PROJECT REPORT - FASHION SEARCH AI

By Shivani Raut, Sindhu M and Vidhya Manikandan

The Fashion Search AI project aims to develop a generative search system capable of searching through extensive fashion product descriptions to recommend appropriate choices based on user queries. This system leverages advanced machine learning models, specifically utilizing the LlamaIndex for efficient indexing and retrieval of product data. The project is built upon a dataset sourced from Myntra, a popular online fashion retailer, and is designed to enhance the user's shopping experience by providing accurate and relevant fashion recommendations.

PROBLEM STATEMENT

With the vast amount of fashion products available online, users often face difficulty in finding items that match their preferences and needs. Traditional search systems may not effectively understand the nuances of user queries, leading to irrelevant or suboptimal results. Therefore, there is a need for an intelligent search system that can comprehend and process natural language queries to deliver precise recommendations.

Why LlamaIndex?

LlamaIndex was chosen for its efficiency in handling large datasets and its ability to provide fast, relevant search results. It allows for the indexing of extensive collections of documents, facilitating quick retrieval of information. Additionally, LlamaIndex supports advanced search features, including semantic search, which is crucial for understanding and processing natural language queries.

PROJECT GOALS:

- Develop a generative search system capable of understanding complex user queries.
- Utilize the Myntra fashion product dataset to provide accurate and relevant product recommendations.
- Implement a multi-layered system design to handle various aspects of the search and recommendation process.
- Ensure the system is scalable and can handle a large volume of queries efficiently.

DATA SOURCE

The primary data source for this project is the Myntra fashion product dataset from Kaggle. This dataset includes detailed descriptions, prices, colors, brands, ratings, and other relevant attributes of fashion items available on Myntra along with their images.

DATA PREPROCESSING:

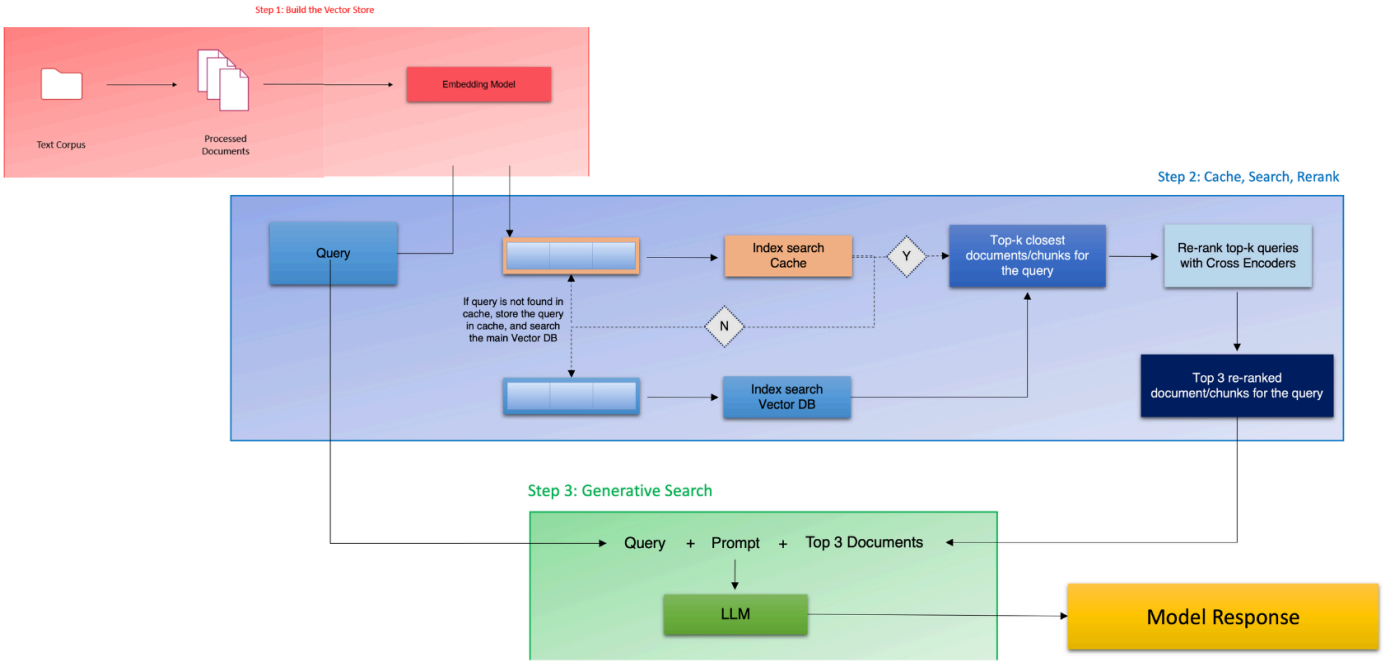
The dataset was processed prior to importing. Key preprocessing steps included:

- Removing unwanted column ‘img’ from the dataset.
- Filling null values in the ‘avg_rating’ and ‘ratingCount’ columns with 0.
- Removing HTML tags from the ‘p_attributes’ column.
- Performing basic data cleaning such as removing white spaces and extra characters.

DESIGN CHOICES:

- Semantic Search: Implemented to understand the context and nuances of user queries.
- Cross-Encoder Re-ranking: Used to refine search results and improve the relevance of recommendations.
- Cache Collection: Employed to speed up repeated queries by storing previously retrieved results.
- Generative Model: Utilized GPT-3.5 for generating comprehensive responses based on search results.
- Not Chunking Data: Chunking was deemed unnecessary as the descriptions and attributes columns did not contain large volumes of data and chunking the data might cause loss of meaning.

ARCHITECTURE:



LAYERS:

1. **Generating and Storing Embeddings Layer (Vector Store):**

The system used OpenAI to generate vector representations for the fashion product descriptions using OpenAI's text-embedding-ada-002 model and these embeddings were stored in ChromaDB for efficient retrieval and semantic search.

2. **Search, Cache and Rerank Layer:**
   Semantic Search with Cache: This layer implemented a semantic search mechanism with a caching layer to optimize search results. The search layer handles the primary search of user query to find top top K closest descriptions in the main collection if the cache doesn't have the results. The cache stores frequently searched queries or query that have been processed before and their results to improve retrieval speed and efficiency.
   Re-ranking Layer: Uses a cross-encoder to re-rank the search results based on relevance. Cross Encoder utilized the `ms-marco-MinLM-6-v2` model for re-ranking the search results. This model helps in improving the relevance of search results by scoring them based on the query-response pairs.
3. **Retrieval-Augmented Generation Layer:**
   The system retrieves the top 3 search results based on the semantic search and cross-encoder scores. Combines the top 3 results with the user query to generate a comprehensive and direct answer, including images of the recommended fashion items using a prompt to the generative model.

Initialization Function

A function was defined to start the conversation with the user, allowing them to input queries and receive responses based on the RAG pipeline. This function continuously processes user input and provides relevant fashion recommendations until the user decides to exit.

Testing Pipeline

- **Testing with 5 Questions:** The system was tested with five different questions to evaluate its performance. User feedback on the responses was collected and analyzed.
- **Feedback Collection:** User feedback was essential in identifying areas for improvement and ensuring the system's responses met user expectations.

CHALLENGES:

1. Metadata Creation: Handling large datasets posed a challenge in creating metadata.
2. System Performance: Ensuring high performance while dealing with extensive datasets was challenging.
3. Data Quality: Inconsistent or incomplete product descriptions in the dataset needed to be addressed to ensure accurate recommendations.

LESSON LEARNED:

1. Memory and Runtime Considerations: Large datasets require significant memory and runtime resources.

2. Importance of Preprocessing: Proper preprocessing of the dataset improves the quality of the results.
3. Advanced AI Models: Utilizing advanced AI models enhances the system's capabilities.
4. Fine-Tuning Parameters: Adjusting semantic search parameters and thresholds is crucial for optimal performance.

FUTURE SCOPE:

- Improving Prompts: Better prompts can enhance the quality of the generated responses.
- Moderation and Iteration Layer: Adding a moderation layer to filter inappropriate content and an iteration layer to gather more information from users can improve the system's performance.
- Content and Product-Based Filters: Implementing these filters can further enhance system performance by providing more tailored recommendations.
- Application Development: The project can be extended to develop a full-fledged application.

CONCLUSION

The Fashion Search AI project successfully implements a search system using the RAG pipeline and semantic search layer. The system efficiently retrieves and generates relevant fashion product recommendations.