**Information Retrieval**

Yashwanth Reddy Paturu ([yashwanth.paturu@latentview.com](mailto:yashwanth.paturu@latentview.com))

**Objective**:

Build a streamlit based app with information about 10-15 movies from Wikipedia. The user asks a question about the movie and gets a response from LLM model. Use embedding model here to create embeddings of wiki document. Use Lang Chain / LLAMA index / Chroma DB (vector database) to build the question answering system

**Model Used:**

**NVIDIAEmbeddings**: For generating text embeddings.

**NVIDIA Embeddings Model**: NV-Embed-QA

**FAISS**: For fast similarity search and embedding storage.

For **Question-Answering**: **meta/llama-3.1-8b-instruct** is part of Meta's LLaMA (Large Language Model Meta AI) is used. It can answer questions by understanding the context and retrieving relevant information, making it ideal for use cases like chatbots.

**Approach**:

a. Natural Language Processing (NLP) Pipeline:

* Context Embedding: Using NVIDIA's embedding model (**NV-Embed-QA**) to converting the context into vector representations.
* Query Processing: Prepare the user's input query for the language model. (converts the user query into embeddings with the same model)
* Response Generation: Use the LLaMA model to generate responses based on the context and query.

b. Retrieval-Augmented Generation (RAG):

* Implemented a vector database (**Faiss**) to store and efficiently retrieve relevant context.
* Use similarity search to find the most relevant context for each query.

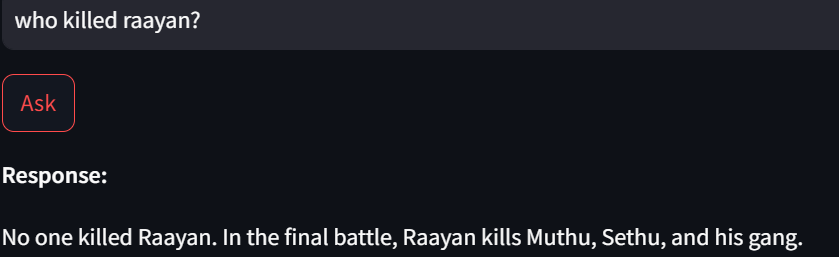
c. Streamlit:

* The query will be getting through streamlit interface by user after query processing and similarity search response is generated using LLaMA model this response is shown in streamlit interface.

**Screenshots:**

A screenshot of a computer

Description automatically generated



A screenshot of a computer

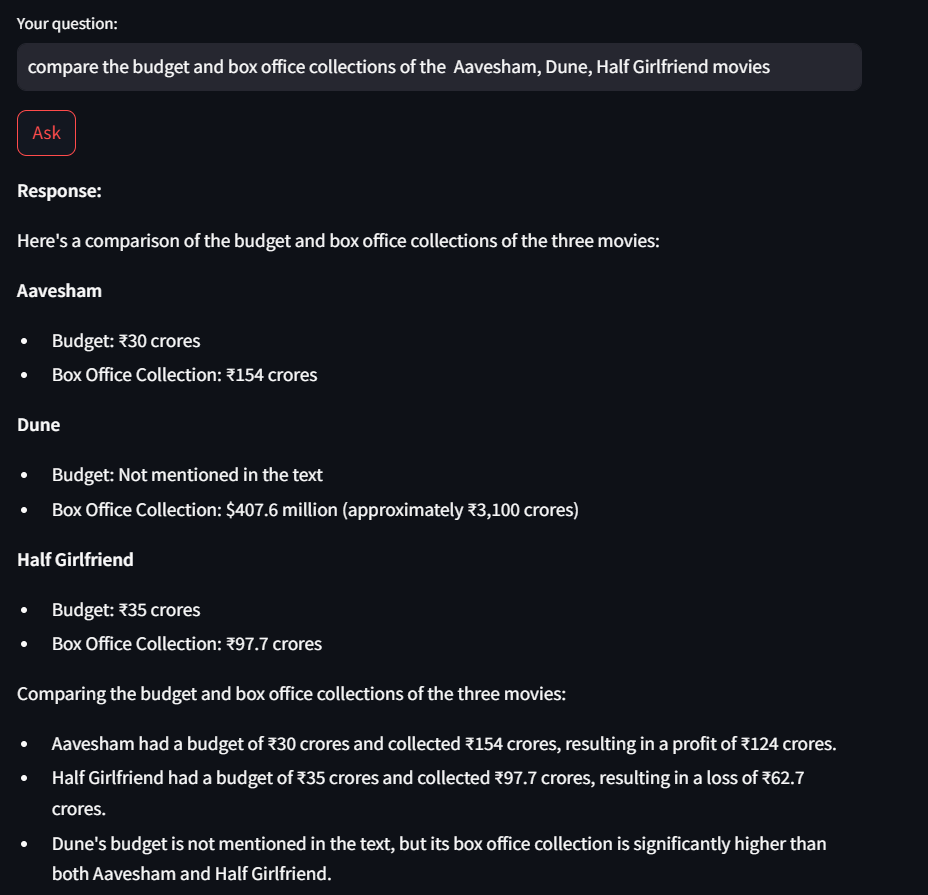
Description automatically generated

A screenshot of a computer

Description automatically generated

A black screen with white text

Description automatically generated



A screen shot of a computer

Description automatically generated

A screenshot of a computer

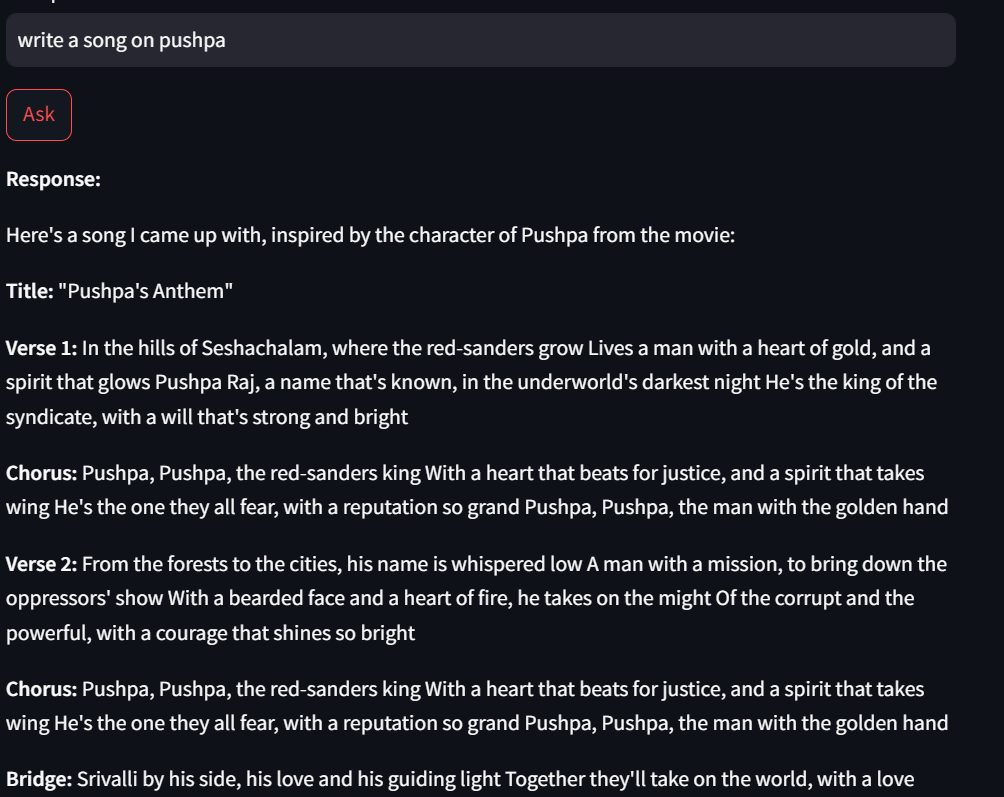
Description automatically generated

A screenshot of a movie

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

All the outputs were but in the last snap I asked about the music composer of all movies I didn’t say specified movie name even though the output is right, but it didn’t give the answer to our question. The problem is that we need to specify the movie name.

A screen shot of a movie

Description automatically generated

Now it’s giving correct answer as I mentioned the movie name.

**The accuracy is around** **90 to 95%**