9 billion.

  

**Sample 5** +

User Input 5 (Semantic Cache hit example - Phase 5): (Assuming a similar query to User Input 2 was asked recently) "Tell me again, what was the Net Income for FY24?"

System Response 5 (from Semantic Cache): "For the fiscal year ended June 30, 2024 (FY24), IFC's Net Income was \$1,485 million. (Retrieved from cache.)"

### Technical Requirements

- LLM: Gemini 2.0 Flash (or newer):
  - Please use streaming, function calling and json/structured output during implementation where appropriate.
- SDK: Google GenAI SDK. (with Vertex AI Auth through GCP project - **NO API Keys, please!**)
  - [Gemini access instructions](#)
- UI: Streamlit or Gradio.
- DB: FAISS, Qdrant.
- PDF Processing: Docing/PyMuPDF/ Gemini Multimodal capabilities.
- RAG Frameworks: LangChain.
- Docker.
- Langfuse for observability, RAGAS for evaluation.

### Functional Requirements

#### Data parsing requirements:

Objective: Learn how to work with data extraction and formatting for multimodal documents.

##### Tasks:

- Extract all textual data from PDF using PyPDF2 / pdfminer / docing:
  - Collect all possible metadata
  - Optional - Consider document structure
- Extract all image data from PDF:
  - Ask Gemini to generate descriptive image captions.
  - Collect all possible metadata.
  - Optional - Figure out if image is related to any text above / below.
- Extract all table data from PDF:
  - Extract table data using LLM or Camelot / Tabula?
  - Save table data in a representative structured format (.md, .html, .json, .csv) or summarise.
  - Collect all possible metadata.
  - Optional - Figure out if table is related to any text or image above / below.

### Phase 1: Text-Based RAG System (Naive RAG)

Objective: Develop a baseline RAG system that processes and queries only the textual content from the IFC Annual Report.

##### Tasks:

- Data Ingestion & Preprocessing:**
  - Extract all textual content from the PDF.
  - Implement text cleaning and chunking strategies. (Optional: work on structure detection for more advanced chunking)
- Embedding & Indexing:**
  - Generate dense embeddings for text chunks.
  - Set up a vector search engine (e.g., FAISS) and index the embeddings.
  - Setup Qdrant locally, populate same embeddings to this vector storage too.
  - Compare the performance of FAISS vs Qdrant. Analyse usecases.
- Retrieval Mechanism:**

Implement a retriever that finds the most relevant text chunks for a given query using vector similarity. (Either with any of the frameworks or custom).
- Generation & UI:**

Integrate retrieved chunks with a Gemini model to generate answers.

Build a simple Streamlit/Gradio interface for querying.
- Observability:** Integrate Langfuse for basic tracing of queries and responses.

Next: Final Assessment ▶

In Progress

Retrieval-Augmented Generation

✓ Retrieval-Augmented Generation (RAG)

○ Final Assessment

Course Tasks 0/1

Course Evaluation