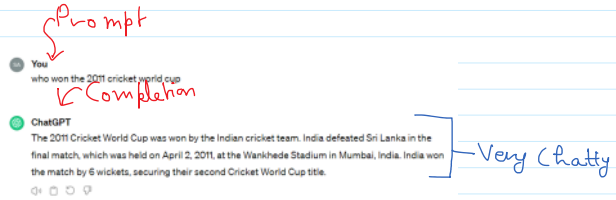
Prompt Engineering

28 March 2024 11:55 PM

Prompt ⇒ The input that you send into a generative model (text, images, videos etc)

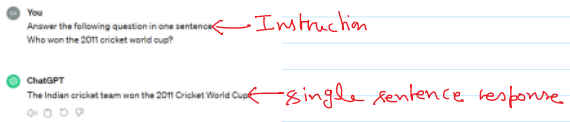
Completion ⇒ Output

Example :

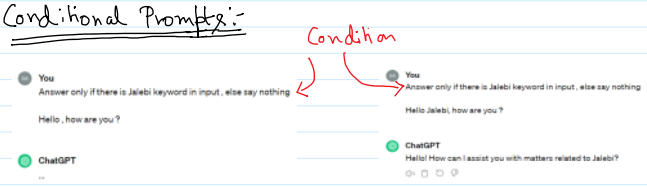


Prompt Structure :-

Instruction ⇒

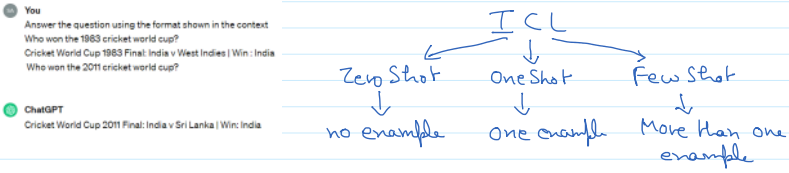


Conditional Prompts :-



In-Context Learning :-

Allows model to perform task with a minimal amount of training examples.

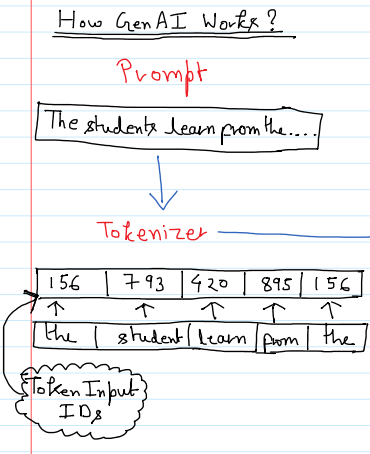


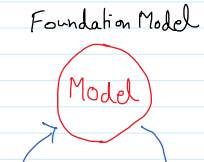
Tokens :-

The smallest unit into which text data can be broken down for all AI model to process.

OpenAI Tokenizer

How GenAI Works ?





Content Window → Detokenizer

Max Amount of tokens, model consider while generating response

In practical terms, the content window limits how much previous dialogue the model can remember during an interaction.

This includes both prompt provided by user & model's generated text. → Short Term Memory

If the interaction exceeds the content window, the model loses access to the earliest part of the conversation.

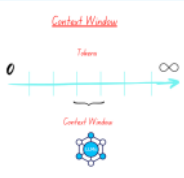
☆ Max Token :-

Total no. of tokens Generative AI can generate in response.

↑ A very basic mechanism to keep model response short.

V.V.I. ⇒ Performance Impact ↓

Tokens ↑ Inference Time ↑



If the interaction exceeds the context window, the model loses access to the earliest part of the conversation.

It is a barrier to long conversation/complex task

Greedy Vs. Random Sampling :-

Prompt → Model → Probability Distribution across all tokens in the model's known vocabulary

↓

Choose a single token

Greedy → Choose next token with heighest probability

The sky is ___

Probability	Token
0.3	blue
0.4	limitless
0.2	Clear
⋮	⋮

Random

The sky is ___

Probability	Token
0.3	blue
0.4	limitless
0.2	Clear
⋮	⋮

Top P & Top K Random Sampling :-

Top K → Choose token randomly from only top k tokens with heighest probability

Probability	Token
0.3	blue
0.4	limitless
0.2	Clear
⋮	⋮

→

Probability	Token
0.4	limitless
0.3	blue
0.2	Clear
⋮	⋮

} k = 2

Top K = 1 ⇒ Greedy Sampling

Top P → Randomly sampling from the set of tokens whose cumulative probability do not exceed p, starting from heighest probability and working down to the lowest.

p = 0.7

Probability	Token
0.3	blue
0.4	limitless
0.2	Clear
⋮	⋮

→

Probability	Token
0.4	limitless
0.3	blue
0.2	Clear
⋮	⋮

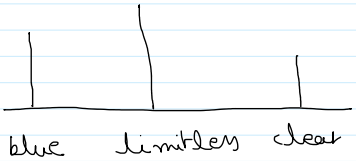
} p = 0.7

Temperature :-

top-p / top-K → affects next token prediction after probability distribution is generated

temperature → changes next-token probability distribution → ultimately affect next token prediction.

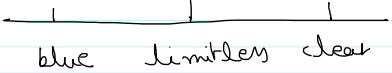
Probability	Token
0.3	blue
0.4	limitless
0.2	Clear
⋮	⋮



Temp ↑ Randomness ↑

Probability	Token
0.01	blue
0.5	limitless
0.02	Clear
⋮	⋮

High Probability Token



Temp ↓ Randomness ↓

low temp ⇒ Strong peaks

Low Temp, Low Top-p ⇒ highly focused on narrow range of high probability token.

Probability	Token
0.02	blue
0.5	limitless
0.01	Clear
⋮	⋮

Sorting →

Probability	Token
0.5	limitless
0.02	blue
0.01	Clear
⋮	⋮

High Temp, High Top-p ⇒ very high randomness.

temp=2, p=1

Probability	Token
0.3	blue
0.4	limitless
0.2	Clear
⋮	⋮

Sorting →

Probability	Token
0.4	limitless
0.3	blue
0.2	Clear
⋮	⋮

p = 1

Low Temp, High Top-p ✓

Probability	Token
0.02	blue
0.5	limitless
0.01	Clear
⋮	⋮

→

Probability	Token
0.5	limitless
0.02	blue
0.01	Clear
⋮	⋮

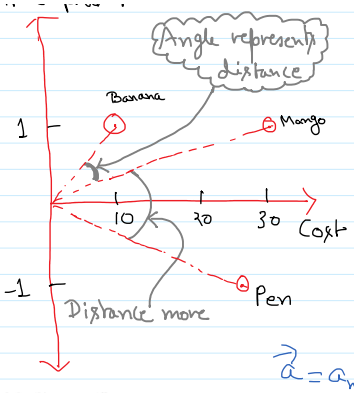
high p

Embedding Vectors :-

Mango → Is it a fruit? 1 | Cost? 30 → [1, 30]

Banana → Is it a fruit? 1 | Cost? 10 → [1, 10]

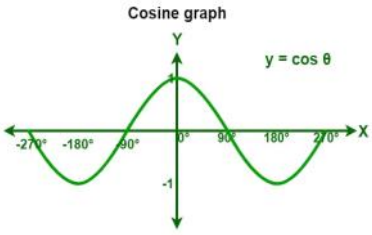
Pen → Is it a fruit? -1 | Cost? 25 → [-1, 25]



$\vec{a} \cdot \vec{b} = |a||b| \cos\theta$ ← Mother Equation

$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|}$

Embedding are numerical, vectorized representation of any type including text, images, audio clips, videos etc.



$\vec{a} = a_x\hat{i} + a_y\hat{j} + a_z\hat{k}$

$\vec{b} = b_x\hat{i} + b_y\hat{j} + b_z\hat{k}$

$\vec{a} \cdot \vec{b} = a_x b_x + a_y b_y + a_z b_z$

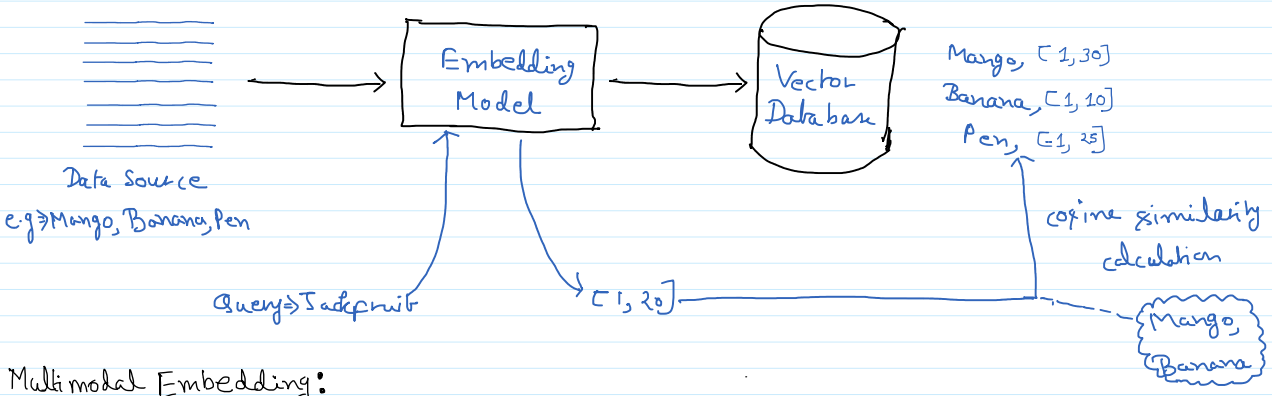
$|\vec{a}| = \sqrt{a_x^2 + a_y^2 + a_z^2}$

$|\vec{b}| = \sqrt{b_x^2 + b_y^2 + b_z^2}$

$\cos\theta = \frac{a_x b_x + a_y b_y + a_z b_z}{\left(\sqrt{a_x^2 + a_y^2 + a_z^2}\right)\left(\sqrt{b_x^2 + b_y^2 + b_z^2}\right)}$

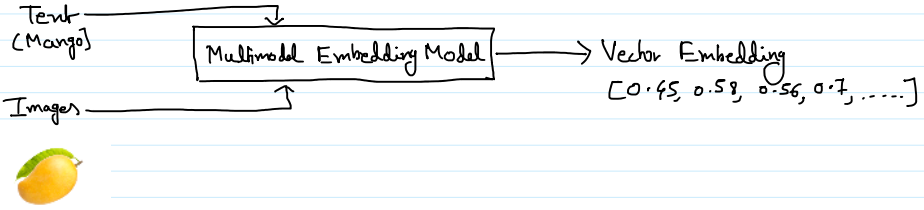
Semantic Search:-

Semantic Search
- Traditional keyword matching based output ✗
- Return output based on meaning & context behind user queries. ✓

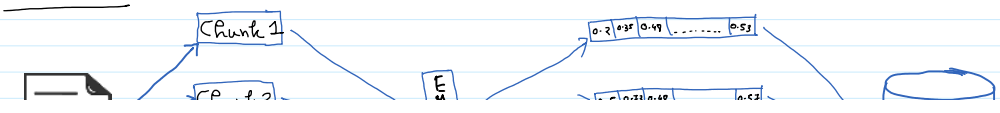


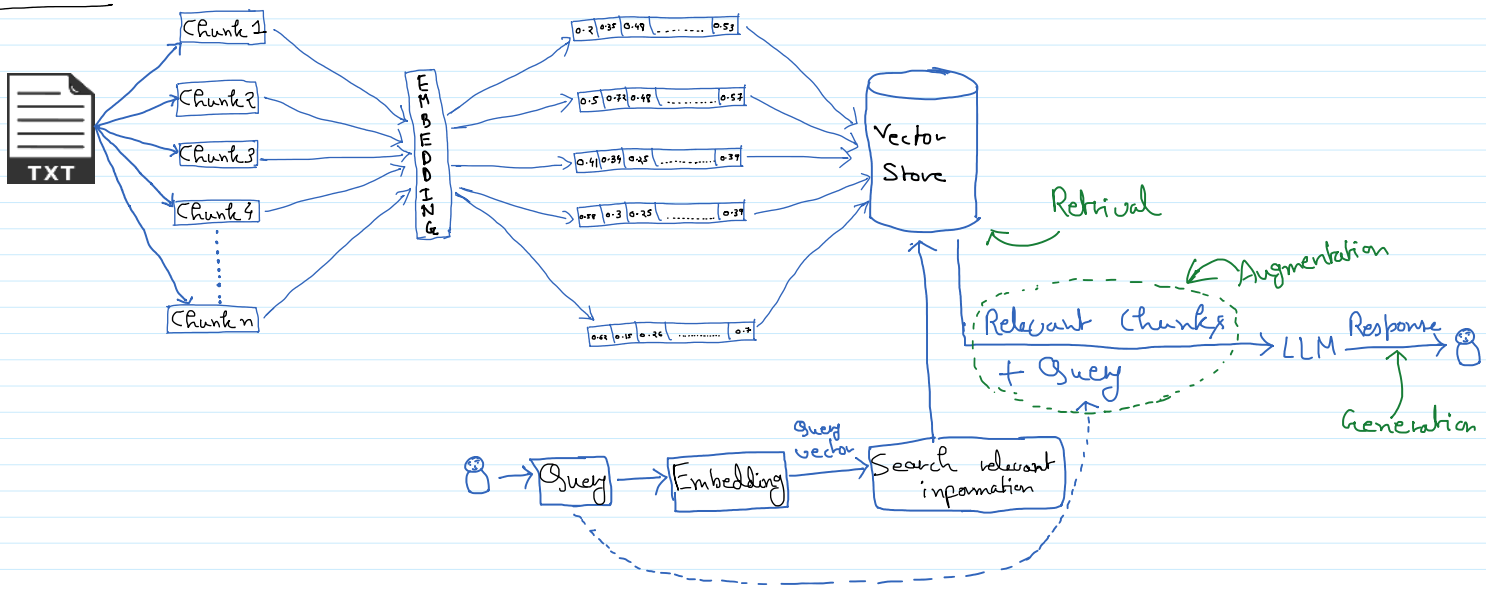
Multimodal Embedding:

↳ Integration of Text, Images & various other datatypes in single vector space.



RAG:-





RAG → Combination of 2 memories

- model's own prior knowledge
- A search engine