AI

Summer of Code

Getting Started with Vector Databases

SAM AYO

Lead AI Engineer & Head of Engineering
AISoC, co-host.

https://www.linkedin.com/in/sam-ayo
https://www.x.com/officialsamayo

About

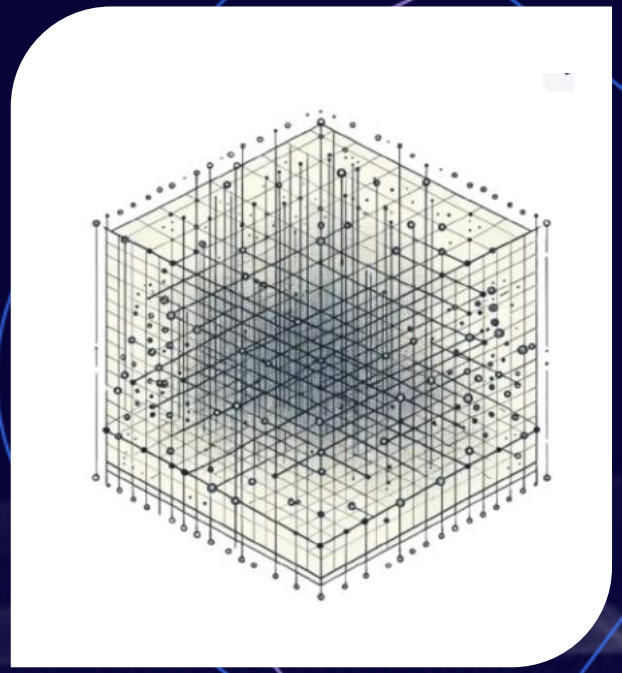
- **Academic background:** Economics, Math, Stats, ARTIBA
- **Areas of Interest:** Core AI, NLP, Audio AI, AI Engineering, probabilistic models, experimentation & system inference design.
- **Programming Languages:** Python, C++, C#, Golang, JavaScript, TypeScript.
- **Recent work:** Real-time Agentic system, near real-time audio signal detection, Semantic relation modelling and search.
- **Industries covered:** Agnostic
- **Fun fact:** Built LangChain equivalent in golang

Content

1. Why Vector Databases?
2. How do Vector Databases work?
3. Vector Databases for LLM Apps
4. Let's code

Why Vector Databases?

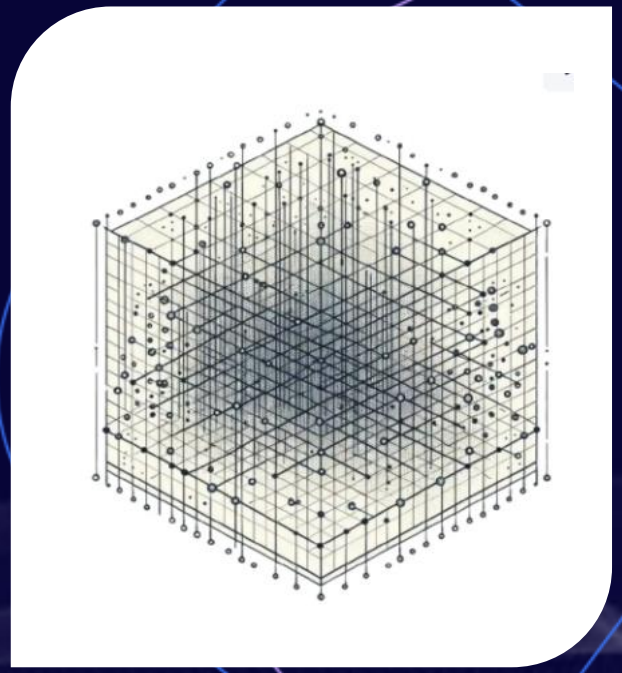
- *Introduction to vector*
- *Unstructured data*
- *Traditional database vs vector database*



Sam Ayo

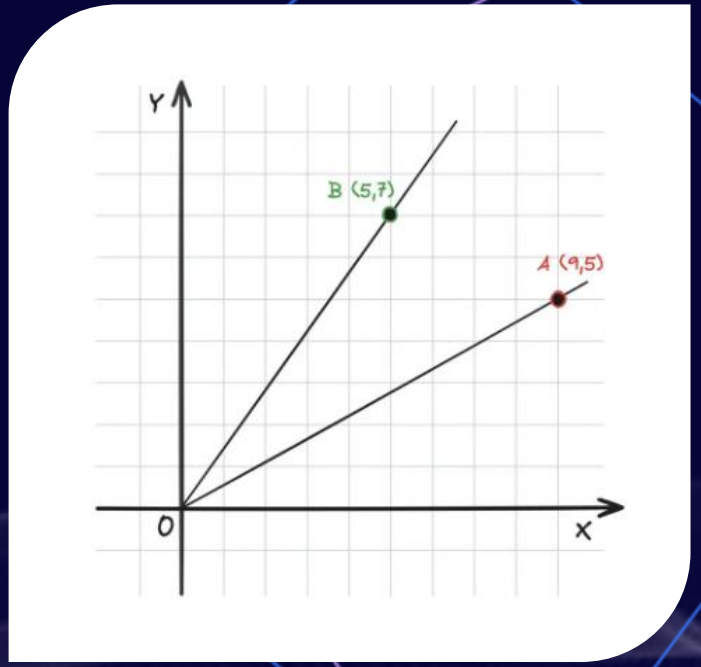
Vector databases aka similarity search engines or approximate nearest neighbour search engines are specialized databases that efficiently store, index and relate entities of data by a quantitative value.

In other words, vector databases are specially designed databases that handle high-dimensional vectors efficiently.



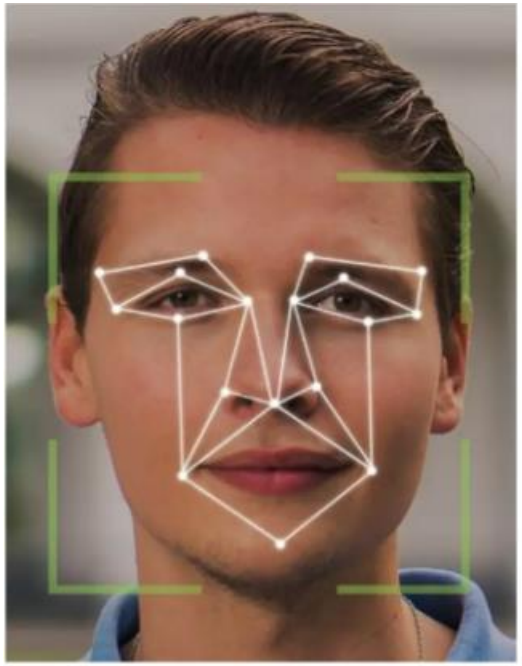
Introduction to Vectors

- **Vectors** are mathematical objects that represent quantities with both magnitude and direction.
- In the context of vector databases, vectors are used to represent data points, where each data point's feature or attributes is represented by the component of that vector.
- In an n-dimensional space, a vector represents data as a coordinate point. For example, on a x-y coordinate plane, A 2-dimensional vector can define a location on that plane.



Introduction to Vectors

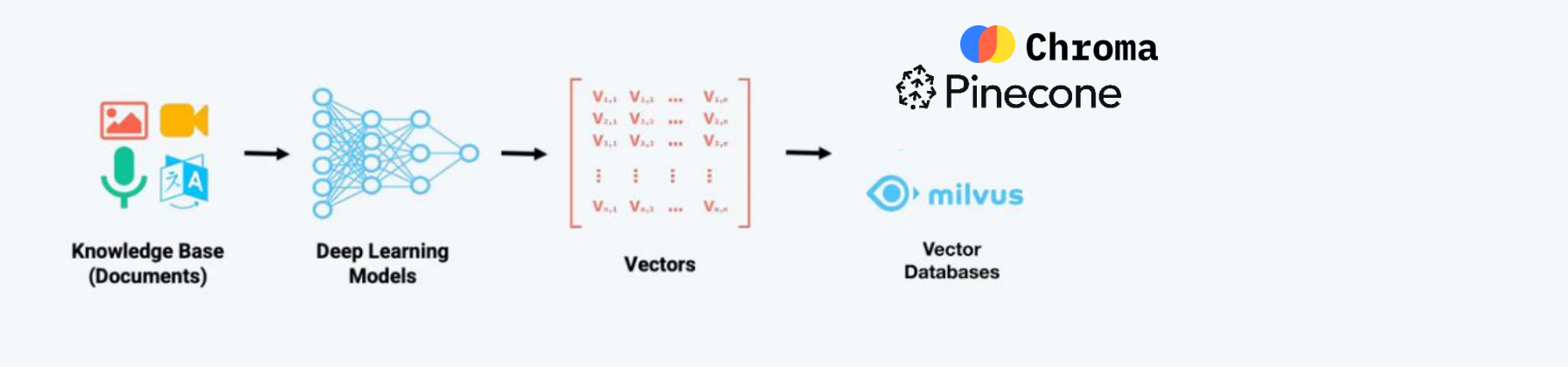
Each element of a **vector** is a feature And the entire vector encapsulates the essence of the data item.



Unstructured Data

Unstructured Data is where it began

Unstructured Data is any data that does not conform to a predefined data model.
Vectors are the generated numerical representation of unstructured data.



Text

Images

Video

Traditional Databases vs Vector Databases

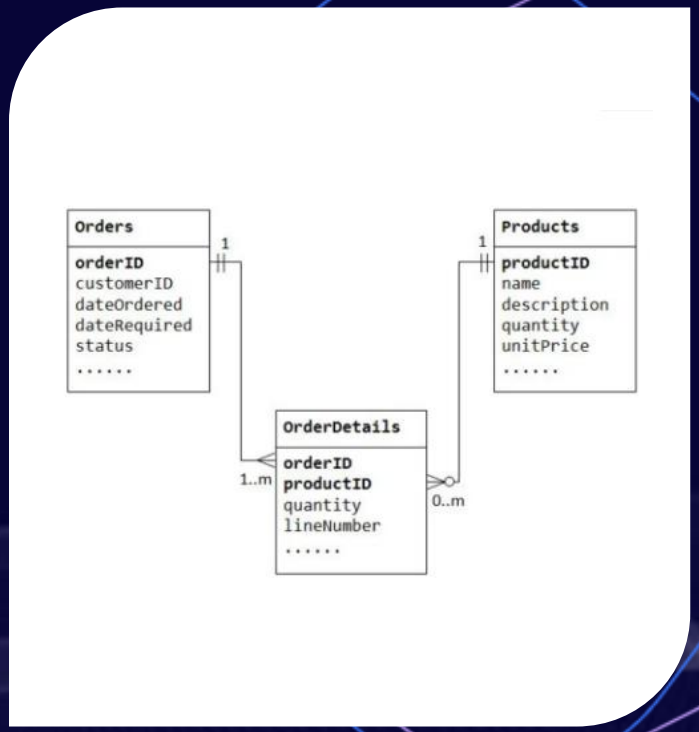
- Compare data you couldn't compare before - generalist
- Use math to quantify relationships between entities – generalist
- Optimized for handling unstructured, high-dimensional data such as images, text documents and user embeddings.
- Find semantically similar data - generalist
- Give LLMs fine-context and improved accuracy in response quality - LLM
- Control Hallucination - LLM

Vespa
Weaviate
Chroma
Pinecone
drant
Milvus
SingleStore

Traditional Databases vs Vector Databases

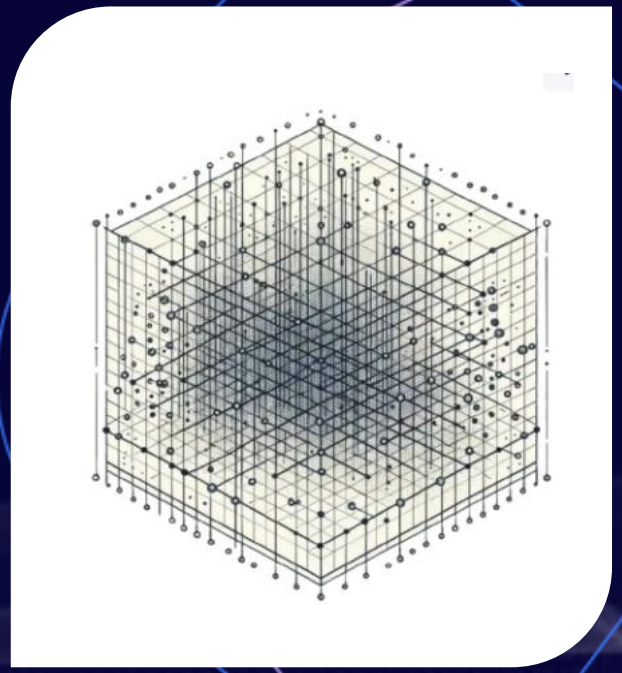
Why can't I just use a SQL/NoSQL Database?

- Limited analytics capabilities
- Data conversion issues
- Suboptimal indexing
- Inefficiency in high-dimensional spaces
- Traditional databases are not optimized for the computationally intensive nature of vector operations.
- Traditional databases store data in structured tables and focus on ACID(Atomicity, consistency, isolation and durability) properties for transactional data integrity.

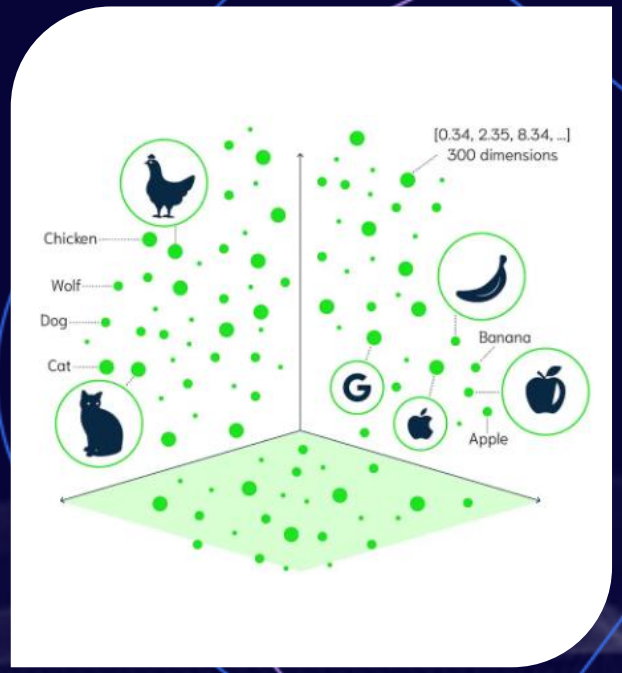


How Do Vector databases Work?

*The answer is simple – **semantic similarity search***



Similarity search is the process of retrieving data points that are similar to a given query point based on a chosen distance metric or similarity measure.



How Do Vector Databases Work?

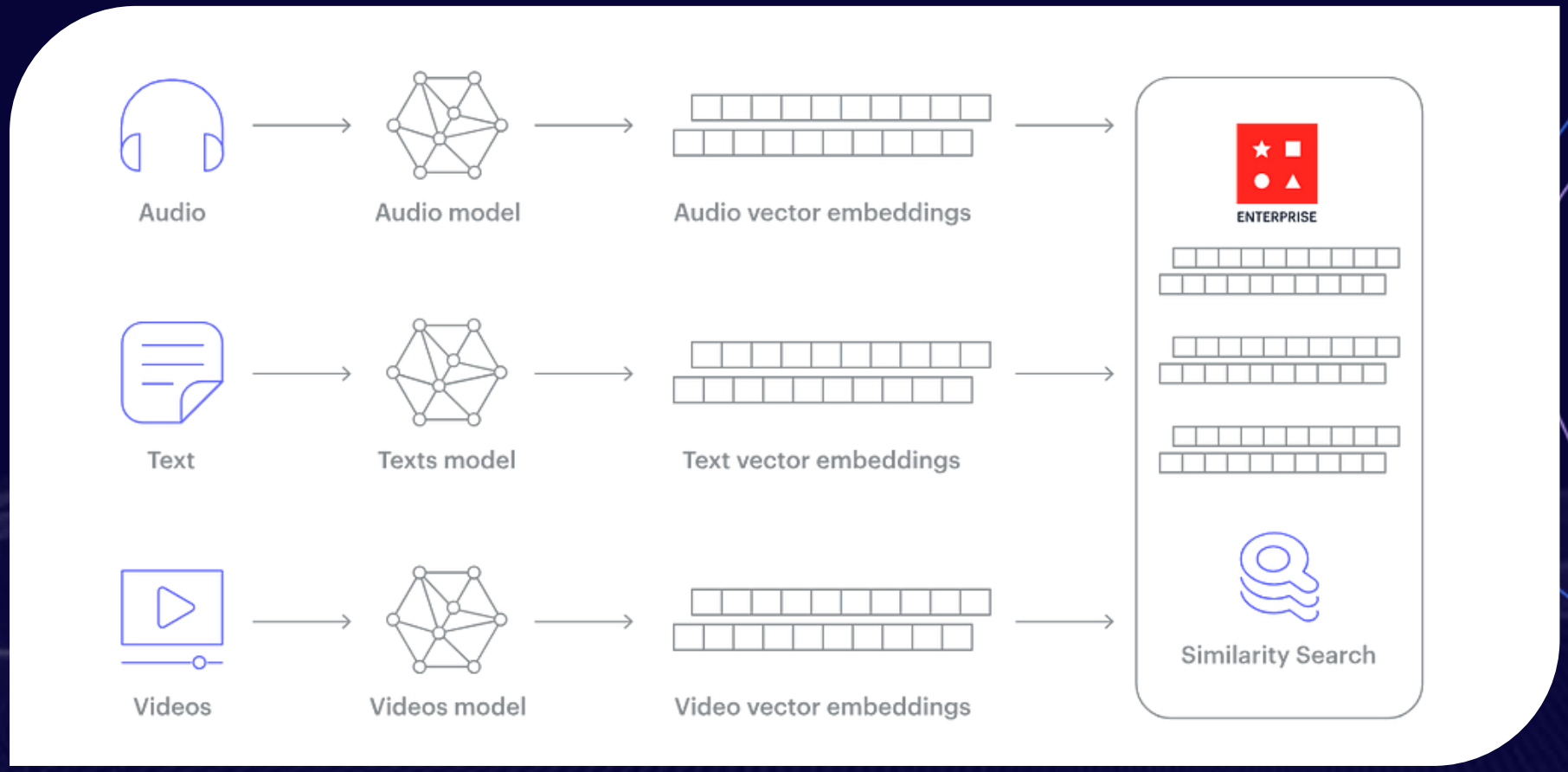
Vector similarity is a mathematical measure of how close two vectors are

Vector similarity metrics include:

- Euclidean(L2 norm) – spatial distance
- Manhattan(L1 norm) – spatial distance
- Cosine – Orientational distance
- Inner Product – (Euclidean and cosine)

Vespa
Weaviate
Chroma
Pinecone
drant
Milvus
SingleStore

How Do Vector Databases Work?



How Do Vector Databases Work?

Use cases for vectors beyond LLMs and RAG

LLM Augmented Retrieval
Expand LLMs' knowledge by incorporating external data sources into LLMs and your AI applications.

Recommender System
Match user behavior or content features with other similar behaviors or features to make effective recommendations.

Text/ Semantic Search
Search for semantically similar texts across vast amounts of natural language documents.

Image Similarity Search
Identify and search for visually similar images or objects from a vast collection of image libraries.

Video Similarity Search
Search for similar videos, scenes, or objects from extensive collections of video libraries.

Audio Similarity Search
Find similar audios from massive amounts of audio data to perform tasks such as genre classification, or recognize speech.

Molecular Similarity Search
Search for similar substructures, superstructures, and other structures for a specific molecule.

Question Answering System
Interactive QA chatbot that automatically answers user questions

Multimodal Similarity Search
Search over multiple types of data simultaneously, e.g. text and images

Vector Databases for LLM Apps

- Concept of embeddings
- Vector indexing, chunking strategy and embedding strategy
- Making technology choices on vector databases

Vespa
Weaviate
Chroma
Pinecone
drant
Milvus
SingleStore

Vector Databases for LLM Apps

You know I'm talking about RAG right?

So, let's begin with **vector embeddings.**

Vector embeddings are numerical representation of vector data in a continuous space.
The sole purpose is to capture semantic meaning between words, phrases or long-form documents.

- There are several dozen embedding models.
- They range in complexity from 384 – 1536 dimensions
- They range in max sequence length from 512 to 8191 tokens

Vector Indexing

Vector indexing is a technique used in vector databases to intelligently organize vector embeddings to enable fast and accurate search/retrieval process.

There are different index strategy and when you should use them, so of them are:

- **HNSW(Hierarchical Navigable Small World)** – for very large dataset where query speed is more important
- **Product Quantization** – when storage or memory is limited
- **Flat** – for small datasets where precision is critical
- **IVF(Inverted File Index)** – for medium sized dataset where there's a tradeoff between precision and speed.

Vector Databases for LLM Apps

You know I'm talking about RAG right?

Chunking strategy

Your chunking strategy depends on what your data looks like and what you need from it.

What you must consider:

- Chunk size (fixed size, paragraph, semantic)
- Chunk overlap
- Chunk splitters

Embedding strategy

Your embedding strategy depends on your accuracy, cost and use case needs.
It involves:

- Embedding chunks directly
- Embedding sub and super chunks
- Incorporating chunking metadata

What you must consider:

- Accuracy
- Appropriateness for task
- Speed of computation
- Length of output vector
- Size of input

Vector Indexing

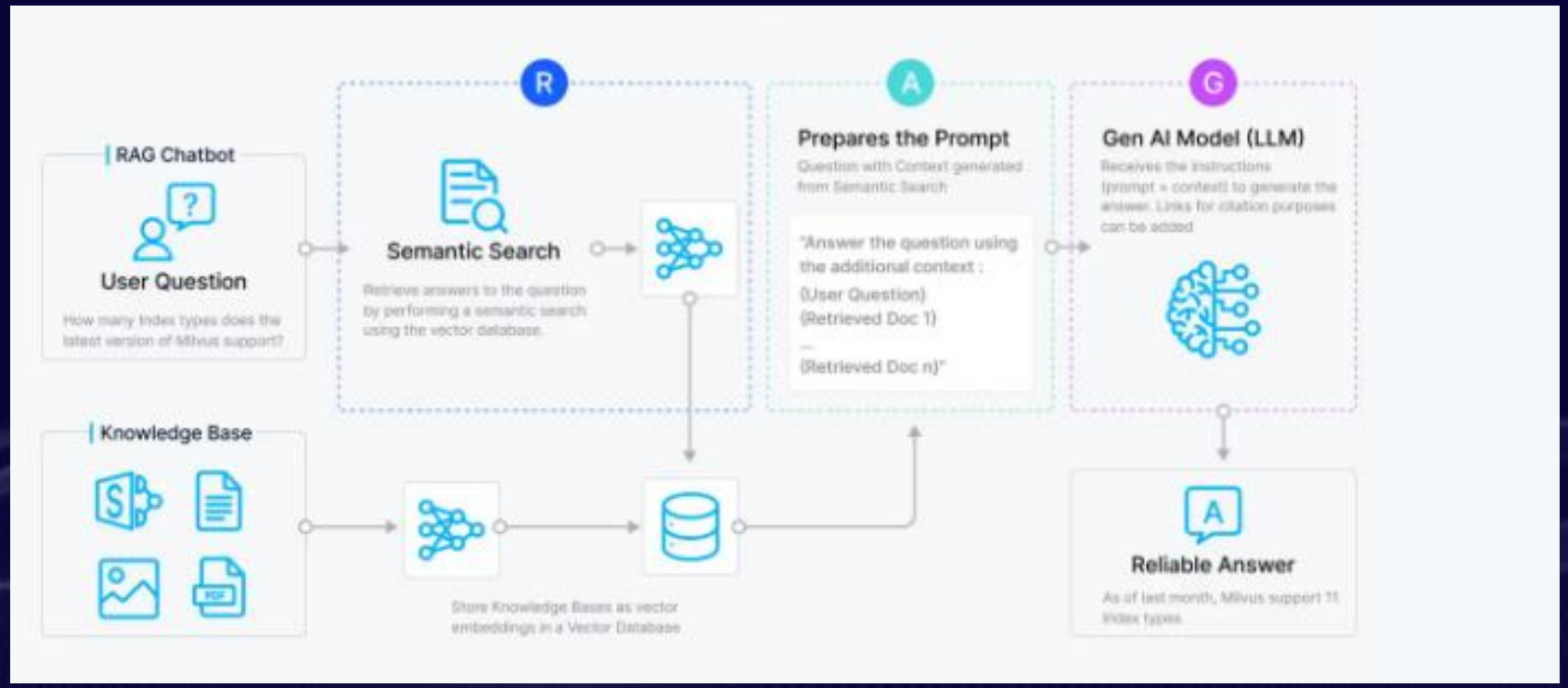
How do I pick the right embedding model for my RAG?

HUGGING FACE

OpenAI

Vector Databases for LLM Apps

Vector Databases are core components for Retrieval Augmented Generation (RAG)



Let's Code

Choose your vector database

Open source	Closed source
Chroma	Pinecone
Qdrant	mongoDB
Vespa	
Milvus	
LanceDB	
Weaviate	

QUESTIONS


WHAT?
WHY?
WHERE?
WHEN?
WHO?
HOW?
WHICH?
WHOSE?
WHO? WHERE? WHAT? HOW?
WHAT? HOW? WHY?
WHERE? HOW? WHO?
WHO? WHERE? WHAT? HOW?
WHAT? HOW? WHY?

AI

Summer of Code