

Types of RAG



Naive RAG — The Basic Approach

- Naive RAG follows the simplest RAG pipeline: retrieve relevant information and give it to the LLM for generation.
- Documents are typically chunked, embedded, stored, and retrieved using similarity search.
- The retrieved chunks are added to the prompt as context for the LLM.
- It is simple to build and works well for straightforward question-answering tasks.
- Its main weakness is that poor retrieval can directly lead to poor answers.



Advanced RAG — Improving the Retrieval Pipeline

- Advanced RAG improves the basic pipeline with techniques before and after retrieval.
- Query rewriting, query expansion, better chunking, metadata filtering, and improved embeddings can increase retrieval quality.
- Retrieved documents can also be re-ranked to place the most relevant information first.
- Context compression can remove unnecessary information before it reaches the LLM.
- The goal is to make the same basic RAG pipeline more accurate, efficient, and reliable.



Modular RAG — Building RAG from Components

- Modular RAG breaks the RAG pipeline into independent components that can be combined and replaced.
- Different modules can handle retrieval, query rewriting, routing, re-ranking, generation, or memory.
- The system can use different workflows depending on the type of question.
- This makes RAG systems easier to customize for different business and technical requirements.
- Modular RAG moves RAG from a fixed pipeline toward a more flexible architecture.



Graph RAG — Connecting Knowledge Through Relationships

- Graph RAG represents information using entities and relationships rather than relying only on isolated text chunks.
- A knowledge graph can connect people, organizations, products, events, and concepts.
- Retrieval can then follow relationships between entities to discover connected information.
- This is particularly useful for questions requiring connections across multiple pieces of information.
- Graph-based retrieval can improve performance on complex, multi-hop questions.



Multimodal RAG — Beyond Text

- Multimodal RAG retrieves information from different data types such as text, images, tables, charts, audio, and video.
- Instead of treating a document as text only, the system can retrieve the most relevant visual or structured information.
- For example, a financial assistant could retrieve both the text explanation and the relevant chart from a report.
- This makes RAG useful for documents where important information is not represented purely as text.
- Multimodal RAG extends retrieval from a text-based problem to a multi-format knowledge problem.



Corrective RAG — Checking and Fixing Retrieval

- Corrective RAG introduces mechanisms to evaluate whether the retrieved information is actually useful.
- If retrieval quality is poor, the system can take corrective actions instead of blindly generating an answer.
- It may refine the query, retrieve additional information, or use another knowledge source.
- This reduces the risk of generating an answer from irrelevant or insufficient context.
- The key idea is: retrieve first, evaluate the evidence, and correct the retrieval when necessary.



Self-RAG — The Model Reflects on Retrieval

- Self-RAG allows the model to determine when retrieval is needed and reflect on the retrieved information.
- Instead of retrieving information for every query, the system can decide whether external knowledge is necessary.
- It can evaluate retrieved passages and assess aspects of its own generated response.
- This creates a more adaptive retrieval-and-generation process.
- The goal is to improve factuality and relevance while avoiding unnecessary retrieval.



Agentic RAG — RAG Meets AI Agents

- Agentic RAG combines retrieval with agent-style reasoning, planning, and tool use.
- Instead of following one fixed retrieval pipeline, the agent can decide what information it needs and where to find it.
- It can perform multiple retrieval steps, use different data sources, call tools, and refine its search based on intermediate results.
- This makes it suitable for complex tasks that require multi-step reasoning and information gathering.
- Agentic RAG represents a shift from "retrieve once and generate" toward "reason, retrieve, evaluate, and act."





Shoaib Akthar

Data Scientist

I share things on Data & AI
follow if you wanna see more!