

MOVIE SEARCH APPLICATION

Big Data Capstone Project – August 2024

ARUN KUMAR SUBRAMANIAM

CARLOS MUNOZ EBRATT

KAMALDEEP KAUR

LUIS ALEJANDRO GUTIERREZ HAYEK

FELIPE FERNANDEZ



INTRODUCTION

This project is an example of building a semantic search using Python, NLP models, and Atlas Vector Search as a vector database for finding movies using natural language queries and Streamlit as the user interface to interact with the Database.



VECTOR EMBEDDINGS

A vector is a list of floating point numbers (representing a point in an n-dimensional embedding space) and captures semantic information about the text it represents. For instance, an embedding for a string using an open source LLM model called all-MiniLM-L6-v2 would consist of 384 floating point numbers and look like this.

```
▼ plot_embedding_hf : Array (384)  
  0: -0.03985478729009628  
  1: -0.016877852380275726  
  2: -0.051168572157621384  
  3: 0.08319630473852158  
  4: 0.04009385406970978
```



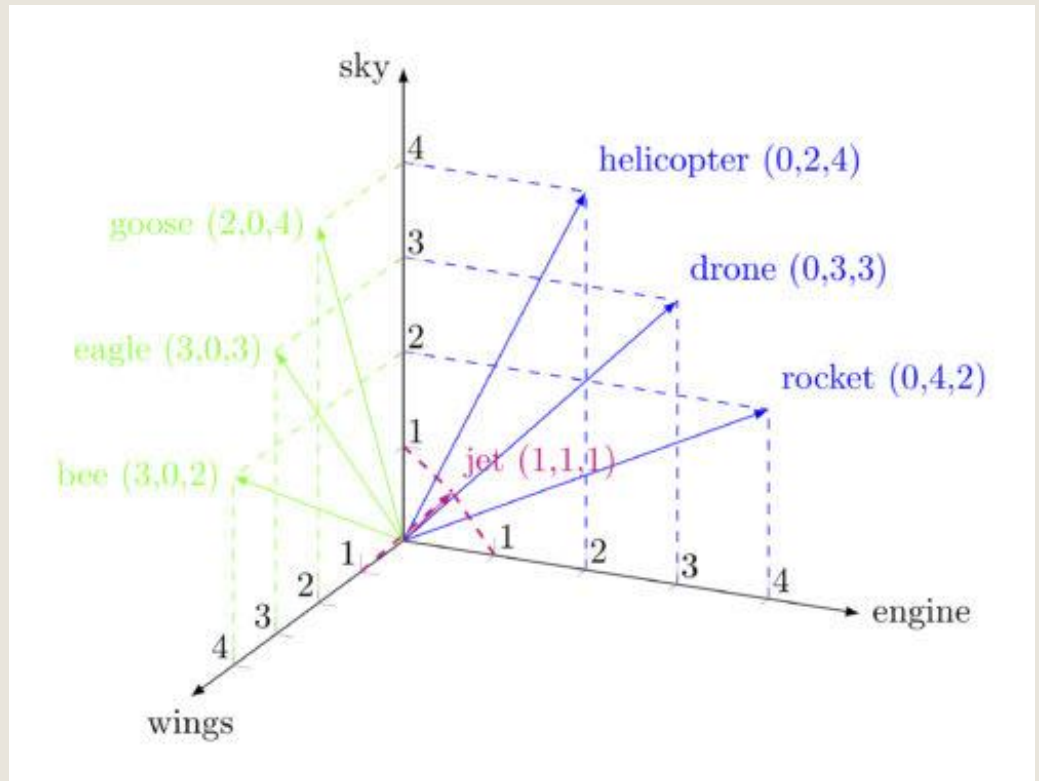
ALL-MINILM-L6-V2

- The **all-MiniLM-L6-v2** is a pre-trained language model developed by Microsoft. It's a smaller and more efficient version of the MiniLM model, specifically designed for tasks like semantic search, sentence similarity, and embedding generation. The model has 6 layers and uses a smaller architecture to provide fast and accurate performance while maintaining a lower computational cost. It is widely used in applications that require embedding text into dense vector spaces for tasks like document retrieval and clustering.
- This is a sentence-transformers model: It maps sentences & paragraphs to a 384 dimensional dense vector space and can be used for tasks like clustering or semantic search
- <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>



COSINE SIMILARITY

- Vector coordinates can be used to calculate the proximities between words. This is done with cosine similarity ($\cos \theta$), the cosine of the angle between two word vectors.



VECTOR SEARCH

- Vector search is a capability that allows you to find related objects that have a semantic similarity. This means searching for data based on meaning rather than the keywords present in the dataset.
- Then, it finds related content by comparing the distances between these vector embeddings, using approximate k nearest neighbor (approximate KNN) algorithms.
- The most commonly used method for finding the distance between these vectors involves calculating the cosine similarity between two vectors.



ATLAS VECTOR DATABASE

- Fully managed service that simplifies the process of effectively indexing high-dimensional vector data within MongoDB and being able to perform fast vector similarity searches.
- MongoDB as a standalone vector database for a new project or augment the existing MongoDB collections with vector search functionality.

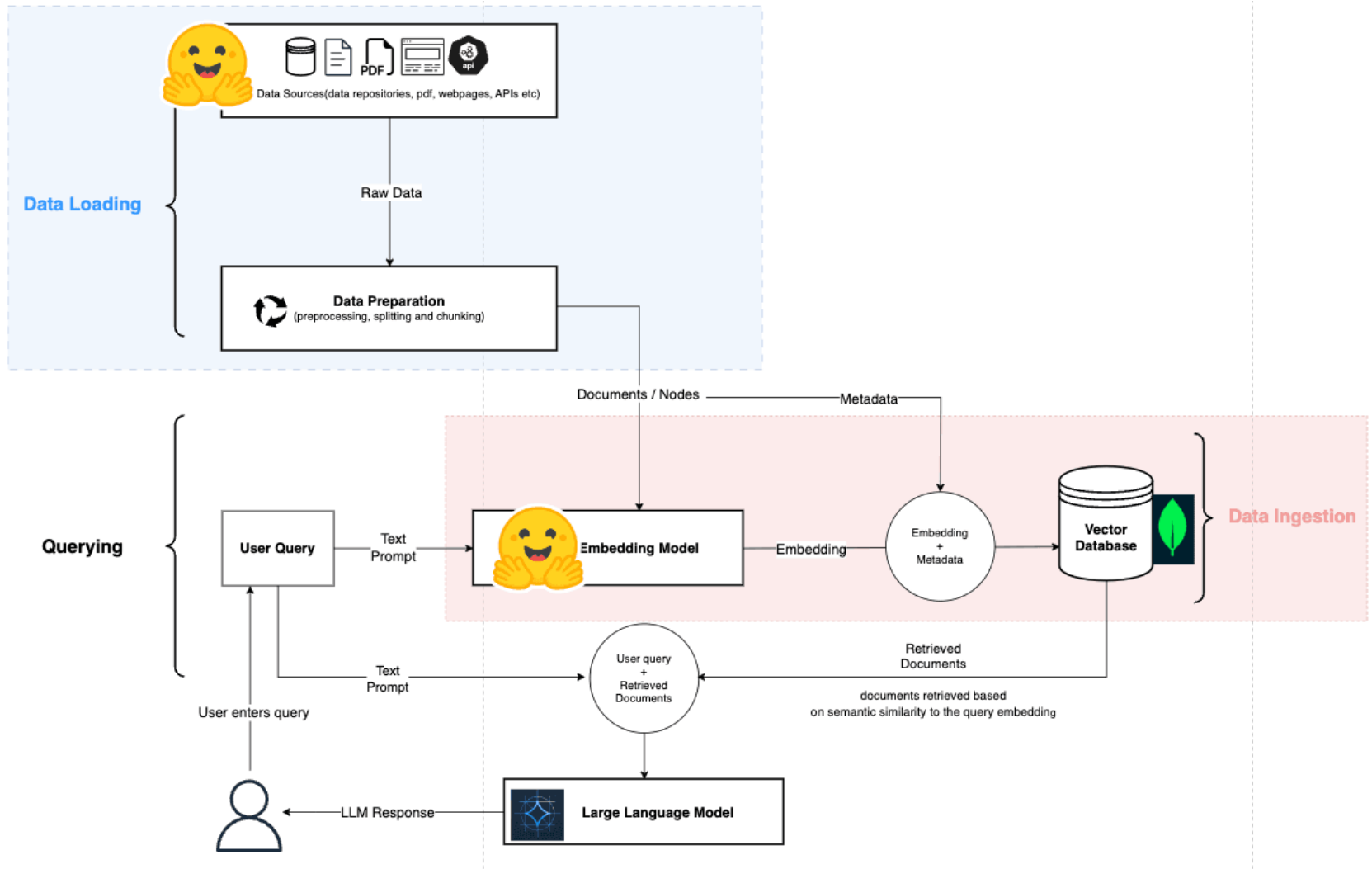


SEMANTIC SEARCH FOR MOVIE RECOMMENDATIONS


- We will be using the all-MiniLM-L6-v2 model from HuggingFace for generating the vector embedding during the index time as well as query time.
- The user query can be converted into vector embedding, and then the vector search can find the most relevant results by finding the nearest neighbors in the embedding space.



SYSTEM ARCHITECTURE



DATA LOADING: STEP 1- CREATE HUGGINGFACE ACCESS TOKENS

**Felipe Fernández**
Felipefe20

[Profile](#)
[Account](#)
[Authentication](#)
[Organizations](#)
[Billing](#)
[Access Tokens](#)
[SSH and GPG Keys](#)

Access Tokens

User Access Tokens

[+ Create new token](#)

Access tokens authenticate your identity to the Hugging Face Hub and allow applications to perform actions based on token permissions. **Do not share your Access Tokens with anyone**; we regularly check for leaked Access Tokens and remove them immediately.

Name ▾	Value	Last Refreshed Date ▾	Last Used Date ▾	Permissions ▾	
🔑 genai	hf_...	May 23	44 minutes ago	READ	⋮
🔑 notebooks	hf_...	Jan 23	-	WRITE	⋮
🔑 space	hf_...	Jan 12	-	WRITE	⋮



STEP 2: CONNECT TO MONGODB AND CREATE EMBEDDINGS

```
vector-search-projects - Generate_embeddings.py

1
2 db = client.sample_mflix
3 collection = db.movies
4 #add counter to the loop
5 counter = 0
6 for doc in collection.find({'fullplot':{'$exists': True}}):
7     #check if the plot embedding already exists
8     if 'plot_embedding_hf' not in doc:
9         try:
10             doc['plot_embedding_hf'] = generate_embedding(doc['fullplot'])
11             collection.replace_one({'_id': doc['_id']}, doc)
12             counter=counter+1
13             print(f"Updated{counter}",)
14         except:
15             pass
16
```

Restrictions in how many vectors per hour, we had to
subscribe to HuggingFace Pro



- Once this step completes, we verify in the database that a new field “plot_embedding_hf” has been created for some collections.

sample_mflix.movies

STORAGE SIZE: 91.25MB LOGICAL DATA SIZE: 125.43MB TOTAL DOCUMENTS: 21349 INDEXES TOTAL SIZE: 17.16MB

[Find](#) [Indexes](#) [Schema Anti-Patterns](#) [Aggregation](#) [Search Indexes](#)

[Generate queries from natural language in Compass](#)

[Filter](#) Type a query: { field: 'value' }

```
{
  "_id": ObjectId('573a1390f29313caabcd5ea4'),
  "plot": "A District Attorney's outspoken stand on abortion gets him in trouble _",
  "genres": Array (1)
    runtime: 62
    rated: "APPROVED"
  "cast": Array (4)
    title: "Where Are My Children?"
    fullplot: "While prosecuting a physician for the death of a client after an abort_"
  "languages": Array (1)
    released: 1916-05-01T00:00:00.000+00:00
  "directors": Array (2)
  "writers": Array (4)
  "awards": Object
    lastupdated: "2015-09-07 00:51:32.560000000"
    year: 1916
  "imdb": Object
  "countries": Array (1)
    type: "movie"
  "tomatoes": Object
    num_mflix_comments: 0
  "plot_embedding_hf": Array (384)
    0: -0.07599133998155594
    1: 0.11164375394582748
    2: -0.07571761057820710
```



STEP 3: CREATE A VECTOR SEARCH INDEX

To perform vector search on the data in Atlas, we create an Atlas Vector Search index, which are used to efficiently retrieve documents that contain vector embeddings.

FELIPE > VECTOR-SEARCH > DATABASES

FelipeMongoDBCluster

VEI
7.0

Overview Real Time Metrics Collections **Search & Vector Search** Performance Advisor Online Archive Cmd Line Tools

← PlotSemanticSearch

Index Overview

Search Tester

Query Analytics

Quick Start Tutorials

[View Atlas Search Docs](#)

Edit search index "PlotSemanticSearch" for sample_mflix.movies [View Atlas Search Docs](#)

```
1  {
2    "mappings": {
3      "dynamic": true,
4      "fields": {
5        "plot_embedding_hf": [
6          {
7            "dimensions": 384,
8            "similarity": "dotProduct",
9            "type": "knnVector"
10         }
11       ]
12     }
13   }
14 }
```



QUERYING THE DATA

```
vector-search-projects - user_interface.py

1 pipeline = []
2
3 pipeline.append({
4     "$vectorSearch": {
5         "queryVector": query_vector,
6         "path": "plot_embedding_hf",
7         "numCandidates": 5000,
8         "limit": num_movies,
9         "index": "PlotSemanticSearch",
10    }
11 })
12
13 # Add a match stage to limit vector search to the initially filtered results
14 pipeline.append({"$match": {"_id": {"$in": matching_ids}}})
15
16 pipeline.append({"$project": {"score": {"$meta": "vectorSearchScore"}}})
17
18 results = collection.aggregate(pipeline)
```



USER INTERFACE AND DEMO

- <https://capstone-project-movies-vector-search.streamlit.app/>

Movie Recommender

Select genres:
Choose an option ▾

Select year range:
1998 2023

Select rating:
Choose an option ▾

Select IMDB rating:
8.00 10.00

Ask for movie recommendations
Superheroes

Number of movies to recommend:
2

1

Score: 9.8416839347648621

Title: Superheroes


imdb rating: 6.5

Genre: ['Documentary', 'Action', 'Comedy']

Year: 2011

rated: N/A

Plot: 'Superheroes' will introduce us to several of the country's most famous masked heroes including, Mr. Xtreme, a 33-year-old security guard officer by day, but a goon's worst nightmare by night. We'll follow Mr. Xtreme on his nightly patrols through the streets of San Diego, as he tries to stop evildoers and protect the innocent. We'll also meet the New York Initiative, a fantastic foursome of real life superheroes living together that tackle crime fighting, one Brooklyn borough at a time. Lead by Zimmer, we'll watch as they take to the streets and try to lure criminals out of hiding with their controversial Bait-Patrols. With over 300 registered superheroes in the United States, we'll definitively uncover the 'Real-Life Superhero' cultural phenomenon and discover what inspired these everyday citizens to take the law in to their own hands as they try to make the world a better and safer place for all.





CONCLUSIONS:

- Effective Use of Transformer Models
- Cosine Similarity as a Reliable Metric
- Scalable and User-Friendly System
- Challenges with Data Quality and API Limits



RECOMMENDATIONS:

- Use a LLM to interact with the data and create a chatbot or RAG (retrieval-augmented generation) system with memory
- App depends on the data quality and size, instead of the model used.
- Fine-Tuning Models: Further fine-tuning of the transformer models on movie-specific data could improve the quality of the embeddings, leading to even more accurate similarity scores and recommendations.



QUESTIONS & ANSWERS

