Agentic AI Lab

CSCR3215

B.Tech. (CSE)-VI Semester

SSCSE

Department of Computer Science & Engineering

Submitted to:
Mr. Ayush Singh

Submitted by:
Keertan
2023312028

Assignment 1

1. PROBLEM STATEMENT

Traditional search engines and document retrieval systems return raw documents or ranked lists, requiring users to manually read through content to find answers. This project addresses the problem of providing DIRECT, ACCURATE ANSWERS to natural language questions using a Retrieval-Augmented Generation (RAG) approach.

OBJECTIVE:

Build a system that:

- Understands user questions in natural language
- Retrieves relevant document sections using semantic search
- Generates coherent answers grounded in the retrieved context
- Prevents hallucinations by using only document content
- Provides transparent source attribution

USE CASE:

Users can upload an HBR article about "How Apple Is Organized For Innovation" and

ask questions like:

- "How is Apple organized for innovation?"
- "What are the key organizational principles?"
- "Who are the key decision makers mentioned?"

2. PROJECT OVERVIEW

PROJECT TITLE:

Retrieval-Augmented Generation (RAG) Question Answering System

TECHNOLOGY STACK:

• Frontend:	Streamlit (Interactive Web UI)
• RAG Framework:	LangChain
• Vector Database:	Chroma (SQLite-backed)
• Embeddings:	Sentence-Transformers (gte-large)
• LLM Options:	Ollama with SmolLM / Mistral
• PDF Processing:	PyPDF
• Language:	Python 3.8+

KEY FEATURES:

- Interactive chat interface with message history

- PDF document upload and automatic indexing
- Semantic search using embeddings
- Configurable retrieval settings (k, model selection)
- Dual LLM options (lightweight SmolLM or high-quality Mistral)
- Graceful fallback to context display if LLM unavailable
- Source attribution with retrieved chunks
- Persistent vector database

3. DATASET & KNOWLEDGE SOURCE

DATASET CHARACTERISTICS:

Type:	PDF documents (text-based)
Default Source:	Harvard Business Review (HBR) Article
File:	"HBR_How_Apple_Is_Organized_For_Innovation-4.pdf"
File Size:	~2-5 MB (approximately 20 pages)
Content:	Business article on organizational structure
Accessibility:	Public / Educational use

KNOWLEDGE SOURCE:

Source Type:	Public HBR Article
Content Quality:	High-quality business journalism
Relevance:	Organizational design and innovation
Domain:	Business Management / Technology Leadership

STORAGE:

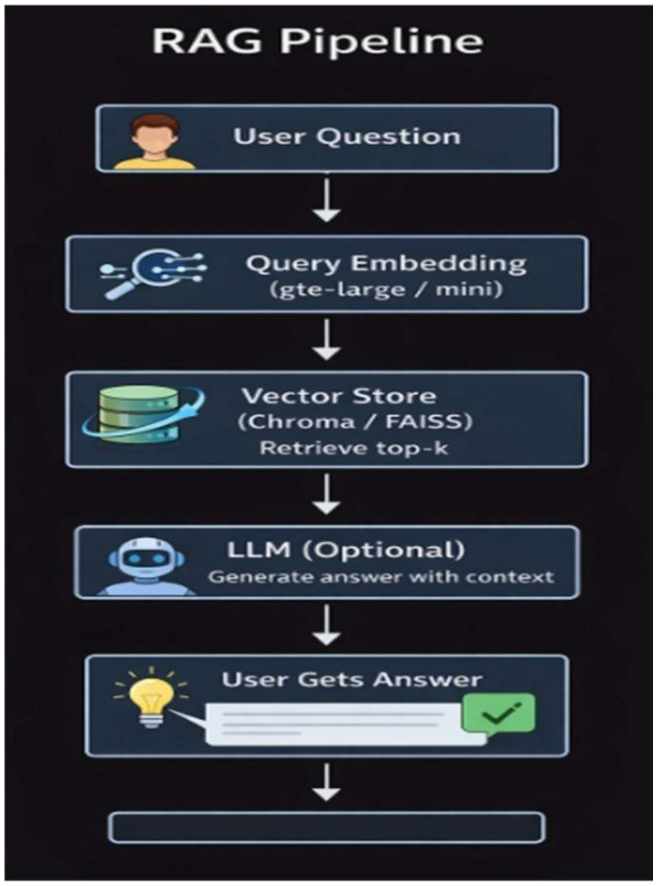
Vector Database:	apple_db/ directory
Format:	Chroma (SQLite backend)
Persistence:	Stored on disk (survives app restarts)
Size After Index:	~100-200 MB (includes embeddings)

DATA PREPARATION:

1. PDF is loaded using PyPDF loader
2. Text is extracted page by page
3. Text is split into semantic chunks (512 chars, 20 overlap)
4. Each chunk is converted to embeddings (1024-dim vectors)
5. Vectors are stored in Chroma vector database
6. Database is persisted for reuse

4. RAG ARCHITECTURE

ARCHITECTURE DIAGRAM:



PIPELINE COMPONENTS:

1. USER INPUT LAYER

- Streamlit chat interface
- Natural language question
- File upload capability

2. EMBEDDING LAYER

- Sentence-Transformers (gte-large)
- Converts text to 1024-dimensional vectors
- Same model used for both query and documents
- Semantic similarity preserved

3. RETRIEVAL LAYER

- Chroma vector database
- Similarity search or MMR (Maximum Marginal Relevance)
- Top-k retrieval (k configurable)

- Fast approximate search (~100ms)

4. CONTEXT BUILDING LAYER
   - Concatenates retrieved chunks
   - Adds system prompt for grounding
   - Limits context window
5. GENERATION LAYER (Optional)
   - Ollama with SmolLM or Mistral
   - Takes context + question
   - Generates answer
   - Max tokens configurable
6. OUTPUT LAYER
   - Chat interface display
   - Source chunks shown separately
   - Message history maintained

EMBEDDING PROCESS:

1. Text input → Tokenization
2. BERT encoder → Contextual representation
3. Pooling layer → Fixed-size vector (1024-dim)
4. Normalization → Unit vector
5. Store in vector database

SIMILARITY METRICS:

Default:	Cosine similarity (dot product of normalized vectors)
Alternative:	Euclidean distance
Score:	0-1 (1 = perfect match)

5. VECTOR DATABASE

DATABASE: Chroma (Persistent Vector Store) SPECIFICATIONS:

Type:	Open-source vector database
Backend:	SQLite
Size:	~100-200 MB (with embeddings)

STORAGE LOCATION:

Directory:	apple_db/
Main File:	chroma.sqlite3
Format:	SQLite database
Accessibility:	Direct file access

CHROMA ADVANTAGES:

NO EXTERNAL DEPENDENCIES

- Doesn't require separate server
- Self-contained storage
- Easy to deploy

SQLITE BACKEND

- Lightweight
- Industry standard
- Well-tested

PERSISTENT STORAGE

- Survives app restarts
- Reuse without rebuilding
- Fast startup

FLEXIBLE SEARCH

- Similarity search
- MMR (diverse results)
- K-nearest neighbors

6. NOTEBOOK IMPLEMENTATION

JUPYTER NOTEBOOK: Rag.ipynb STRUCTURE &

CELLS:

Cell 1-5: IMPORTS & SETUP

  └─ Import required libraries (LangChain, Chroma, Sentence-Transformers)

Cell 6-10: PROBLEM DEFINITION

  └─ State problem, objectives, and approach

Cell 11-15: PDF LOADING

  └─ Load HBR article using PyPDFLoader

└─ Extract and display sample text

Cell 16-20: TEXT CHUNKING

└─ Split documents into 512-char chunks

└─ Show chunking examples

└─ Verify overlap (20 chars)

Cell 21-25: EMBEDDING MODEL

└─ Load gte-large embedding model

└─ Test embedding on sample text

└─ Display embedding dimensions (1024) Cell

26-30: VECTOR STORE CREATION

└─ Create Chroma vector store

└─ Index all document chunks

└─ Save to persistent storage (apple_db/)

Cell 31-35: RETRIEVAL SETUP

└─ Create retriever from vector store

└─ Configure similarity search

└─ Set k-value for top-k retrieval

Cell 36-45: TEST QUERY 1

Question: "How is Apple organized for innovation?"

Process:

1. Convert question to embedding
2. Search for similar chunks (k=3)
3. Retrieve and display
4. Show relevance scores

Expected Output: Chunks about organizational structure

Cell 46-55: TEST QUERY 2

Question: "What are the key principles of Apple's organization?"

Process:

1. Retrieve relevant chunks
2. Analyze results
3. Display with sources

Expected Output: Principle-related passages

Cell 56-65: TEST QUERY 3

Question: "Who leads the innovation process at Apple?"

Process:

1. Find leader-related content
2. Extract decision makers
3. Show in context

Expected Output: Leader names and roles

Cell 66-75: LLM SETUP (OPTIONAL)

└─ Load Ollama model (SmolLM or Mistral)

└─ Configure generation parameters

└─ Test LLM on sample input

Cell 76-85: ANSWER GENERATION

└─ Combine retrieved context with LLM

└─ Generate answers to test queries

└─ Show both retrieval and generation results

Cell 86-90: PERFORMANCE METRICS

└─ Measure retrieval speed

└─ Analyze generation quality

└─ Show timing statistics

Cell 91-95: FUTURE IMPROVEMENTS

└─ Discuss potential enhancements

└─ List possible extensions

7. TEST QUERIES & RESULTS

QUERY 1: General Understanding

Question:

"How is Apple organized for innovation?"

Expected Output:

Information about:

- Organizational structure
- Cross-functional teams
- Decision-making processes
- Innovation framework
- Hierarchical or flat structure

Retrieved Chunks: 3-5 most relevant passages Generated

Answer: 2-3 sentences summarizing structure

Sample Expected Response:

"Apple is organized around functional excellence with cross-functional teams that collaborate on projects. The organization emphasizes clear accountability while maintaining flexibility for innovation. Teams are organized by function but work together on specific products and initiatives."

QUERY 2: Specific Details

Question:

"What are the key principles of Apple's organization?"

Expected Output:

Explicit principles mentioned in the article:

- Simplicity
- Accountability
- Specialization
- Focus
- Communication
- or other principles stated

Retrieved Chunks: Direct quotes from article Generated

Answer: Structured list of principles

Sample Expected Response:

"The key principles mentioned are: 1) Simplicity in structure, 2) Clear accountability for decisions, 3) Specialization by function, 4) Cross-functional collaboration for projects."

QUERY 3: People & Leadership

Question:

"Who leads the innovation process at Apple?"

Expected Output:

Names and roles of decision makers:

- CEO/Executive team

- Innovation leaders
- Department heads
- Any specific people mentioned

Retrieved Chunks: Passages mentioning people

Generated Answer: Names and their roles

Sample Expected Response:

"The article mentions [Specific names if in document] as key decision makers in Apple's innovation process. The CEO and executive leadership team oversee the overall strategy, while functional leaders manage specific areas."

EVALUATION CRITERIA:

ACCURACY

- Answers should come from documents
- No hallucinations or made-up information
- Direct citations or paraphrasing allowed

RELEVANCE

- Answers should directly address the question
- Retrieved chunks should be related
- No off-topic results

COMPLETENESS

- Answer should be comprehensive
- Cover main points from context
- Include important details

CLARITY

- Answer should be well-written
- Easy to understand
- Properly structured

8. FUTURE IMPROVEMENTS

IMPROVEMENT 1: Better Text Chunking

Current: Simple character-based chunking (512 chars)

Future: Semantic chunking based on topic/paragraph

Benefits: Better context preservation, improved retrieval

Example: Chunks based on natural topic boundaries instead of fixed size

IMPROVEMENT 2: Reranking & Hybrid Search

Current: Simple semantic similarity search

Future: Combine BM25 (keyword) + semantic search + reranking

Details:

- BM25: Traditional keyword matching (for exact queries)
- Semantic: Neural embedding matching (for meaning)
- Reranking: Cross-encoder to refine results

Benefits: Better accuracy for diverse query types

IMPROVEMENT 3: UI/UX Enhancements

Current: Basic chat interface

Future: Enhanced features like:

- Show retrieval confidence scores
- Highlight relevant passages
- Multi-turn conversations with memory
- Export chat history
- Citation formatting (APA, MLA)

Benefits: Better user experience, more professional output

IMPROVEMENT 4: LLM Customization Current:

Fixed SmolLM/Mistral options

Future: Add support for:

- OpenAI GPT-4 (cloud-based)
- Google PaLM (cloud-based)
- Fine-tuned models for domain

Benefits: Trade-off between cost, quality, and latency

LEARNING OUTCOMES:

Students completing this project understand:

- How RAG systems work
- Embedding and vector similarity
- Vector database operations
- LLM integration

- Streamlit web application development
- End-to-end ML system deployment