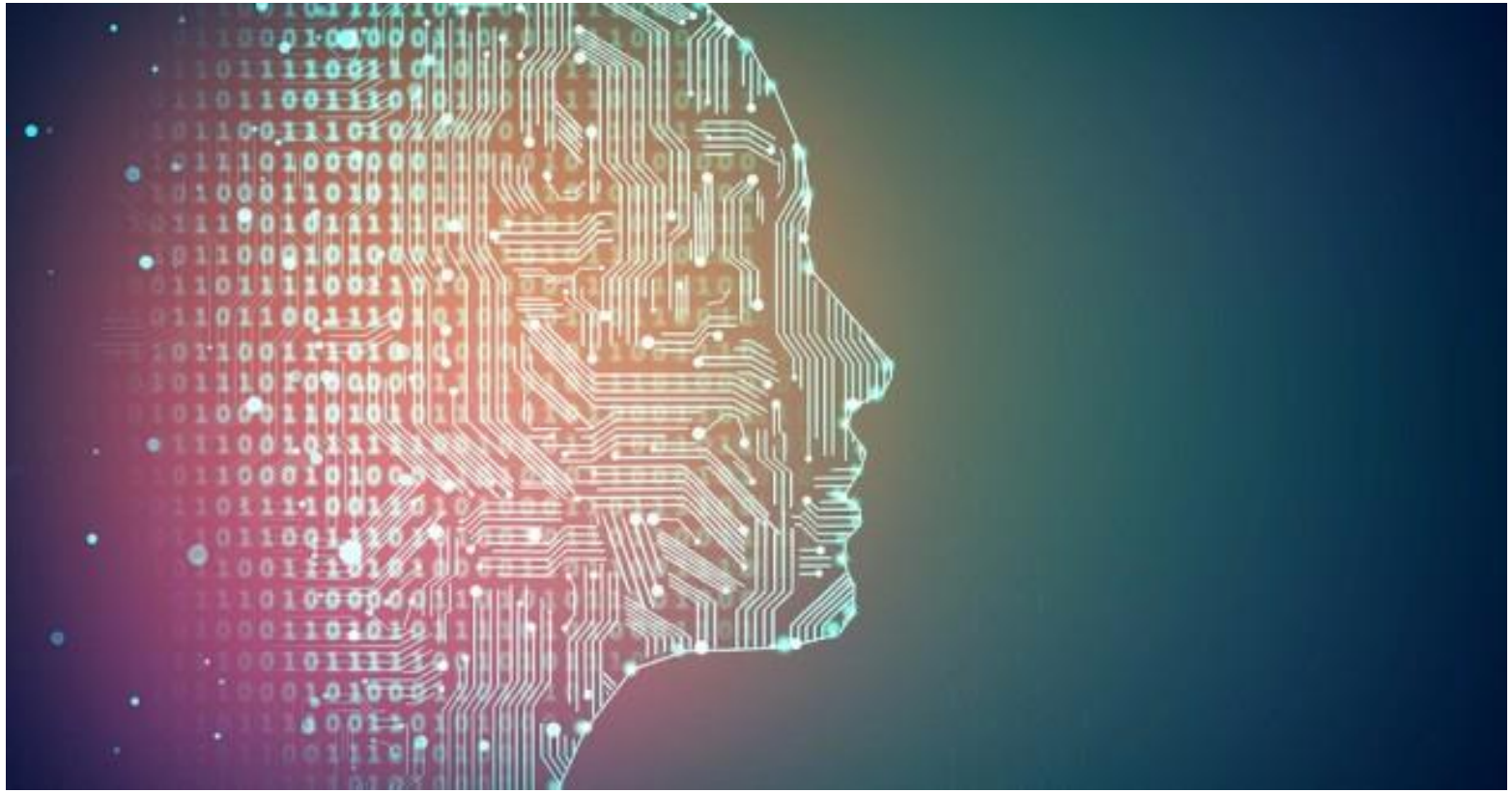
Retrieval-Augmented Generation (RAG)

Thuan L Nguyen, PhD

2: Generative AI: LLM: Retrieval-Augmented Generation (RAG)



AI Deep learning (Source: mindovermachines.com)

3: Generative AI: LLM: Retrieval-Augmented Generation (RAG)

1. Generative AI: Retrieval-Augmented Generation (RAG): Overview

2. Generative AI: RAG: Why Do We Need It?

3. Generative AI: RAG: A Bit of History

4. Generative AI: RAG: Core Concepts & RAG Process

5. Generative AI: RAG: Main Parameters & Properties

6. Generative AI: RAG: Sequence Models vs Token Models

7. Generative AI: RAG: PROs & CONs

4: Generative AI: LLM: Retrieval-Augmented Generation (RAG)

Artificial Intelligence: Generative AI

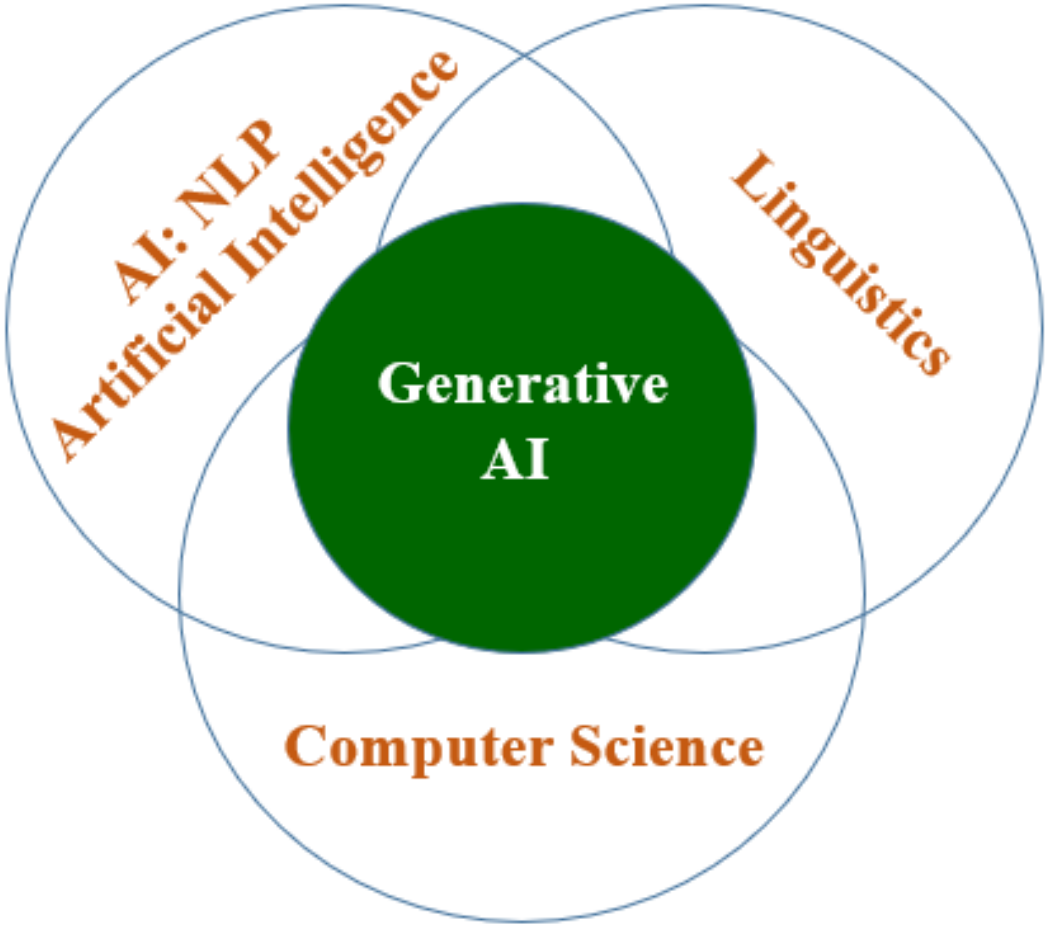
What is It?

Generative AI: A category of artificial intelligence focused on using AI deep learning models to generate new contents, including text, images, audio, video, and more. The contents are novel but look realistic and may be indistinguishable from human-created ones.

5: Generative AI: LLM: Retrieval-Augmented Generation (RAG)

Artificial Intelligence: Generative AI: LLM

What is It?



Generative AI is based on the NLP technologies such as Natural Language Understanding (NLU) and Conversational AI (AI Dialogues) - Those among the most challenging tasks AI needs to solve.

6: Generative AI: LLM: Retrieval-Augmented Generation (RAG)

Artificial Intelligence: Generative AI: LLMs

Large Language Models

Large Language Models (LLMs) are revolutionary AI **Deep Learning neural networks** that excel in natural language understanding (NLU) and content generation.

- “LARGE" in LLMs refers to the vast scale of data and parameters used to train them, allowing LLMs to develop a comprehensive understanding of language.
- Being particularly transformer-based models trained on massive text datasets using deep learning techniques,
- Able to learn complex language patterns, capture nuances like grammar and tone, and generate coherent and contextually relevant text

7: Generative AI: LLM: Retrieval-Augmented Generation (RAG)

Artificial Intelligence: Generative AI: LLMs

Large Language Models

Large Language Models (LLMs) are revolutionary AI **Deep Learning neural networks** that excel in natural language understanding (NLU) and content generation.

- “LARGE" in LLMs refers to the vast scale of data and parameters used to train them, allowing LLMs to develop a comprehensive understanding of language.
- Being particularly transformer-based models trained on massive text datasets using deep learning techniques,
- Able to learn complex language patterns, capture nuances like grammar and tone, and generate coherent and contextually relevant text

8: Generative AI: LLM: Retrieval-Augmented Generation (RAG)

RAG: Overview

- Retrieval-Augmented Generation (RAG) is a technique that **augments** traditional language model generation with the ability to retrieve and integrate information from a knowledge base (like Wikipedia or a company-specific database). This technique improves the **factual accuracy** and **consistency** of the generated text.
- **RAG** cleverly integrates information retrieval with powerful generative language models to enhance **reliability** and **accuracy**.
- The **key idea** behind RAG is to **augment** the **text generation process** with relevant facts and information retrieved from a **knowledge base**. This knowledge base could be a structured database, Wikipedia, or a collection of documents.

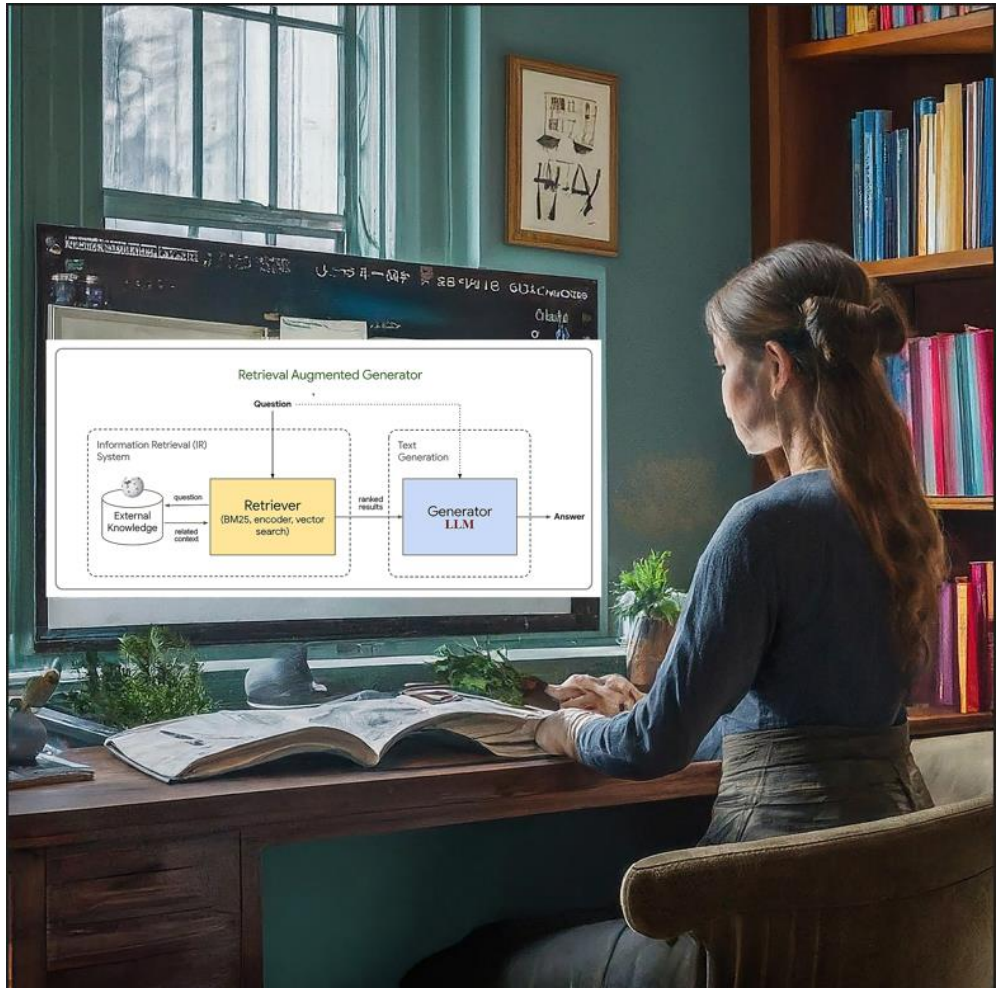
***Source**: Thuan L Nguyen – AI Generated images using Google-DeepMind Imagen 2.0*

9: Generative AI: LLM: Retrieval-Augmented Generation (RAG)

RAG: Why Do We Need It?

- **Large Language Models (LLMs)** have achieved extraordinary strides in natural language processing (**NLP**).
  - They can generate remarkably human-like text, translate between languages, and answer questions with seemingly deep understanding.

  - Large Language Models (LLMs) have improved quantitatively and qualitatively. They can learn new abilities without being directly trained on them.



***Source**: Thuan L Nguyen – AI Generated images using Google-DeepMind Imagen 2.0*

10: Generative AI: LLM: Retrieval-Augmented Generation (RAG)

***Source**: Thuan L Nguyen – AI Generated images using Google-DeepMind Imagen 2.0*

RAG: Overview

- Large Language Models (LLMs) have achieved extraordinary strides in natural language processing.
- However, **LLMs have limitations**:
  - **Factual Inconsistency**: They sometimes provide incorrect or "hallucinated" information, as their knowledge is largely derived from the patterns within their training data.
  - **Static Knowledge**: LLMs struggle to adapt to rapidly changing information, as they don't inherently connect to real-world knowledge sources.
    - LLM are unaware of events **after training** and it is almost impossible to trace the sources to their responses.

Retrieval-Augmented Generation (RAG) was introduced to **address these issues**.

11: Generative AI: LLM: Retrieval-Augmented Generation (RAG)

***Source**: Thuan L Nguyen – AI Generated images using Google-DeepMind Imagen 2.0*

RAG: A Bit of History

- The concept was first proposed in the 2020 paper "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks" by researchers at Facebook AI Research (FAIR).
    - RAG cleverly integrates information retrieval with powerful generative language models to enhance reliability and accuracy.

12: Generative AI: LLM: Retrieval-Augmented Generation (RAG)

RAG: Core Concepts

Non-Parametric vs. Parametric Knowledge:

- **Parametric Knowledge**:
  - **LLMs** primarily rely on parametric knowledge – the patterns and relationships stored within their vast number of parameters during training.
- **Non-Parametric Knowledge**:
  - **RAG** introduces the concept of **non-parametric** knowledge, which lives externally in the knowledge base and is retrieved on demand.

Tokenization and Marginalization :

- **Tokenization** and **Marginalization**:
  - RAG models often use a special token (like **"[SEP]**") to separate the original query from the retrieved documents.
  - The **probability** of generating a response is calculated by marginalizing (summing) over **all** possible positions of these separator tokens – a **clever way** to combine knowledge seamlessly into the process.

13: Generative AI: LLM: Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation Process

Query Formulation: The user provides a query or prompt (e.g., "What is the capital of France?").

Document Retrieval: A retrieval component searches a knowledge base (frequently Wikipedia is used) and identifies the most relevant documents or passages based on the query. This retrieval often uses techniques like dense vector search for semantic similarity.

Document Encoder: The retrieved documents are encoded into a format the language model can understand.

***Source**: Thuan L Nguyen – AI Generated images using Google-DeepMind Imagen 2.0*

14: Generative AI: LLM: Retrieval-Augmented Generation (RAG)

***Source**: Thuan L Nguyen – AI Generated images using Google-DeepMind Imagen 2.0*

Retrieval-Augmented Generation Process

Answer Generation: The retrieved documents, along with the original query, are fed into a generative language model. This model is responsible for producing the final answer.

Output: The language model generates the final answer, leveraging both its own pre-trained knowledge and the specific retrieved knowledge from the external source.

15: Generative AI: LLM: Retrieval-Augmented Generation (RAG)

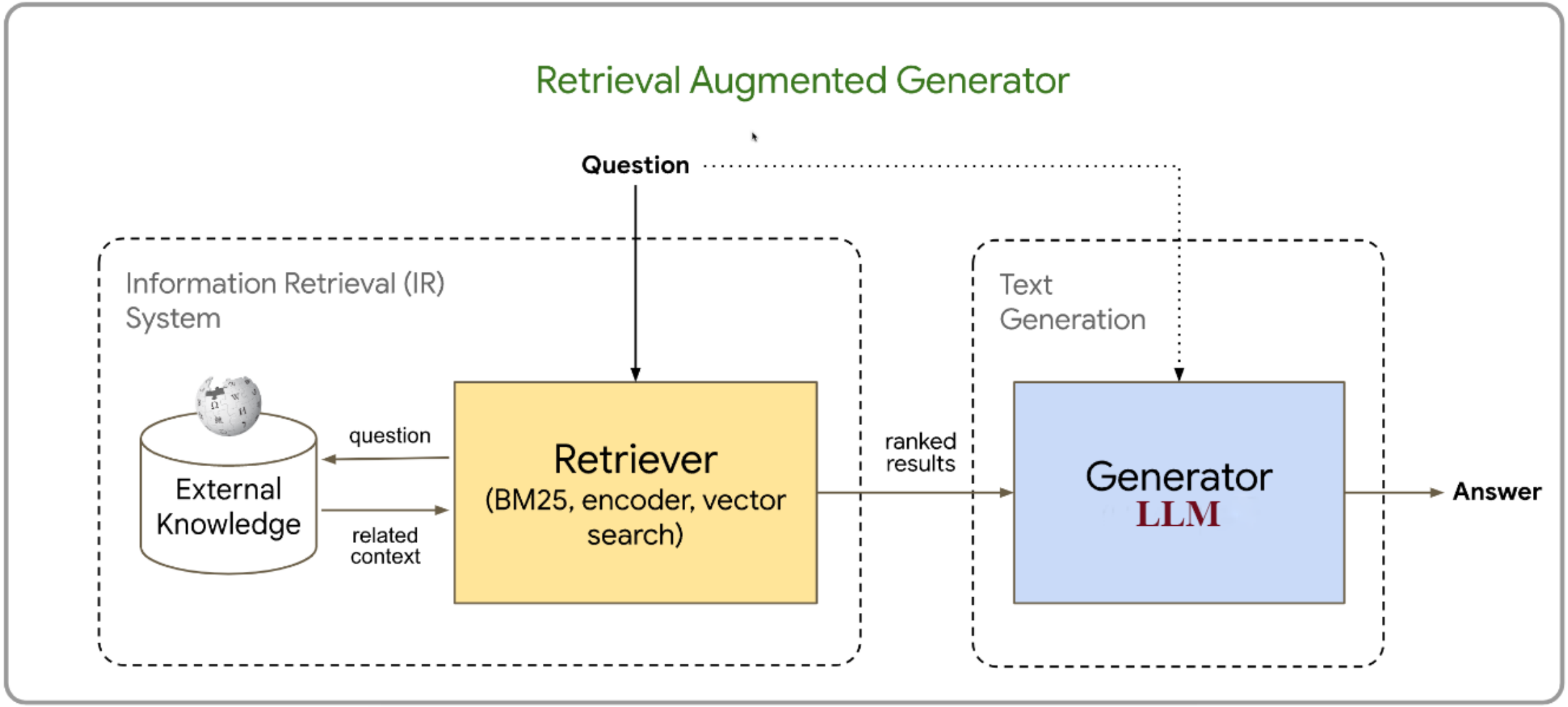
***Source**: Thuan L Nguyen – AI Generated images using Google-DeepMind Imagen 2.0*

Retrieval-Augmented Generation: Actors

RAG involves **two core components** (actors):

- **Retriever**: Responsible for finding relevant documents from the knowledge base.
- **Generator**: Generates text conditioned on both the original input and retrieved documents.

16: Generative AI: LLM: Retrieval-Augmented Generation (RAG)



***Source**: Original Image by Alphabet/Google: Updated by Thuan L Nguyen*

17: Generative AI: LLM: Retrieval-Augmented Generation (RAG)

RAG: Main Parameters and Properties

- **Knowledge Base**: RAG can use various structured (e.g., databases) or unstructured (e.g., Wikipedia, text documents) knowledge bases. The choice depends heavily on the nature of the task.

- **Retrieval Model**: The performance of RAG is often dependent on the quality of the retrieval model. Common techniques include traditional keyword-based search, dense vector representations (like BM25), and neural retrievers.

- **Generative Model**: RAG is flexible and can work with different generative language models (e.g., LLMs like Gemini, GPT, Claude, and other transformer-based models like BART or T5).

- **Sequence vs. Token Models**: RAG has two primary variations:
  - RAG-Sequence: Generates the entire output sequence in one go.
  - RAG-Token: Generates the output one token at a time, allowing the retrieval model to provide new context after each generated token.