Flipkart

GRiD 5.0

Problem Statement Title: Conversational Fashion Outfit Generator powered by GenAI.

Team Name: Wingardium Laviosa

Team members details

Team Name	WINGARDIUM LEVIOSA		
Institute Name/Names	INDIAN INSTITUTE OF TECHNOLOGY, JAMMU		
Team Members >	1-Dakshi Goel	2-Ashutosh Chauhan	3- Sulabh Srivastava
Name	Dakshi Goel	Ashutosh Chauhan	Sulabh Srivastava
Batch	2024	2024	2024

Problem Statement: Conversational Fashion Outfit Generator powered by GenAI.

Deliverables:

The proposed solution compromises of a web application and LLM pipeline which provides a conversational experience to the user for purchasing, getting recommendations and generate outfits as per the user’s requirements. The web application is build using Reactjs, Django framework and PostgreSQL database. The LLM pipeline is powered by finetuned meta’s Llama 2 chat model with 13 billion parameters. Llama_index and huggingface libraries made it possible for the LLM to interact with both the structured product data and unstructed fashion trends data. The trends data is scrapped from vogue india’s website.

It aims to move the users from traditional static search box experience to a more conversational and personalized shopping experience.

Glossary

1. LLM - Large Language Model
2. PEFT - Parameter Efficient Fine Tuning
3. RAG - Retrievel Augmented Generation
3. NL- Natural Language
4. DB - Database
5. SQL - Structured Query Language
6. UI - User Interface
7. URL - Uniform Resource Locator
8. GPT - Generative Pretrained Transformer

USE-CASES

P0 - Personalised Recommendations by considering both user’s demographics and app usage history.

P1 - Matching the recommendations in accordance with the current fashion trends.

P2- Complete outfit suggestion to provide products which go with each other.

P3 - Conversational experience while shopping in natural language as one would be talking to a friend.

Fig-1: The use cases showing the diversity in the nature of query from user

DATASETS:

There are four types of datasets required for the pipeline :

1. **Users demographics and app-usage data** of the user to filter out products using a query engine.

2. **Product meta data** to provide the details about the available products.

3. **Trending fashion data** from social media like Twitter, Instagram, and fashion blogs to understand the current fashion trends in the market.

4. **Data for fine-tuning** the LLM for better answering capabilites of the LLM.

S.no.	Dataset	Format	Source	Description
1	Demographics and Browsing Data of Users	Postgres DB	Collected by App usage	Demographics: Age, Gender, Location. App-usage: Purchased Items, Favourite Items, Previously Viewed Items
2	Product Meta Data	Postgres DB	Synthetically generated using GPT 3.5 (Reference: Flipkart_Fashion_Products dataset [1])	Product Id, Product_name, Brand, Description, Ratings, Age, Gender, Discount, Pocket, Fabric, Seller, Title etc.
3	Trends Data	Html URLs, Data loaders using Llama-Index	Scrapping of Fashion Blogs (Vogue-India) using BeautifulSoup Twitter Scraping	The data of current fashion trends is available in the form of documents or text in natural language.
4	Data for fine-tuning the LLM	Instruction Based dataset	Custom dataset Generated using GPT 3.5	Question answering data set describing recommendation behaviour and patterns.

Fig-2: The table showing the details of the dataset

Solution statement/ Proposed approach

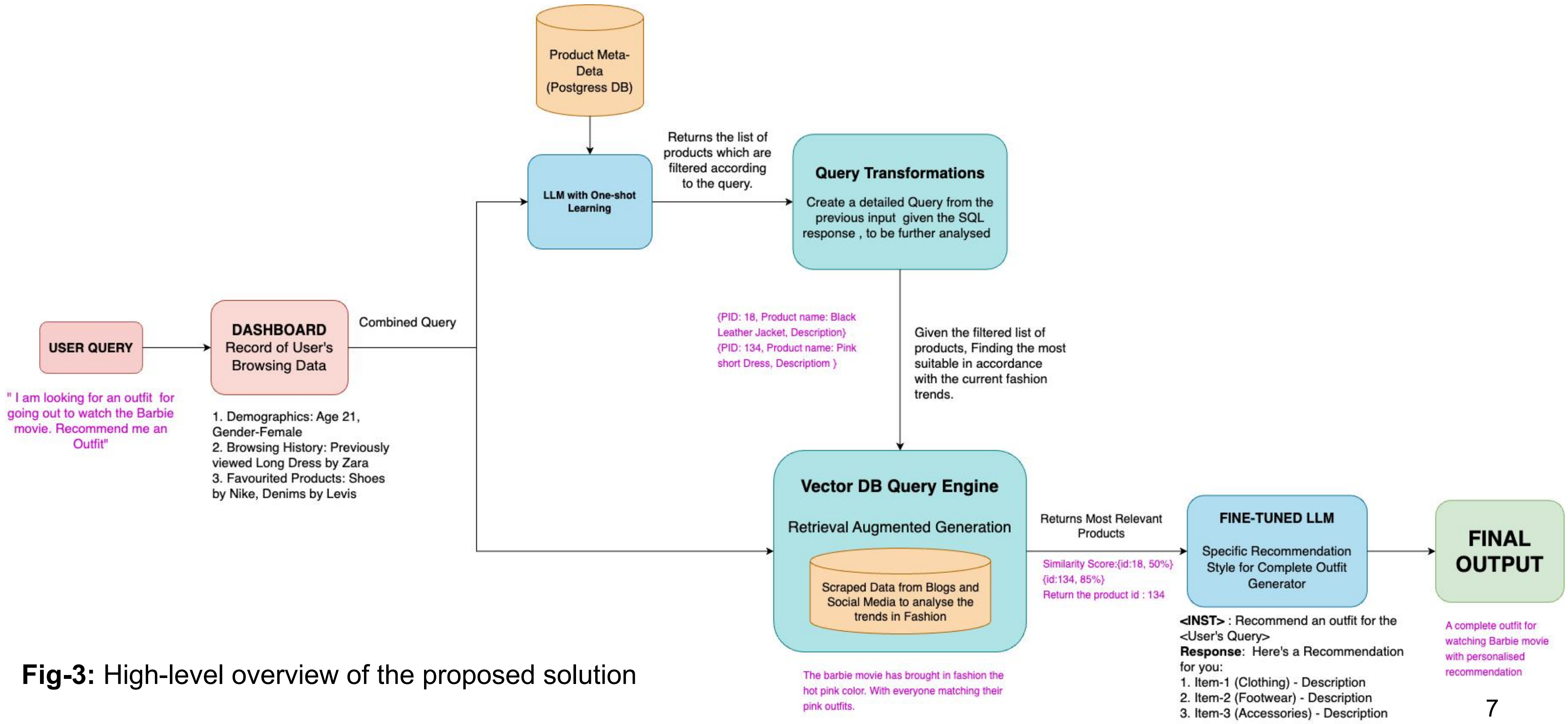


Fig-3: High-level overview of the proposed solution

Step-2: Data Collection and Query :

1. **The Application:**
User can login using their google account and can view, favourite, add to cart, purchase and query products. The UI is user friendly and intuitive.

2. **Data Collection:**
The viewed products by user are tracked and stored in the client's browser and used for recommendation during inference. The favourited, purchased and added to cart products are stored in the database. The latest fashion data is scraped from Vogue India's website.

3. Data Storage:
The user and product data is stored in PostgreSQL database and embedding of scrapped data is stored in redis stack server for semantic search.

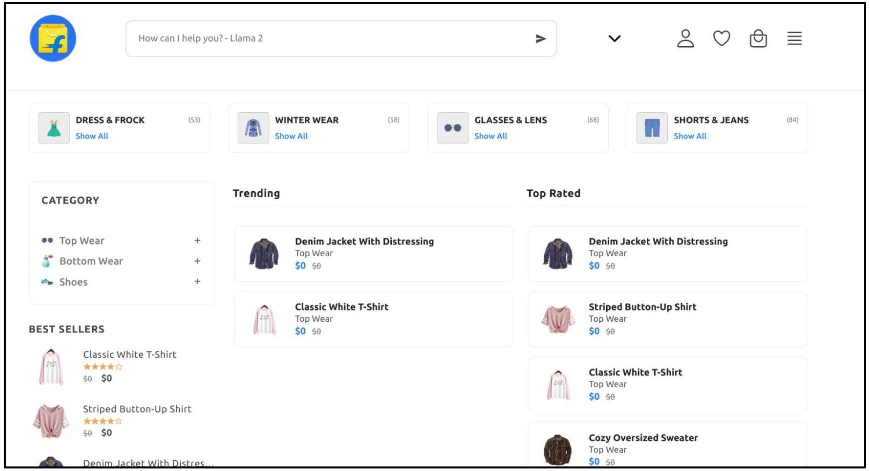


Fig 4. The solution's User Interface

Step-2: LLM for TexttoSQL (Structured Data) :

Given the user query, the users data and demographics are united to form a **combined query**.

The combined query is sent to the LLM model to define two tasks:

- **Task-1 :** Form a query to retrieve the products id, name and description relevant to the combined query. the model queries the product meta data stored in PostgreSQL format. The output of the query contsists of products that are most suitable according to a particular user.

- **Task-2:** Build a query on top of the result obtained fro mthe task -1 to filter the products which are in line with the current fashion trends.This query asks the Llama model to find the similarity score between the description of each product to the vector store (to be further explained).

For this Meta's open-source Llama -13 B model is used and Instruction-based prompt engineering (Few shot learning) is done for the model to perform the above two tasks.

Step-3: LLM for Retrieval Augmented Generation (for Unstructured Data):

The output query of the task 2 is used to find the similarity score between the query embeddings and the vector store embeddings. For this below steps have been performed.

i. **Scrapping of files:** The blogs of Vogue India are parsed using beautifulsoup and the URLs are extracted.

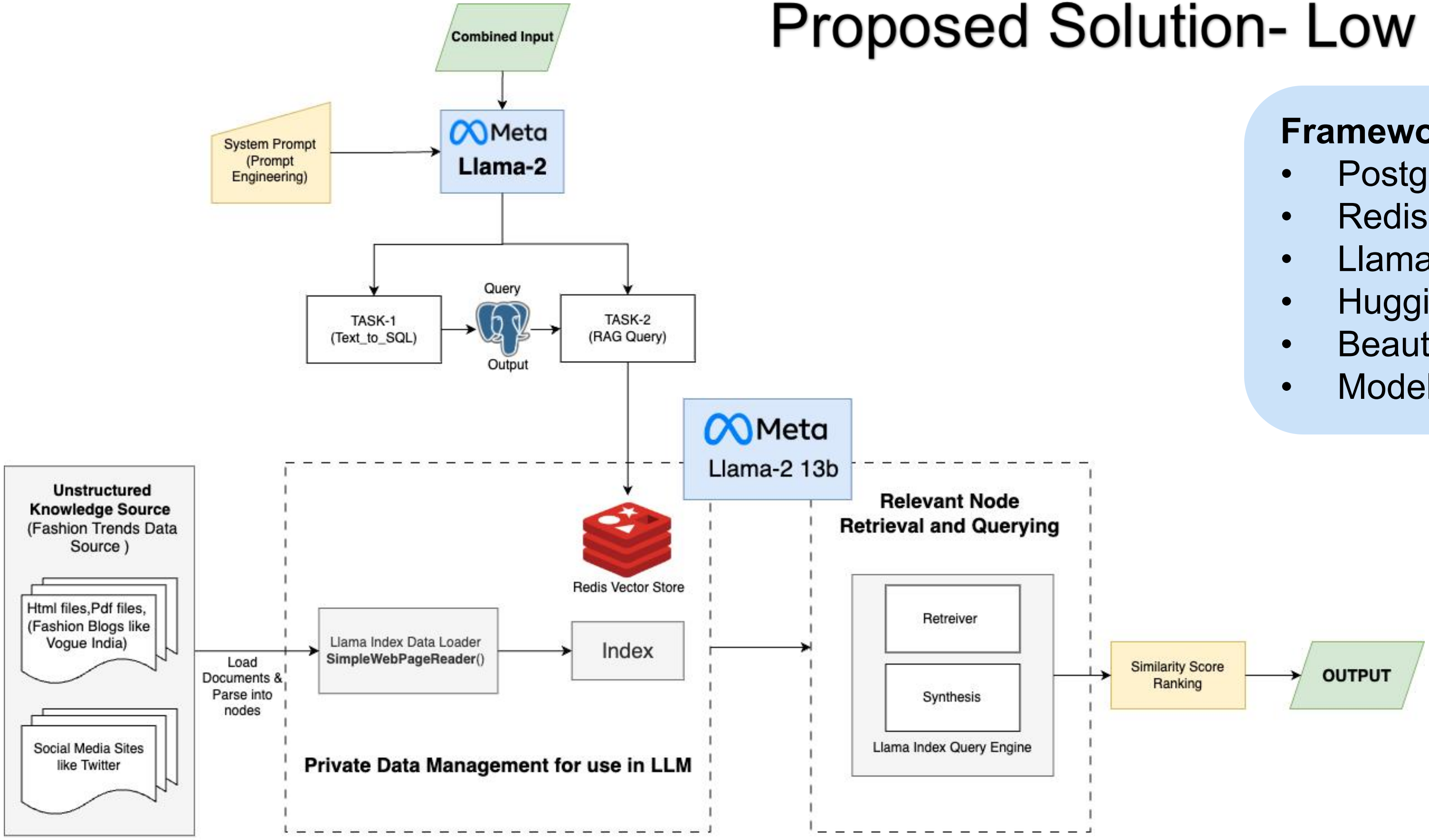
ii. **Data Loaders using Llama Index Hub**: The extracted URLs are fed to the SimpleWebPageReader provided by the llama index.

iii. **Embedding Creation:** Sentence transformers library from Hugging Face is used to find the embeddings of the data.

iv. **Indexing and Vector Data Store:** The embeddings are stored in Redis vector store and are retrieved during inference.

v. **Query Engine** :The query engine find products which have maximum similarity score between the product description and the vector store embeddings.

Proposed Solution- Low Level View



Frameworks used:

- PostgreSQL
- Redis
- Llama_Index
- Hugging Face
- Beautifulsoup
- Model: Llama-2-13b

Fig-3: Low-level view of the model showing flow of data between step-1 and step-2

Step-4: Fine-tuning of the model

1. The llama 2 7b chat model is finetuned to perform outfit recommendation tasks.

2. State-of-the-art Parameter Efficient Fine-Tuning (PEFT) is used for which uses less resources for fine tuning and results in smaller size models.

3. A custom question answering dataset is created using GPT 3.5 model containing 446 datapoints.

4. The fine tuned model showed good results for outfit recommendation tasks.

Limitations

1. **Data limitations**: Due to the creation of synthetic data via another LLM (GPT-3.5) the quality of the dataset can be compromised and limited in diversity.

2. **Hallucinations**: Hallucinations by the LLM are possible due to unavailability of proper data.

3. **High computational resources**: Due to high number of parameters in the Llama-2 model (7B,13B,70B) it require powerful GPU.

4. **Dynamic and Short-lived data**: As the fashion industry is fast paced and always changing, it makes it necessary to continuously update the knowledge base and the vector store.

Future Scope

1. Modeling of LLM with **faster inference capabilities** and **greater context window size**.
2. LLMs capable for analyzing the trends in the users’ changing buying patterns.
3. Fine-tuning on a larger real world dataset instead of synthetic dataset generated by GPT.
4. LLMs capability to analyze multiple similar users to give more probabilistic output.
5. The capability of an LLM to update its knowledge base own its own to handle the dynamic nature of market trends and user preferences.

Flipkart

GRiD 5.0

Thank You