RAG Whiteboard: Enhancing LLM Responses

Introduction to RAG

- **RAG (Retrieval Augmented Generation)** is a technique to improve LLM responses by incorporating retrieved information.
- It addresses the challenge of making an LLM aware of your own content.
- **How it works:** Inlines relevant content from a knowledge store based on the user's prompt, then sends this augmented prompt to the LLM.

RAG vs. Fine-tuning/Retraining

- **Fine-tuning/Retraining:**
    - Changes a model's functionality through training.
    - Computationally expensive, but done only once.
    - Generally harder than RAG.
- **RAG:**
    - Adds context to an LLM call **without changing the model itself.**
    - Requires additional setup outside of the LLM, but fairly easy to do.
    - Performed **every time** a call is made to the LLM.
    - **Recommendation:** Start with RAG.

Core Concepts: Vectors and Embeddings

Vectors/Tensors/Matrices

- In mathematics: Distinct but related concepts.
- In programming: Can be thought of as **arrays**.
- **Key:** You can perform mathematical operations on these arrays.

Embeddings (Vector Embeddings)

- The process of converting content (e.g., web pages, images) into **vector points**.
- These vector points are "embedded" into a **vector space**.
- Allows for **vector operations** on these points.
- Enables operation **conceptually** rather than literally.

Search Mechanisms

Keyword Search

- **Simple terms:** Break content into **tokens** and store them in an index.
    - Tokens can be words, parts of words, depending on the algorithm.

    - Example: "the happy dog is walking" -> tokens: "happy", "dog", "walk".
- **Searching:** Look for **literal matches** with these tokens.
    - Example: "walking dog" -> tokens: "walk", "dog". Matches "happy dog is walking".
- **Algorithm Example:** BM25.
- **Drawback:** Only looks for literal matches, not conceptual.
    - Example: "running puppy" (tokens: "run", "puppy") won't match "happy dog walk" (from "the happy dog is walking") because there's no literal match. Adding synonyms doesn't scale well.

Vector Search (Semantic Search)

- Leverages **vector embeddings plus math** for **conceptual operation**.
- **How it works (simplified example):**
    - Imagine a vector space (e.g., Y-axis: Age (0-100 years), X-axis: Color (visible light spectrum)).
    - Plot points for content (e.g., Green 1961 Jaguar, Purple 2004 Lamborghini).
    - When searching (e.g., "yellow 1957 cars"), the query is also converted to a vector point.
    - Perform a **nearest neighbor lookup** to find conceptually similar content (e.g., the Green 1961 Jaguar, even if not literally yellow or 1957).
    - This allows understanding the *meaning* of the query, not just keywords.

High-Level RAG Setup Overview

1. **Load Phase:**
    - **Load and Process Content:** Ingest your data.
    - **Chunking:** Break content into smaller chunks (due to LLM context window limitations).
    - **Calculate Embeddings:** Generate vector embeddings for all content chunks.
    - **Store:** Store embeddings in a **Vector Store** along with any necessary metadata.
2. **Running the Application (Query Phase):**
    - **Embed User Prompt:** Calculate embeddings for the user's query/prompt.
    - **Retrieve Content:** Use **Vector Search** to find relevant content in the vector store.
    - **Augment Prompt:** Inline the retrieved relevant content into the LLM prompt call.
    - **Generate Response:** Send the augmented prompt to the LLM to generate a better, context-aware response.