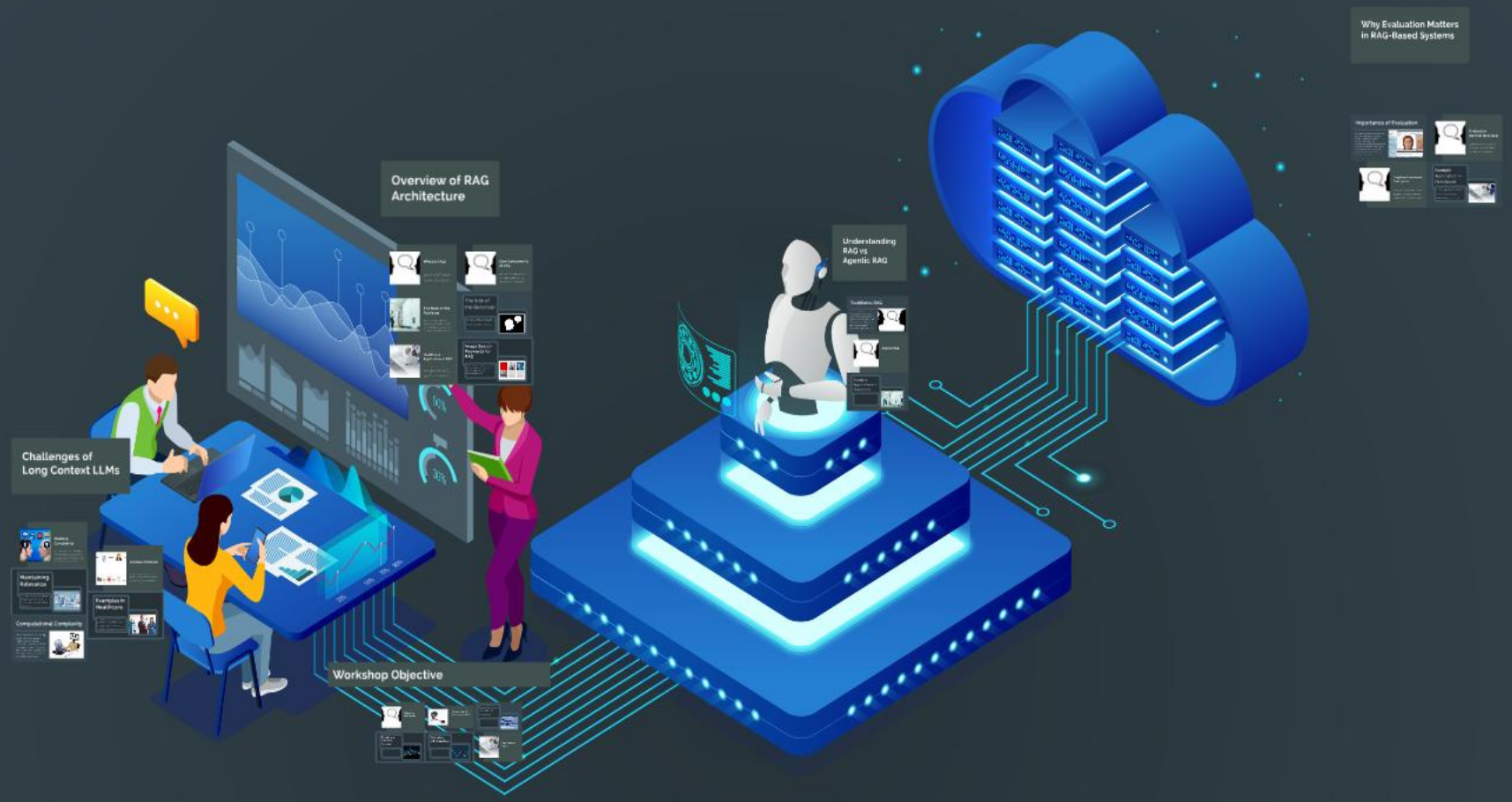
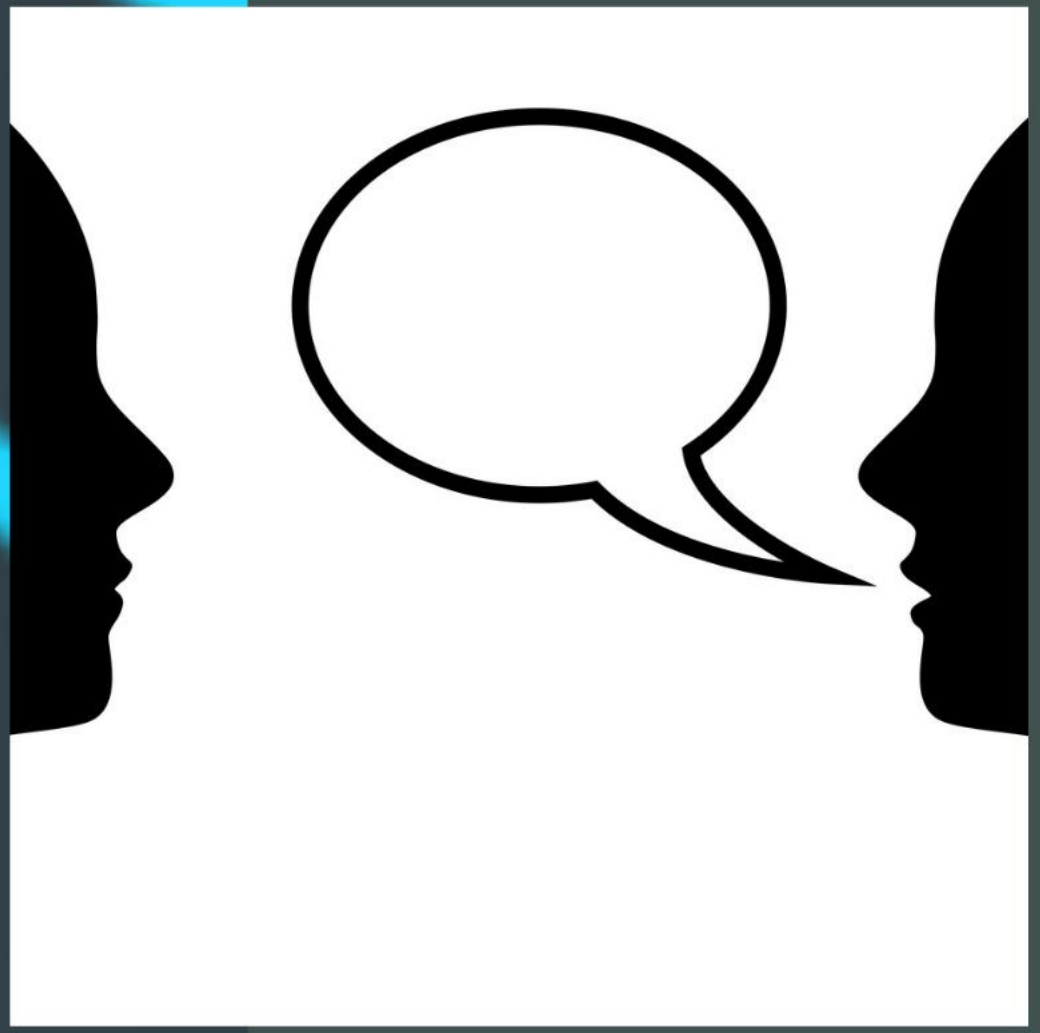


Workshop on Conversational Advanced Multi-PDF RAG System with DeepEval

Building and Evaluating RAG-Based Conversational Systems in Healthcare

Workshop Objective



Objective Statement

To build and evaluate a RAG-based conversational system using single and multiple PDF sources on Azure platform leveraging Azure OpenAI.

Workshop Activities

- Creating a multi retrieval system from PDFs.
- Storing embeddings in ChromaDB without duplication.
- Evaluating performance using DeepEval

Example Learning Outcomes in Healthcare

Learners will:

- Develop a system to answer medical queries using a single clinical guideline document (PDF).
- Expand this system to handle complex queries involving multiple medical research studies, evaluating conversational accuracy and retrieval efficiency using DeepEval.

Challenges of Long Context LLMs

Memory Constraints

Long-context LLMs struggle with memory constraints, making it difficult to store and access extensive information effectively. This limitation can negatively impact the model's performance, particularly when dealing with large datasets or historical records.

Maintaining Relevance

Ensuring responses are relevant becomes increasingly challenging as context expands. Long contexts can lead to an overflow of details, making it difficult for models to deliver concise and pertinent information.

Computational Complexity

The complexity of processing longer contexts escalates computational demands. Increased computational load can result in slower response times, which may impede real-time applications essential in sectors like healthcare.

Context Dilution

As context lengthens, critical details can become overshadowed, leading to context dilution. This can result in significant information loss, affecting the accuracy of AI-generated responses in specialized fields.

Examples in Healthcare

In healthcare scenarios, AI systems can struggle with extensive patient histories. For instance, a 10-year medical history may lead to missed critical details or critical findings in research documents due to context dilution or slow response times.

Workshop Objective

Memory Constraints

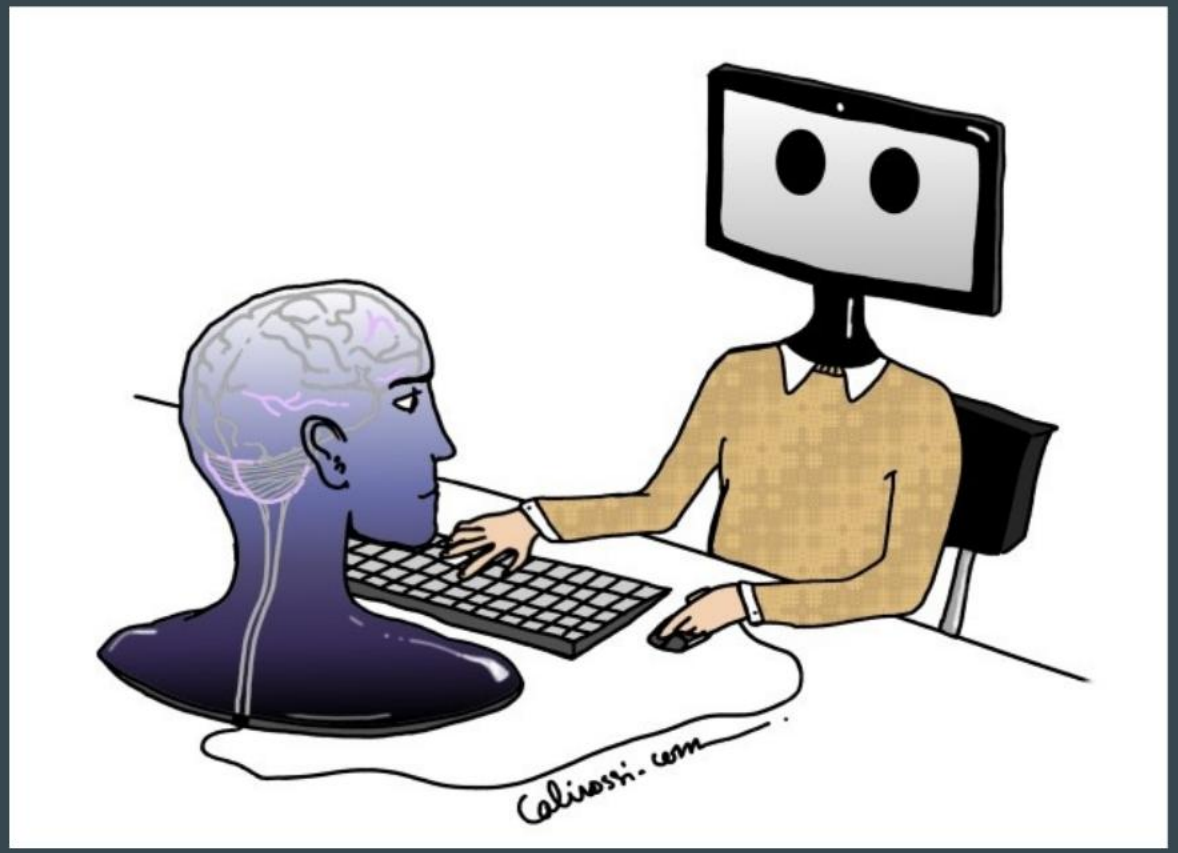
Difficulty in storing and accessing extensive contextual information within the model.

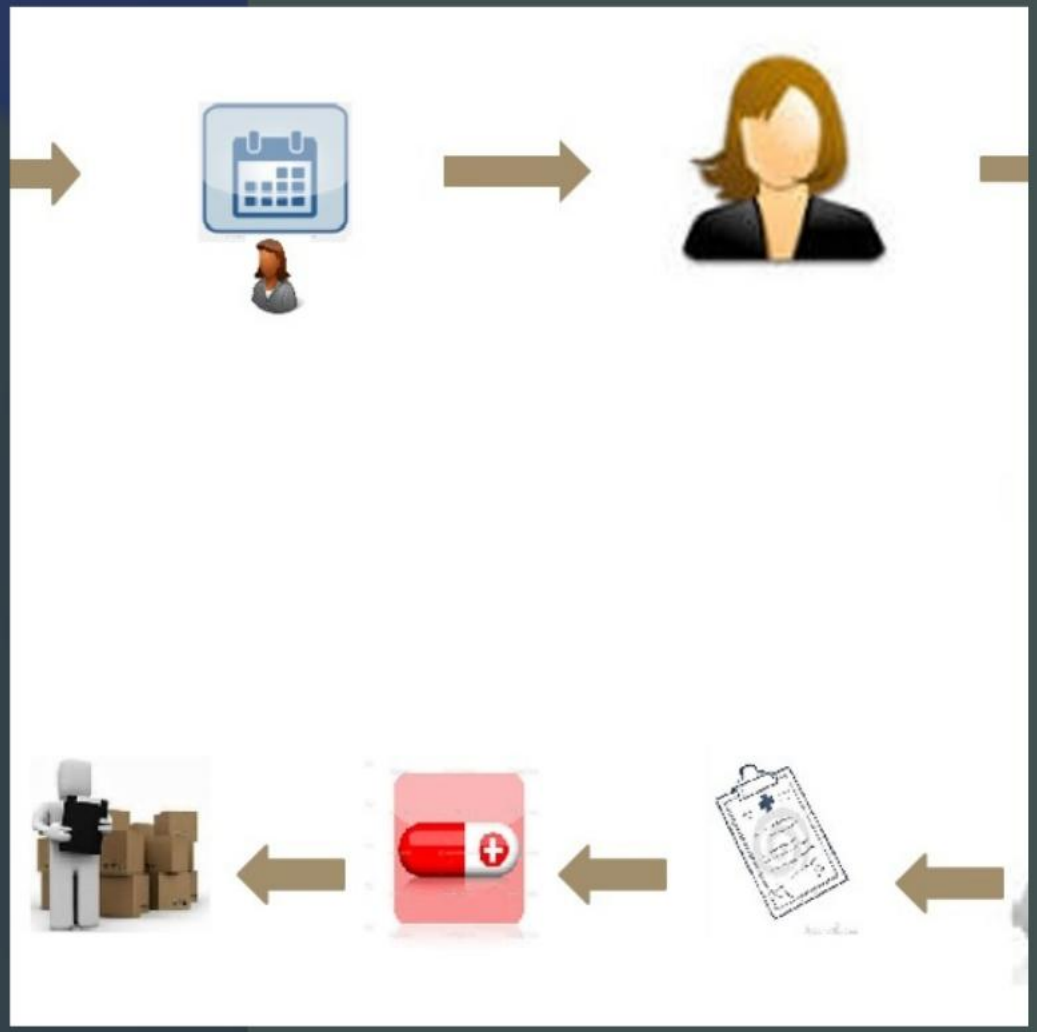
Maintaining Relevance

Challenges in ensuring responses remain directly relevant when context becomes very extensive.

Computational Complexity

Longer contexts increase computational load, causing slower response times.





Context Dilution

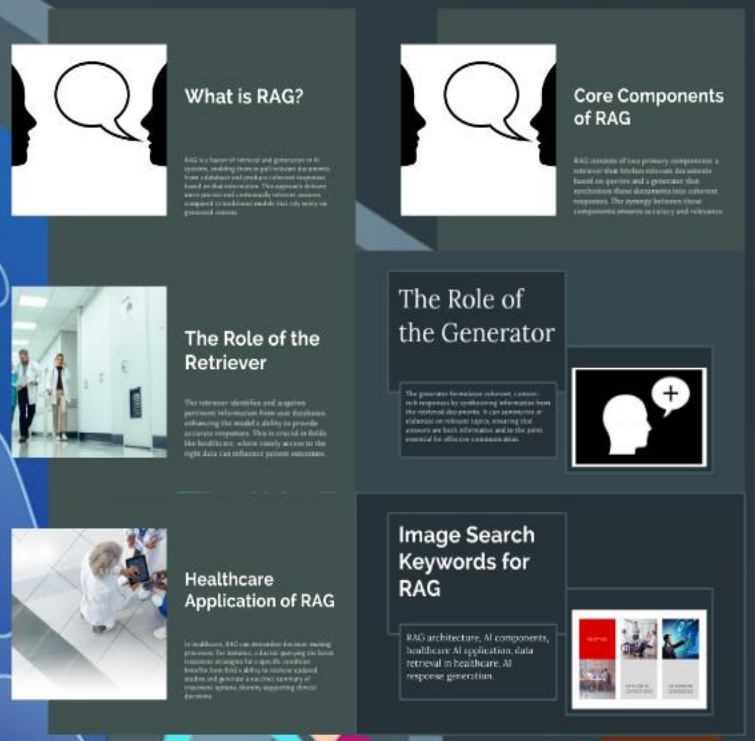
Important details may become overshadowed or diluted in extensive contexts.

.

Examples in Healthcare

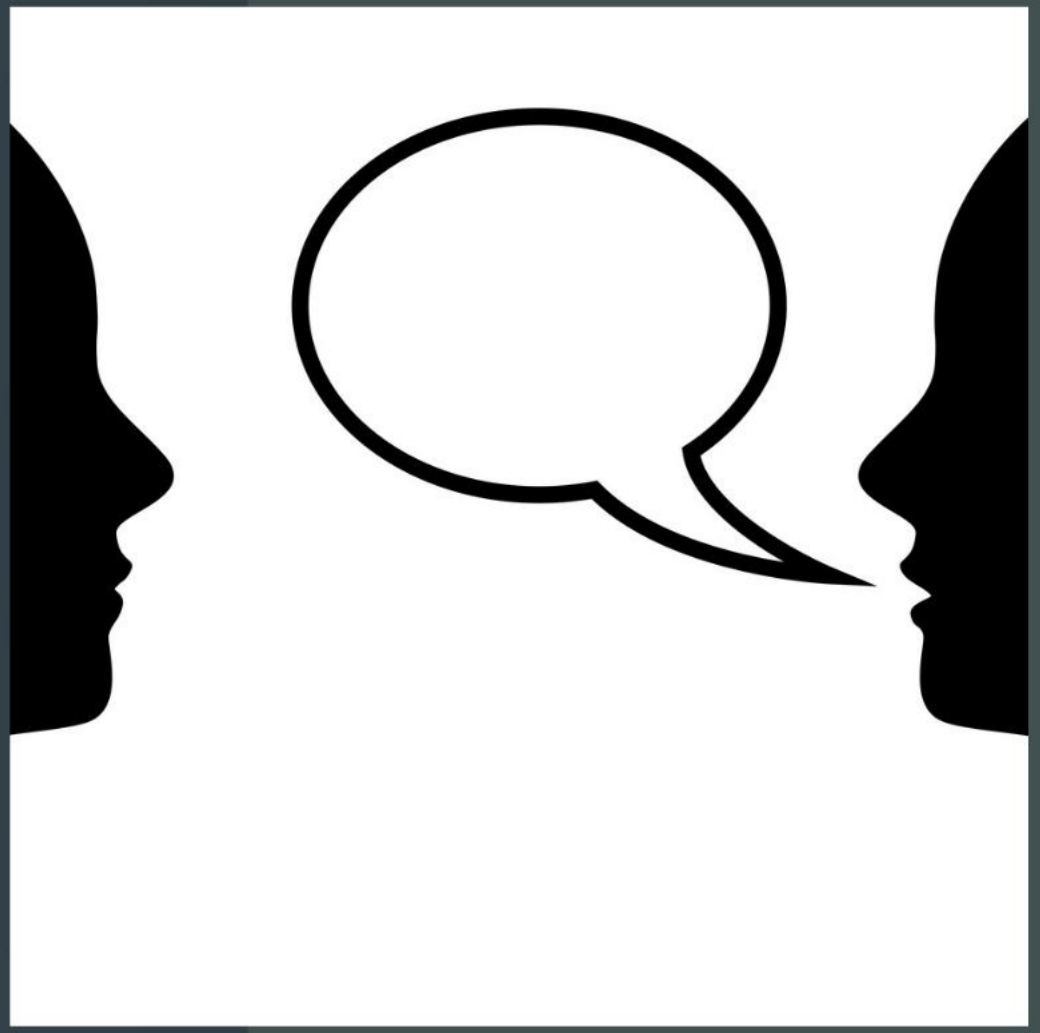
A doctor consults an AI for insights from a patient's extensive 20-year medical history; the system might miss crucial details from past medical incidents due to context dilution.

Overview of RAG Architecture



What is RAG?

Retrieval-Augmented Generation (RAG) combines retrieval mechanisms with generative language models to provide accurate, context-specific responses.



Core Components of RAG

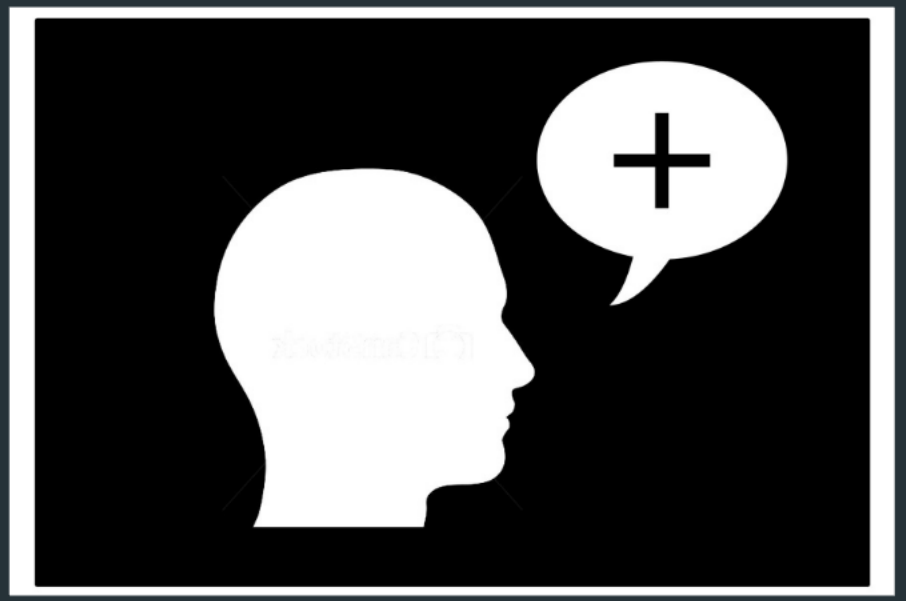
- Retriever (fetches relevant documents)
- Generator (creates responses based on retrieved content)

The Role of the Retriever

- Fetches relevant documents or context based on user queries.
- Improves accuracy and efficiency of responses by providing targeted information to LLMs.

The Role of the Generator

- Generates coherent and context-aware responses using retrieved information.
- Transforms user queries and retrieved data into natural language output.



Healthcare Application of RAG

Example (Healthcare):
Imagine a doctor asking, "What are the latest treatments for diabetes?“

The retriever finds relevant medical studies.

The generator summarizes these documents into a concise recommendation for treatment options.

Understanding RAG vs. Agentic RAG

Traditional RAG

Traditional RAG systems focus on retrieving relevant information and generating responses directly based on this information. They operate with a straightforward retrieval mechanism, limiting their scope for complex reasoning and user interaction.

Agentic RAG

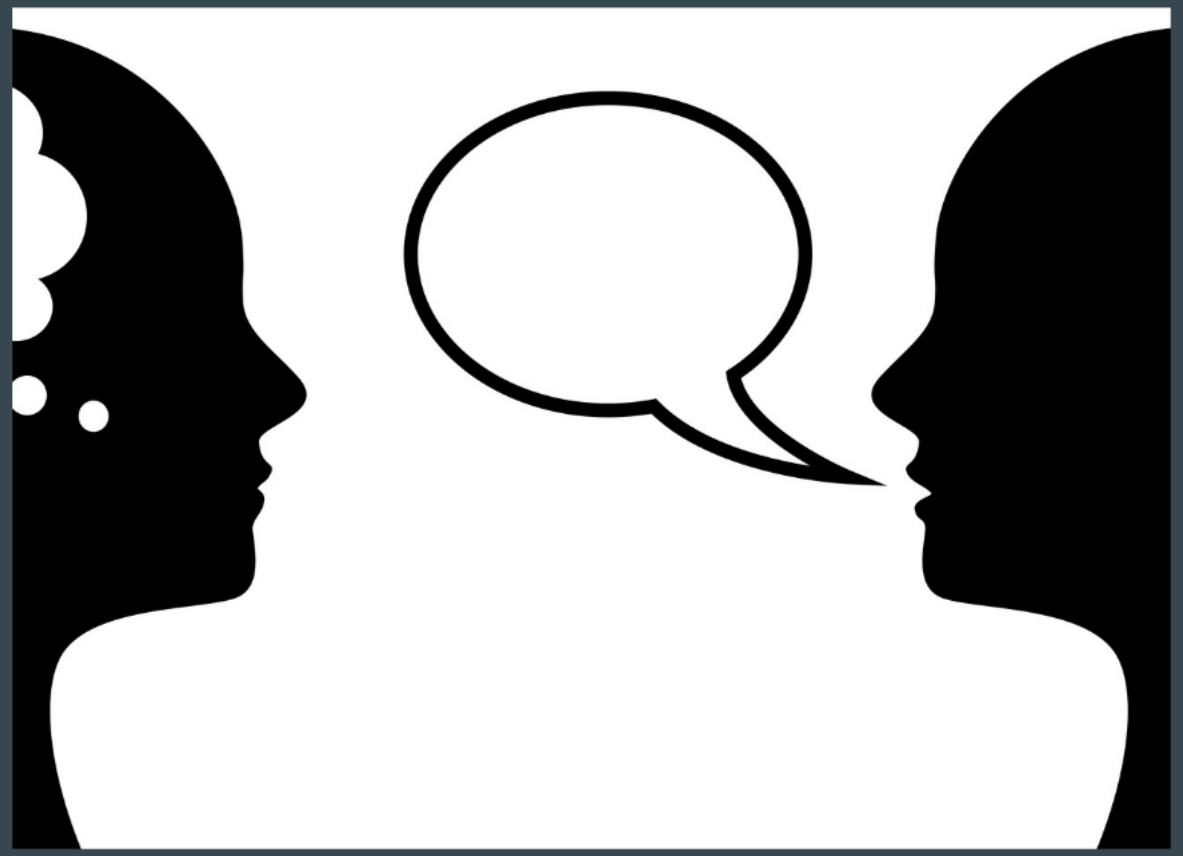
Agentic RAG enhances traditional retrieval methods by incorporating reasoning and decision-making capabilities, allowing for dynamic interaction with users. This transformation allows for personalized responses based on user context and needs.

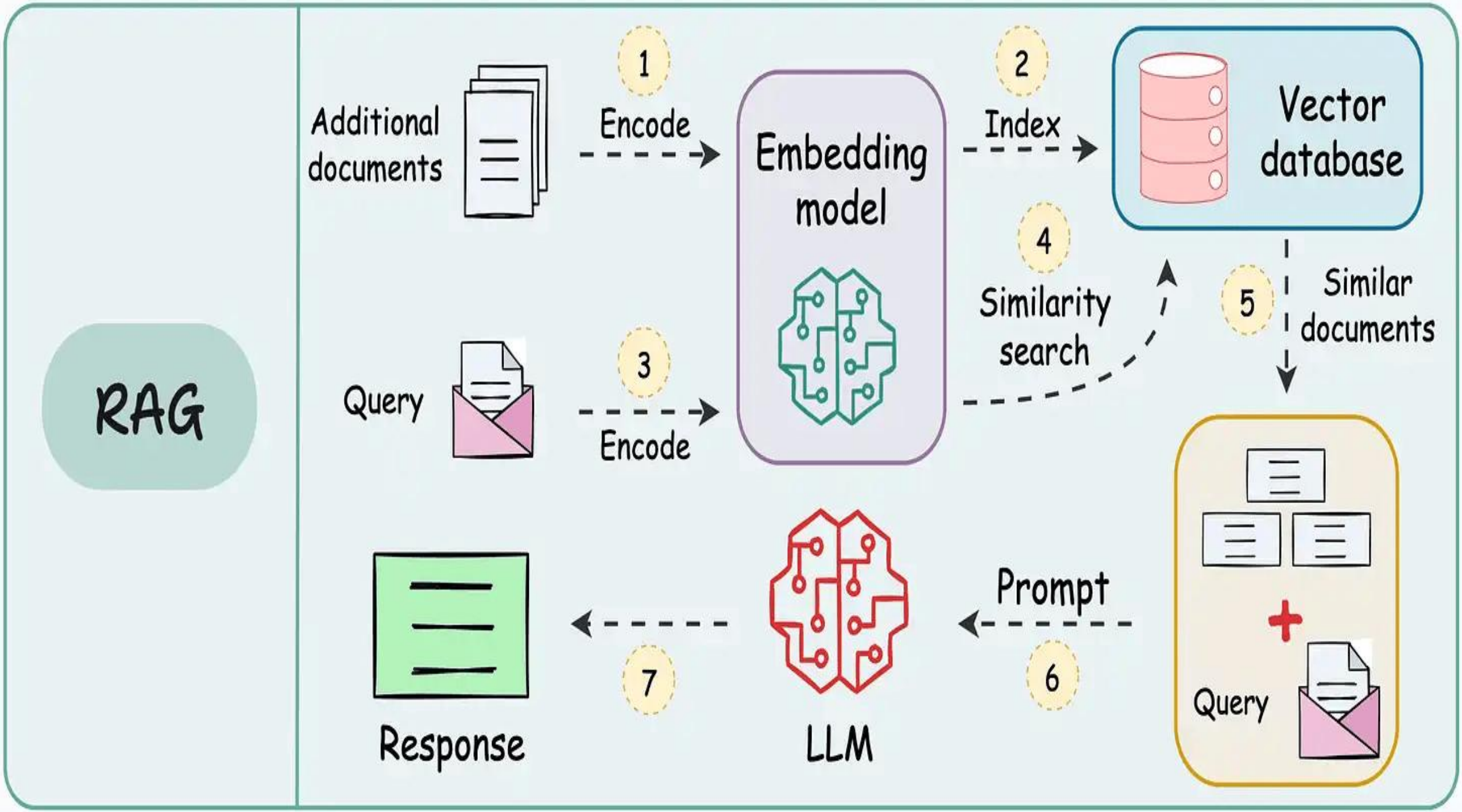
Example Applications in Healthcare

In healthcare, Traditional RAG might answer simple questions like listing symptoms, while Agentic RAG can evaluate complex patient cases, offering tailored treatment plans and continuous recommendation updates as new information emerges.

Traditional RAG

Retrieves relevant information and generates responses directly.

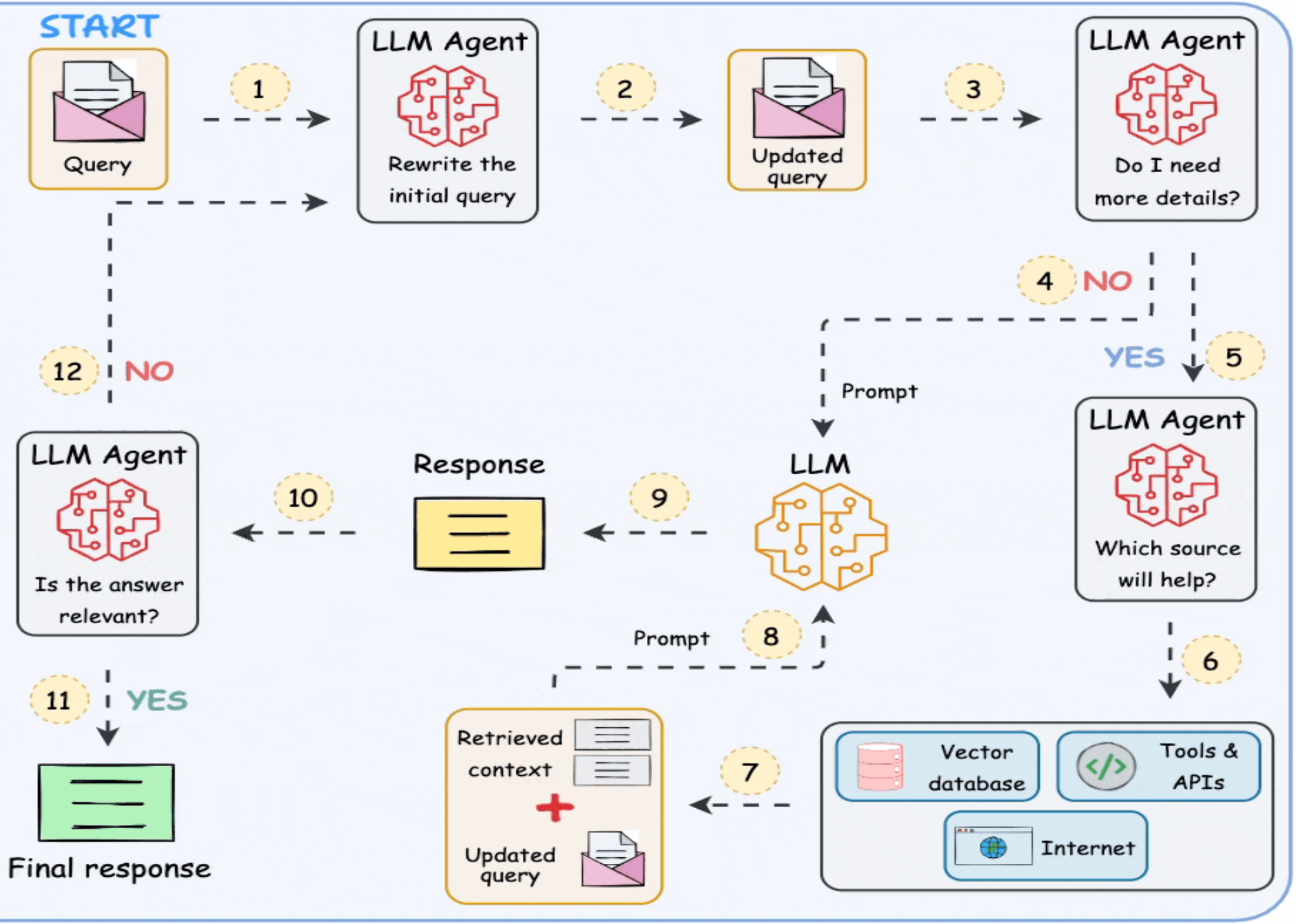



RAG
Additional documents
1
Encode
Embedding model
2
Index
Vector database
4
Similarity search
5
Similar documents
Query
3
Encode
Prompt
6
Query
7
Response
LLM

Agentic RAG

Incorporates reasoning and decision-making, allowing iterative retrieval and deeper interaction with the user.

Agentic RAG


START
Query
1
LLM Agent
Rewrite the initial query
2
Updated query
3
LLM Agent
Do I need more details?
4
NO
YES
5
LLM Agent
Which source will help?
6
Vector database
Tools & APIs
Internet
7
Retrieved context
Updated query
Prompt
8
LLM
9
Response
10
LLM Agent
Is the answer relevant?
11
YES
Final response
12
NO
Prompt

Example Applications in Healthcare

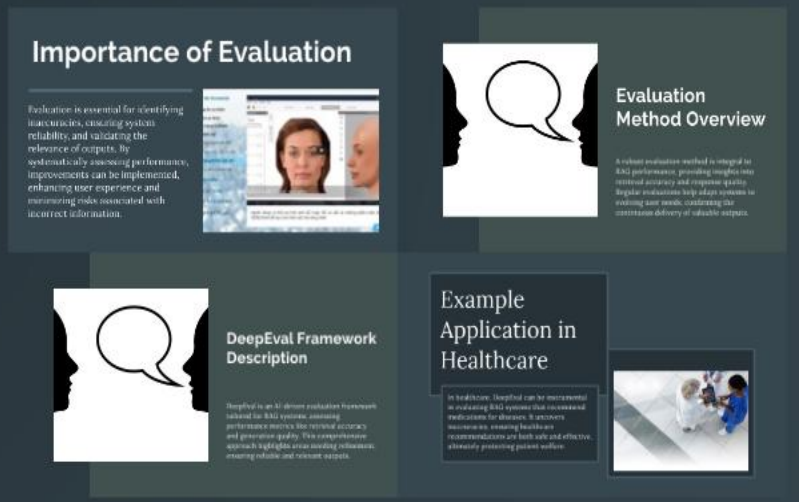
Traditional RAG:

User asks, "What are the symptoms of hypertension?" Immediate retrieval: list of symptoms.

Agentic RAG:

User (doctor) asks, "Diagnose and suggest a treatment plan for a patient exhibiting symptoms A, B, and C." The agent retrieves medical records, guidelines, and iteratively refines answers, suggesting follow-up diagnostics or personalized treatment options.

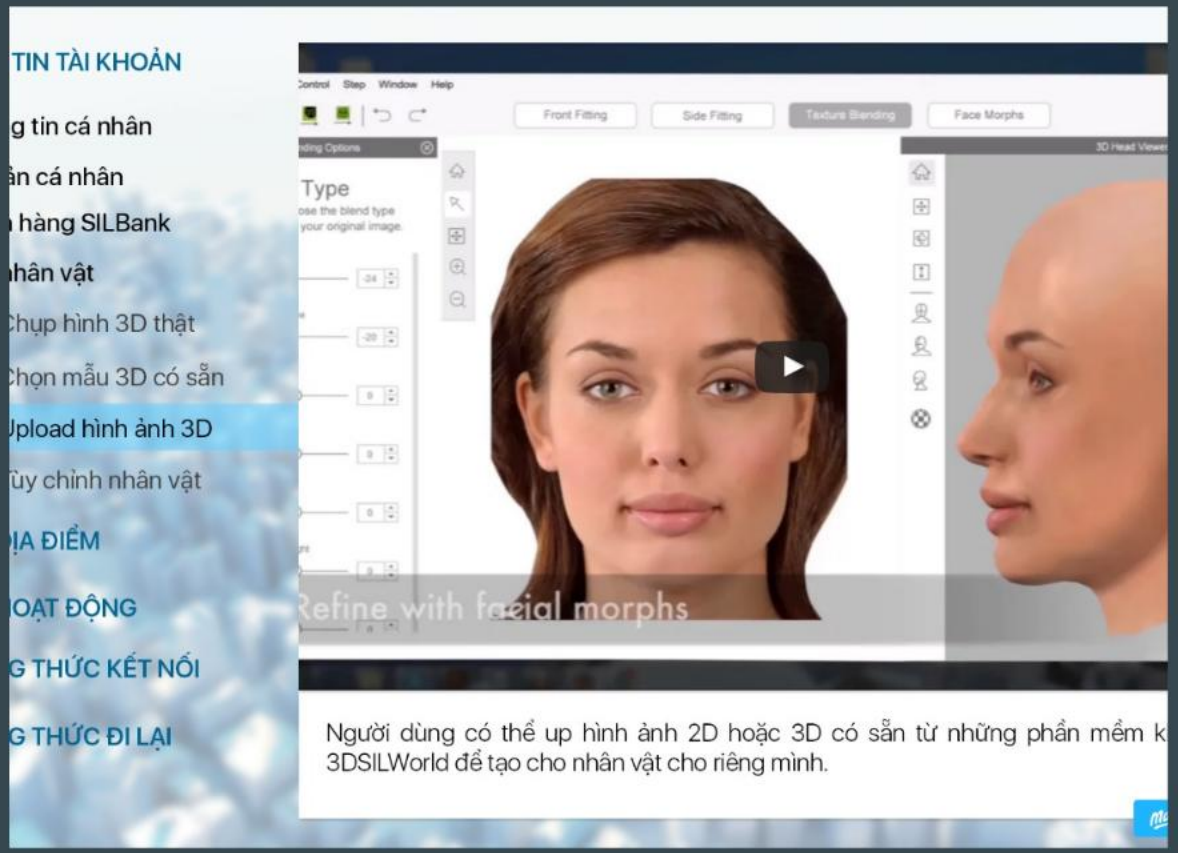
Why Evaluation Matters in RAG-Based Systems

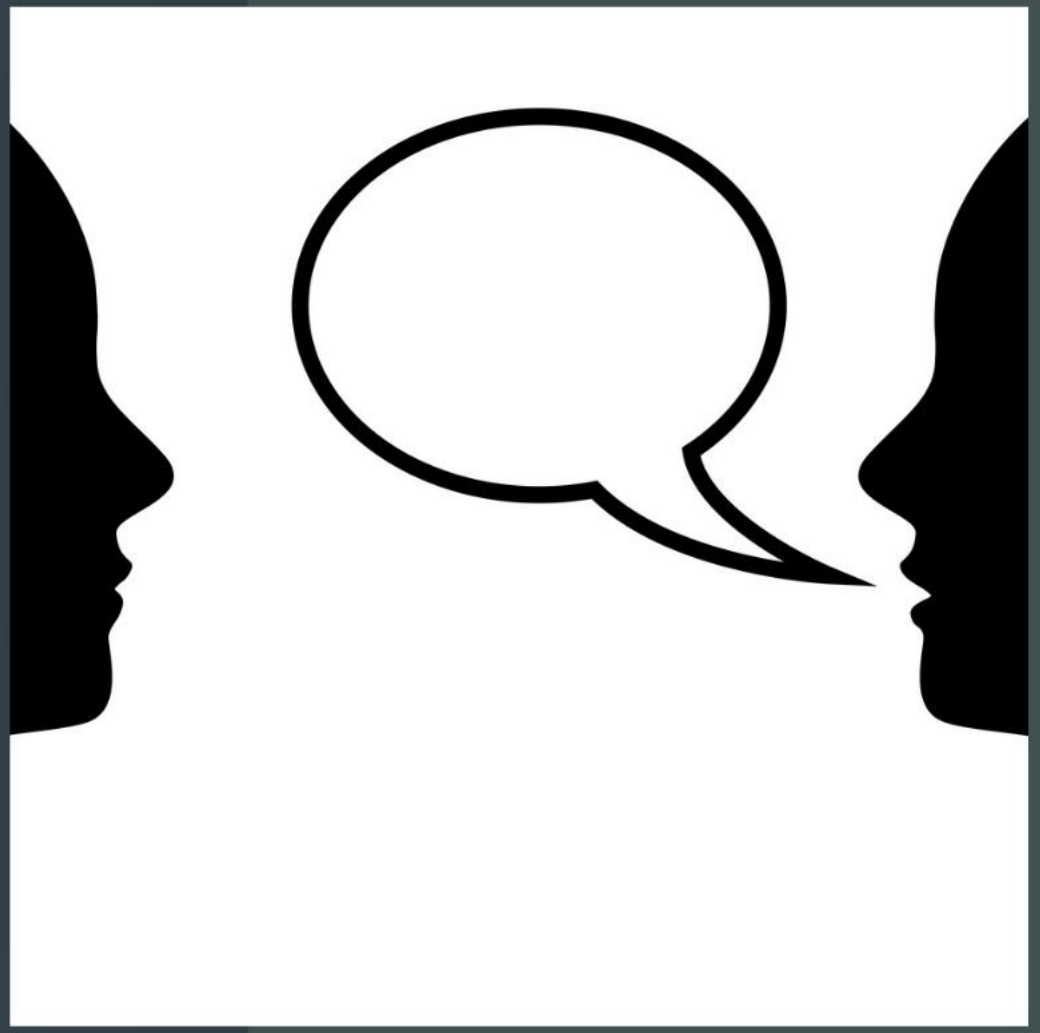


Importance of Evaluation

Ensures accuracy, relevance, and quality of generated responses.

Highlights limitations and informs improvements.





Evaluation Method Overview

DeepEval:
An AI-driven evaluation framework that assesses retrieval accuracy, generation quality, relevance, and overall performance.

Example Application in Healthcare

- Using DeepEval to evaluate a RAG system identifying medications for specific diseases:
- DeepEval highlights inaccuracies or outdated information, ensuring safe and effective healthcare recommendations, reducing risks to patients.