RAG : Turning Knowledge into conversations

29/11/2024

GDG Carthage

By Mohammed Arbi Nsibi

GDG Carthage

MOHAMED ARBI NSIBI

- Final year ICT engineering student@ SUP'COM
- GDG Carthage member
- GDG On Campus SUP'COM & ISAMM Mentor

mohammedarbinsibi@gmail.com

https://huggingface.co/Goodnight7

https://www.linkedin.com/in/mohammed-arbi-nsibi-584a43241/

GDG Carthage

Agenda

- Introduction to Retrieval Augmented Generation (RAG)
  - Why do we need RAG?
  - Concept of RAG
  - Framework of RAG
  - Naive RAG vs Advanced RAG
- Configuring your environment
- simple RAG demo

GDG Carthage

Introduction

Challenge: LLM has limited knowledge, causing hallucinations

Retrieval Augmented Generation (RAG)

- RAG extends the powerful capabilities of LLMs to specific domains or an organization's internal knowledge base, all without the need to retrain the model.
- It is a cost-effective approach to improving LLM output so it remains relevant, accurate, and useful in various contexts.

Why use RAG?

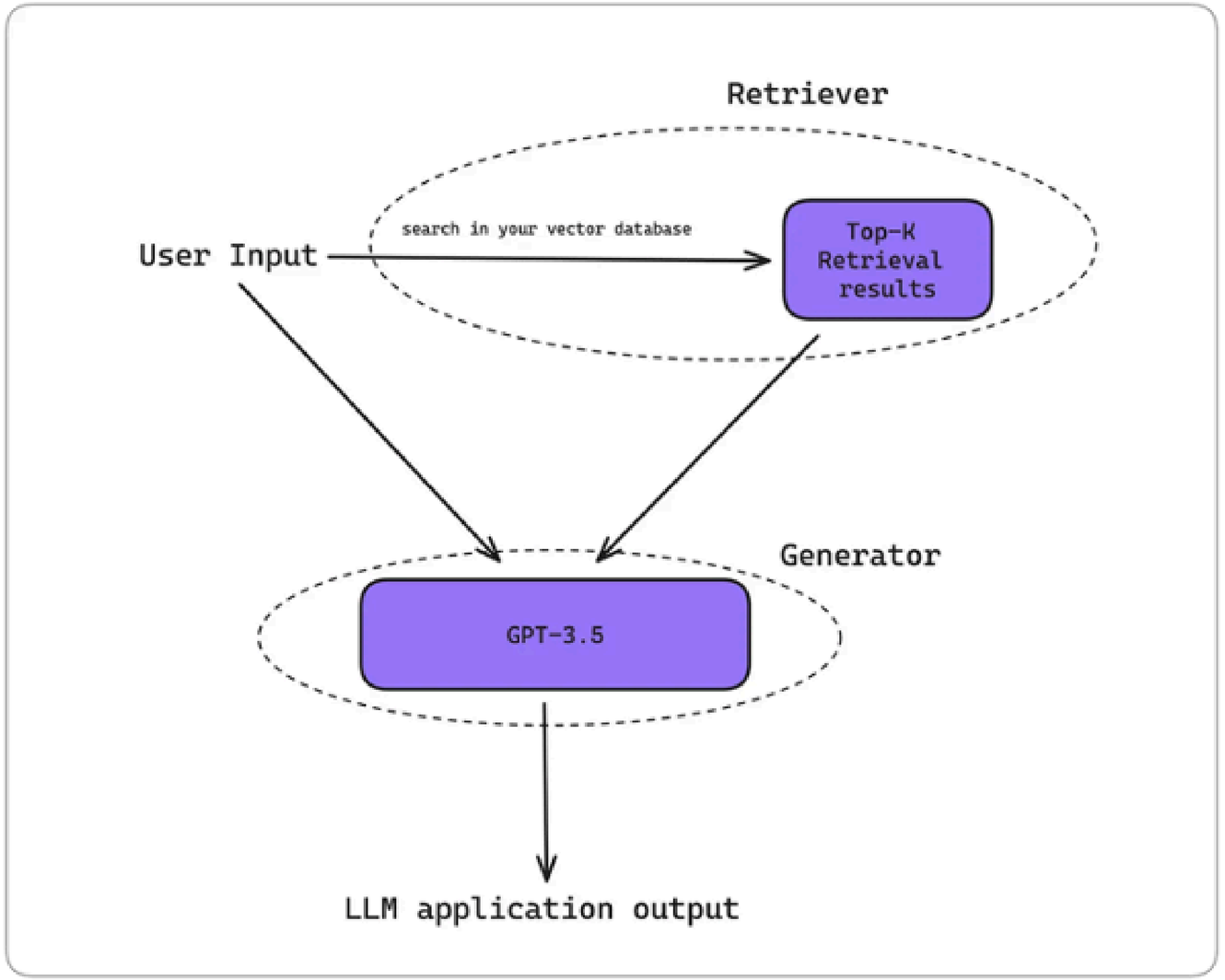
There are three ways an LLM can learn new data.

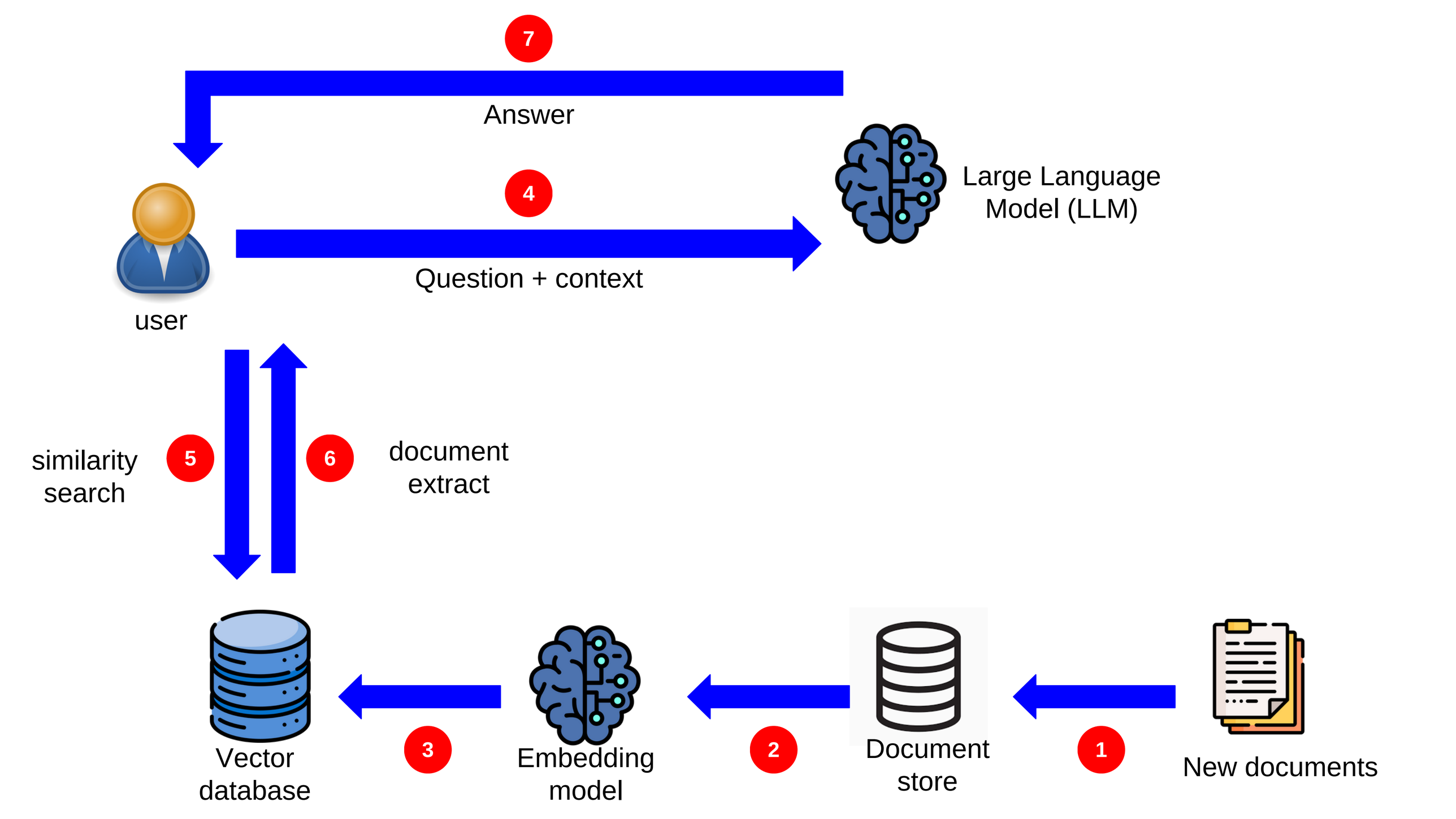
1. **Training:** A large neural networks is trained over trillions of tokens with billions of parameters to create Large Language Models. To train a model like GPT-4 costs hundreds of millions of dollars.
2. **Fine-tuning:** fine-tune a model on existing data. Fine-tuning involves using a pre-trained model as a starting point during training. We use the knowledge of the pre-trained model to train a new model on different data sets.
3. **Prompting:** Prompting is the method where we fit new information within the context window of an LLM and make it answer the queries from the information given in the prompt.

Retrieval Augmented Generation (RAG)

- A RAG system consists of two primary components: **retriever** and **generator**.
- The **retriever** is responsible for searching through the knowledge base for the most relevant pieces of information that correlate with the given input.
- The **generator** utilizes these retrieval results to craft a series of prompts based on a predefined prompt template to produce a coherent and relevant response to the input.

Retrieval Augmented Generation (RAG)




7
Answer
Large Language Model (LLM)
4
Question + context
user
similarity search
5
6
document extract
Vector database
3
Embedding model
2
Document store
1
New documents

Embedding

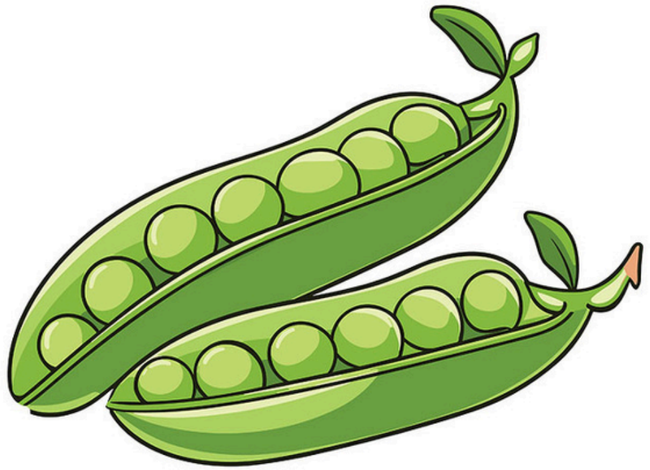
- Machine Learning models only able to process numbers
- Word embedding is mathematical representation of texts
- Embeddings aim to represent the work in a dense vector, while making sure that similar words are close to each other in the embedding space

tea

coffee

tea

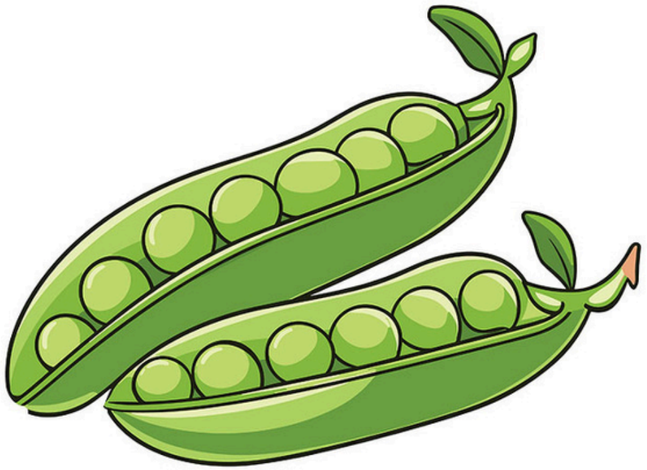
≠



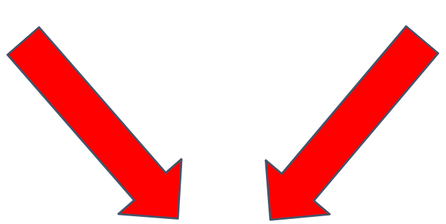
pea

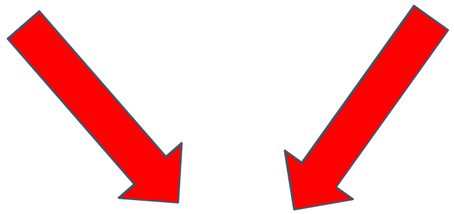
tea

coffee



pea



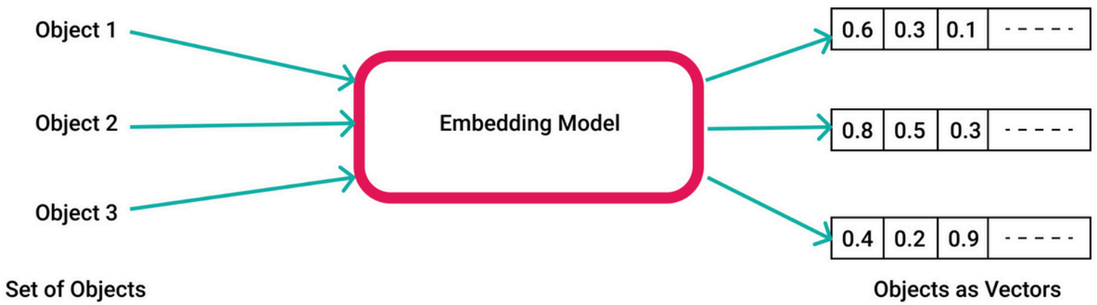


distance = 0.3

distance = 0.7

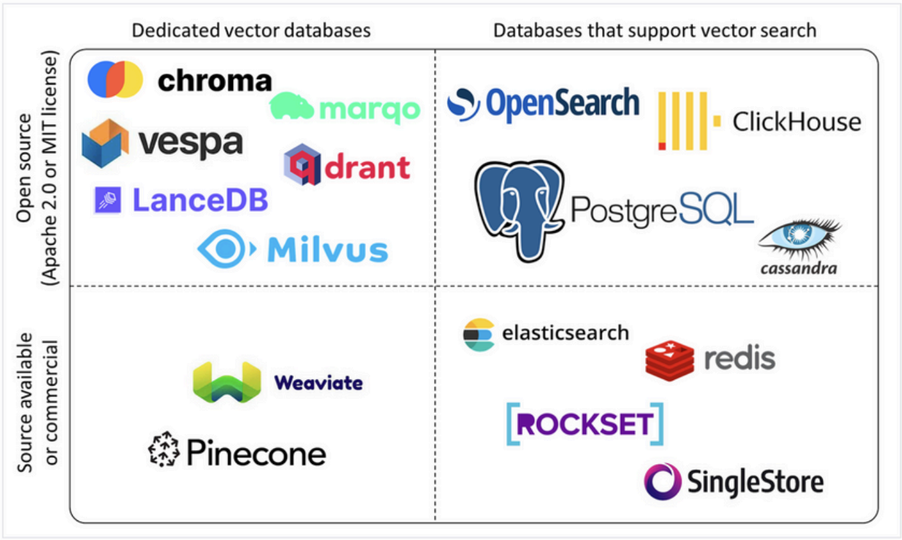
Embedding Model

- These vectors live in a high-dimensional space where the proximity between vectors reflects the relatedness of the original items.
- **Embedding model** trained along LLM and learn to **produce representation(vectors) based on context** in word appear.

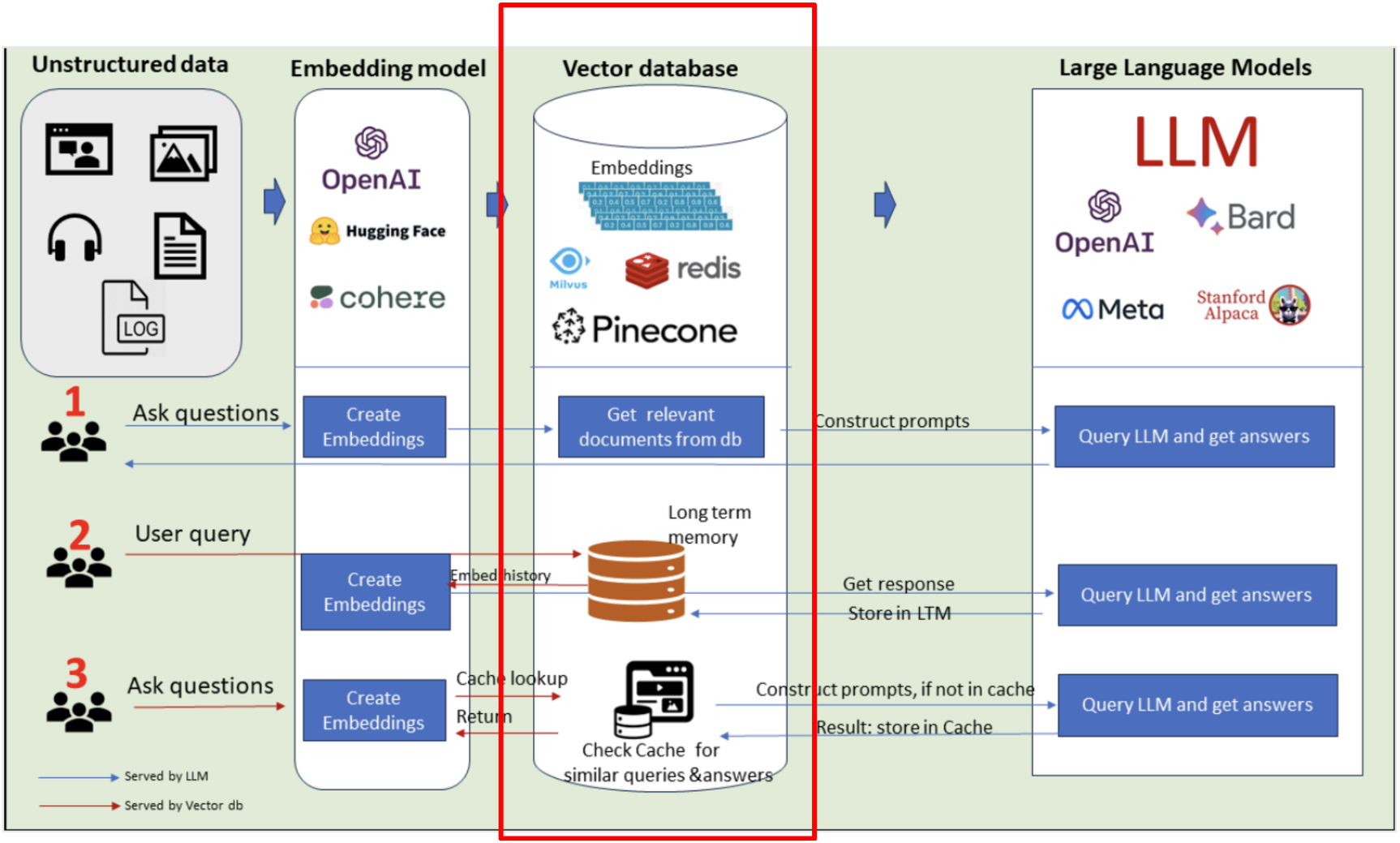


Vector Database

- A **vector database** is a type of database that **stores data as high-dimensional vectors**, which are mathematical representations of features or attributes.
- Each vector has a certain number of dimensions, which can range from tens to thousands, depending on the complexity and granularity of the data.



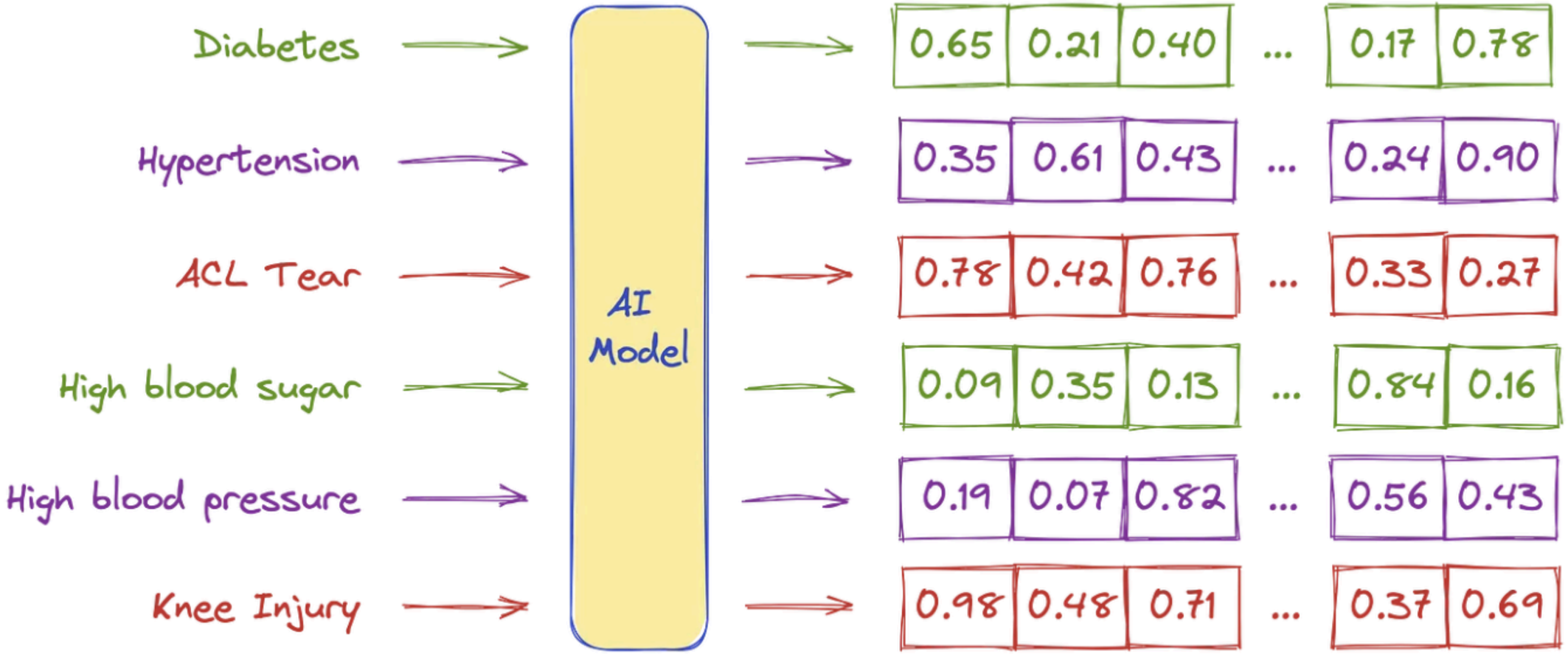
Vector Database



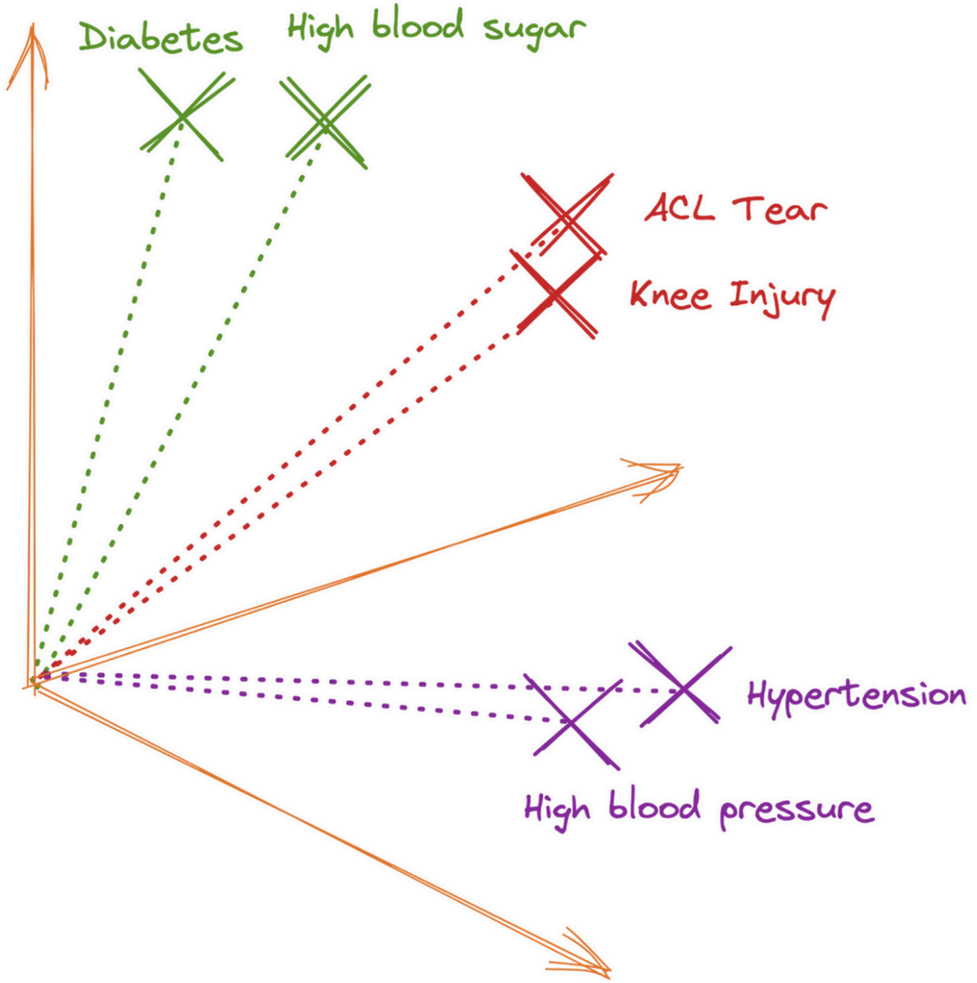
Embedding Space

- **Transformation:** It is a concept where complex objects like text, images, or audio are transformed into numerical vectors space.
- **Dimension Reduction:** This new space usually has fewer dimensions than the original data, making it easier for algorithms to process and analyze.
- **Similarity preservation:** Ideally, similar objects in the original data remain close together in the embedding space. For example, words with similar meanings would have nearby vectors.

Embedding Space



Embedding Space



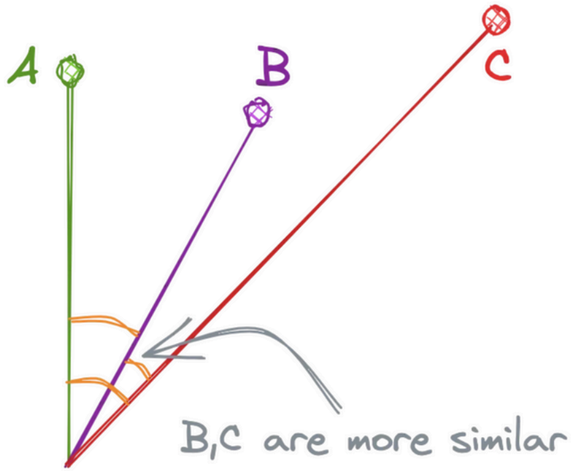
Semantic Search & Similarity Search

- While both semantic and similarity search deal with finding related information, they approach it in slightly different ways:
  - **Semantic Search:** focuses on meaning and tries to understand the intent/meaning behind the query. Eg. the best laptop for a student: affordability, portability, battery life etc
  - **Similarity Search:** focuses on similar features. Eg. picture of a cat: find similar species, similar settings etc
- **Objective:** Semantic search aims to find relevant information, while similarity search aims to find similar items.

Similarity Calculation

- For calculating similarity, there are several methods:
  - measuring distance - euclidean distance
  - cosine similarity or inner product

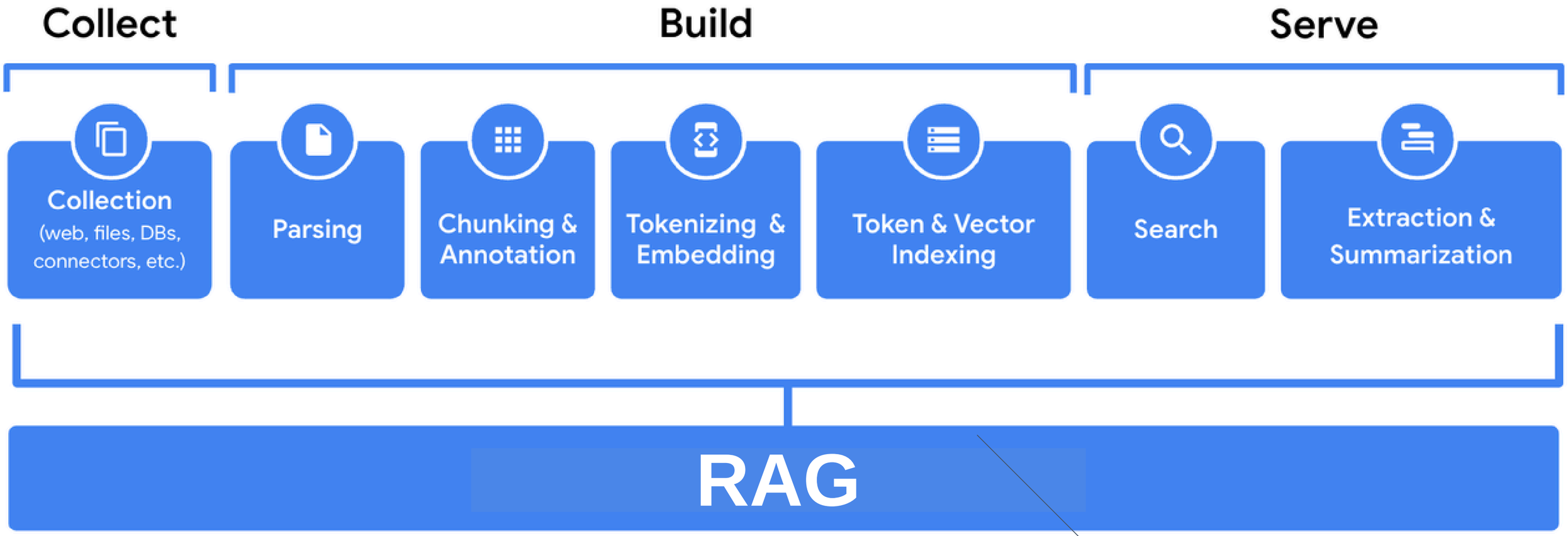
Euclidean distance



Cosine similarity

Dot Product

RAG Framework



Benefits of RAG

- **Accuracy**
- **Relevance**
- **Transparency**
- **Cost-effectiveness**

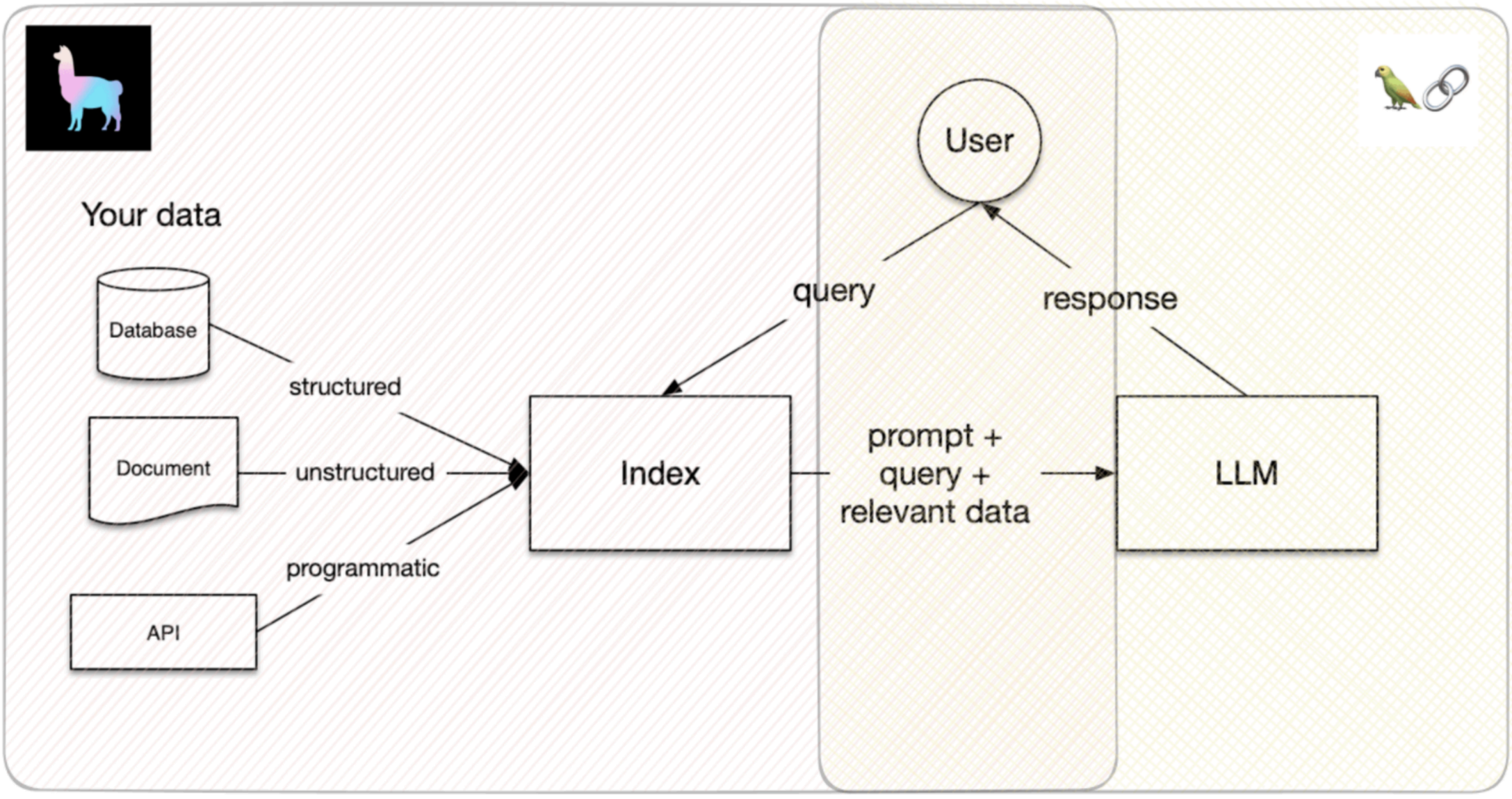
Limitations of RAG

- **Model complexity:** RAG models combine a retrieval-based model with a generative model, more complex and computationally expensive to train and deploy.
- **Data requirements:** RAG models require a large dataset of text and code to train the retrieval-based model, and a large dataset of text to train the generative model.
- **Performance trade-off:** lower latency this can come at the cost of some accuracy.

Where do I start?

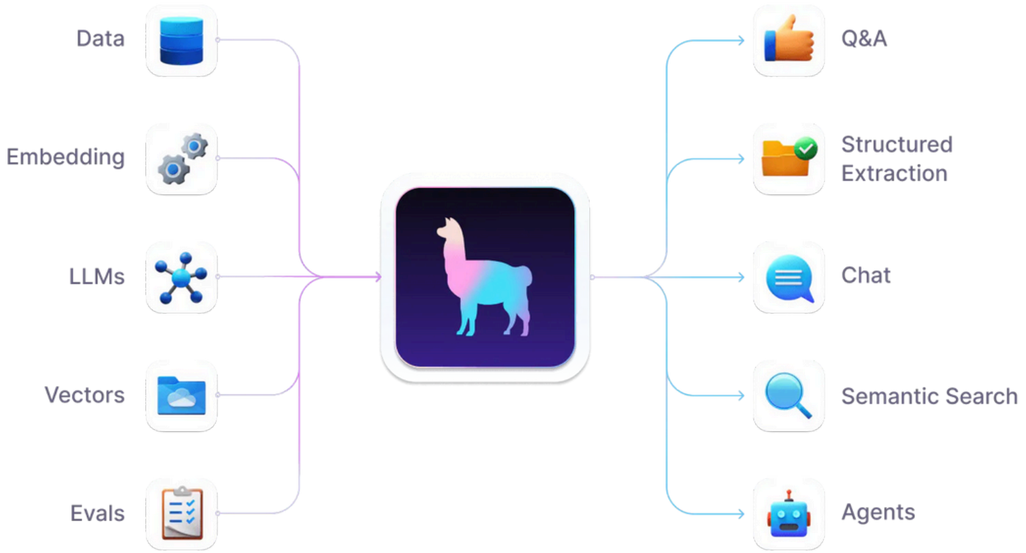
LangChain

- LangChain is a framework designed to simplify the creation of applications using large language models.
- It is based on LCEL (LangChain Expression Language)
- Use-cases including chatbots, RAG, document summarization and synthetic data generation.


Your data
Database
Document
API
structured
unstructured
programmatic
Index
User
query
response
prompt +
query +
relevant data
LLM

LlamaIndex

- LlamaIndex is a handy tool that acts as a bridge between your custom data and large language models (LLMs) which are powerful models capable of understanding human-like text.
- Since majority applications are RAG, LlamaIndex provides the right tools to build RAG



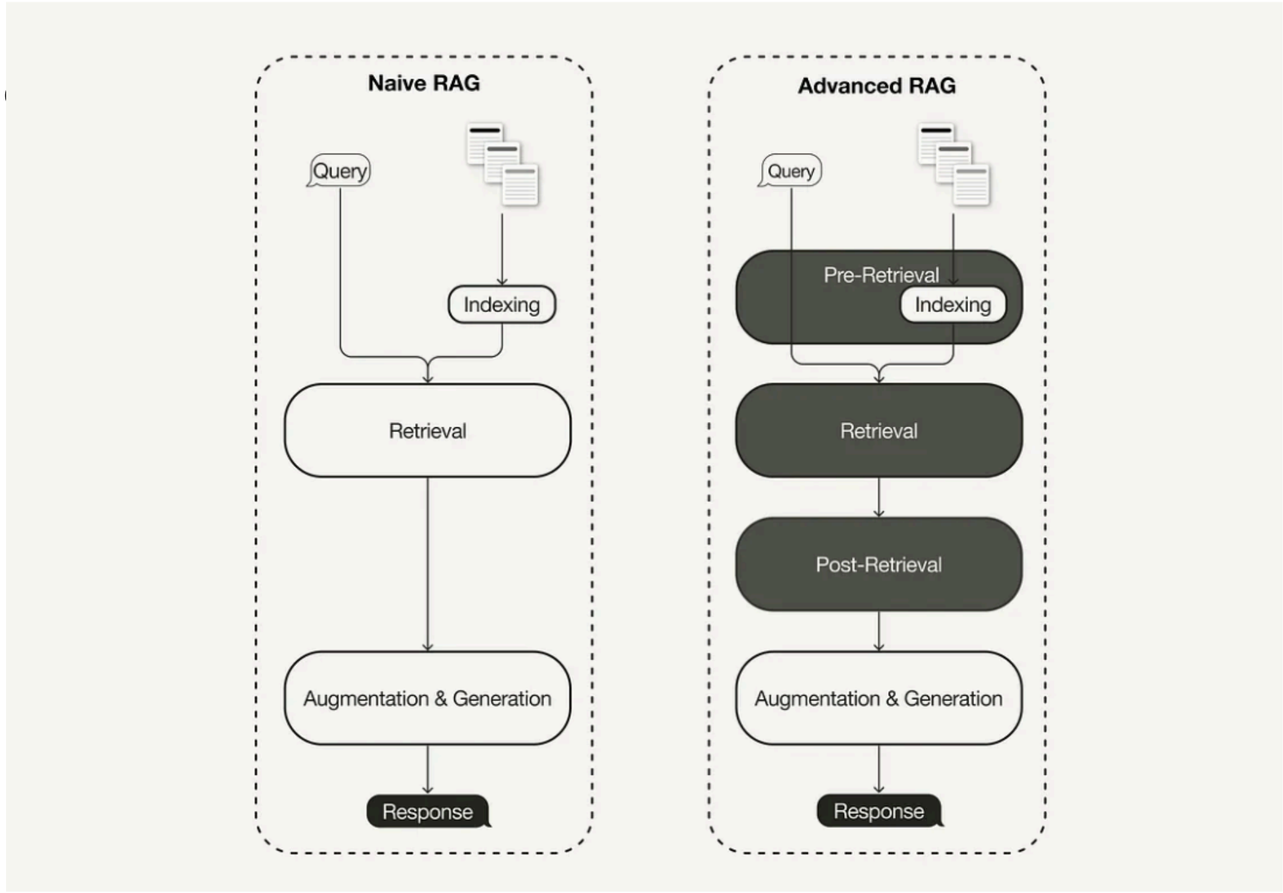
LlamaIndex

- Components of an RAG application or workflow:
  - **Knowledge Base (Input):** The knowledge base is like a library filled with useful information such as FAQs, manuals, and other relevant documents. When a question is asked, this is where the system looks to find the answer.
  - **Trigger/Query (Input):** Question/Prompt asked by user
  - **Task/Action (Output):** After understanding the trigger or query, the system then performs a certain task to address it.

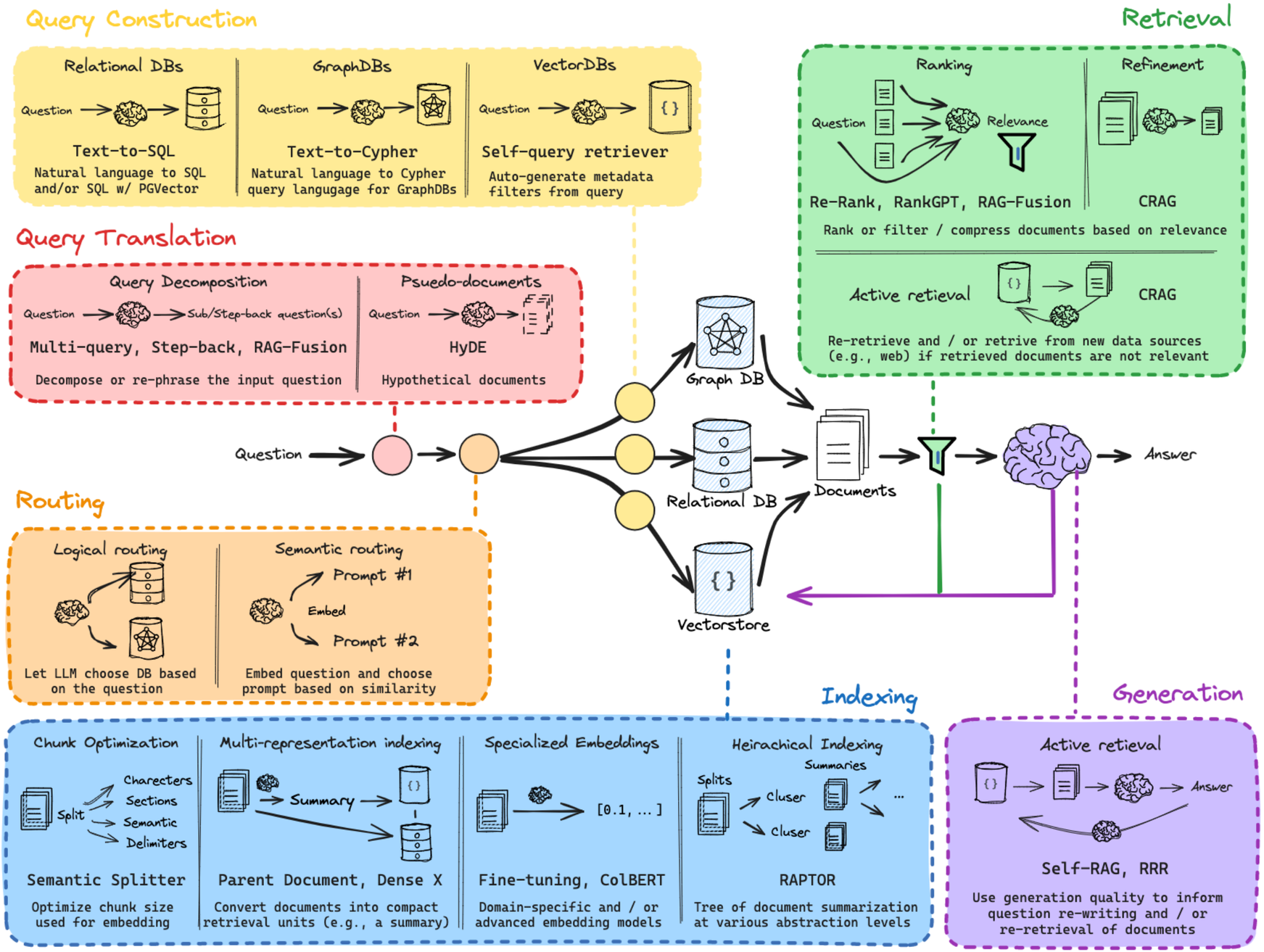
Naive RAG vs Advanced RAG

- There are many implementation to further improve performance Naive RAG.
- Advanced RAG has evolved as a new paradigm with targeted enhancements to address some of the limitations of the naive RAG paradigm.

- Advanced RAG techniques can be categorized into
  - pre-retrieval optimization,
  - retrieval optimization, and
  - post-retrieval optimization



Difference between Naive and Advanced RAG (Image by the author, inspired by [1])



https://github.com/langchain-ai/rag-from-scratch/tree/main?tab=readme-ov-file

Resources

- Your RAG powered by Google Search Technology
- Let’s talk about LlamaIndex and LangChain
- Retrieval-Augmented Generation (RAG) framework in Generative AI
- Retrieval-Augmented Generation (RAG): From Theory to LangChain Implementation
- Advanced Retrieval-Augmented Generation: From Theory to LlamaIndex Implementation
- Retrieval-augmented generation for large language models: A survey [arXiv]

Q&A

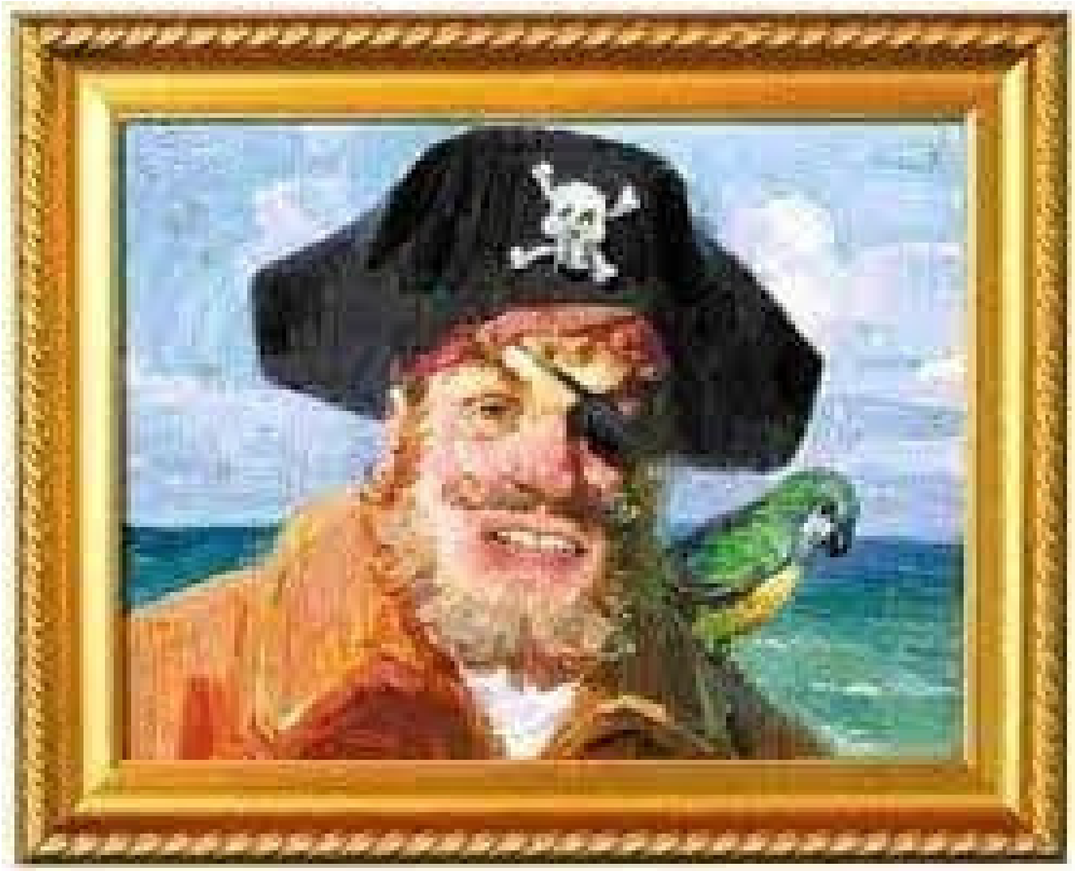
GDG Carthage

By Mohammed Arbi Nsibi

THANK YOU for you attention!!


https://www.linkedin.com/in/mohammed-arbi-nsibi-584a43241/

QUIZ TIME 😄



GDG Carthage

By Mohammed Arbi Nsibi