Applied Generative AI : LLM Application Development

Bhattaraprot Bhabhatsatam, Ph.D.

https://bhattaraprot.com

Dates: 12 & 19 September 2025

Retrieval-Augmented Generation

What is RAG?

Definition:

Retrieval-Augmented Generation (RAG) is a technique that combines:

- **Information Retrieval** - Finding relevant documents/data
- **Text Generation** - Creating responses using language models

Key Concept:

- RAG enhances language models by providing them with <u>**external information**</u> to generate more accurate and contextual responses.
- Up-to-date ?

LLM Enhancement

Method	Type	Cost	Complexity	Model Impact
Prompt Engineering *Prompt Engineering*	External	Low	Low	Better input formatting to optimize responses
RAG *Retrieval-Augmented Generation*	External	Medium	Medium	Provides context via retrieval from external knowledge
Tool Calling *Tool Calling*	External	Low	Medium	Extends capabilities via external APIs
Multi-Agent *Multi-Agent Systems*	External	Medium	High	Orchestrates multiple models working together
Memory Systems *Memory Systems*	External	Medium	Medium	Manages context externally for long-term interactions
Ensembles *Model Ensembles*	External	High	Medium	Combines multiple model outputs for better performance
MCP *Model Context Protocol*	External	Low	Medium	Standardized tool and data integration
Fine-Tuning *Fine-Tuning*	Internal	High	High	Modifies model weights for specific tasks
PEFT (LoRA) *Parameter-Efficient Fine-Tuning (Low-Rank Adaptation)*	Internal	Medium	Medium	Adds trainable parameters efficiently
RLHF *Reinforcement Learning from Human Feedback*	Internal	High	High	Retrains model with human feedback for alignment
Distillation *Knowledge Distillation*	Internal	Medium	Medium	Creates new optimized model from teacher model

RAG Advantages

Knowledge Cutoff Problem

- Language models are trained on data up to a specific date
- Cannot access real-time or recent information
- **Example**: ChatGPT trained in 2023 won't know 2024 events

Hallucination Issues

- Models sometimes generate plausible but incorrect information
- RAG provides factual grounding from reliable sources

RAG Advantages

Domain-Specific Knowledge

- Generic models lack specialized knowledge for specific industries
- RAG allows access to company documents, technical manuals, etc.

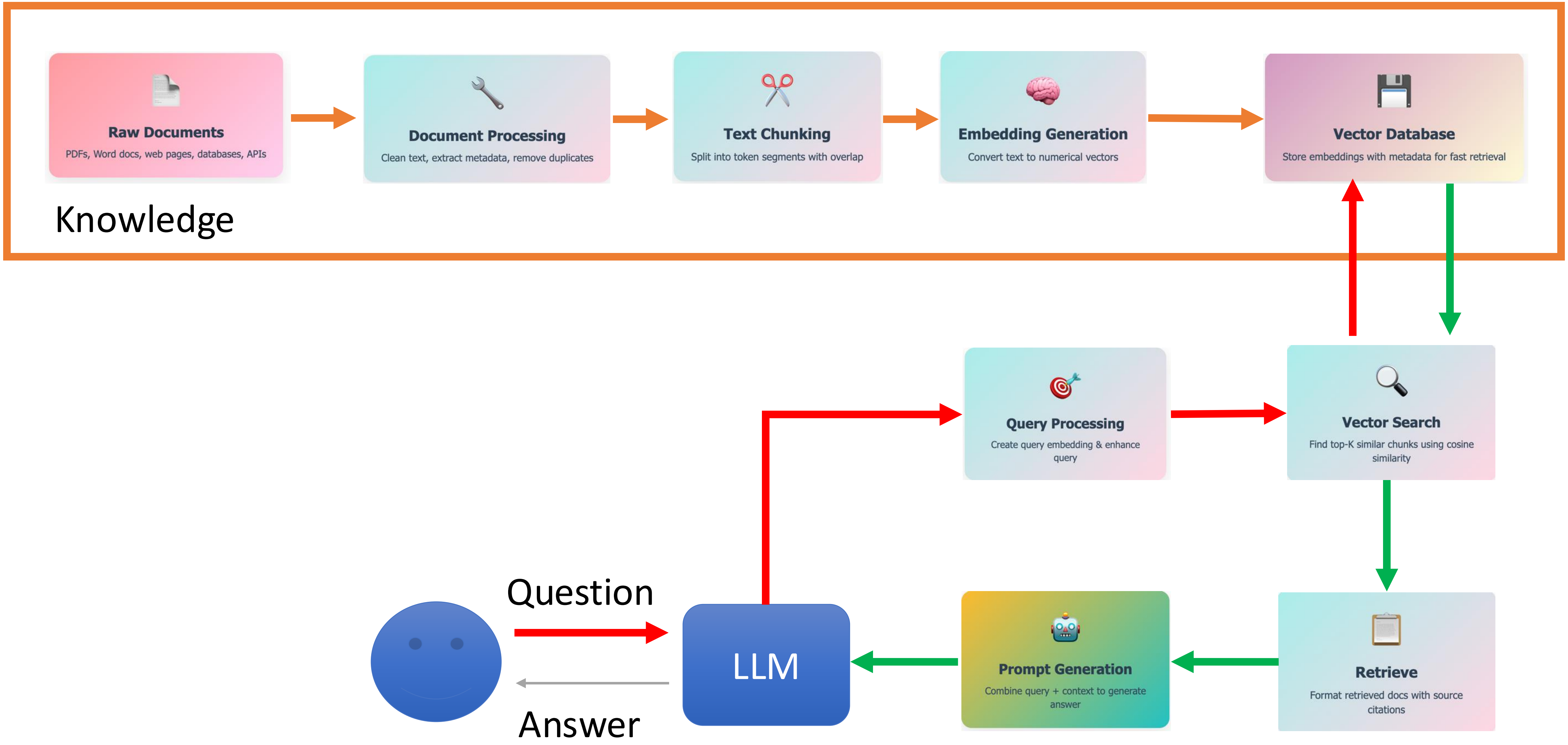
Cost and Scalability

- Retraining large models is expensive and time-consuming
- RAG allows updating knowledge without retraining

RAG Pipeline

Main Components

1. **Document Store/Knowledge Base**
   1. Collection of documents, databases, or information sources
   2. Pre-processed and indexed for efficient searching
2. **Retrieval System**
   1. Searches and finds relevant information based on user query
   2. Uses techniques like semantic search, keyword matching
3. **Language Model**
   1. Generates response using both the query and retrieved information
   2. Combines retrieved context with its trained knowledge


Raw Documents
PDFs, Word docs, web pages, databases, APIs
Document Processing
Clean text, extract metadata, remove duplicates
Text Chunking
Split into token segments with overlap
Embedding Generation
Convert text to numerical vectors
Vector Database
Store embeddings with metadata for fast retrieval
Knowledge
Query Processing
Create query embedding & enhance query
Vector Search
Find top-K similar chunks using cosine similarity
Question
LLM
Answer
Prompt Generation
Combine query + context to generate answer
Retrieve
Format retrieved docs with source citations

Use-Cases

Question: "What are the latest COVID-19 vaccination guidelines?"

- **Traditional Model**: Might give outdated 2021 information
- **RAG System**: Retrieves current CDC guidelines and provides up-to-date answer

Use-Cases

Customer Support

- **Use Case**: Automated helpdesk with company knowledge base
- **Benefit**: Consistent, accurate answers from documentation

Legal Research

- **Use Case**: Finding relevant cases and precedents
- **Benefit**: Faster research with proper citations

Medical Diagnosis Support

- **Use Case**: Retrieving relevant medical literature for symptoms
- **Benefit**: Evidence-based recommendations

Enterprise Q&A

- **Use Case**: Company-wide knowledge sharing system
- **Benefit**: Employees get instant access to policies, procedures

Content Creation

- **Use Case**: Research-backed article and report generation
- **Benefit**: Factually accurate, well-sourced content

Type of RAG

1. Dense Retrieval RAG

- Uses neural embeddings for semantic similarity
- Better at understanding context and meaning
- Examples: FAISS, Pinecone, Weaviate

2. Sparse Retrieval RAG

- Uses traditional keyword-based search [Term Frequency – Inverse Document Frequency (TF-IDF), Best Match 25]
- Good for exact matches and specific terms
- Examples: Elasticsearch, Solr

3. Hybrid RAG

- Combines dense and sparse retrieval
- Best of both worlds - semantic and keyword matching
- More robust retrieval performance

4. Multi-Modal RAG

- Works with text, images, audio, video
- Can retrieve and process different content types
- Emerging area with growing applications

Aspect	Dense Retrieval RAG	Sparse Retrieval RAG
Representation	Dense vectors from an embedding model (e.g., MiniLM, E5, BGE)	Bag-of-words weights (TF-IDF / BM25)
Index/Score	Vector index (e.g., FAISS), cosine/dot similarity	Inverted index, lexical scoring (BM25)
Strengths	Catches semantics & synonyms; good for paraphrases and long, natural questions	Nails exact terms, IDs, codes, numbers, quoted phrases; fast, transparent
Weaknesses	Can miss rare strings/IDs; depends on embedding quality/domain match; larger vectors	Misses paraphrases/synonyms; brittle to typos/inflection unless normalized
Example	“Explain trade-offs of chunk overlap” (conceptual)	“Find CVE-2025-1234” / “Section 3.2.1” / product codes
Typical use in RAG	Lab 4 currently (FAISS + sentence embeddings)	Lab 3: Next -> Add TF-IDF/BM25 retriever for keywords/IDs

RAG Tools & Technologies

Vector Databases

- **Pinecone**: Managed vector database service
- **Weaviate**: Open-source vector search engine
- **Chroma**: Lightweight embedding database
- **FAISS**: Facebook's similarity search library

Frameworks & Libraries

- **LangChain**: Popular RAG development framework
- **LlamaIndex**: Data framework for LLM applications
- **Haystack**: Open-source NLP framework
- **Semantic Kernel**: Microsoft's SDK for AI orchestration

Measuring Quality

- **Retrieval**: Hit@K, Recall@K, nDCG, duplicate rate.
- **Answers**: Faithfulness/groundedness, Exact Match (EM)/F1 for factoids, and human preference.
  Tune chunk size/overlap/Top-K first; then try **MMR or reranking**.”

MMR = **Maximal Marginal Relevance**
Goal: pick a diverse, relevant top-K by balancing **relevance to the query** and **novelty vs. already-selected items**.

nDCG = **normalized Discounted Cumulative Gain**.
Goal: Evaluating retrievers/rerankers with multi-level relevance labels.