Myntra Product Semantic Search using Sentence Transformers and ChromaDB

Author: Nitish Narayanan

Abstract

This project enables semantic search on Myntra's product catalog using transformer-based embeddings and vector similarity.
By converting product descriptions into dense embeddings, users can perform natural language queries and retrieve contextually relevant results.

Introduction

E-commerce platforms like Myntra host millions of products, making it challenging for users to find exactly what they are looking for.
Traditional keyword-based search often fails to capture semantic meaning or intent behind queries.
This project introduces a semantic search approach powered by Sentence Transformers and ChromaDB to enable context-aware retrieval.

System Architecture

The architecture consists of data preprocessing, embedding generation, vector storage, semantic retrieval, and LLM-based response generation.

Methodology

1. **Data Loading**: Myntra dataset loaded using Pandas.
2. **Preprocessing**: Combined relevant fields (product name, description, brand, etc.) into a unified semantic text field.
3. **Embedding Generation:** Used SentenceTransformer ('all-MiniLM-L6-v2') to generate dense embeddings.
4. **Vector Storage:** Stored embeddings in ChromaDB for fast similarity-based retrieval.
5. **Retrieval:** User queries are embedded and matched using cosine similarity.
6. **LLM Integration:** Retrieved results are summarized using Perplexity API or GPT for natural responses.

Implementation

- Environment: Google Colab
- Libraries: pandas, numpy, sentence-transformers, chromadb, langchain-core, openai
- Vector DB: ChromaDB
- Embedding Model: SentenceTransformer ('all-MiniLM-L6-v2')
- LLM Integration: Perplexity API for semantic responses

Results & Demo

Example queries executed:
- “Find cotton shirts under 999 from brand U.S Polo Assn”
- “Show me red dresses suitable for party.”
- “List running shoes for men under ₹2000.”

Each query retrieves semantically relevant products from the dataset with contextual responses from the LLM.

Future Enhancements

- Integration of multilingual support.
- Deployment of API endpoint for production.
- Addition of fine-tuned models for e-commerce-specific embeddings.
- Better visualization dashboards using Streamlit or Gradio.

References

1. https://www.sbert.net/ – Sentence Transformers
2. https://docs.trychroma.com/ – ChromaDB Documentation
3. https://www.perplexity.ai/ – Perplexity API
4. LangChain Documentation – https://python.langchain.com/