CHENNAI
DATA CIRCLE

Building Next-Gen AI Apps with LLMs

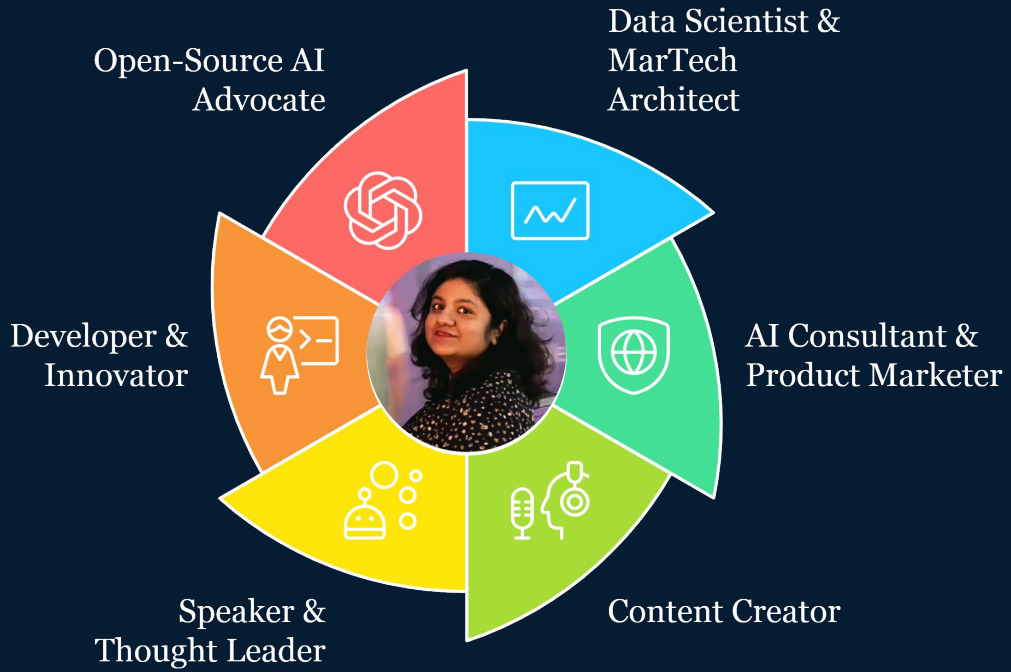
Arthi Rajendran
AI Research Engineer & Advocate

Disclaimer

The views and opinions expressed in this presentation are solely those of the presenter and do not necessarily reflect the official policy or position of any affiliated organization or entity.

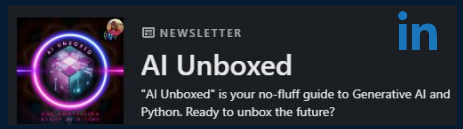
About: Arthi Rajendran

Professional Profile Overview



YouTube

Spotify



CHENNAI DATA CIRCLE

Agenda

- A quick travel through history
- Understanding Embedding Vectors
- DocIQ App Demo
- DocIQ Architecture Overview
- Vector Databases
- Introduction to Retrieval-Augmented Generation (RAG)
- DocIQ’s Technology Stack
- Wrap-up
- Q&A

The Basics

Evolution of Gen AI

CHENNAI DATA CIRCLE

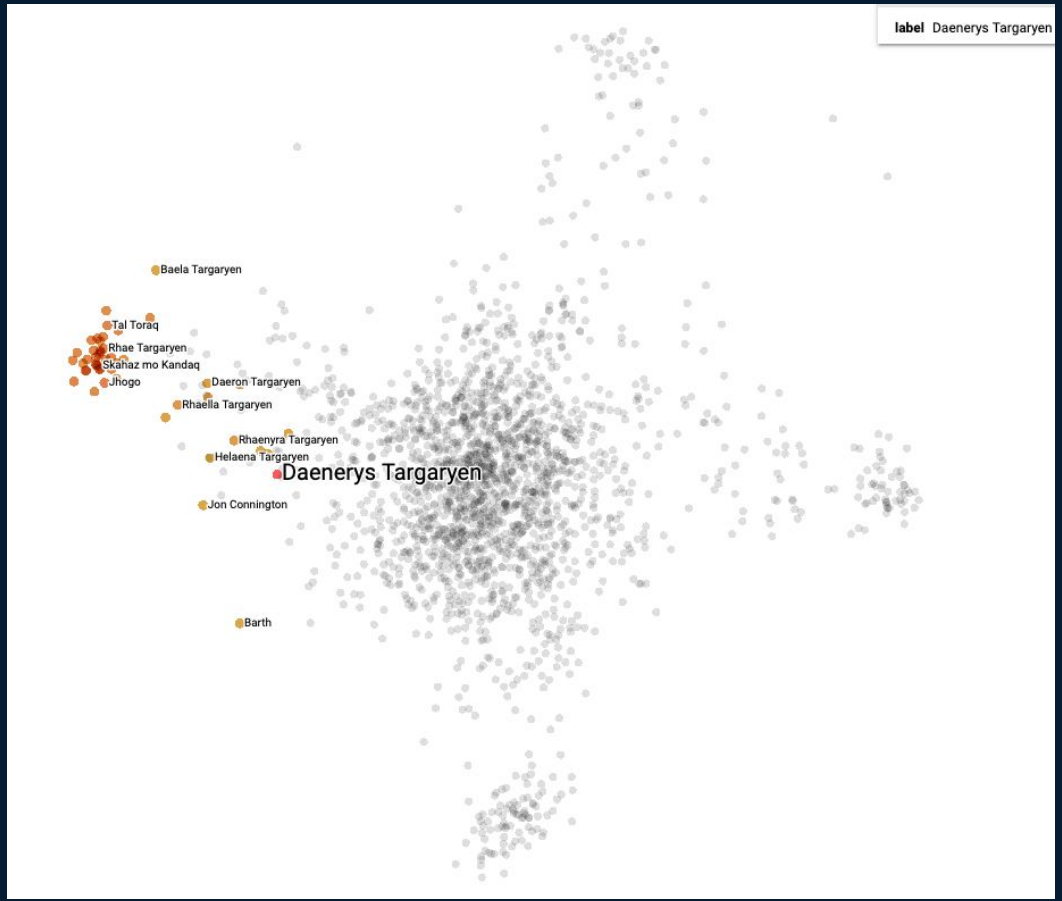
The Basics

How does BERT Work?

(Or just any other transformer model)

CHENNAI DATA CIRCLE

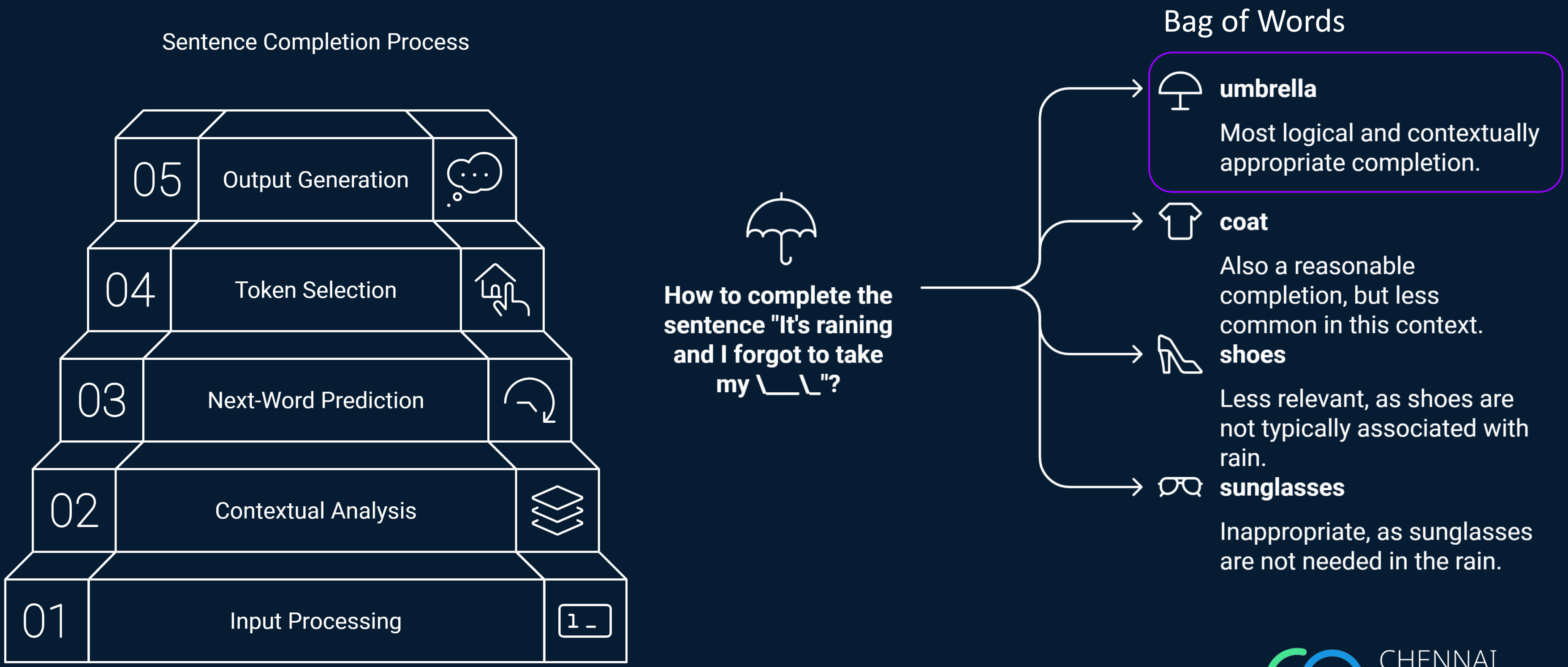
The Embedding vector



Source:
https://docs.ampligraph.org/en/2.0.0/tutorials/AmpliGraphBasicsTutorial.html

CHENNAI
DATA CIRCLE

Sentence Completion



CHENNAI DATA CIRCLE

DocIQ : Quick Demo

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

Your Personal Document Assistant

Navigate

Chatbot

Chatbot Interface (Llama 3.2 RAG)

Upload Document

Upload a PDF

Drag and drop file here
Limit 200MB per file • PDF
Browse files

s41591-024-03302-1.pdf 9.1MB

File Uploaded Successfully!

Filename: s41591-024-03302-1.pdf

File Size: 9581255 bytes

Embeddings

Create Embeddings

Embeddings successfully created and stored in Chroma!

Chat with Document

Type your message here...

what is this article about? give me short summary

Based on the context provided, it appears that this article discusses a study evaluating the effectiveness and consistency of AI-generated reports in medical imaging, specifically chest X-ray reports. The authors compare human-written reports (A) with AI-generated reports (B), assessing their accuracy, completeness, and clarity.

Thanks for asking!

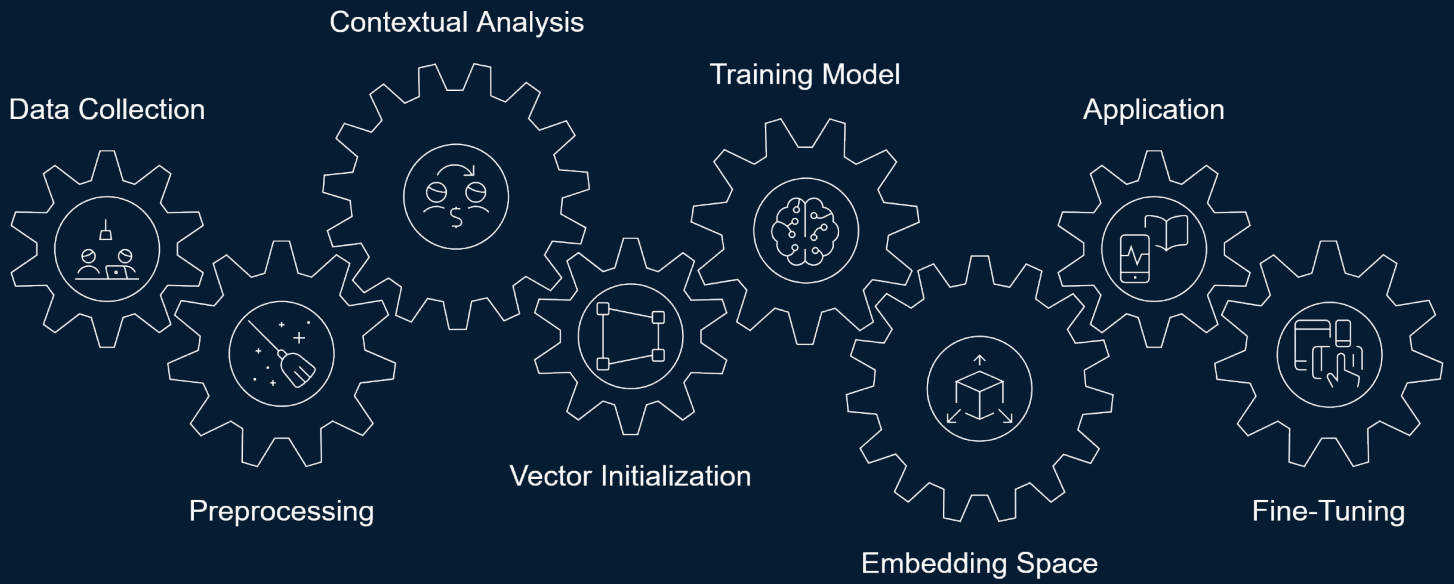
CHENNAI
DATA CIRCLE

DocIQ : What goes on BTS

- **Document Uploader:** Supports various formats with a preview feature.
- **Embeddings Engine:** Transforms text into numerical representations for machine understanding.
- **Vector Database (Chroma):** Stores embeddings for efficient retrieval.
- **LLM (LLaMA 3.2):** Powers the question-answering capabilities.
- **Ollama:** Ensures smooth operation on devices with limited computing power.
- **LangChain:** Orchestrates the interaction between components.

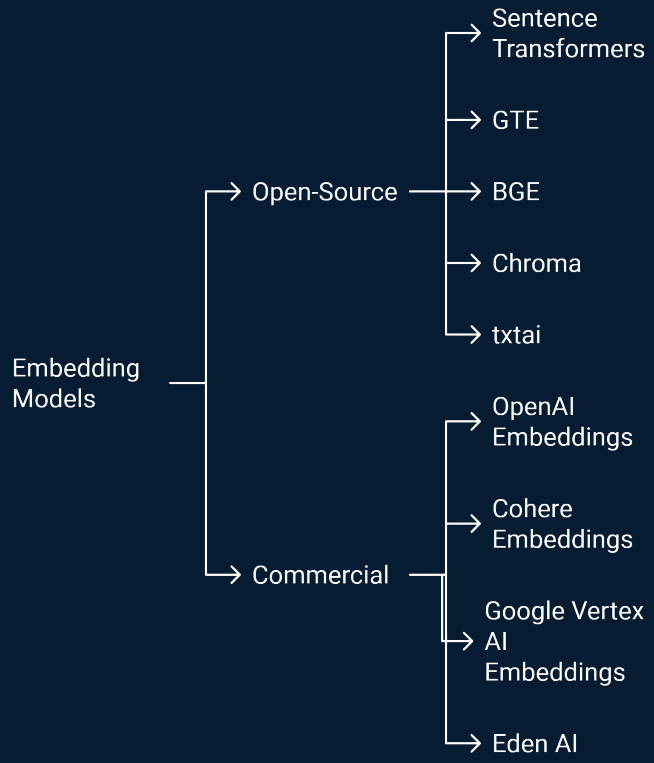
Embeddings Engine

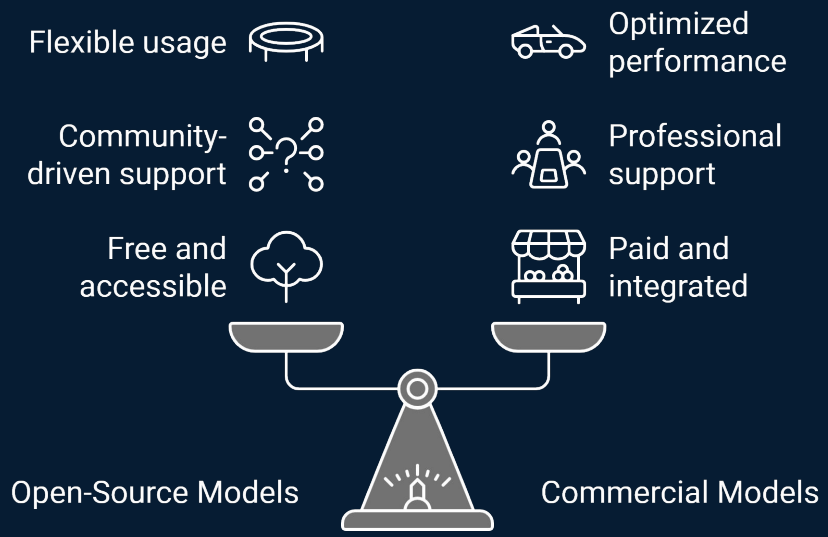
Journey to Effective Embeddings



CHENNAI
DATA CIRCLE

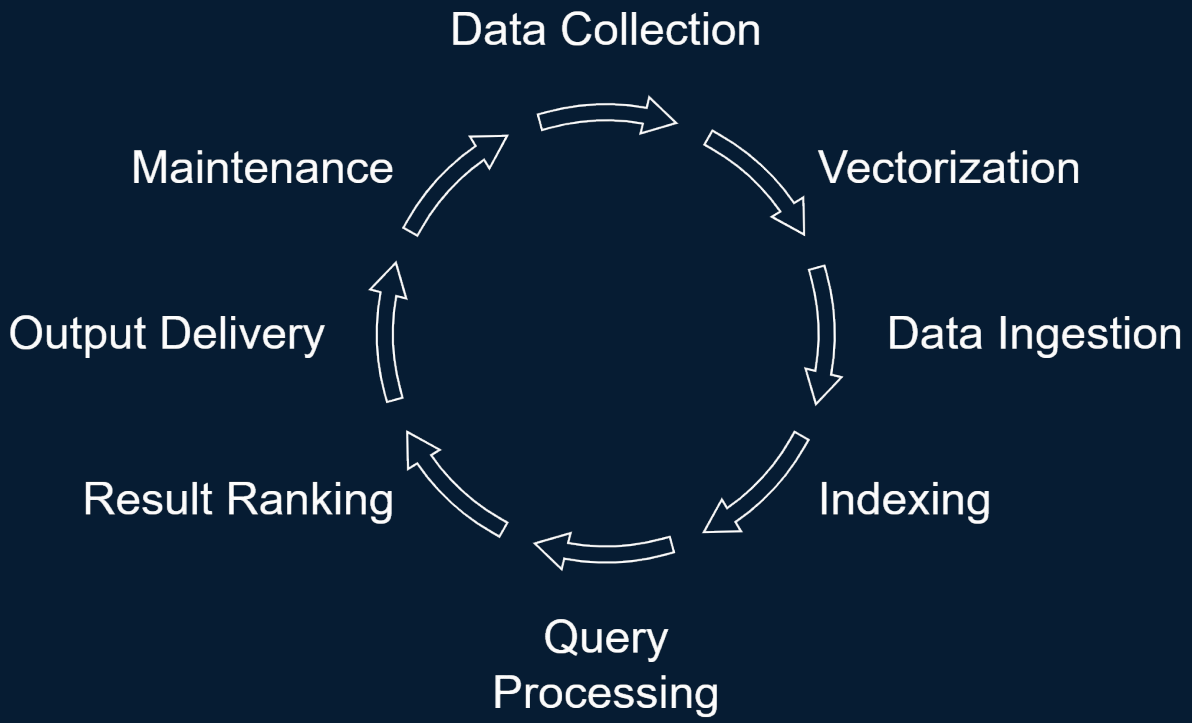
Different Embedding models





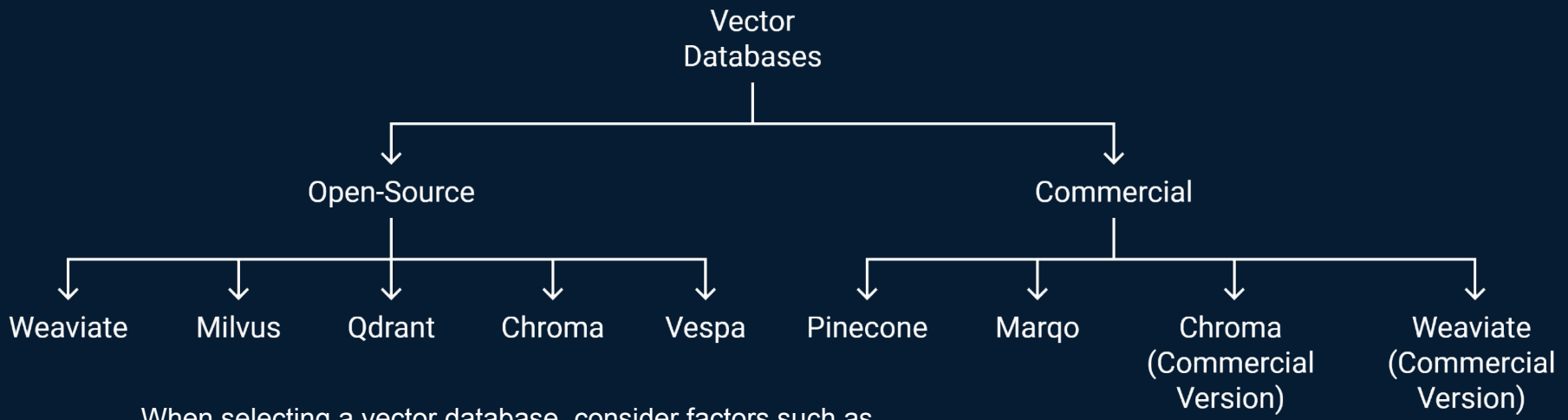
CHENNAI DATA CIRCLE

Vector Database



CHENNAI DATA CIRCLE

Different Vector Databases

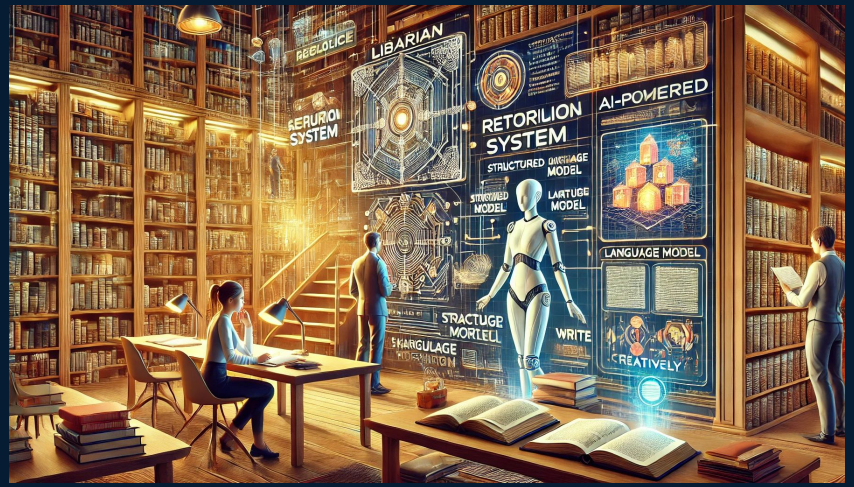


When selecting a vector database, consider factors such as scalability, performance, integration capabilities, community support, and licensing to ensure it aligns with your project's requirements.

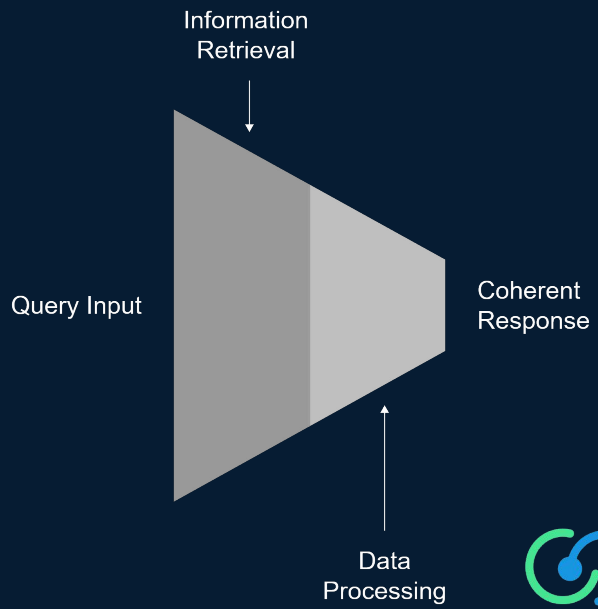
Understanding RAG

Retrieval Augmented Generation - enhances the capabilities of large language models (LLMs) by integrating external knowledge sources into the generation process.

Imagine a knowledgeable librarian (retrieval system) assisting a brilliant author (LLM) to craft precise and informed content.



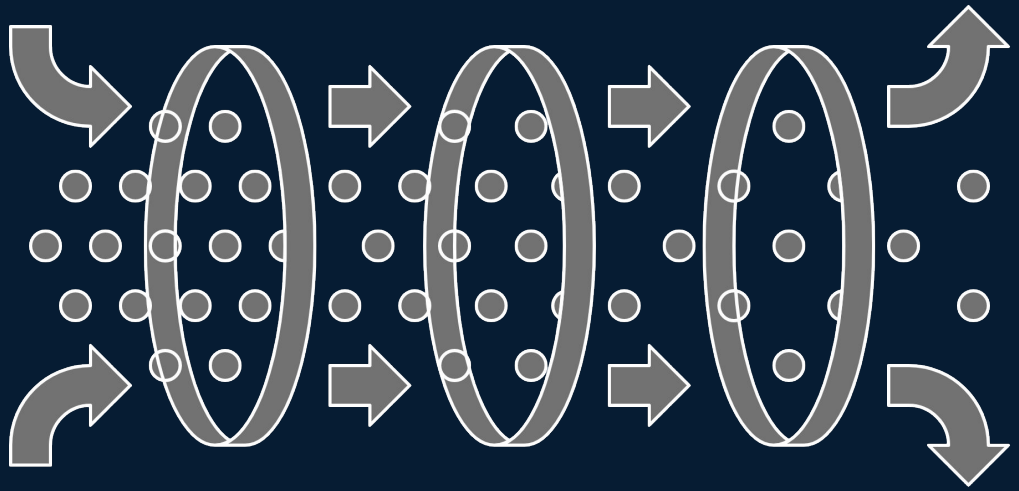
RAG Process Flow



CHENNAI DATA CIRCLE

Understanding RAG

Transforming Queries into Informed Responses



Retrieval

Searching external knowledge bases for information

Augmentation

Combining retrieved data with the query

Generation

Producing a coherent and informed response

CHENNAI DATA CIRCLE

Understanding RAG

Benefits of RAG:

- **Up-to-Date Information:** By accessing external sources, RAG ensures that responses are based on the most current data available.

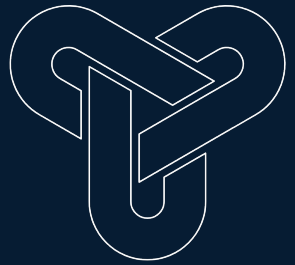
- **Reduced Hallucinations:** Grounding responses in real data minimizes the chances of generating incorrect or nonsensical information.

- **Domain Adaptability:** RAG allows models to tap into specialized knowledge bases, making them adaptable to various domains without extensive retraining.

RAG Implementation

Overview of Knowledge Base Sources

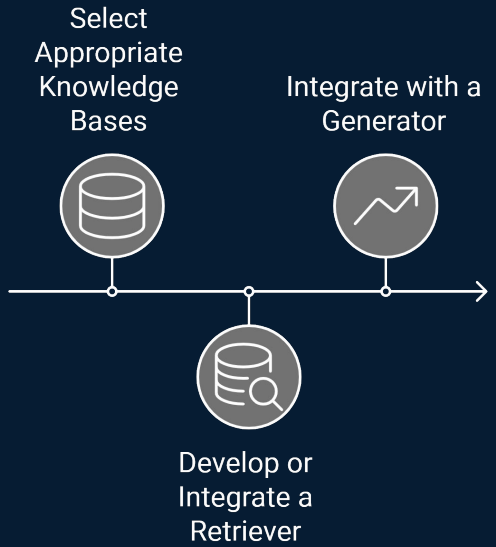
Data Marketplaces
Comprises Kaggle Datasets and AWS Data Exchange for diverse data offerings



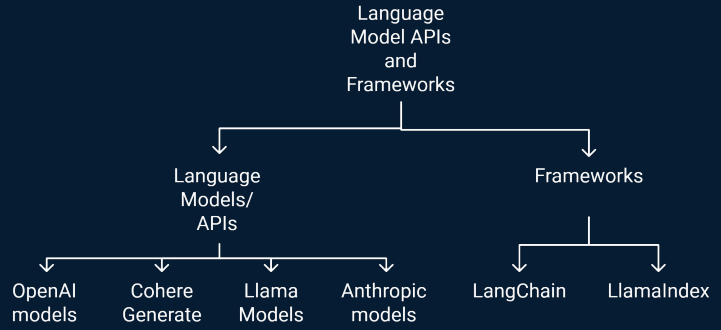
Public Datasets
Includes Common Crawl and Wikipedia Dumps for general data access

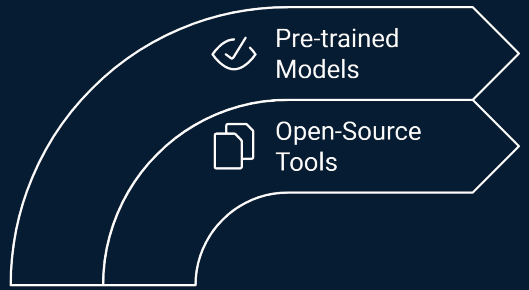
Domain-Specific Databases

Features PubMed and ArXiv for specialized academic and scientific literature



Components of Information Retrieval





Models like DPR and BM25 for dense retrieval tasks and traditional information retrieval

Tools like Haystack and FAISS for building search systems and similarity search

CHENNAI DATA CIRCLE

Tech Stack

The DocIQ App leverages a combination of cutting-edge technologies to deliver a seamless and efficient user experience. Here's a breakdown of the technologies and tools used:

LangChain: Utilized as the orchestration framework to manage the flow between different components, including embeddings creation, vector storage, and chatbot interactions.

Unstructured: Employed for robust PDF processing, enabling the extraction and preprocessing of text from uploaded PDF documents.

BGE Embeddings from HuggingFace: Used to generate high-quality embeddings for the processed documents, facilitating effective semantic search and retrieval.

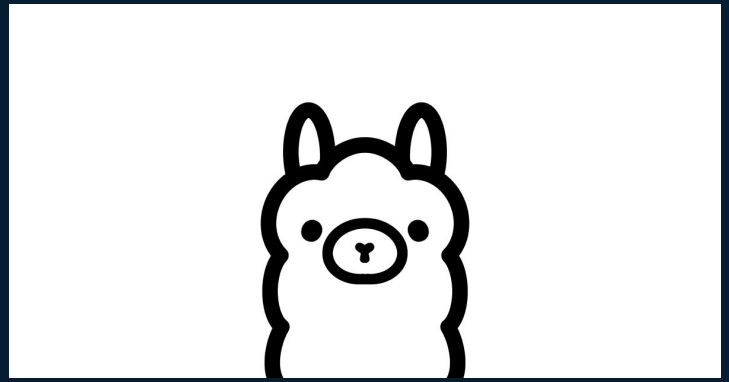
Qdrant: A vector database running locally via Docker, responsible for storing and managing the generated embeddings for fast and scalable retrieval.

LLaMA 3.2 via Ollama: Integrated as the local language model to power the chatbot, providing intelligent and context-aware responses based on the document embeddings.

Streamlit: The core framework for building the interactive web application, offering an intuitive interface for users to upload documents, create embeddings, and interact with the chatbot.

Ollama

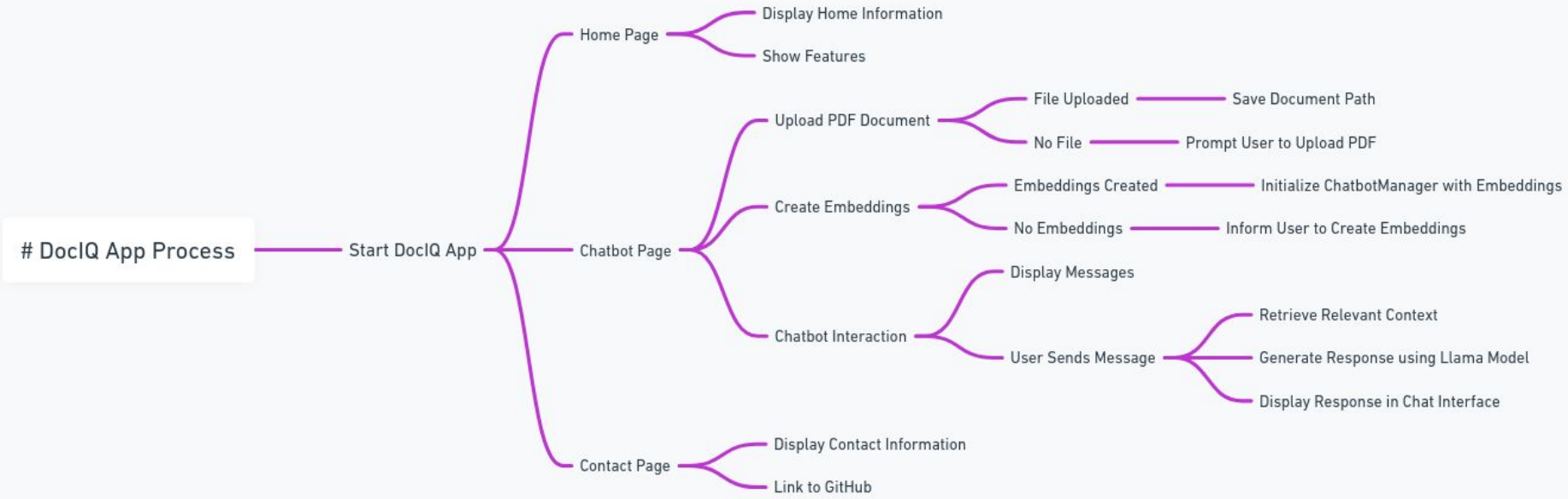
```
(venv) PS [redacted]\venv> ollama pull llama3.2:3b
pulling manifest
pulling dde5aa3fc5ff... 100% ▕████████████████████▏ 2.0 GB
pulling 966de95ca8a6... 100% ▕████████████████████▏ 1.4 KB
pulling fcc5a6bec9da... 100% ▕████████████████████▏ 7.7 KB
pulling a70ff7e570d9... 100% ▕████████████████████▏ 6.0 KB
pulling 56bb8bd477a5... 100% ▕████████████████████▏  96 B
pulling 34bb5ab01051... 100% ▕████████████████████▏ 561 B
verifying sha256 digest
writing manifest
success
```



Ollama in action

```
llm_load_print_meta: arch             = llama
llm_load_print_meta: vocab type       = BPE
llm_load_print_meta: n_vocab          = 128256
llm_load_print_meta: n_merges         = 280147
llm_load_print_meta: vocab_only       = 0
llm_load_print_meta: n_ctx_train      = 131072
llm_load_print_meta: n_embd           = 3072
llm_load_print_meta: n_layer          = 28
llm_load_print_meta: n_head           = 24
llm_load_print_meta: n_head_kv        = 8
llm_load_print_meta: n_rot            = 128
llm_load_print_meta: n_swa            = 0
llm_load_print_meta: n_embd_head_k    = 128
llm_load_print_meta: n_embd_head_v    = 128
llm_load_print_meta: n_gqa            = 3
llm_load_print_meta: n_embd_k_gqa     = 1024
llm_load_print_meta: n_embd_v_gqa     = 1024
llm_load_print_meta: f_norm_eps       = 0.0e+00
llm_load_print_meta: f_norm_rms_eps   = 1.0e-05
llm_load_print_meta: f_clamp_kqv      = 0.0e+00
llm_load_print_meta: f_max_alibi_bias = 0.0e+00
llm_load_print_meta: f_logit_scale    = 0.0e+00
llm_load_print_meta: n_ff             = 8192
llm_load_print_meta: n_expert         = 0
llm_load_print_meta: n_expert_used    = 0
llm_load_print_meta: causal attn      = 1
llm_load_print_meta: pooling type     = 0
llama_new_context_with_model: n_batch    = 2048
llama_new_context_with_model: n_ubatch   = 512
llama_new_context_with_model: flash_attn = 0
llama_new_context_with_model: freq_base  = 500000.0
llama_new_context_with_model: freq_scale = 1
llama_kv_cache_init:        CPU KV buffer size =   896.00 MiB
llama_new_context_with_model: KV self size  =  896.00 MiB, K (f16):  448.00 MiB, V (f16):  448.00 MiB
llama_new_context_with_model:        CPU  output buffer size =     2.00 MiB
llama_new_context_with_model:        CPU compute buffer size =   424.01 MiB
llama_new_context_with_model: graph nodes  = 902
llama_new_context_with_model: graph splits = 1
time=2024-11-10T07:20:06.692+05:30 level=INFO source=server.go:601 msg="llama runner started in 5.54 seconds"
[GIN] 2024/11/10 - 07:22:43 | 200 |         2m42s |       127.0.0.1 | POST     "/api/chat"
```

Recap


DocIQ App Process
Start DocIQ App
Home Page
Display Home Information
Show Features
Chatbot Page
Upload PDF Document
File Uploaded
Save Document Path
No File
Prompt User to Upload PDF
Create Embeddings
Embeddings Created
Initialize ChatbotManager with Embeddings
No Embeddings
Inform User to Create Embeddings
Chatbot Interaction
Display Messages
User Sends Message
Retrieve Relevant Context
Generate Response using Llama Model
Display Response in Chat Interface
Contact Page
Display Contact Information
Link to GitHub

Questions?

Mail to: arthirajendran24@gmail.com
Linkedin:
https://www.linkedin.com/in/arthirajendran/

CHENNAI
DATA CIRCLE