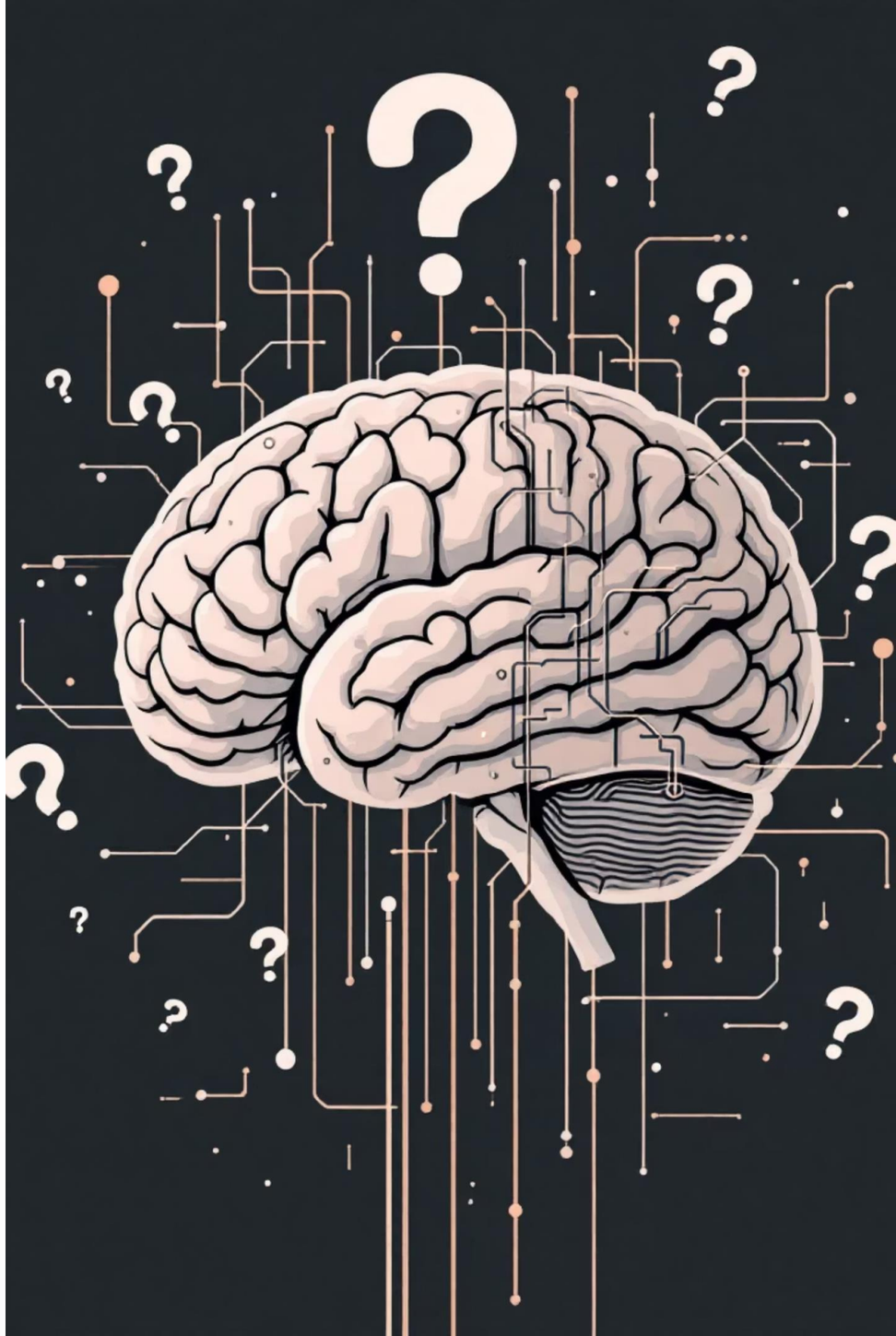
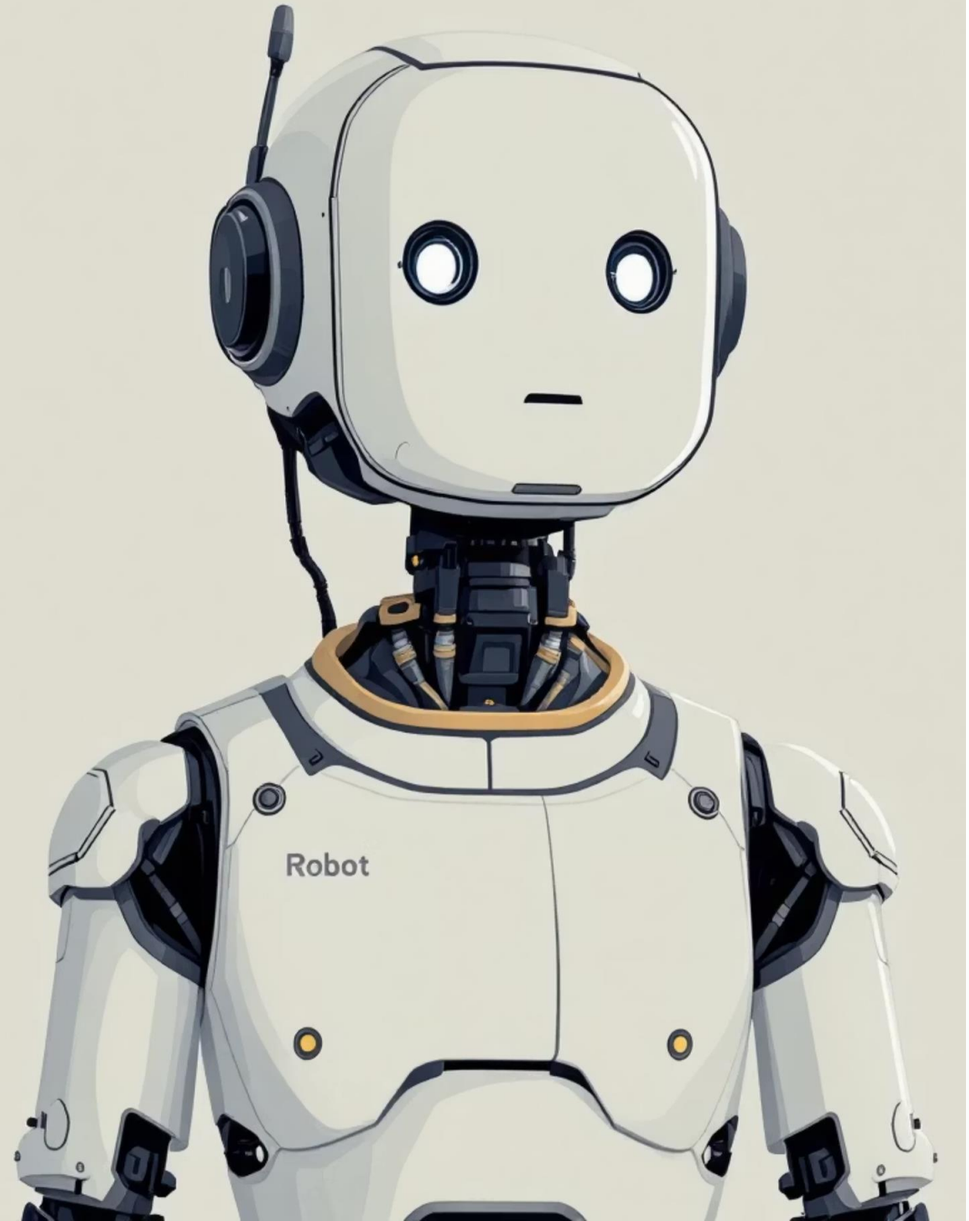
Day 7
Retrieval-Augmented Generation (RAG)

Why LLMs Go Wrong: Understanding Hallucinations

Large Language Models (LLMs) operate by predicting the next token based on vast patterns learned during their training. However, they lack direct access to the latest information or private knowledge bases . This fundamental limitation often leads them to confidently guess when faced with queries outside their training data, resulting in what we call hallucinations .

Example: If you ask an LLM, "Who won IPL 2025?", it might invent a winner and match details, despite the event not having occurred yet.





What Exactly is an LLM Hallucination?

Sounds Confident

The LLM delivers information with conviction, as if it were factual.

Factually Incorrect

The generated content is false, misleading, or unsupported by real-world data.

Common Causes:

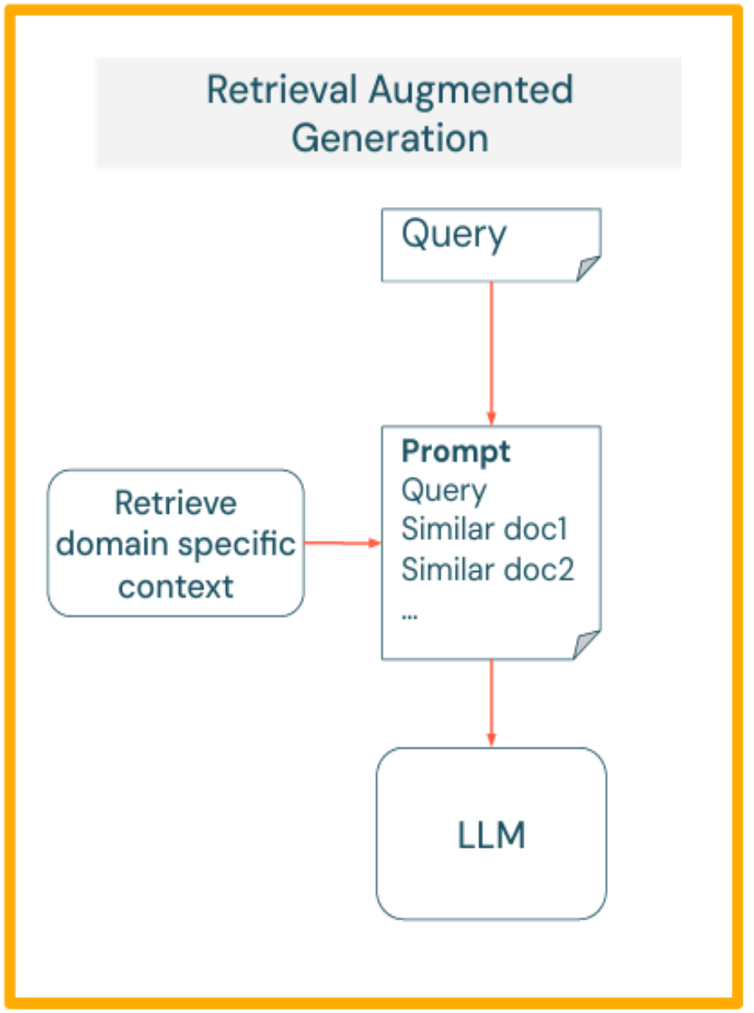
- Missing Knowledge: The LLM's training data doesn't cover the specific information requested.
- Ambiguous Prompts: Vague or unclear user queries can lead the model to make assumptions.
- Over-Generalization: The model applies patterns too broadly, leading to incorrect inferences.

RAG is...

Retrieval **A**ugmented **G**eneration, or RAG;

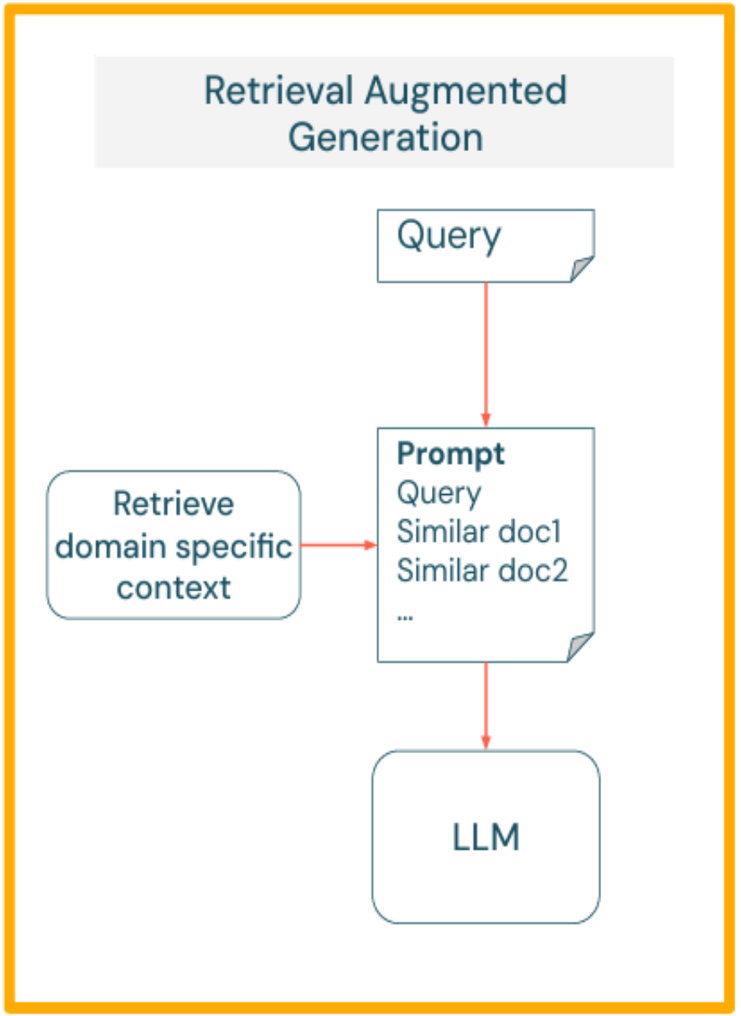
- Is a pattern (Lewis et al. 2020) that can improve the **efficacy** of large language model (LLM) applications by **leveraging custom data**.

- Is done by **retrieving** data/documents relevant to a question or task and providing them as context to **augment** the prompts to an LLM to improve **generation**.



RAG is...

- The main problem that is solved with RAG architecture is the **knowledge gap**. This approach enhances the accuracy and relevance of responses.



Main Concepts of RAG Workflow

Components and concepts in search and RAG workflow

Index & Embed:

An embedding model used for creating **vector representation** of the documents and user queries.

Vector Store:

Specialized to store unstructured data indexed by vectors. Vectors can be stored with a **vector DB, library, or plugin**.

Retrieval:

Search stored vectors using similarity search to efficiently **retrieve relevant information**.

Filtering & Reranking:

The process of selecting or ranking retrieved documents before passing as context.

Main Concepts of RAG Workflow

Components and concepts in search and RAG workflow

Prompt Augmentation:

Prompt engineering workflow to **enhance context via injection of data retrieved** from Vector store

Generation:

A large language model used for generating a response for the user's request.

RAG Architecture: A High-Level Overview

Retrieval Augmented Generation combines the strengths of information retrieval with the generative power of LLMs. Here's how the pipeline works:

01

User Query

The user submits a question or prompt to the RAG system.

02

Retrieval from Knowledge Base

The system searches a vast knowledge base (documents, databases) to find relevant context.

03

Context + Query to LLM

The retrieved context is combined with the original query and sent to the LLM.

04

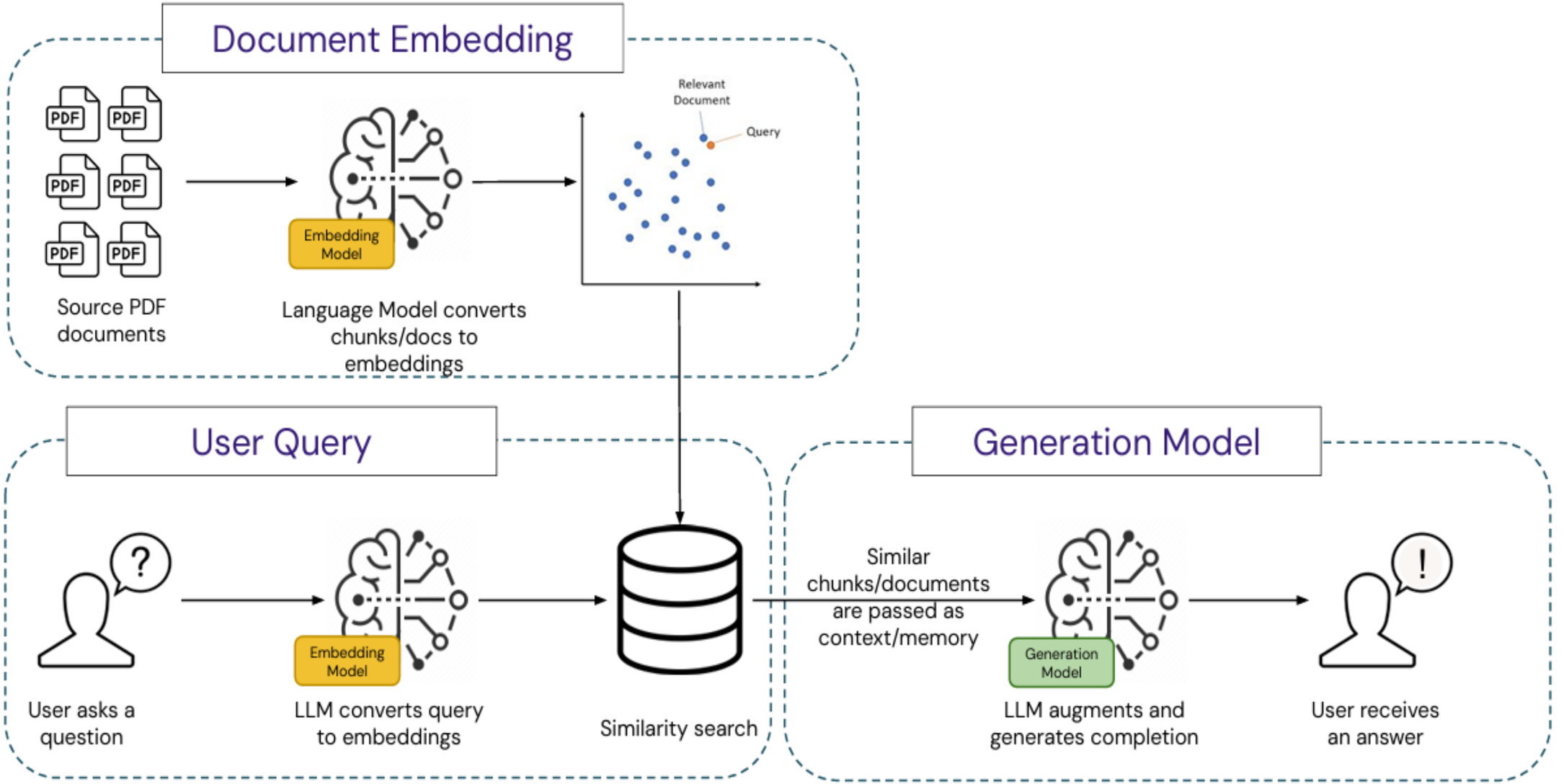
Response with Citations

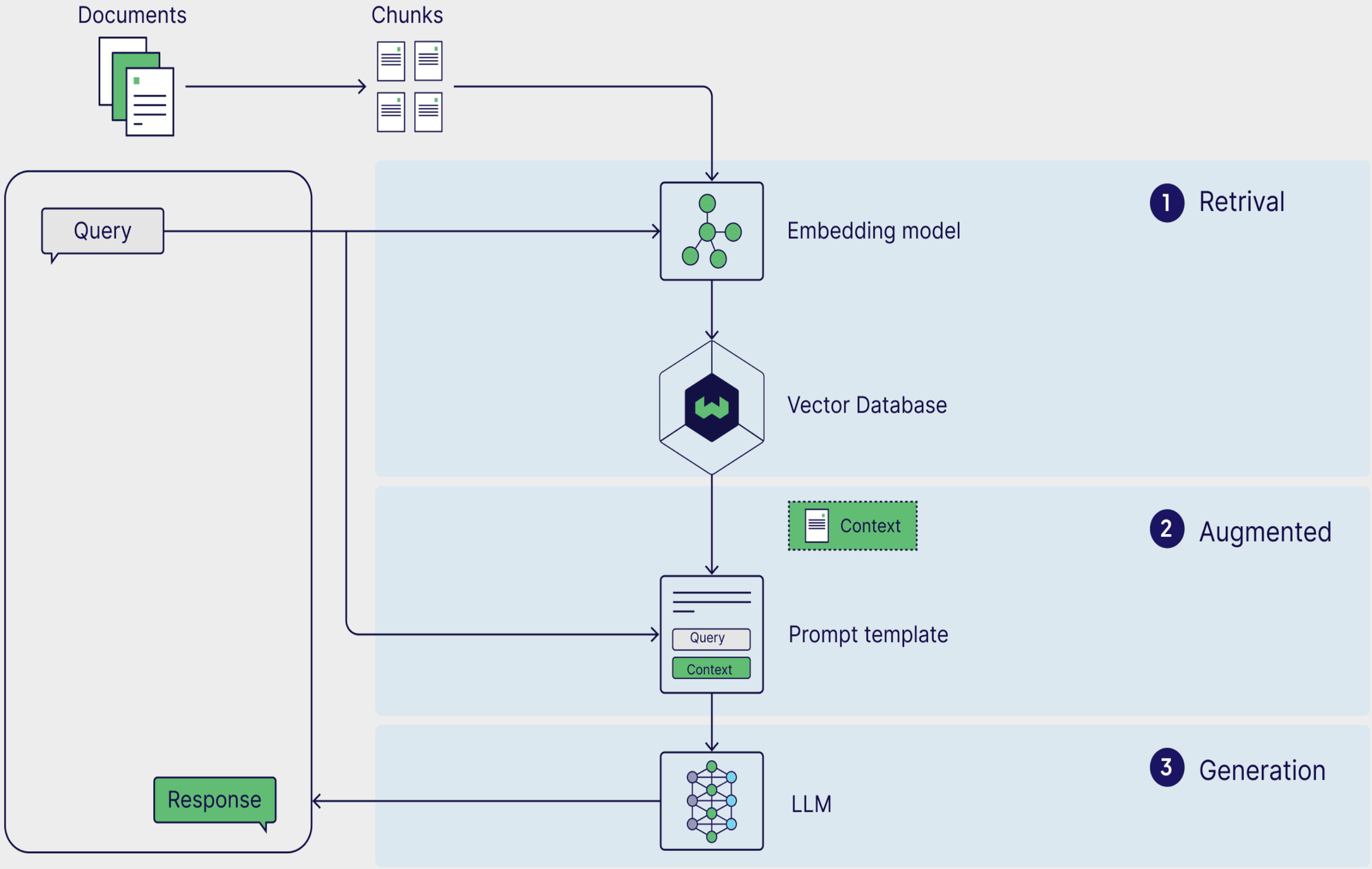
The LLM generates a response, referencing the sources from which the information was retrieved.

▯ Retrieval + Generation — This synergistic approach offers the best of both worlds: factual accuracy and fluent, coherent text generation.

Retrieval Augmented Generation (RAG)

A sample architecture




Documents
Chunks
Query
Embedding model
1 Retrival
Vector Database
Context
2 Augmented
Query
Context
Prompt template
LLM
3 Generation
Response

Why We Need Retrieval Augmented Generation (RAG)

The core challenge with traditional LLMs is that their internal parameters are frozen once training is complete. They cannot inherently access new information in real-time.

> ### The RAG Solution:
>
> - Integrates real, up-to-date documents into the generation process.
> - Provides domain-specific knowledge crucial for specialized applications.
> - Enables source citations, allowing users to verify information.

The result: The LLM becomes more truthful and grounded in verifiable data, drastically reducing hallucinations.

RAG vs. Fine-Tuning: Choosing the Right Approach

When enhancing LLM performance, RAG and fine-tuning are two primary strategies. Understanding their differences is key to selecting the optimal solution.

Feature	RAG	Fine-Tuning
Speed	Fast deployment & updates	Slow (requires retraining)
Data Size	Few documents needed	Thousands of examples required
Cost	Low computational cost	High computational cost
Dynamic Updates	Very easy to refresh data	Requires re-training the model
Best Use	Private or latest factual data	Adapting to new behaviors or styles

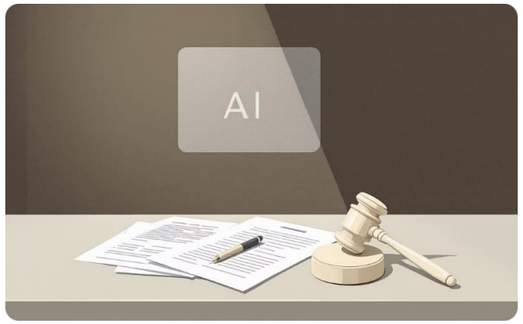
Conclusion: Always consider RAG as your initial strategy before opting for the more resource-intensive fine-tuning process.

RAG in the Real World: Practical Applications

RAG's ability to provide accurate, up-to-date, and verifiable information makes it indispensable across various industries.

Enterprise Chatbots

Enhances customer service in banking and healthcare with precise information.

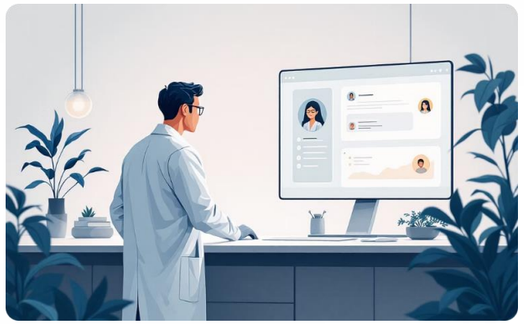


Legal & Policy Assistants

Provides rapid access to legal precedents and policy documents for quick analysis.

Code Documentation Bots

Helps developers navigate complex codebases and generate accurate documentation.



Research Assistants

Expedites literature reviews and data synthesis across scientific domains.

Key Takeaway: If an LLM leverages company-specific or proprietary data for its responses, it is almost certainly employing a RAG architecture.

Diverse Retrieval Sources for RAG Systems

The power of RAG lies in its flexibility to integrate knowledge from almost any digital source, transforming raw data into searchable intelligence.

PDFs

Websites

SQL Databases

APIs

SharePoint / Drive

Essentially, any form of text-based information can be processed and indexed to become part of the LLM's dynamic knowledge base, enabling robust and relevant responses.

What Makes a Good RAG System?

An effective RAG system is built on several critical components that ensure accuracy, relevance, and reliability in LLM-generated responses.

→ Correct & Relevant Retrieval

Ensuring the system pulls the most accurate and pertinent information.

→ Structured Chunking

Breaking down documents into optimally sized segments for efficient retrieval.

→ Top-k Contextual Grounding

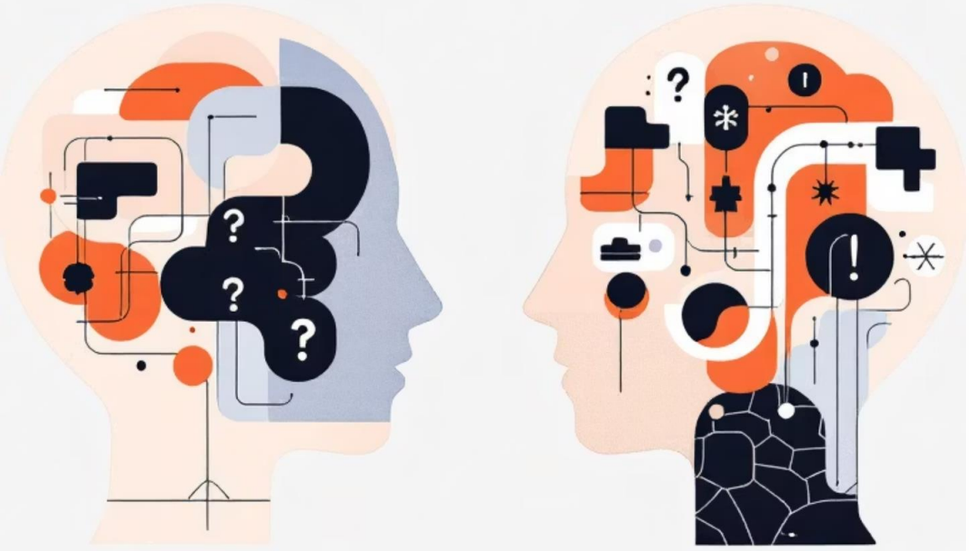
Selecting the most relevant 'k' chunks to provide sufficient context to the LLM.

→ Citations + Verifiable Sources

Providing clear references to allow users to cross-reference and build trust.

→ Hallucination Fallback Strategies

Implementing mechanisms to detect and mitigate potential hallucinations.



Fixing Hallucinations with Retrieval

Let's illustrate the direct impact of retrieval on mitigating LLM hallucinations using a simple example:

LLM-only

When asked a question beyond its training data (e.g., "What were the latest Q3 earnings for Company X?"), an LLM without retrieval might:

- Invent financial figures.
- State outdated information.
- Generate a plausible but completely false report.

Result: Hallucinates and provides incorrect information.

Retrieval-enabled

With retrieval, the same question triggers a search across real-time financial reports or company databases. The LLM then:

- Accesses the latest Q3 report.
- Extracts accurate earnings data.
- Cites the source of the information.

Result: Provides a correct and verifiable response.

When RAG Systems Fall Short

While powerful, RAG systems are not foolproof. Their effectiveness hinges on the quality of their components and implementation.

Poor Chunking

Documents are split too broadly or too narrowly, hindering retrieval accuracy.

Wrong Search Algorithm

Ineffective algorithms fail to match queries with relevant document sections.

Missing Documents

Critical information is not included in the knowledge base, leading to gaps.

Low-Quality Embeddings

Poor semantic representation of text leads to inaccurate retrieval matches.

Understanding these common pitfalls is the first step. We will explore strategies to improve and optimize these aspects in our upcoming sessions.