Retrieval-Augmented Generation (RAG) Pipeline

Why RAG? The Real Problem

Problem: LLMs often lack domain-specific or latest knowledge.

- LLM answer: "I don't know your internal company policies"
- LLM hallucination: invents answers if uncertain
- Expensive fine-tuning is not always possible

RAG Fixes This: Injects external knowledge at runtime.

RAG: Core Idea

RAG enhances an LLM by enriching its prompt with external data.

- Uses private or latest documents
- Reduces hallucinations
- Avoids fine-tuning cost

Example: User asks: "What does Section 4 of our HR policy say?" Model retrieves your HR PDF and answers correctly.

RAG Pipeline Overview

A RAG system has 3 components, each solving a different problem:

1. **Ingestion:** How do we convert raw data into searchable formats?
2. **Retrieval:** How do we find relevant information?
3. **Generation:** How do we create answers grounded in the data?

Ingestion Pipeline

Goal: Convert raw documents into structured, searchable vector format.
Problem Solved: Raw PDFs, text, websites are unstructured and hard to search.

- Collect data (PDFs, websites, docs)
- Clean (remove formatting issues)
- Chunk text (so search is efficient)
- Embed chunks (turn into vectors)
- Store in Vector DB

Example: A 200-page research PDF is chunked into 300 vector-searchable pieces.

Retrieval Pipeline

Goal: Fetch relevant context for a user query.
Problem Solved: LLMs don't know which document is relevant.

- Convert query into embedding
- Search Vector DB
- Retrieve most similar chunks

Example: Query: "Show YOLOv5 results for drone action detection." Retrieved chunks: model performance table from your documents.

Generation Pipeline

Goal: Construct the final answer using retrieved context.
Problem Solved: Without retrieved context, LLM hallucinates.

- Add chunks to prompt
- Pass augmented prompt to LLM
- LLM produces grounded answer

Example: Instead of: "YOLOv5 has poor drone accuracy." RAG answer: Cites real accuracy from retrieved document.

Vanilla RAG Flow (Step-by-Step)

1. Load data into Vector DB
2. User asks question
3. Retrieve relevant chunks
4. Add chunks to prompt
5. Send to LLM
6. Generate answer

Outcome: LLM answers based on your data, not guesses.

Before vs After RAG

Before: LLM: "I think YOLOv5 accuracy is around 60%." (Wrong)
After RAG: LLM (after retrieval): "YOLOv5 achieved 84.2% drone action detection accuracy according to the internal dataset." (Correct)

RAG Principle Summary

Retrieval

Find relevant data chunks from external sources.

Augmentation

Insert retrieved chunks into the model prompt.

Generation

LLM uses the inserted context to produce grounded answers.

Bottom line: RAG = Less hallucination + More trust + Real data.