SEMANTIC SEARCH CHATBOT

Problem Statement:

The goal is to develop a chatbot capable of providing detailed responses about fashion products from a Myntra dataset. This chatbot must effectively understand user queries, retrieve relevant product information, and generate comprehensive explanations. The solution involves preprocessing the dataset, creating a vector-based index for efficient querying, and implementing Retrieval-Augmented Generation (RAG) using LlamaIndex to generate detailed responses. The chatbot should also manage conversational context and memory across interactions.

Why LlamaIndex:

LlamaIndex is an ideal framework for this project because it provides robust tools for creating vectorbased search indices and implementing Retrieval-Augmented Generation. It supports efficient querying and retrieval of documents, which is crucial for generating detailed and contextually relevant responses by combining retrieval with generated content.

Project Goals

1. **Preprocess and Clean Dataset:** Prepare the fashion product dataset by filling missing values and cleaning text fields.
2. **Build Vector-Based Index:** Create a vector store index using LlamaIndex for efficient querying and retrieval of product information.
3. **Implement RAG**: Build Retrieval-Augmented Generation using LlamaIndex to provide detailed, context-aware responses by combining retrieval with generated content.
4. **Develop Conversational Interface:** Build a chatbot interface that manages conversational history and provides relevant responses based on user input.
5. **Feedback Mechanism:** Implement a feedback system to evaluate the quality of responses and improve the chatbot's performance over time.

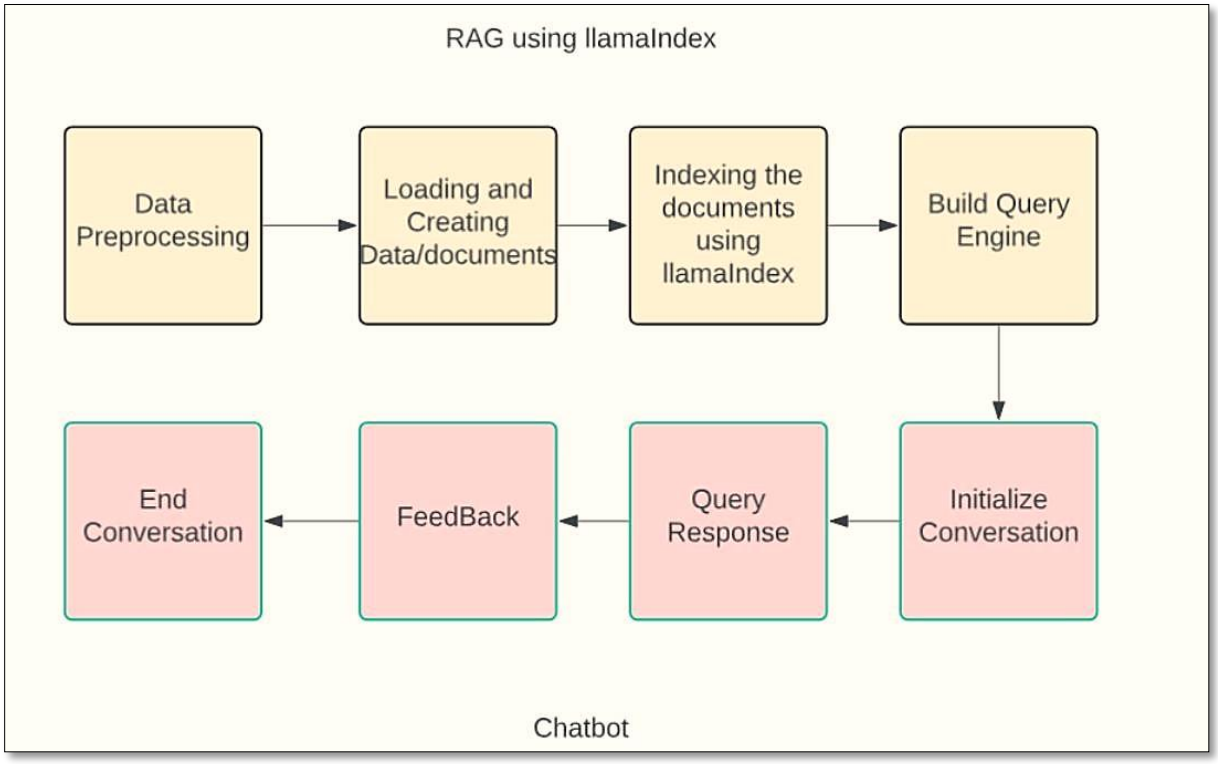
Data Sources

- Myntra Fashion Dataset: The primary data source is a CSV file containing fashion product information such as descriptions, attributes, and ratings etc.,

p_id	name	products	price	colour	brand	img	ratingCount	avg_rating	description	p_attributes
17048614	Khushal K Women B	Kurta, Palazzos, Dup	5099	Black	Khushal K	http://assets.myntasse	4522	4.418398939	black printed kurta wi	addons na body sh
16524740	InWeave Women Ora	Kurta, Palazzos, Flor	5899	Orange	InWeave	http://assets.myntasse	1081	4.11933395	orange solid kurta wi	addons na body sh
16331376	Anubhutee Women I	Kurta, Trousers, Dupa	4899	Navy Blue	Anubhutee	http://assets.myntasse	1752	4.16152968	navy blue embroider	addons na body sh
14709966	Nayo Women Red Fl	Kurta, Trouser, Dupa	3699	Red	Nayo	http://assets.myntasse	4113	4.088986142	red printed kurta with	addons na body sh
11056154	AHIKA Women Blacl	Kurta	1350	Black	AHIKA	http://assets.myntasse	21274	3.978377362	black and green prin	body shape id 424
18704418	Soch Women Red T	Anarkali Kurta	3498	Red	Soch	http://assets.myntasse	184.3122511	4.10104415	colour red solid wove	body shape id 333
14046594	Libas Women Navy E	Kurta, Palazzos, Dup	3599	Navy Blue	Libas	http://assets.myntasse	8695	4.435652674	stately and versatile	addons na body sh

SEMANTIC SEARCH CHATBOT

System Design:



1. Data Preprocessing:

- This step involves cleaning and preparing the raw data to ensure it is suitable for further processing and analysis.
- Removing null values, handling missing data, normalizing text, and any other data cleaning steps.

2. Loading and Creating Data/Documents:

- In this phase, the pre-processed data is loaded and organized into a format that can be indexed.
- Loading the dataset, creating documents for indexing.

3. Indexing the Documents Using LlamaIndex:

- This step involves creating an index of the documents, which allows for efficient querying.
- Parsing the documents into nodes and creating an index.

4. Build Query Engine:

- Here, a query engine is constructed to handle and process user queries.
- Setting up the query engine based on the indexed data.

5. Initialize Conversation:

- This step involves setting up the initial state for the chatbot conversation.
- Initializing variables to store conversation history and other relevant data.

6. Query Response:

- The query engine processes user inputs and generates responses based on the indexed data, this data is then passed in API calls and the relevant response is generated.
- Handling user queries, generating detailed responses, and updating conversation history.

7. Feedback:

- Collecting feedback from users about the responses provided by the chatbot to improve future interactions.
- Asking users for feedback and storing their responses.

8. End Conversation:

- This step concludes the interaction with the user.
- Providing final outputs, saving conversation logs, and exiting the interaction.

Design Choices

1. **Data Preprocessing:** Text fields are cleaned to remove HTML tags, punctuation, and extra spaces for better indexing and retrieval.
2. **Indexing:** LlamaIndex is used to create a vector-based index of the documents to facilitate efficient querying.
3. **Query Engine:** Query Engine is utilized to enhance response quality by combining retrieved data with generated explanations.
4. **Conversational Memory:** The chatbot maintains a conversation history to provide contextually relevant responses.

Challenges Faced

1. **Data Quality:** Handling missing values and cleaning text data to ensure high-quality input for indexing.
2. **Context Management:** Ensuring that the chatbot maintains conversational context over multiple interactions.
3. **RAG by LlamaIndex:** Effectively crafting retrieval and generation in LlamaIndex to produce detailed and relevant responses.

4. **User Feedback:** Collecting and analyzing feedback to continuously improve the chatbot's performance.

README File Fashion Product Chatbot

This project implements a fashion product chatbot using LlamaIndex. The chatbot provides detailed responses about fashion products by leveraging vector-based indexing and Retrieval-Augmented Generation (RAG).

Installation

1. Install dependencies:

```
!pip install llama-index openai pandas
```

Usage

1. Mount Google Drive and Set API Key:

- Ensure Google Drive is mounted and the OpenAI API key is set.

2. Run the Chatbot: Dialogue_Management()

3. Interaction:

- Type queries related to fashion products.

- Type "exit" to end the session.

4. Feedback:

- Optionally provide feedback on responses after the session ends.

Data

- CSV File: `fashion_dataset.csv` from Myntra.