# RAG & Flow

RAG and Flow are two distinct concepts that are often used together in building AI applications. RAG (Retrieval-Augmented Generation) is an AI technique used to improve the accuracy of large language models (LLMs) by providing them with relevant, up-to-date data from external sources.

A flow is the sequence of steps and logic that defines a process, such as how data moves through a system or how a prompt is processed. In the context of AI, RAG is a capability you build, and the flow is the process that utilizes that capability.

What is RAG?

Large Language Models (LLMs) are trained on massive, but finite and static, datasets. This can lead to inaccurate or outdated information in their responses, a problem known as "hallucination". RAG addresses this limitation by inserting a data retrieval step into the generation process.

The RAG process works as follows:

1. Retrieve: When a user enters a query, the system first retrieves relevant documents, articles, or other data from a private or external knowledge base.
2. Augment: The retrieved information is then used to create a new, more informed prompt for the LLM. The LLM receives both the original query and the newly retrieved context.
3. Generate: The LLM uses this augmented prompt to generate a more accurate, relevant, and context-aware answer.

What is a flow?

In software and AI, a flow, also known as a workflow or pipeline, is a blueprint for how a task is executed. It is a structured process that defines the sequence of operations needed to achieve a specific outcome.

In AI applications, a flow can define:

* A data pipeline, which outlines how data is ingested, processed, and stored for use by a model.
* An agent's decision-making logic, which specifies the steps an AI agent should take, including which tools to call and what path to follow.
* The user interaction path, which defines the sequence of prompts, model calls, and tool usage to answer a user's request.

How RAG and flow work together

RAG is typically implemented as one or more steps within a larger AI workflow or "flow." The flow orchestrates the entire process, including the retrieval and generation steps that make up RAG.

For example, to build a chatbot that answers questions based on an internal company knowledge base, you would design a flow with the following steps:

1. Receive user input: The flow starts when a user sends a message.
2. Trigger RAG: The user's message triggers the RAG component of the flow.
3. Search knowledge base: The RAG component retrieves relevant documents from the company's internal knowledge base.
4. Augment prompt: The flow combines the user's message with the retrieved documents to create a new, enhanced prompt.
5. Call LLM: The augmented prompt is sent to the LLM.
6. Return response: The LLM's generated answer is delivered back to the user.

Developers can use interactive editors like Prompt Flow in Azure Machine Learning to build, debug, and test these RAG-based AI applications.

# Vector DB

# A vector database stores, indexes, and queries data as high-dimensional numerical vectors, also known as embeddings. Unlike traditional databases that rely on exact matches, vector databases use similarity search to find data points that are semantically or visually similar, making them ideal for AI applications like natural language processing, computer vision, and recommender systems. They excel at handling unstructured data like text, images, and audio by representing it as vectors where similar concepts are located closer together in a multi-dimensional space.

Key features and function

* Stores vectors: They are designed to handle high-dimensional vectors, which are arrays of numbers that represent data.
* Similarity search: They use algorithms like cosine similarity or Euclidean distance to find the most similar vectors to a query vector, enabling semantic search rather than just keyword matching.
* Handles unstructured data: They are crucial for managing unstructured data like images, text, and audio by first converting them into vector embeddings.
* AI and machine learning: They are a core component of modern AI systems, providing a way to store and retrieve data for applications like large language models, recommendation engines, and generative AI.
* Specialized indexing: To perform fast and efficient searches across large datasets, vector databases use specialized indexing techniques, such as Approximate Nearest Neighbor (ANN) algorithms.