**Types of RAG**

**RAG Overview**

RAG (Retrieval-Augmented Generation) is a framework that enhances large language models (LLMs) by combining them with external knowledge retrieval systems.

The core concept of RAG is simple but powerful:

1. **Retrieval**: When a user asks a question, the system first searches through a knowledge base (like documents, databases, or websites) to find relevant information.
2. **Augmentation**: The retrieved information is then added to the user's prompt as context.
3. **Generation**: Finally, the LLM uses both the original query and the retrieved context to generate a response.

This approach offers several key benefits:

* **Reduced** hallucinations: By grounding responses in retrieved facts, RAG helps prevent LLMs from making up information.
* **Up-to-date information**: RAG can access knowledge beyond the LLM's training cutoff.
* **Source attribution**: Responses can reference where information came from.
* **Domain-specific knowledge**: By creating specialized knowledge bases, RAG systems can become experts in particular fields.

**Types of RAG**

There are many types of RAGs are releasing periodically based on the evolution of LLM applications.

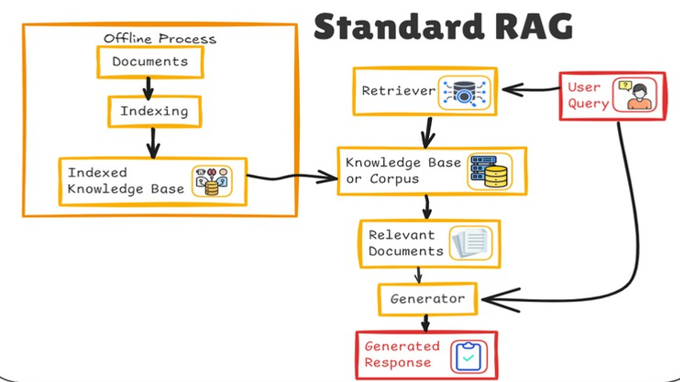
The following RAG types widely used in LLM applications.

1. **Standard RAG -**forms the foundation with its straightforward approach
2. **Iterative RAG-** improves accuracy through multiple retrieval rounds
3. **Graph RAG -**leverages relationship structures for better context
4. **Self RAG-** introduces self-evaluation capabilities
5. **Corrective RAG-** focuses on error detection and correction
6. **Rule RAG-** adds explicit constraints for more controlled outputs

**1.Standard RAG**

This works well for straightforward information retrieval approach

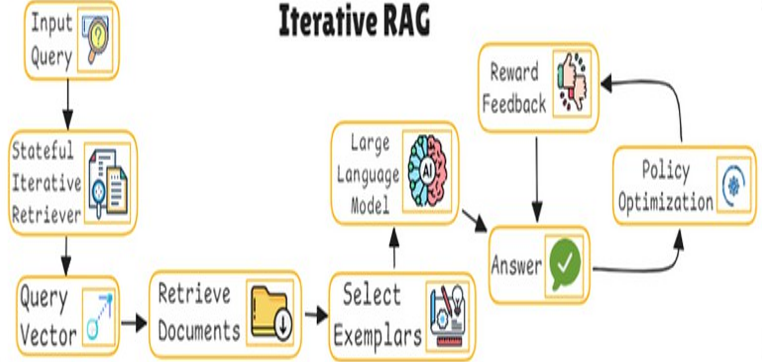
* Combines retrieval with large language models for accurate, context-aware responses.
* Breaks documents into chunks for efficient information retrieval.
* Aims for 1-2 second response times for real-time use.
* Enhances answer quality by leveraging external data sources.



**2.Iterative RAG**

This type of RAG excels at complex queries requiring multiple information retrieval steps

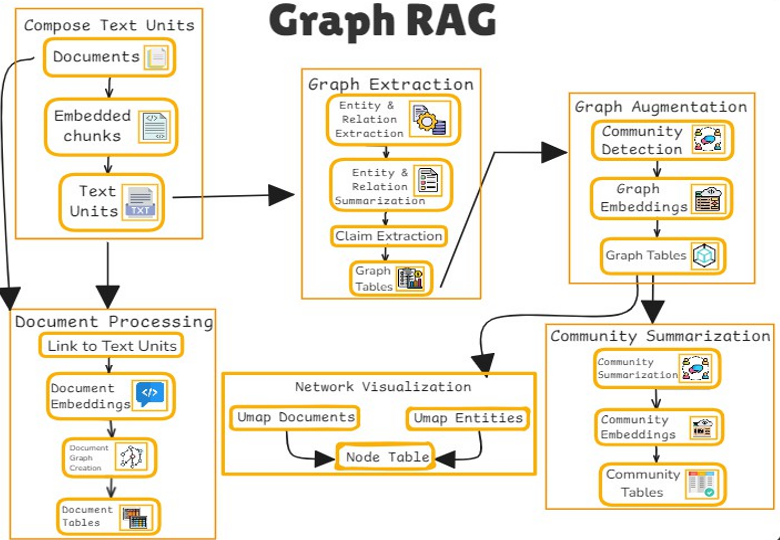
* Unlike traditional retrieval, iterative RAG performs multiple retrieval steps, refining its search based on feedback from previously selected documents.
* Retrieval decisions follow a decision process.
* Reinforcement learning improves retrieval performance.
* The iterative retriever maintains an internal state, allowing it to adjust future retrieval steps based on the accumulated knowledge from previous iterations.



**3.Graph RAG**

A variation of Retrieval-Augmented Generation (RAG) where the retrieval process utilizes graph structures.

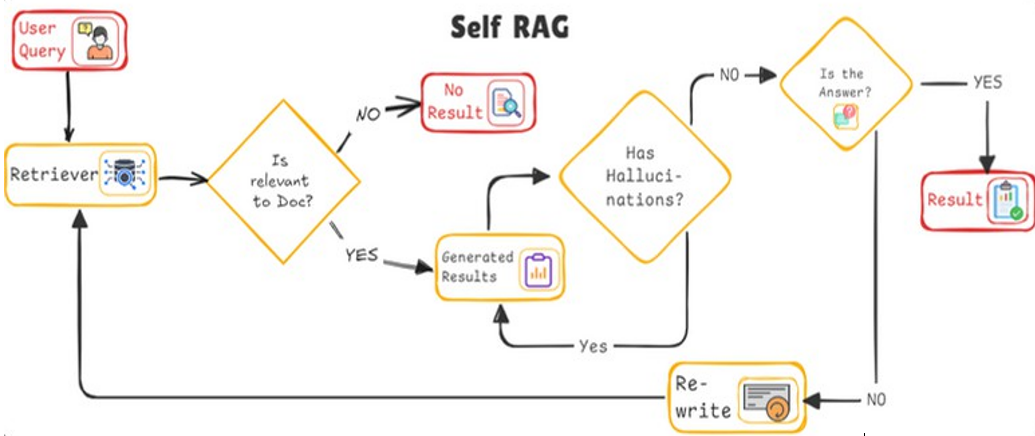
* Graphs represent entities and relationships, enhancing context-aware retrieval.
* Nodes in the graph can represent concepts, while edges denote relationships or contextual
* The graph ensures that retrieved documents or pieces of information are linked logically.
* Leveraging graph traversal enables more coherent reasoning over interrelated entities.
* This approach improves efficiency and response accuracy by keeping the knowledge graph compact and relevant.



**4.Self RAG**

Self-RAG evaluates its own retrievals and generations, deciding whether to use retrieved content or not. Focus on confidence assessment and continuous improvement.

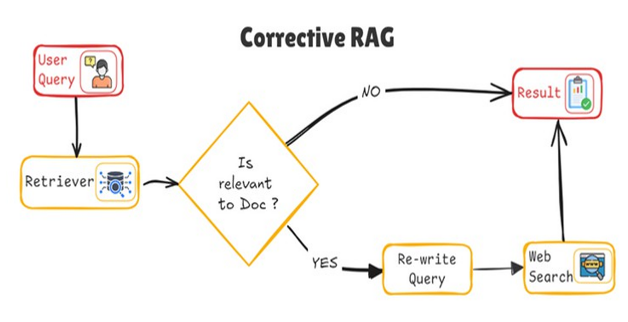
* Refines responses iteratively, improving consistency and coherence.
* Grounds responses in prior outputs for increased accuracy.
* Adapts retrieval strategies based on the conversation's evolving context.



**5.Corrective RAG**

Corrective RAG focuses on identifying and fixing errors in generated responses. This is the best type for high-accuracy requirements where factual correctness is critical

* Uses multiple passes to improve outputs based on feedback.
* Aims for higher precision and user satisfaction compared to standard RAG.
* Leverages user feedback to enhance the correction process.



**6.Rule RAG**

Rule-RAG enhances question answering by adding rule-based guidance to RAG. Focuses on delivering consistent, controlled outputs in highly regulated environments

* It retrieves documents logically relevant to queries using predefined rules.
* Rules are also used to guide answer generation for accuracy and context.
* It includes in-context learning (ICL) and a fine-tuned version (FT) for better retrieval and generation.

