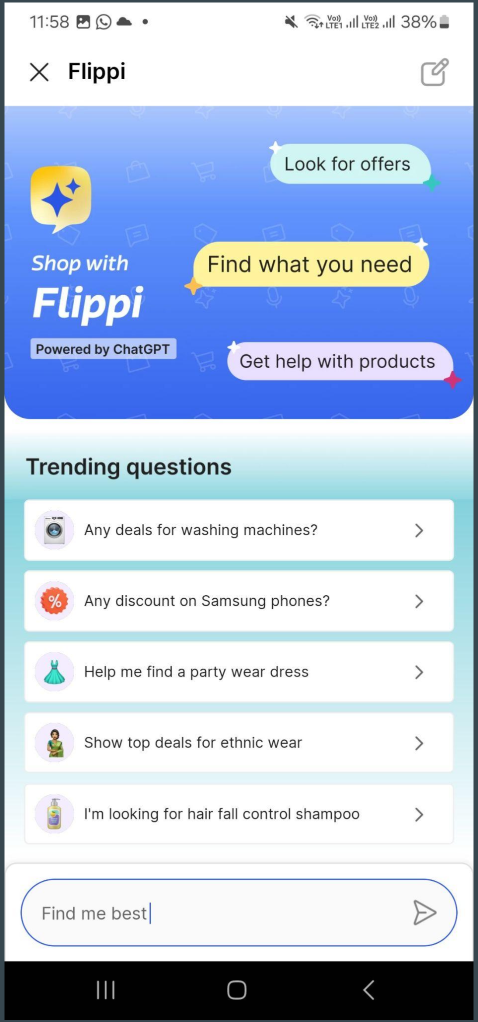
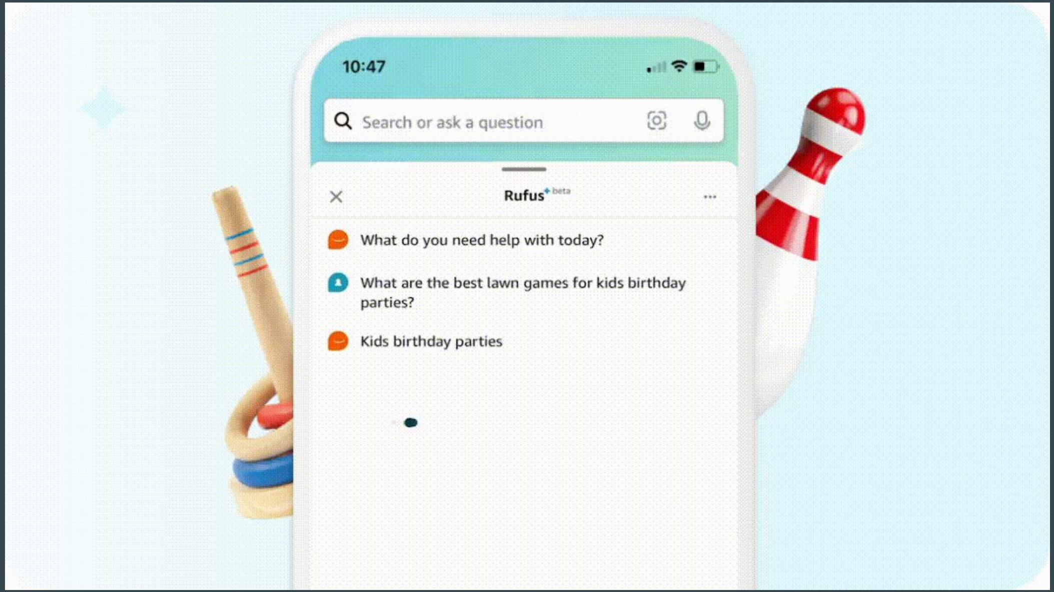
Intuitive Product Discovery with Conversational AI in E-commerce

- V G Samvardhan

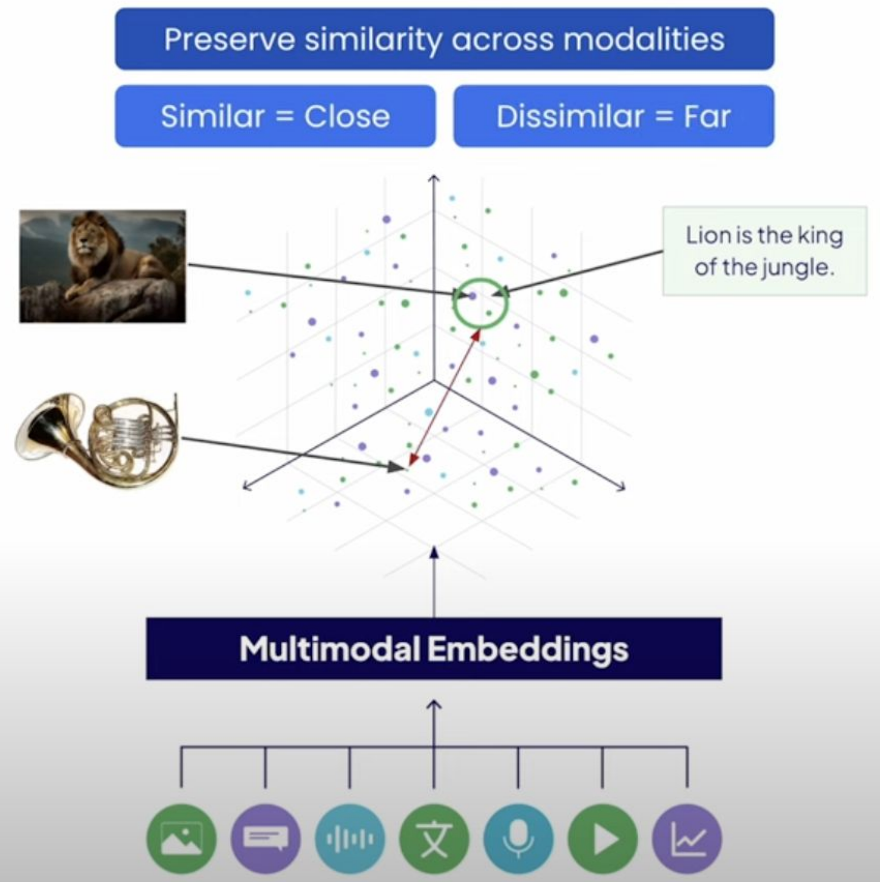
Breakdown of the Topic


11:58
38%
Flippi
Look for offers
Shop with
Flippi
Powered by ChatGPT
Find what you need
Get help with products
Trending questions
Any deals for washing machines?
Any discount on Samsung phones?
Help me find a party wear dress
Show top deals for ethnic wear
I'm looking for hair fall control shampoo
Find me best


10:47
Search or ask a question
Rufus beta
What do you need help with today?
What are the best lawn games for kids birthday parties?
Kids birthday parties

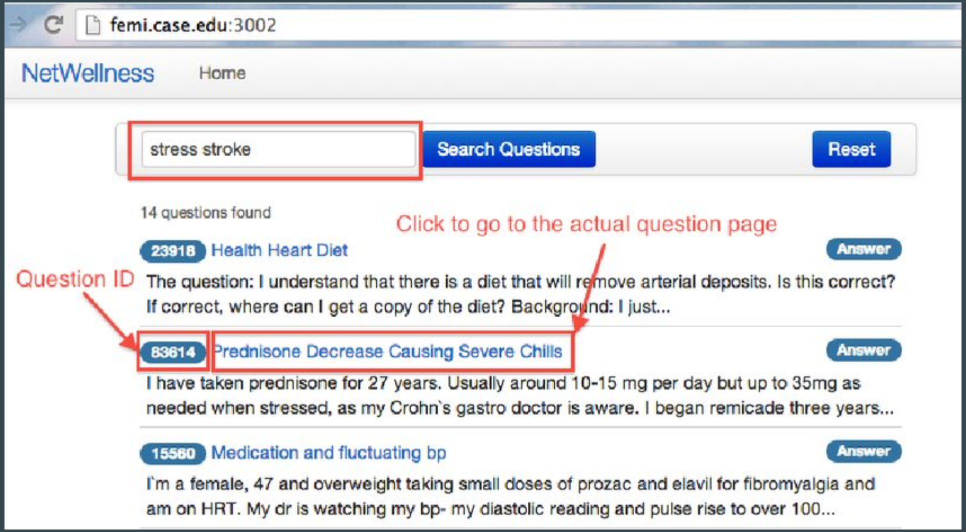
What is MultiModal ?

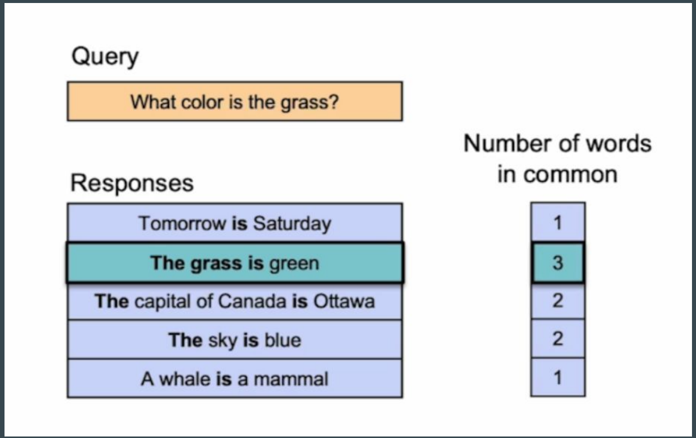
Multimodal Embedding Models



Types of Search

Keyword-based search

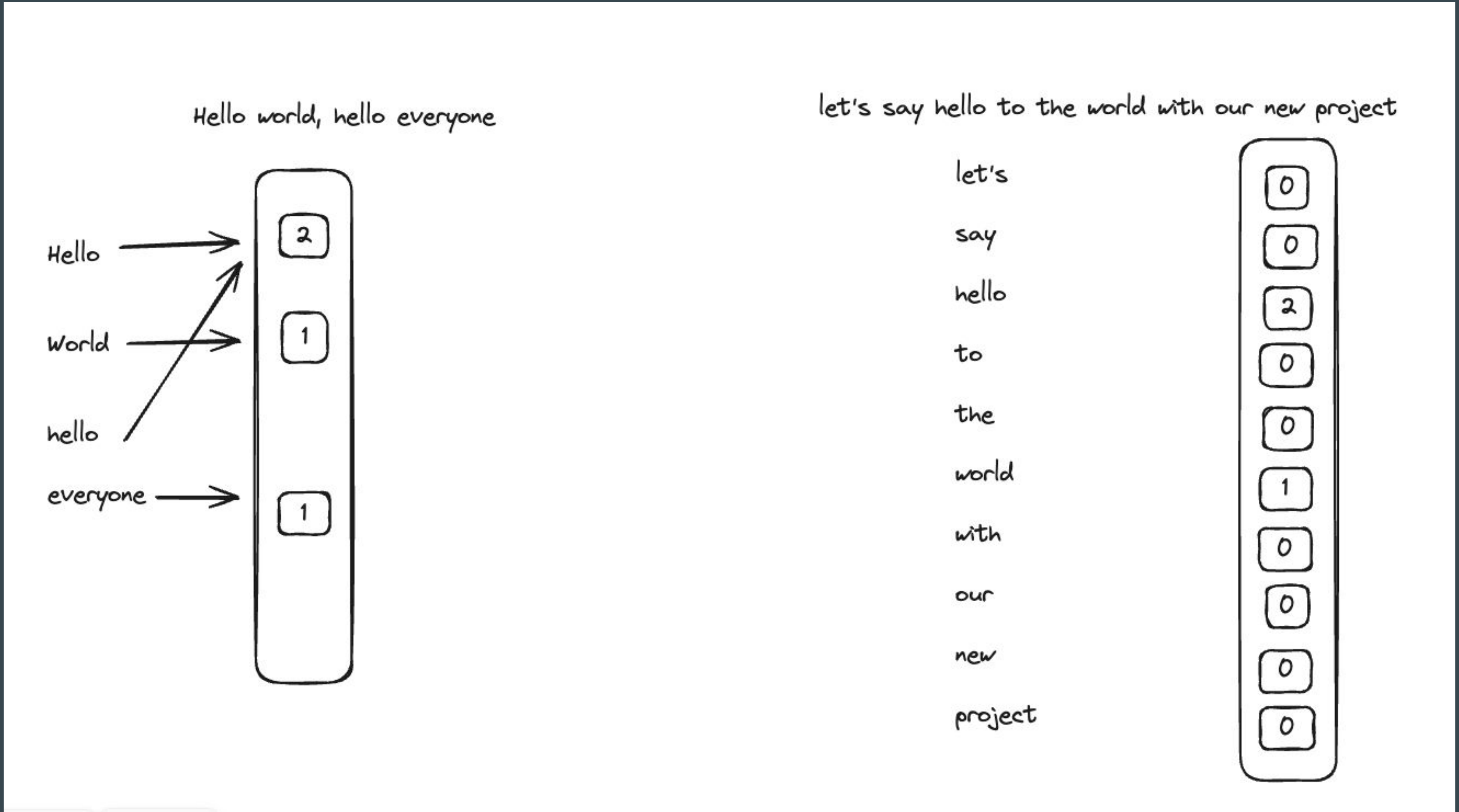




Sparse Vectors

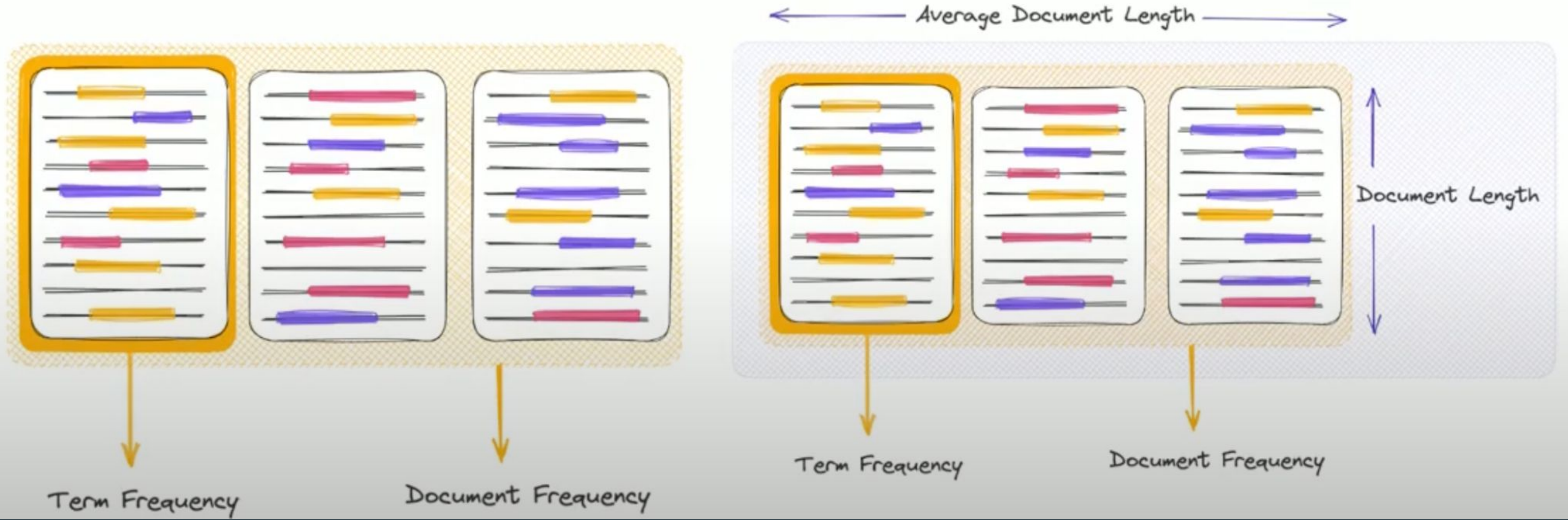
Keyword-based search in the context of hybrid search often uses a representation called sparse embeddings, which is why it is also referred to as sparse vector search. Sparse embeddings are vectors with mostly zero values with only a few non-zero values, as shown below.

Sparse Vectors



TF-IDF

BM-25



Average Document Length
Document Length
Term Frequency
Document Frequency
Term Frequency
Document Frequency

TF-IDF

Corpus D

d1 → Apple is good for health

d2 → Banana is good for eating

d3 → Apple and banana both are fruits

TF-IDF

Term Frequency - Inverse Document Frequency

Search: Apple

TF Calculation:

TF("Apple",d1)= 1/5 =0.2

TF("Apple",d2) 0/5 =0

TF("Apple",d3) = 1/6 = 0.167

IDF Calculation:

IDF("Apple")=log(3/2) = 0.4054651081081644

TF-IDF Calculation

TF-IDF("Apple",d1) = 0.405 * 0.2 = 0.081 (approx)

TF-IDF("Apple",d2) = 0.405 * 0 = 0

TF-IDF("Apple",d3) = 0.405 * 0.167 = 0.068

Result: The word apple has more relevant to Document 1

BM-25

BM-25 (Best Match)

Document Length Normalization and non-linear term frequency scaling (saturation)

Given a query Q, containing keywords $q_1, \ldots, q_n$, the BM25 score of a document D is:

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

where $f(q_i, D)$ is q_i's term frequency in the document D, $|D|$ is the length of the document D in words, and avgdl is the average document length in the text collection from which documents are drawn. k_1 and b are free parameters, usually chosen, in absence of an advanced optimization, as $k_1 \in [1.2, 2.0]$ and $b = 0.75$.[1] $\text{IDF}(q_i)$ is the IDF (inverse document frequency) weight of the query term q_i. It is usually computed as:

$$\text{IDF}(q_i) = \ln\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1\right)$$

where N is the total number of documents in the collection, and $n(q_i)$ is the number of documents containing q_i.

BM-25

BM-25 (Best Match)

$$IDF = \log \left[\frac{N - n(t) + 0.5}{n(t) + 0.5} + 1 \right]$$

IDF calcumation

Total Number of documents in the crpus =3,

n(Apple)=2

IDF(Apple) = log (((3-2+0.5)/(2+0.5))+1) = 0.47

TF Calculation

TF("Apple",d1)= 1/5 =0.2

TF("Apple",d2) 0/5 =0

TF("Apple",d3) = 1/6 = 0.167

BM-25 Calculation

k1 and b b are parameters that control the influence of TF and document length.
Common values are
k 1 =1.5 and b=0.75.

$$score(D,t) = IDF(t) \cdot \frac{TF(t,D) * (k1+1)}{TF(t,D) + k1 * \left[1 - b + b * \frac{|D|}{avgdl} \right]}$$

score(d1,Apple) = 0.470

score(d2,Apple) = 0

score(d3,Apple) = 0.470

The formula effectively balances the frequency of the term in the document against the frequency of the term in the entire corpus, adjusted for document length,

Key Takeaway

Memory Efficiency: They use less memory due to high proportion of zeros.

Interpretability: Dimensions often correspond directly to specific features, enhancing transparency.

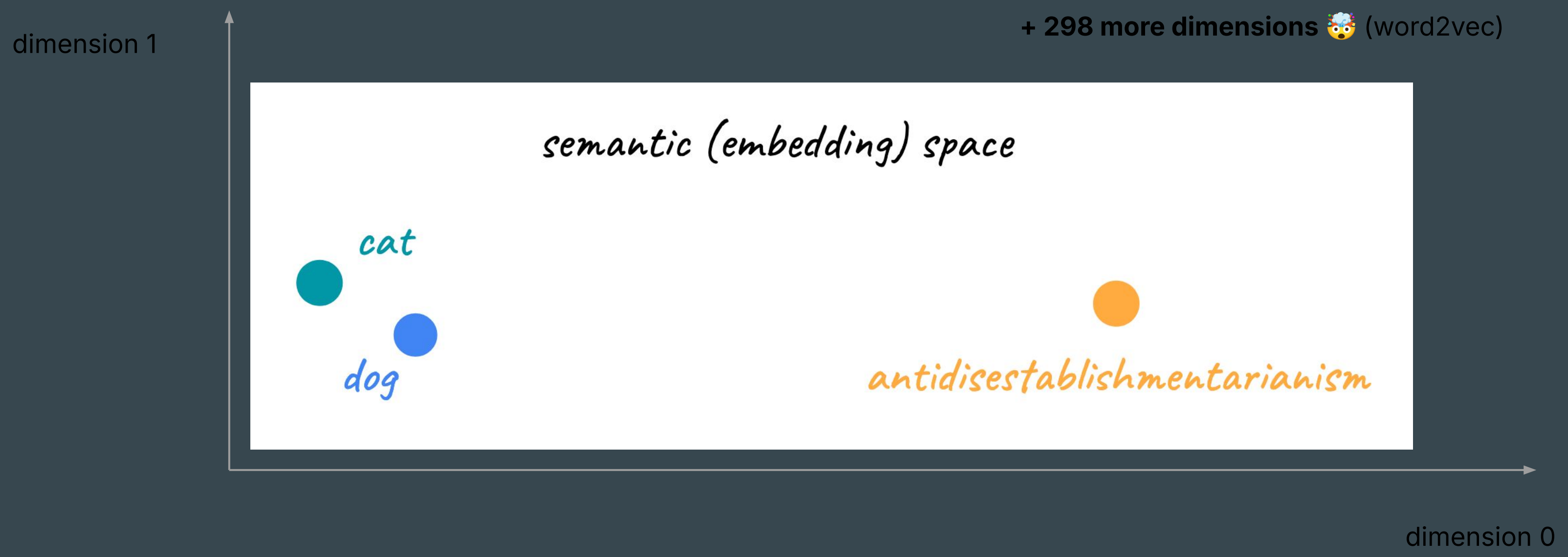
High Dimensionality: May result in computational challenges and inefficiencies.

Difficulty in Capturing Complexity: Sparse nature can make it hard to represent complex patterns effectively.

Embeddings

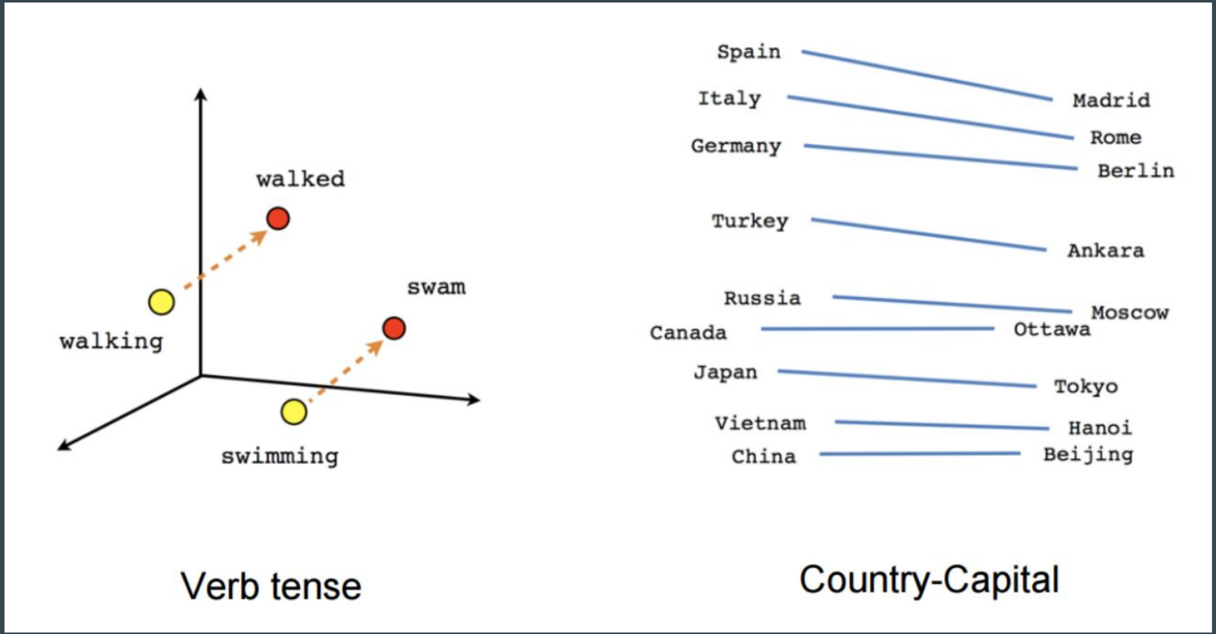
A list (vector) of numbers [0.5, 0.798, 0.03, ...]

Similar things are close together

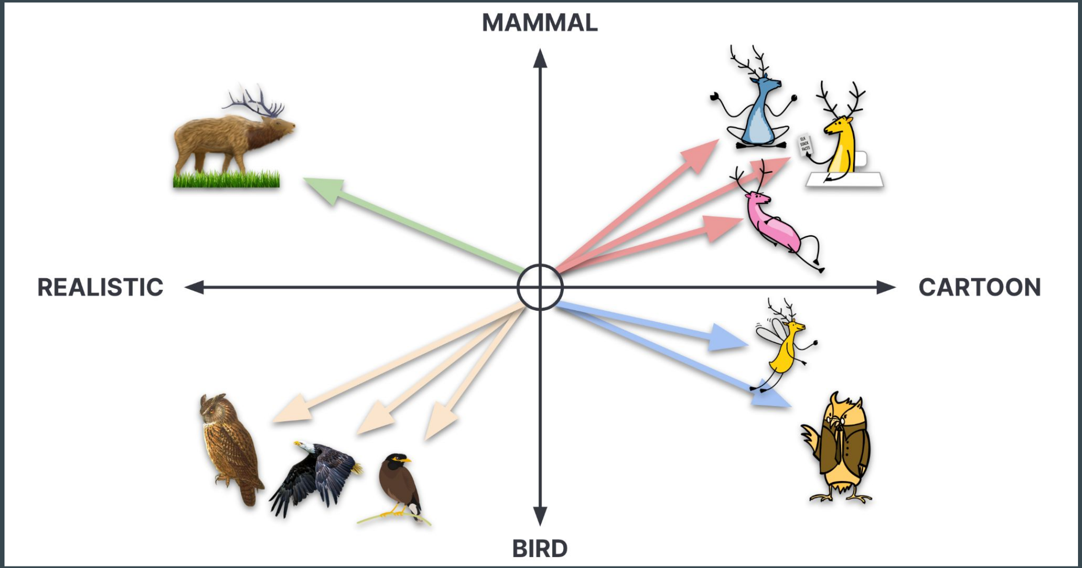


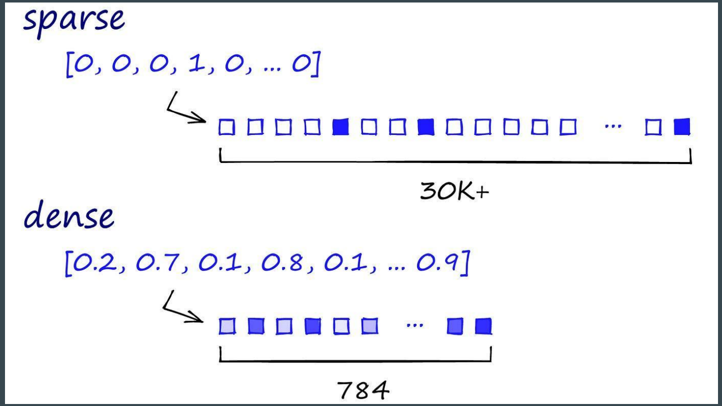
Semantic algebra with embeddings

Kinda handy, like we might do with vectors



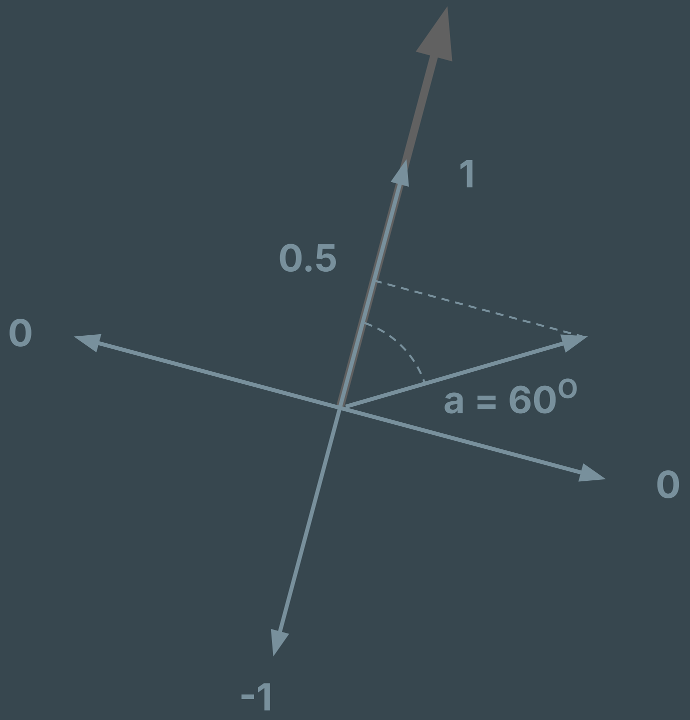
Semantic search





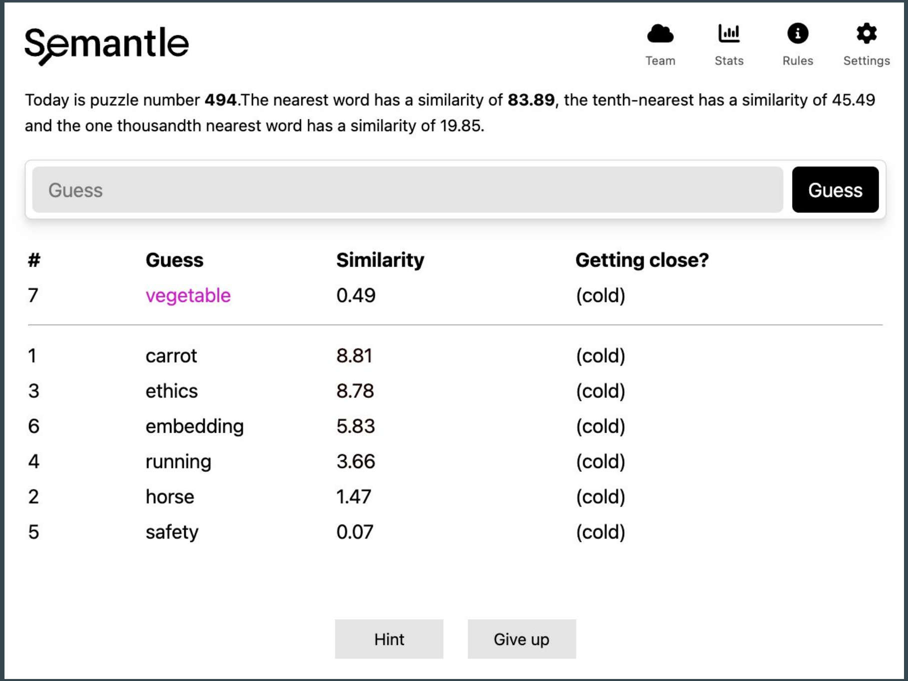
Dense Vectors

Measuring similarity with cosine

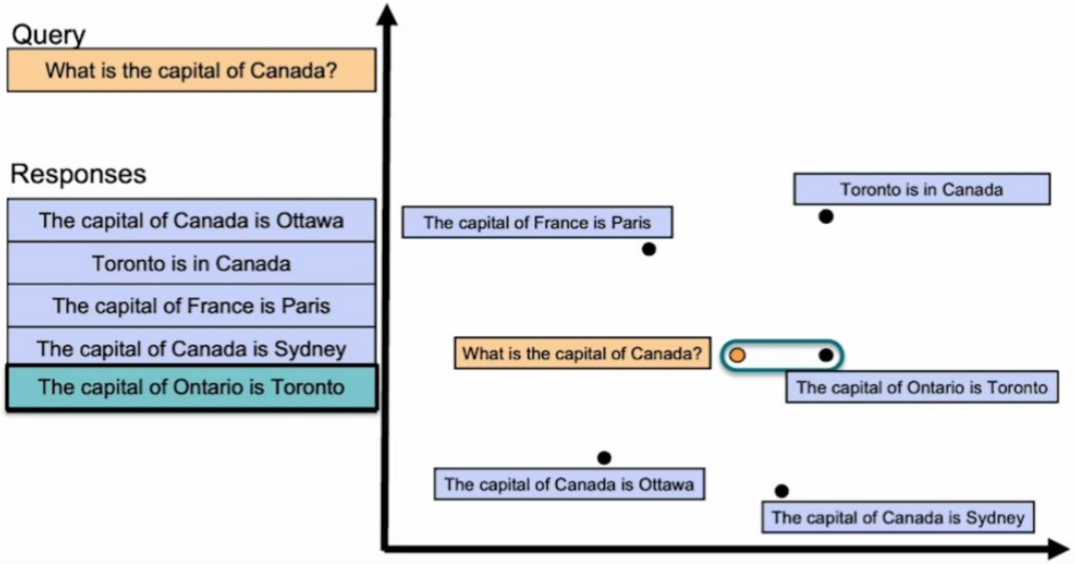


Cosine similarity puns are sticky

Semantle - a game of semantic

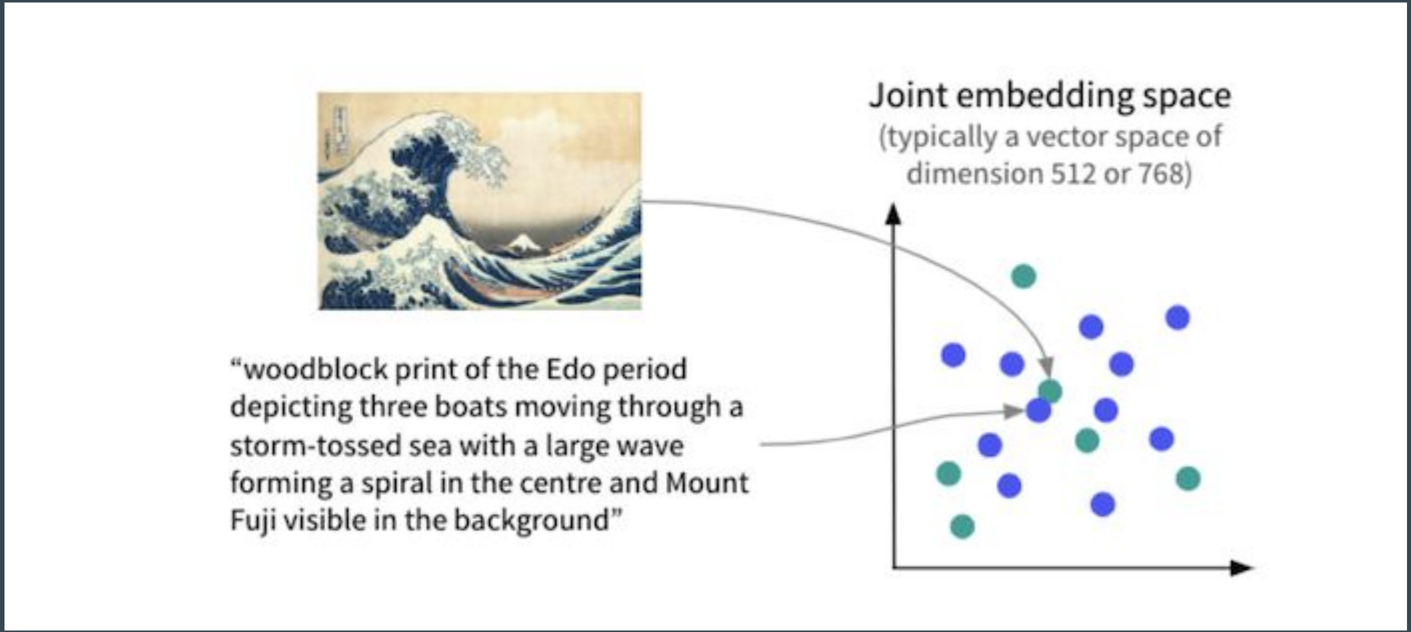


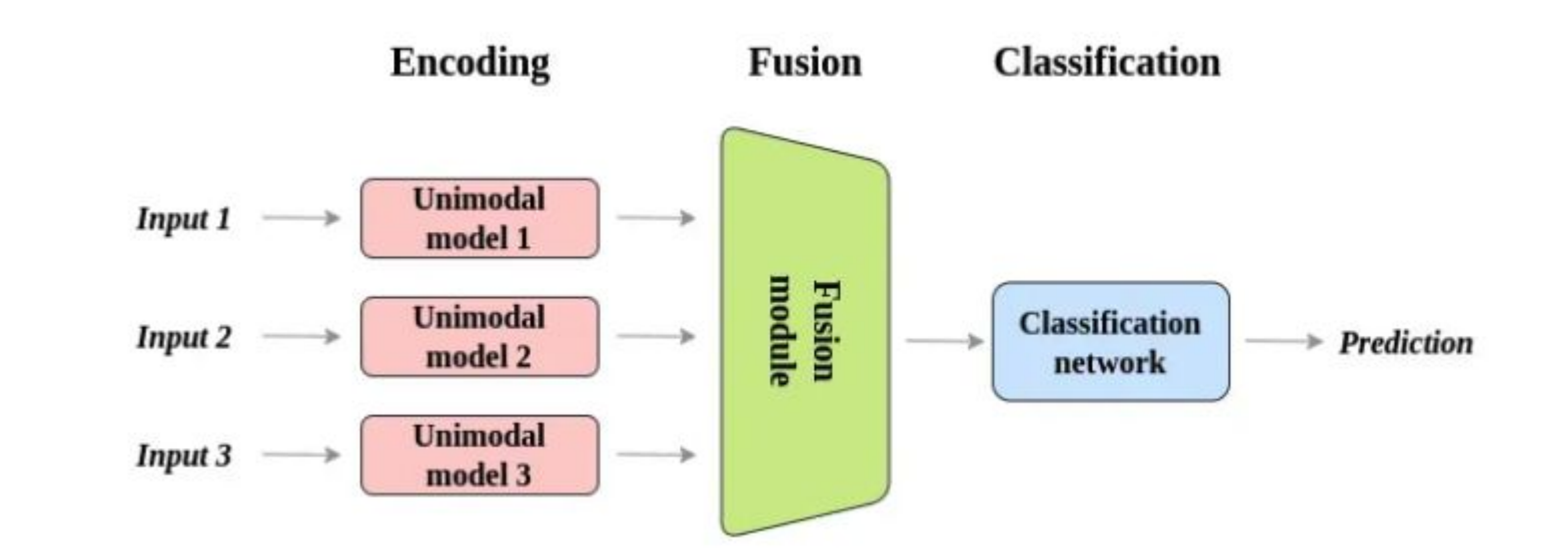
Dense Retrieval is also not perfect



Multimodal Embedding

Joint Embedding Space




Encoding
Fusion
Classification
Input 1
Input 2
Input 3
Unimodal model 1
Unimodal model 2
Unimodal model 3
Fusion module
Classification network
Prediction

Examples:
CLIP ,SigLIP etc

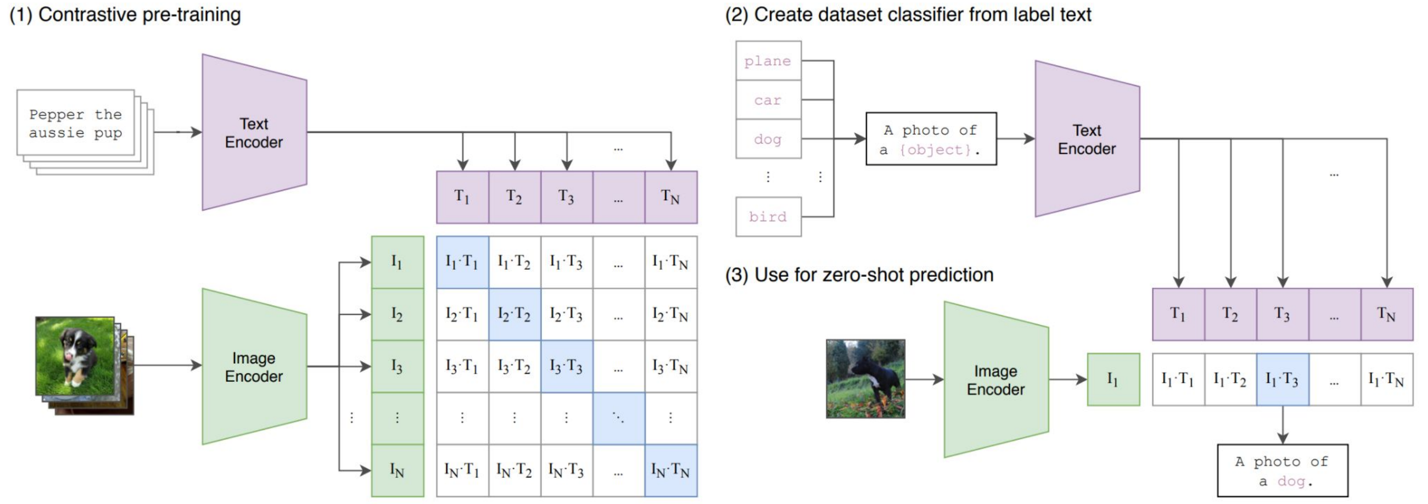
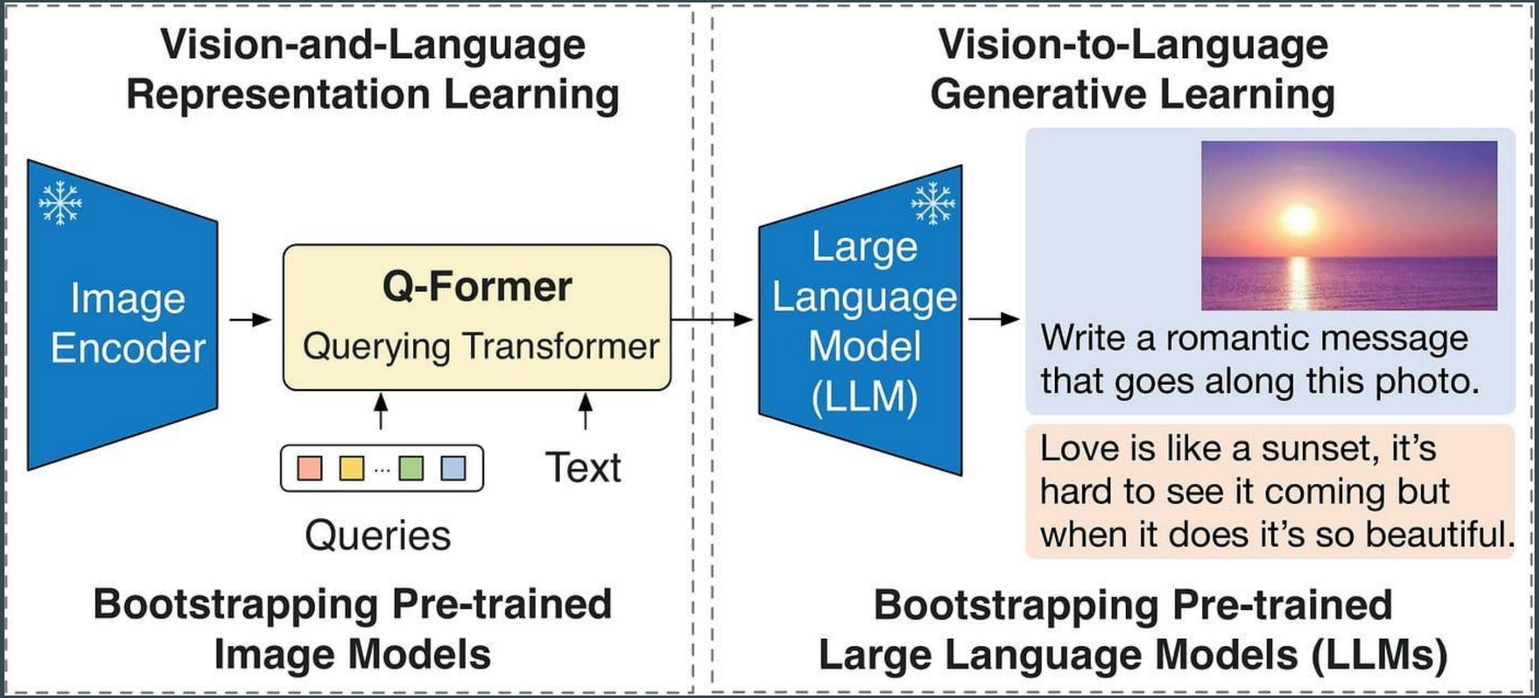


Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

MultiModel Search
(Using BLIP2 and Lavis)

BLIP-2 Architecture



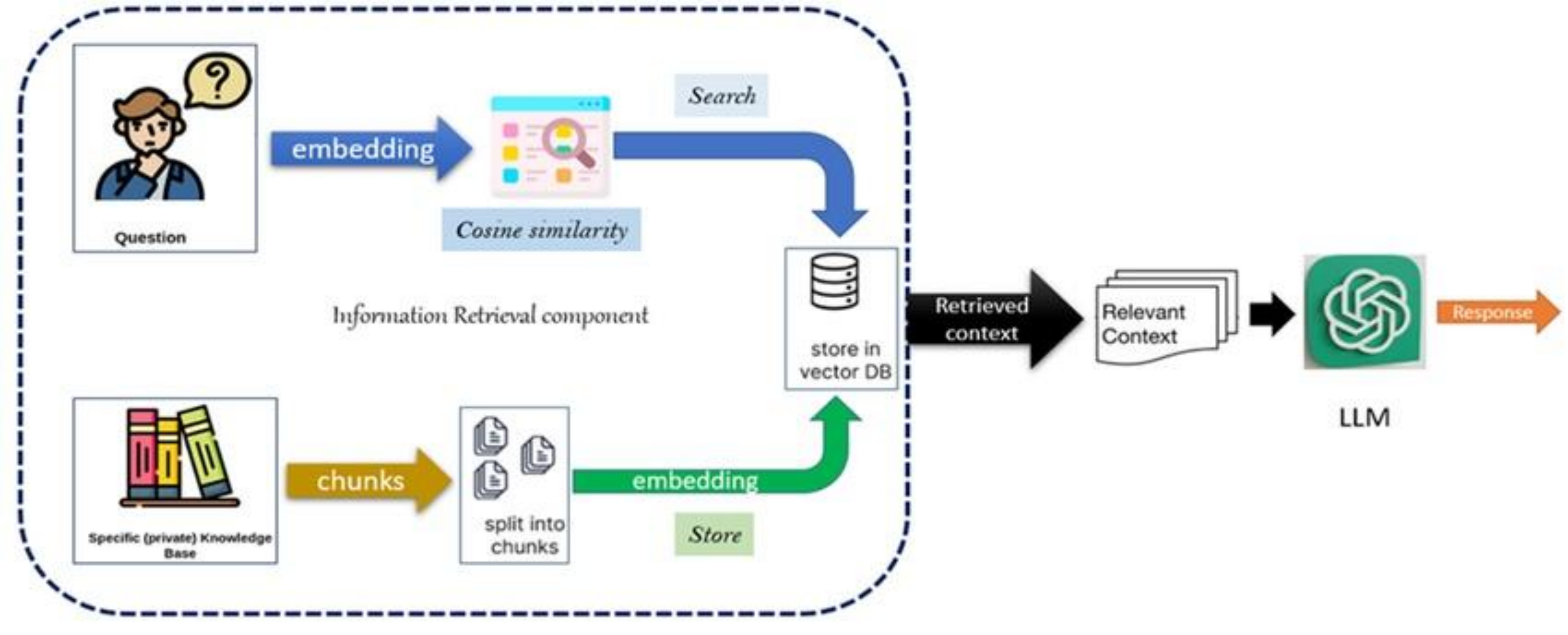
Demo

Core Components for Conversational Search

- Choosing correct Multimodal Model (like Llava)

- Conversational History Handling

Retrieval-Augmented Generation (RAG)



Using LlamaIndex,Ollama using Llava model

Examples

Multimodal Ollama Cookbook

Open in Colab

This cookbook shows how you can build different multimodal RAG use cases with LLaVa on Ollama.

- Structured Data Extraction from Images
- Retrieval-Augmented Image Captioning
- Multi-modal RAG

Setup Model

```
!pip install llama-index-multi-modal-llms-ollama
!pip install llama-index-readers-file
!pip install unstructured
!pip install llama-index-embeddings-huggingface
!pip install llama-index-vector-stores-qdrant
!pip install llama-index-embeddings-clip
```

```
from llama_index.multi_modal_llms.ollama import OllamaMultiModal
```

```
mm_model = OllamaMultiModal(model="llava:13b")
```

Table of contents

Repo Link