Introduction to RAG

LLM → Large Language Models → Deep Learning Models

↓

Weights + Biases

{ Transformers } Encoder - Decoder Architecture

BERT, T5, BART

GPT → Decoder

BERT → Encoder

LLMS

NLP

Upstream :- Pretraining (Masking)

Downstream :- Post Training

↓

Fine Tuning

Text Classification, Summarization, POS, NER

QA, MT, Text Generation

LLM Problems

Base Model, Foundational Model

+

{ Custom Data }

↓

Fine Tuning Task

1) Knowledge Cutoffs
   ↳ Cutoff Training Date

2) Lack of Info in Domain Specific Task — Wrong Answer

3) Lack of Private Data

4) Loses Trusted Source

5) Probabilistic Output Generation.

SLM

Small Language Model

{ RAG } vs { LLM }

Training

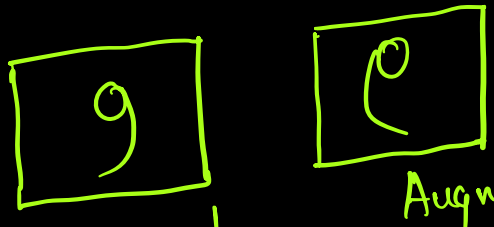
{ Computationally Expensive } { limited }

GPU

Basics of RAG

Retrieval Augmented Generation → LLM

(Original → Modification)



ORiginal Augmented

My name is Paul.

Paul is my name.

99% Context + Modification

6 9

Data → Vector DB → * Sources Original

{ Context Retained }

↓

LLM

↓

* Answer

RAG Simple Architecture

1) Data Ingestion (External knowledge Sources)

Text → PDF, HTML, JSON, CSV, XLX DOCX, PPTX

MultiModal
- Image → GIFS, JPG, PNG, TIF
- Audio → .stt, .mpv, .acc
- Videos → .mp4, .mov

2) Vector Databases

Number Representation

My name is Paul.

0.1 0.8 0.9 0.75

[0.1, 0.8, 0.9, 0.75] →

Algorithm :- Embeddings

1) Bag of Words

2) TF IDF

3) Concurrency Matrix

4) Word 2 Vec < CBOW / Skipgram

5) (Positional Encoding) *

6) One Hot Encodding

7) ELMO

1) Text Data / Multimodal

↓

Numbers (Embeddings)

(Model) → MTEB

1) Open Source → Hugging face — gemini-embedding model / all mini lm 6

2) Paid → Nomic / Titan

Algorithm :- Text / Image

↓

{ Numeric Representation }

Vector DB → Embeddings

↙ Open Source → FAISS, ChromaDB

↘ Paid → Qdrant, Weaviate, Zilliz

3) Retrieval

Neural Retriever — DPR → 2 Encoders

Dense Passage Retrieval

Bi Encoders

Question Encoder → q-embed

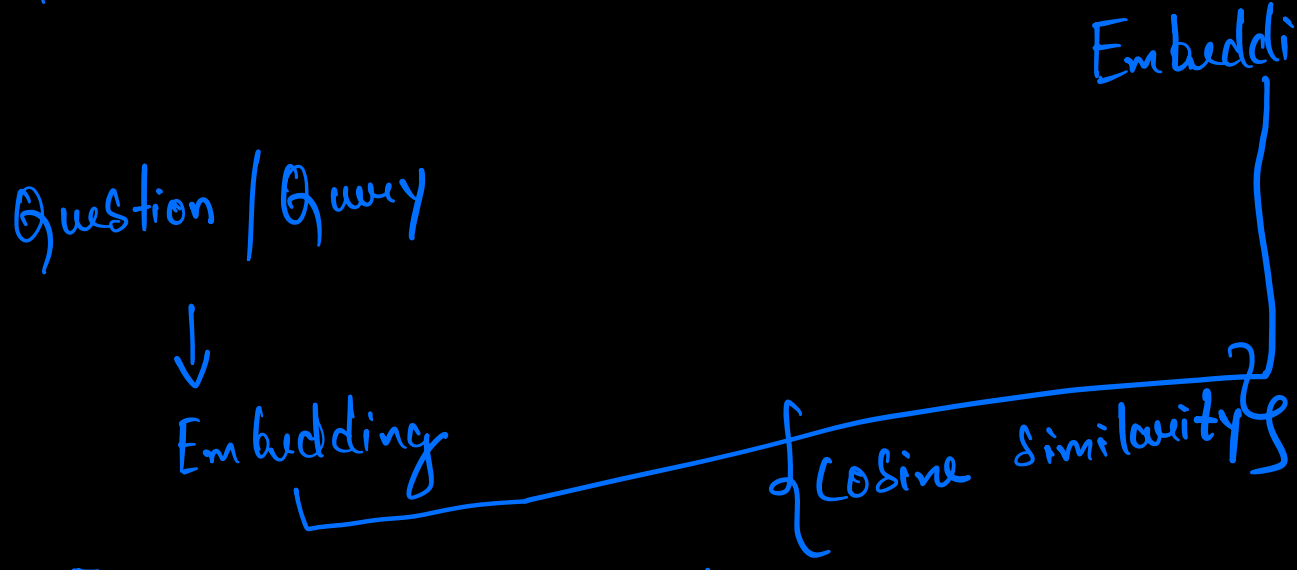
Passage Encoder → p-embed

q-embed, p-embed → Comparing

Comparing:
- Cosine Similarity
- Dot Similarity

* Semantic Similarity

Retrieval :- Fetch relevant data from VectorDB



E_1

↓

Semantic Similarity Result

E_1 → V_{11} → Data Sources (0.4)
V_{12} (0.6)
V_{13} (0.8)

4) Augmentation

1) User Query
2) Retrieved Docs
3) Prompt

5) Generation

LLM -> Text Generation

↳ Context (3 parts)

G, R, P

Phase 1 :- Data Ingestion

1) Docs, (www)

↓

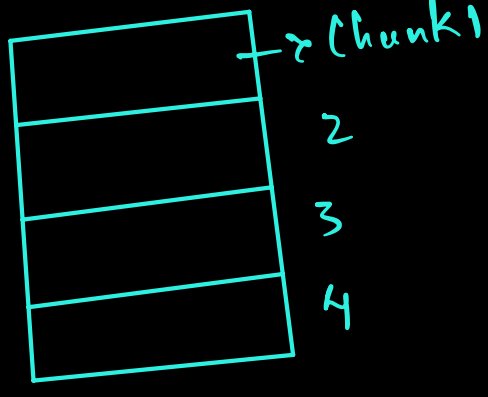
Chunks

↓

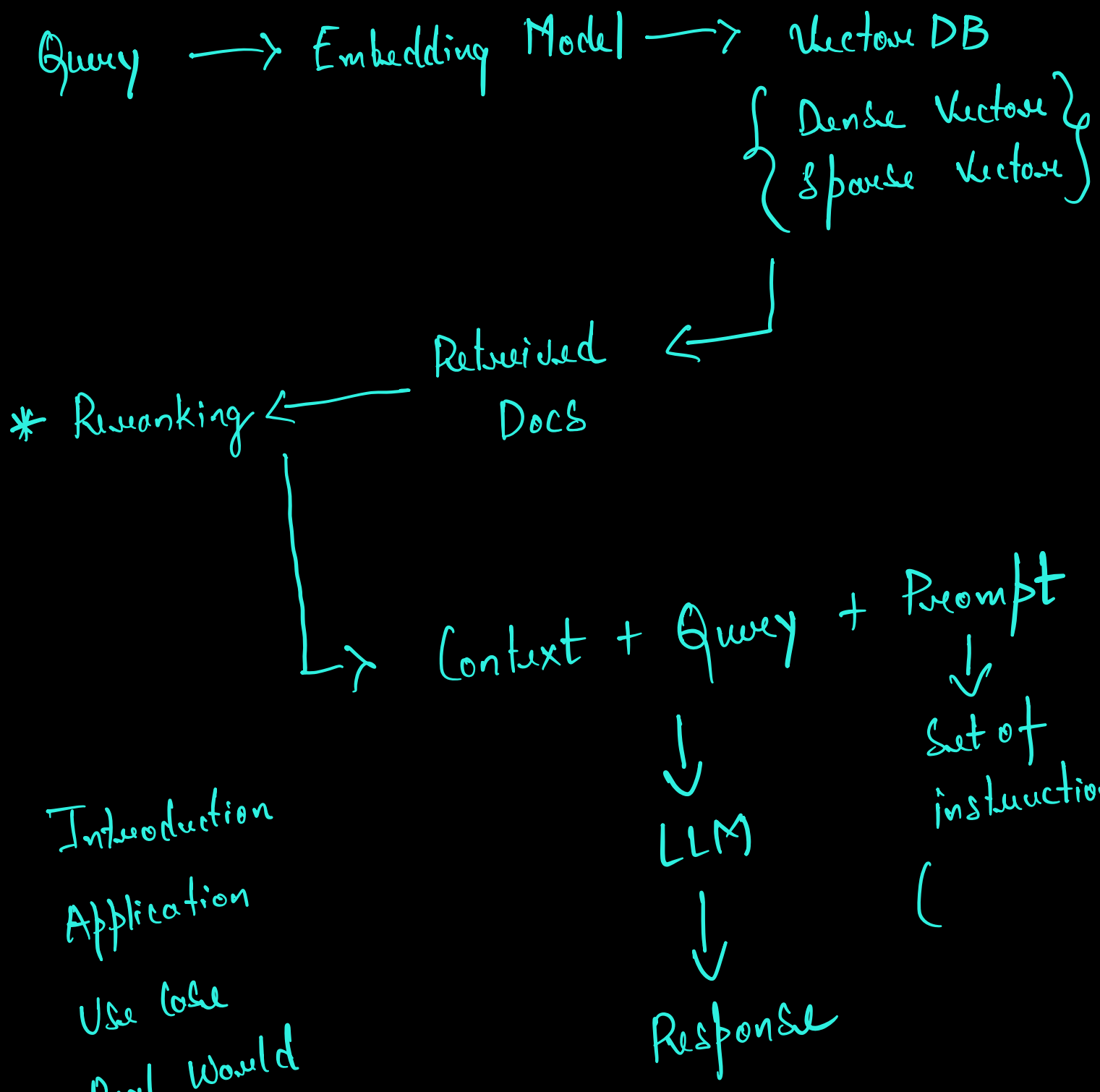
Embedding Model

↓

Vector DB



Phase 2 :- Retrieval



Introduction
Application
Use case
Real World
Conclusion

1) Extractive :- Exact same piece of info from data

2) Abstractive :- Modified info but with same context. like Rephrasing.

Parametric — Pretrained Seq2Seq (BART) that stores knowledge in weights (400M Parameters)

Training Data → Seq2Seq (BART)

↓

Final Model (400M Parameters)

↓

Final Model → NLP Tasks

Parametric Memory

Non Parametric Memory

Dense Vector Index of any type of external data sources (Wikipedia)

Accessed :- Neural Retriever
DPR

2 RAG Recipes

1) RAG Sequence :- Uses the same retrieved docs to generate the entire output sequence

2) RAG Token :- We can use different documents to generate the final output sequence.