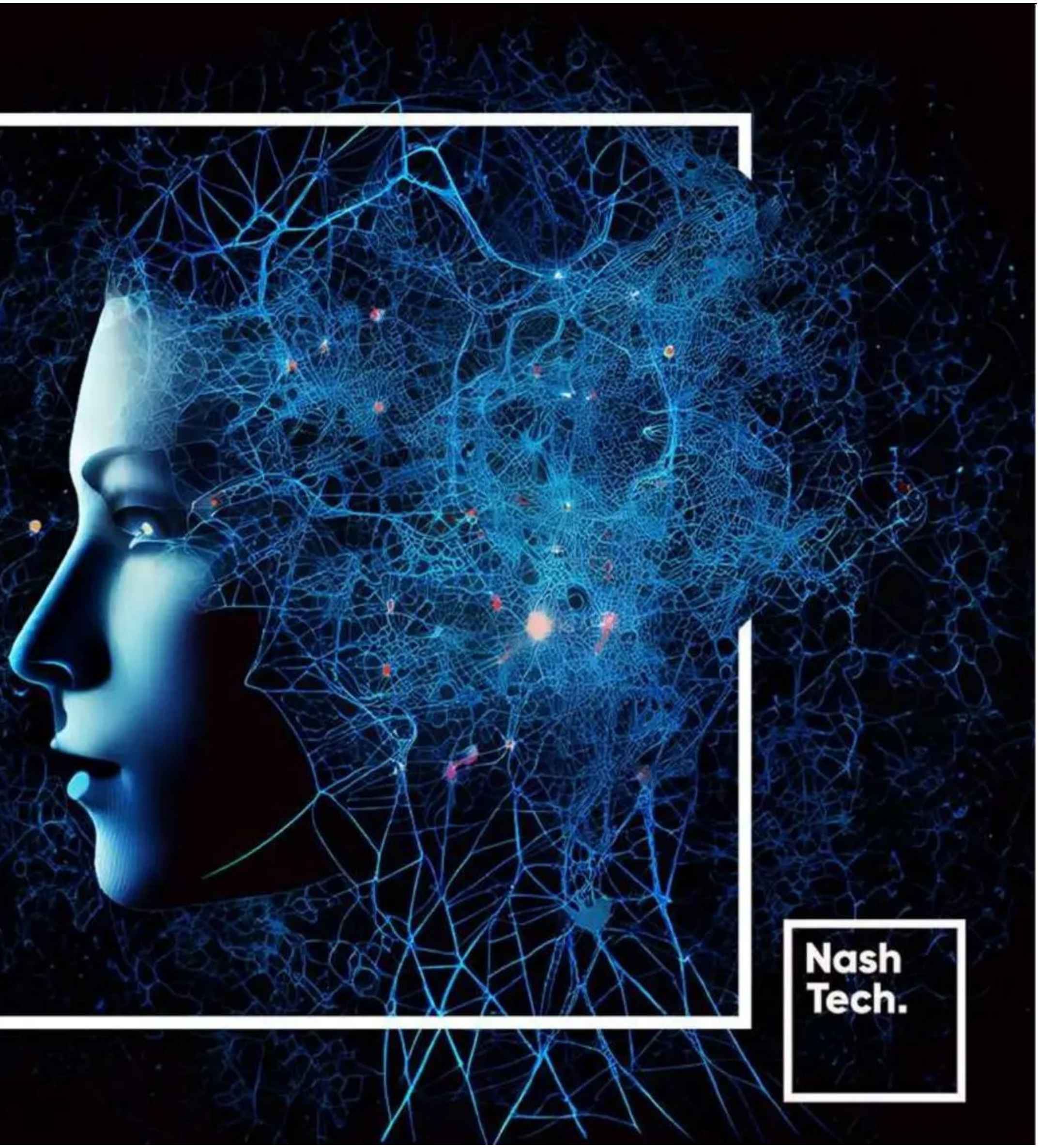
Introduction to RAG and It's Application

Presented By: Aayush Srivastava



Nash Tech.

KnolX Etiquettes

Lack of etiquette and manners is a huge turn off.

- **Punctuality**

  Join the session 5 minutes prior to the session start time. We start on time and conclude on time!

- **Feedback**

  Make sure to submit a constructive feedback for all sessions as it is very helpful for the presenter.

- **Silent Mode**

  Keep your mobile devices in silent mode, feel free to move out of session in case you need to attend an urgent call.

- **Avoid Disturbance**

  Avoid unwanted chit chat during the session.

Agenda

1. **Introduction**
   - What is LLM
   - What is RAG?
2. **LLM And It's Limitation**
   - Why RAG is important?
3. **RAG Architecture**
4. **How Does RAG Work**
5. **RAG Vs Fine-Tuning**
6. **Benefits Of RAG**
7. **Applications**
8. **Demo**

Introduction

What is LLM

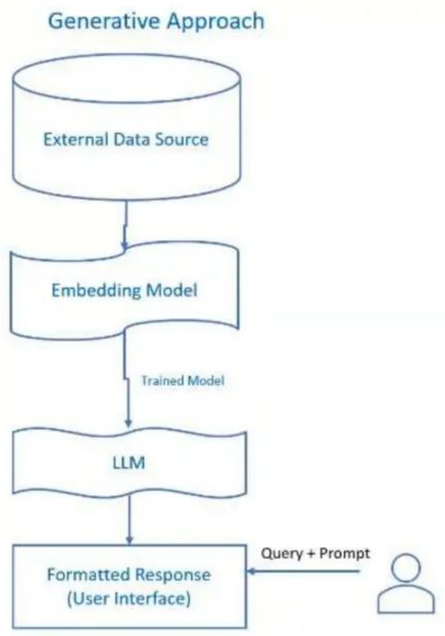
- A large language model (LLM) is a type of artificial intelligence program that can recognize and generate text, among other tasks.
- LLM are very large models that are pre-trained on vast amounts of data.
- Built on transformer architecture is a set of neural network that consist of an encoder and a decoder with self-attention capabilities.
- It can perform completely different tasks such as answering questions, summarizing documents, translating languages and completing sentences.

Open AI's GPT-3 model has 175 billion parameters. Also it can take inputs up to 100K tokens in each prompt

What is LLM

- In simpler terms, an LLM is a computer program that has been fed enough examples to be able to recognize and interpret human language or other types of complex data.

- Quality of the samples impacts how well LLMs will learn natural language, so an LLM's programmers may use a more curated data set.



LLM's And It's Limitations

- **Not Updated to the latest information:** Generative AI uses large language models to generate texts and these models have information only to date they are trained. If data is requested beyond that date, accuracy/output may be compromised.

- **Hallucinations:** Hallucinations refer to the output which is factually incorrect or nonsensical. However, the output looks coherent and grammatically correct. This information could be misleading and could have a major impact on business decision-making.

- **Domain-specific most accurate information:** LLM's output lacks accurate information many times when specificity is more important than generalized output. For instance, organizational HR policies tailored to specific employees may not be accurately addressed by LLM-based AI due to its tendency towards generic responses.

- **Source Citations:** In Generative AI responses, we don’t know what source it is referring to generate a particular response. So citations become difficult and sometimes it is not ethically correct to not cite the source of information and give due credit.

- **Updates take Long training time:** information is changing very frequently and if you think to re-train those models with new information it requires huge resources and long training time which is a computationally intensive task.

- Presenting false information when it does not have the answer.

What is RAG?

- RAG stands for Retrieval-Augmented Generation
- It's an advanced technique used in Large Language Models (LLMs)
- RAG combines retrieval and generation processes to enhance the capabilities of LLMs
- In RAG, the model retrieves relevant information from a knowledge base or external sources
- This retrieved information is then used in conjunction with the model's internal knowledge to generate coherent and contextually relevant responses
- RAG enables LLMs to produce higher-quality and more context-aware outputs compared to traditional generation methods
- Essentially, RAG empowers LLMs to leverage external knowledge for improved performance in various natural language processing tasks

Retrieval Augmented Generation (RAG) is an advanced artificial intelligence (AI) technique that combines information retrieval with text generation, allowing AI models to retrieve relevant information from a knowledge source and incorporate it into generated text.

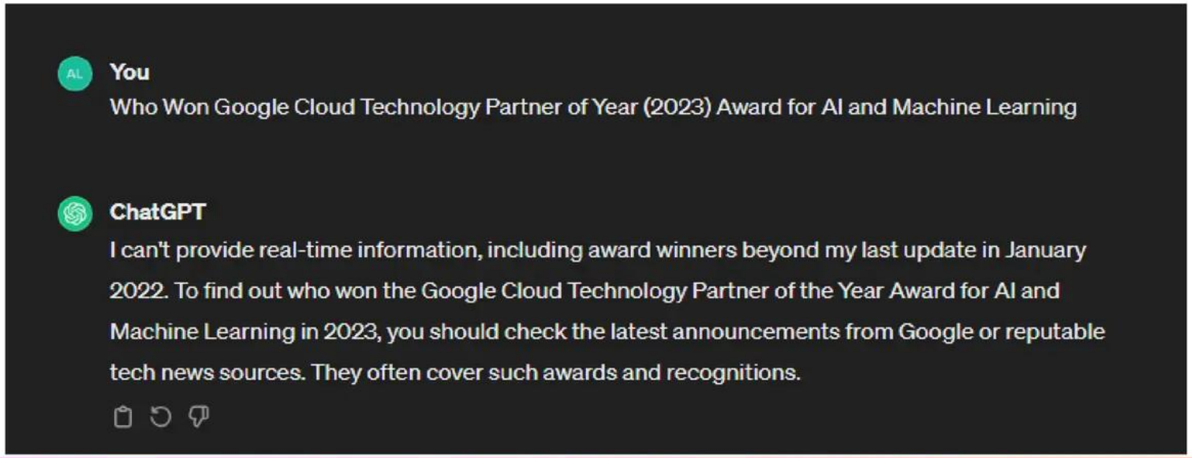
Why is Retrieval-Augmented Generation important

- You can think of the LLM as an over-enthusiastic new employee who refuses to stay informed with current events but will always answer every question with absolute confidence.

- Unfortunately, such an attitude can negatively impact user trust and is not something you want your chatbots to emulate!

- RAG is one approach to solving some of these challenges. It redirects the LLM to retrieve relevant information from authoritative, pre-determined knowledge sources.

- Organizations have greater control over the generated text output, and users gain insights into how the LLM generates the response.



AL **You**

Who Won Google Cloud Technology Partner of Year (2023) Award for AI and Machine Learning

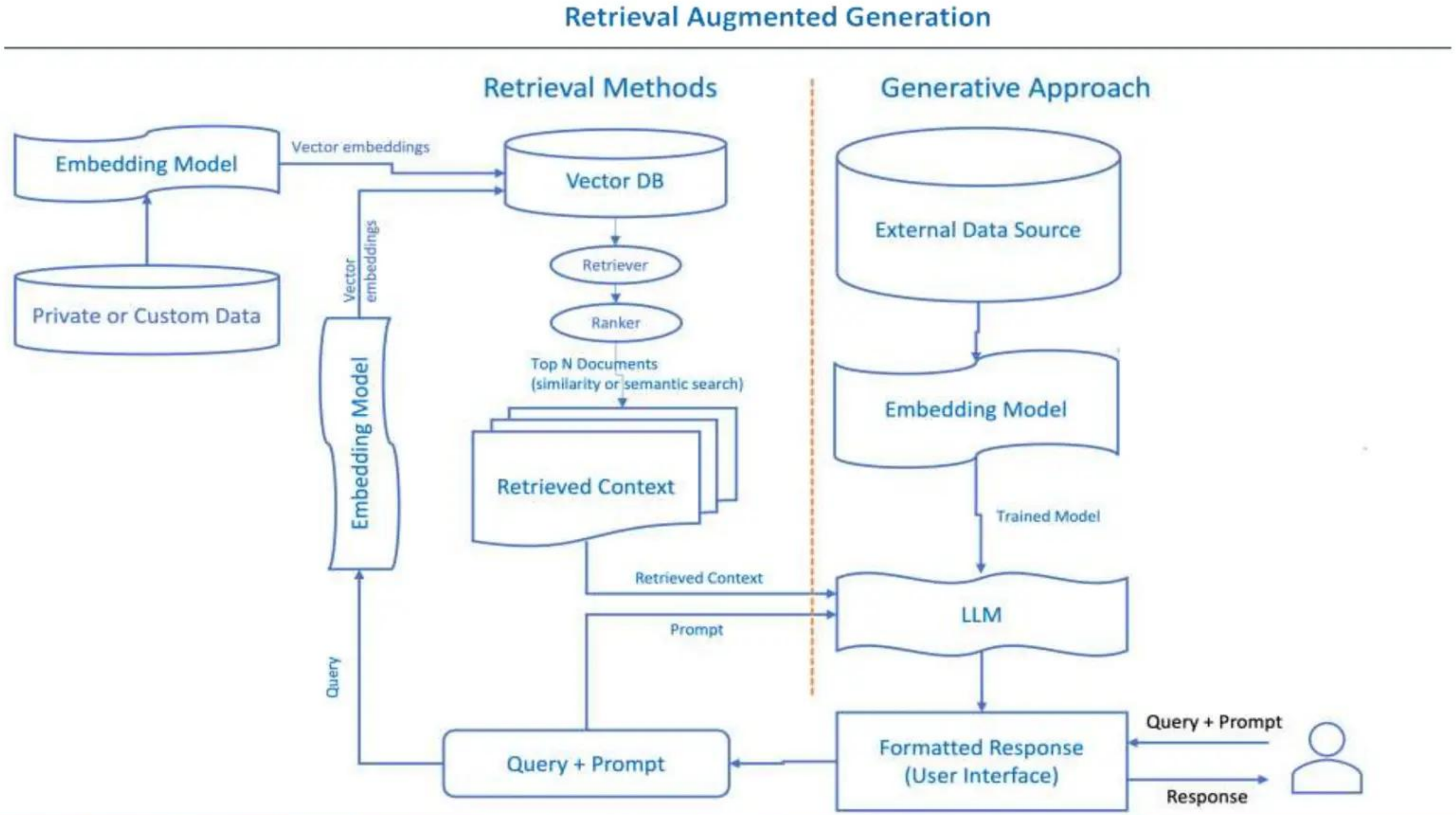
ChatGPT

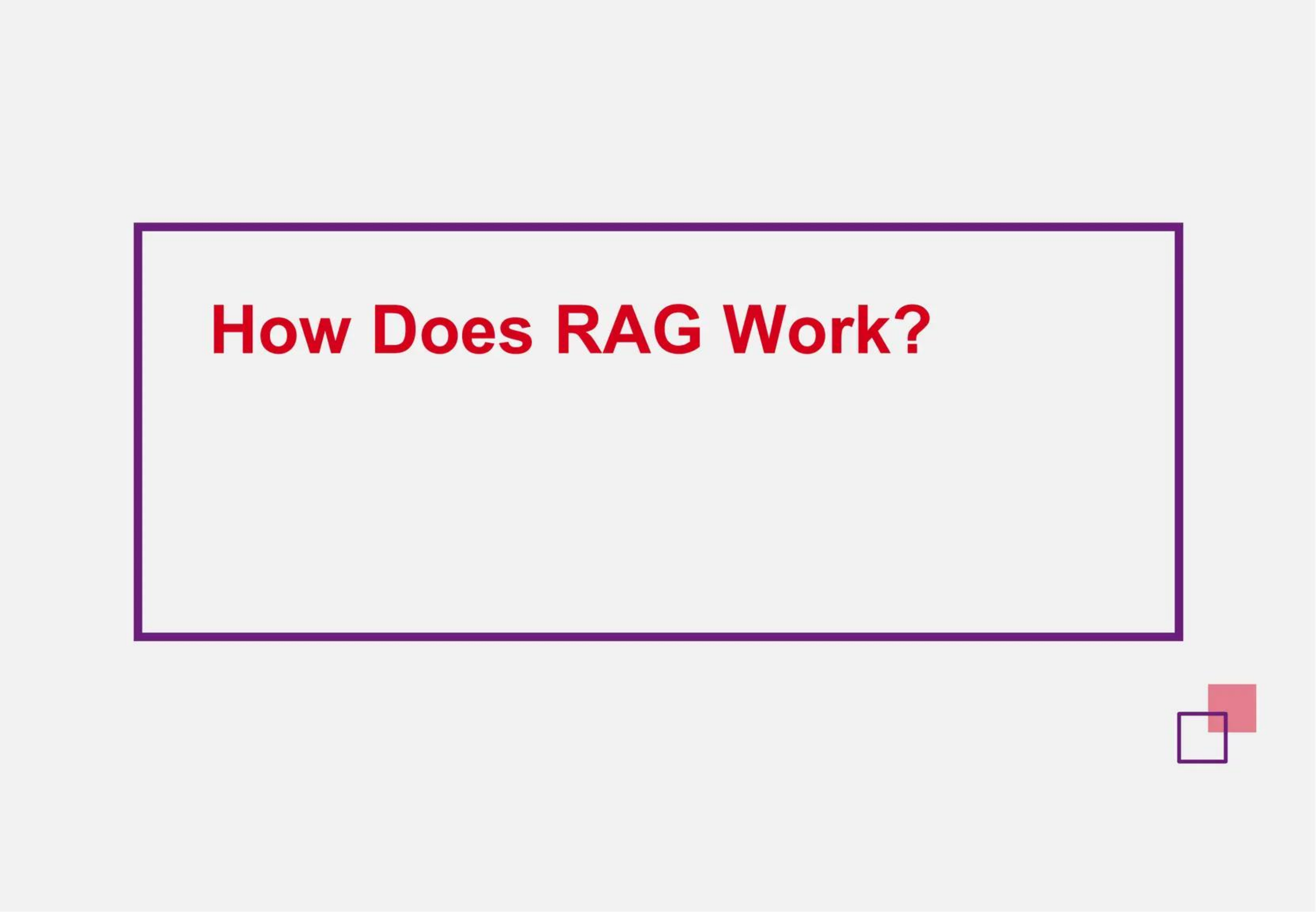
I can't provide real-time information, including award winners beyond my last update in January 2022. To find out who won the Google Cloud Technology Partner of the Year Award for AI and Machine Learning in 2023, you should check the latest announcements from Google or reputable tech news sources. They often cover such awards and recognitions.

RAG Architecture

Generalized RAG Approach

Let's delve into RAG's framework to understand how it mitigates these challenges.





How Does RAG Work?

Overview

- Retrieval Augmented Generation (RAG) can be likened to a detective and storyteller duo. Imagine you are trying to solve a complex mystery. The detective's role is to gather clues, evidence, and historical records related to the case.
- Once the detective has compiled this information, the storyteller designs a compelling narrative that weaves together the facts and presents a coherent story. In the context of AI, RAG operates similarly.
- The **Retriever Component** acts as the detective, scouring databases, documents, and knowledge sources for relevant information and evidence. It compiles a comprehensive set of facts and data points.
- The **Generator Component** assumes the role of the storyteller. Taking the collected information and transforming it into a coherent and engaging narrative, presenting a clear and detailed account of the mystery, much like a detective novel author.

This analogy illustrates how RAG combines the investigative power of retrieval with the creative skills of text generation to produce informative and engaging content, just as our detective and storyteller work together to unravel and present a compelling mystery.

RAG Components

- RAG is an AI framework that allows a generative AI model to access external information not included in its training data or model parameters to enhance its responses to prompts.
- RAG seeks to combine the strengths of both retrieval-based and generative methods.
- It typically involves using a retriever component to fetch relevant passages or documents from a large corpus of knowledge.
- The retrieved information is then used to augment the generative model's understanding and improve the quality of generated responses.

RAG Components

- **Retriever**
- **Ranker**
- **Generator**
- **External Data**

What is a Prompt.

A prompt is the input provided by the user to generate a response. It could be a question, a statement, or any text that serves as the starting point for the model to generate a relevant and coherent continuation

RAG Components

Let's understand each component in detail

External data

The new data outside of the LLM's original training data set is called external data. It can come from multiple data sources, such as a APIs, databases, or document repositories. The data may exist in various formats like files, database records, or long-form text.

Vector embeddings

- ML models cannot interpret information intelligibly in their raw format and require numerical data as input. They use neural network embeddings to convert real-word information into numerical representations called vectors.

- Vectors are numerical values that represent information in a multi-dimensional space
- Embedding vectors encode non-numerical data into a series of values that ML models can understand and relate. Example:

  The Conference (Horror, 2023, Movie) → *The Conference (1.2, 2023, 20.0)*

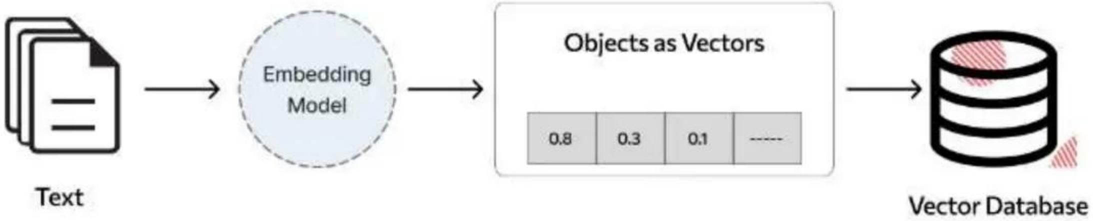
  Tales from the Crypt (Horror, 1989, TV Show, Season 7) → *Tales from the Crypt (1.2, 1989, 36.7)*

- The first number in the vector corresponds to a specific genre. An ML model would find that *The Conference* and *Tales from the Crypt* share the same genre. Likewise, the model will find more relationships

RAG Components

Vector DB

- Vector DB is a database that stores embeddings of words, phrases, or documents along with their corresponding identifiers.
- It allows for fast and scalable retrieval of similar items based on their vector representations.
- Vector DBs enable efficient retrieval of relevant information during the retrieval phase of RAG, improving the contextual relevance and quality of generated responses.
- **Data Chunking**: Before the retrieval model can search through the data, it's typically divided into manageable "chunks" or segments.
- Vector DB Examples : Chroma, Pinecone, Weaviate, Elasticsearch

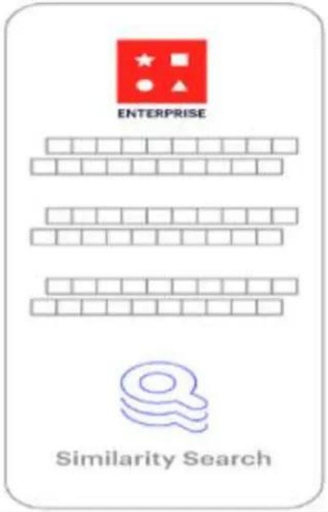


RAG Components

RAG Retriever

- The next step is to perform a relevancy search. The user query is converted to a vector representation and matched with the vector databases
- The retriever component is responsible for efficiently identifying and extracting relevant information from a vast amount of data.

For example, consider a smart chatbot for human resource questions for an organization. If an employee searches, *"How much annual leave do I have?"* the system will retrieve annual leave policy documents alongside the individual employee's past leave record. These specific documents will be returned because they are highly-relevant to what the employee has input. The relevancy was calculated and established using mathematical vector calculations and representations



RAG Components

RAG Ranker

- The RAG ranker component refines the retrieved information by assessing its relevance and importance. It assigns scores or ranks to the retrieved data points, helping prioritize the most relevant ones.
- The retriever component is responsible for efficiently identifying and extracting relevant information from a vast amount of data.

For example, consider a smart chatbot that can answer human resource questions for an organization. If an employee searches, *"How much annual leave do I have?"* the system will retrieve annual leave policy documents alongside the individual employee's past leave record.

Augment the LLM prompt

- Next, the RAG model augments the user input (or prompts) by adding the relevant retrieved data in context. This step uses prompt engineering techniques to communicate effectively with the LLM.
- The augmented prompt allows the large language models to generate an accurate answer to user queries.

RAG Components

RAG Generator

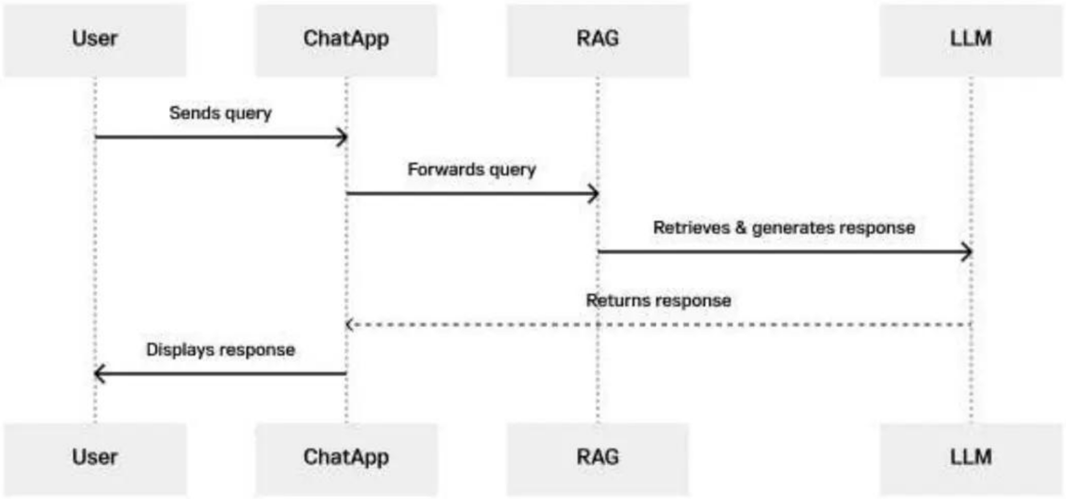
- The RAG generator component is basically the LLM Model such a (GPT)
- The RAG generator component is responsible for taking the retrieved and ranked information, along with the user's original query, and generating the final response or output.
- The generator ensures that the response aligns with the user's query and incorporates the factual knowledge retrieved from external sources.

Update external data

- To maintain current information for retrieval, asynchronously update the documents and update embedding representation of the documents.
- **Automated Real-time Processes**: Updates to documents and embeddings occur in real-time as soon as new information becomes available. This ensures that the system always reflects the most recent data.
- **Periodic Batch Processing**: Updates are performed at regular intervals (e.g., daily, weekly) in batches. This approach may be more efficient for systems with large volumes of data or where real-time updates are not necessary.

RAG Based Chat Application

Simplified sequence diagram illustrating the process of a RAG chat application



Step 1 - User sends query: The process begins when the user sends a query or message to the chat application.

Understanding RAG Architecture

Step 2 - Chat App forwards query: Upon receiving the user's query, the chat application (Chat App) forwards this query to the Retrieval Augmented Generation (RAG) model for processing.

Step 3 - RAG retrieves + generates response: The RAG model, which integrates retrieval and generation capabilities, processes the user's query. It first retrieves relevant information from a large corpus of data, using the LLM to generate a coherent and contextually relevant response based on the retrieved information and the user's query.

Step 4 - LLM returns response: Once the response is generated, the LLM sends it back to the chat application (Chat App).

Step 5 - Chat App displays responses: Finally, the chat application displays the generated response to the user, completing the interaction.

RAG Vs Fine Tuning

- **Objective**
  - Fine-tuning aims to adapt a pre-trained LLM to a specific task or domain by adjusting its parameters based on task-specific data.
  - RAG focuses on improving the quality and relevance of generated text by incorporating retrieved information from external sources during the generation process.

- **Training Data**
  - Fine-tuning requires task-specific labeled data /examples to update the model's parameters and optimize, leading to more time & cost
  - RAG relies on a combination of pre-trained LLM and external knowledge bases

- **Adaptability**
  - Fine-tuning makes the LLM more specialized and tailored to a specific task or domain
  - RAG maintains the generalizability of the pre-trained LLM by leveraging external knowledge allowing it to adapt to a wide range of tasks

- **Model Architecture**
  - Fine-tuning typically involves modifying the parameters of the pre-trained LLM while keeping its architecture unchanged.
  - RAG combines the retrieval and generation components, with the standard LLM architecture to incorporate the retrieval mechanism.

RAG Benefits

- **Enhanced Relevance**:
  - Incorporates external knowledge for more contextually relevant responses.
- **Improved Quality**:
  - Enhances the quality and accuracy of generated output.
- **Versatility**:
  - Adaptable to various tasks and domains without task-specific fine-tuning.
- **Efficient Retrieval**:
  - Leverages existing knowledge bases, reducing the need for large labeled datasets.
- **Dynamic Updates**:
  - Allows for real-time or periodic updates to maintain current information.
- **Trust and Transparency**
  - Accurate and reliable responses, underpinned by current and authoritative data, significantly enhance user trust in AI-driven applications.
- **Customization and Control:**
  - Organizations can tailor the external sources RAG draws from, allowing control over the type and scope of information integrated into the model's responses
- **Cost Effective**

Applications

- **Conversational AI:**
  - RAG enables chatbots to provide more accurate and contextually relevant responses to user queries..
- **Advanced Question Answering:**
  - RAG enhances question answering systems by retrieving relevant passages or documents containing answers to user queries.
- **Content Generation:**
  - In content generation tasks such as summarization, article writing, and content recommendation, RAG can augment the generation process with retrieved information, incorporating relevant facts, statistics, and examples from external sources.
- **Healthcare:**
  - RAG can assist healthcare professionals in accessing relevant/latest medical literature, guidelines.

Demo

Thank you

Nash
Tech.