Recommender with Hybrid Embeddings on MovieLens 32M

Repository: https://github.com/tsukayaa/SRIN-Assignment

A. Objective

Building a recommendation system for movie based with vector database

B. Introduction : Vector Databases

Vector databases stores complex data like images as high-dimensional numerical array called vector embeddings. We can use this embeddings to find contextually similar items using method like cosine similarity etc.It can also be used on the recommendation system because engineer can store item as an embeddings on the vector database.Why Milvus? I try 3 vector databases for this project, the first one is Pinecone and its have a free starter plan but it comes with some limitations. Each upsert batch is capped at 2 MB and each vector can only hold up to 40kb of metadata, which is for dataset as big as MovieLens 32M(approximately 260 MB on size) these limitations is really hard to scale.

The 2nd is Supabase,I had some experience with this vector database in te past, its nice because we can mix SQL queries with vector search, when I've experimented with Supabase on the dataset I realized it doesnt perform really good with large dataset. From my opinion the performance not good because its basically Postgres with pgvector and it doesn't have advanced indexing for similarity search. The last one is Milvus, its open source and very scalable with big data and built for large scale vector search. This vector databases comes with built-in approximate nearest neighbour(ANN) method like HNSW which makes similarity seach much faster and scalable. At the end, I choose Milvus for this project because its fast,scalable and can handle millions of vector(its free of course 😊)

C. Methodology

I use Movielens 32M Dataset,it consist of 3 file:

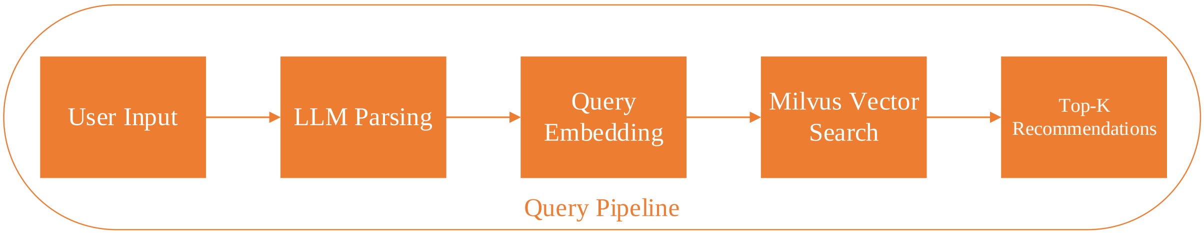
- ratings.csv ,contains user ratings for movies
- movies.csv , contains metadata like movie titles and genres.
- tags.csv , contains user-provided tags describing movies.

For the embeddings of the items(movie in this case),I use 3 hybrid query embeddings as follows :

- Tag embeddings(semantic), i use the tags from tags.csv then preprocess the raw tags by cleaning them(lowercasing,trimming,removing short tokens) and filtering out tags that rare since this kind of tags will add noise to the representation. After that I take the top 10 tags that users frequently associate with that movie. End of process is every movies is described by small set of keywords. Then I use a pretrained sentence transformers model(all-MiniLM-L6-v2) to convert the text into embeddings. Output of this embeddings is 384 dimensional vector

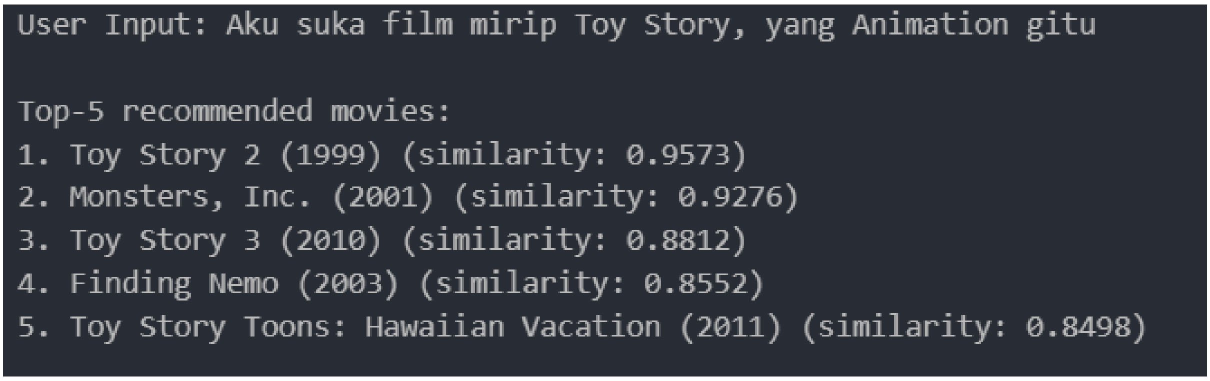
- Genre embeddings,i use the genres from movies.csv and encode them into one-hot vector to capture categorical similiratiy between movies, output of this is 20 dimensional vector
- Collaborative Filtering(CF) embeddings, i use the ratings from ratings.csv then applying matrix factorization on it. First the dataset is filtered to include user with a minimum of 5 ratings and movies with minimum of 10 ratings. Then i use Singular Value Decomposition (SVD) model from the Surprise library to factorize the sparse user-movies rating matrix. The model configured with 64 latent factors which make the dataset turn into a 64 dimensional vector.

The final hybrid embeddings matrix finally consists of 384+20+64 = 468 dimensional vectors.After that I designed the query pipeline,system flow of the query pipeline is as below :



The input is natural language(e.g "Aku suka film mirip Toy Story, yang Animation gitu") and then a Gemini 2.5 Flash model that have been prompted will extract structured information.The Gemini LLM will identify key components such as reference movies and relevant genres then I convert that output inton JSON format. After that i fetch the hybrid embeddings from references movies and performs cosine similarity search in Milvus using .distance . Finally the system will exclude the reference movie itself(of course the reference movies will have similarity =1) and returns top-k most similar movies based on the distance

D. Results Example



```
User Input: Aku suka film mirip Toy Story, yang Animation gitu

Top-5 recommended movies:
1. Toy Story 2 (1999) (similarity: 0.9573)
2. Monsters, Inc. (2001) (similarity: 0.9276)
3. Toy Story 3 (2010) (similarity: 0.8812)
4. Finding Nemo (2003) (similarity: 0.8552)
5. Toy Story Toons: Hawaiian Vacation (2011) (similarity: 0.8498)
```