Lecture 29: What is RAG?

Introduction

- **RAG** stands for **Retrieval Augmented Generation**.
- It is used in **Generative AI** to improve responses by adding context.
- RAG combines **LLM (Large Language Models)** with **external knowledge sources**.

Problems with Only Using LLMs

- **Outdated Knowledge**
  - Models are trained on fixed data (often years old).
  - They cannot access the latest information.

- **Hallucination**
  - If the model does not know an answer, it may generate **incorrect or imaginary responses**.
- **Custom Data Limitation**
  - By default, models do not know your **personal documents or product data**.
  - **Example:** A chatbot cannot answer based on a company's product details without extra context.

Role of RAG:

- RAG helps solve these issues by **retrieving relevant context** from your own data and **augmenting the prompt** before sending it to the model.
- This ensures:
  - More **accurate** responses.
  - **Reduced hallucinations**.
  - Ability to answer from **specific documents or datasets**.

How RAG Works

- **User Prompt**
  - A user sends a query (e.g., "Need details regarding the art kit for kids").
- **Retrieve Context**
  - Documents are **chunked** and converted into **embeddings** (numbers).
  - These embeddings are stored in a **Vector Database**.
  - Relevant chunks are retrieved based on the query.
- **Augment Prompt**
  - The retrieved chunks are added as **context** to the original query.
- **Generate Response**

    - The final prompt (user query + context) is sent to the LLM.
    - The model gives a **specific and accurate** response.

Example Scenario:

- Without RAG:
    - **Query:** “Need details about art kit for kids”
    - **Response**: General suggestions, not from your product file.
- With RAG:
    - **Query** is combined with product file data.
    - **Response**: “Art supplies kit for kids includes …” (from your own product details).

Benefits of RAG:

- Keeps answers **relevant and updated**.
- Reduces chances of **hallucination**.
- Makes chatbots **domain-specific** (works with your data).
- Useful for building **knowledge-driven applications** (e.g., product Q&A, support bots).

Summary:

- RAG = **Prompt + Retrieved Context → Better Generation**.
- It bridges the gap between **static LLM knowledge** and **dynamic, domain-specific information**.