RestoFindAI Chatbot

- **Introduction**

RestoFindAI is a conversational AI chatbot designed to recommend restaurants based on user preferences.

Leveraging the power of large language models (LLMs) and a structured dialogue management system, RestoFindAI aims to provide a seamless and efficient restaurant discovery experience.

This report details the project's objectives, design, and implementation.

- **Dataset**

Sourced **zomato.csv** dataset from Kaggle source - https://www.kaggle.com/datasets/absin7/zomato-bangalore-dataset?select=zomato.csv

Clean up done to remove unintended data and also basic cleanup (Ref:# *zomato_dataset_cleanup.ipynb*)

Clean up dataset (**cleaned_zomato.csv**) is being used in this assignment which have below columns:

* address --> Restaurant Address
* name --> Restaurant Name
* online_order --> Does restaurant have online order facility or not
* book_table --> Does restaurant have facility to book table in advance
* rating --> Rating if restaurant out of 5
* votes --> No. of reviews
* phone --> Contact numbers of restaurant
* location --> Restaurant location/neighborhood
* rest_type --> Type of restaurant like casual dining, cafe
* dish_liked --> Dishes which most people liked
* cuisines --> Cuisines restaurant have
* cost_for_two --> Approx cost of two people

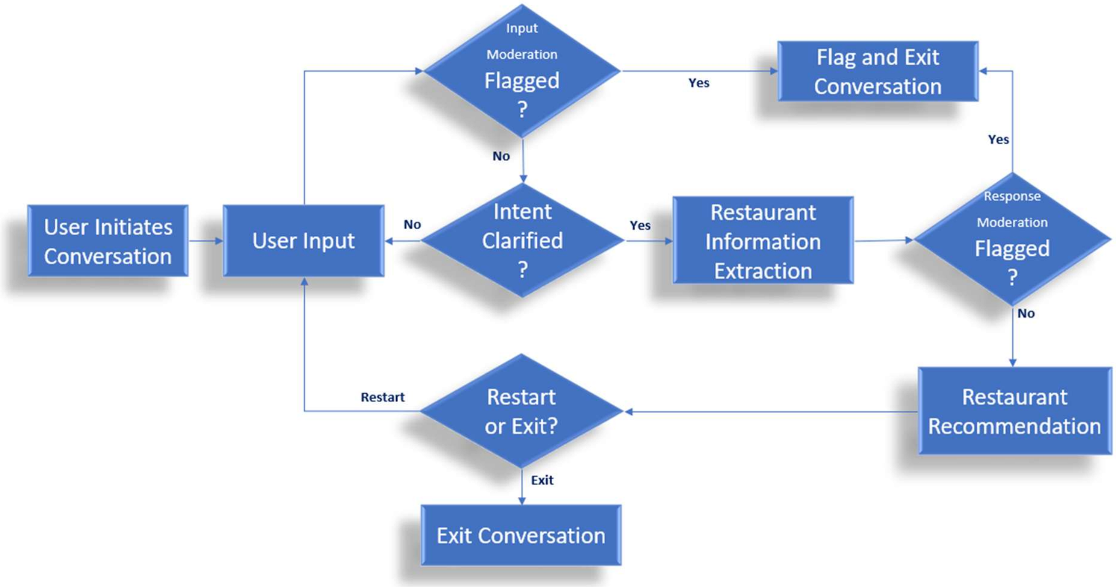
- **Project Objectives**

  - **Develop a conversational AI chatbot**: Create an interactive system that engages users in natural language, allowing them to specify their restaurant preferences.
  - **Accurate Restaurant Recommendation**: Recommend relevant restaurants based on user-provided criteria like cuisine, location, budget, and desired amenities (online ordering, table booking).
  - **Robust Dialogue Management**: Implement a dialogue management system capable of handling various user inputs, maintaining context, and guiding users towards a successful recommendation.

- **User-Friendly Interface**: Provide a simple and intuitive interface (in this case, a Colab notebook environment) for interaction.
- **Moderation and Safety**: Integrate content moderation to prevent inappropriate language and ensure a safe user experience.

- **System Design**

RestoFindAI employs a layered architecture comprising the following components:



1. **Initialization Layer:** Begins the conversation with an introductory message and sets the stage for user interaction.

2. **User Profile Extraction Layer:** Utilizes an LLM to understand and extract key user requirements (cuisine, location, budget, online ordering preference, table booking preference). This layer employs multiple functions:

* `moderation_check()`: Ensures user and assistant messages adhere to safety guidelines.

* `intent_confirmation_layer()`: Validates the completeness and correctness of extracted user information.

* `dictionary_present()`: Confirms the extracted information is in the correct dictionary format.

3. **Restaurant Matching Layer:** Compares the extracted user profile with a restaurant dataset. The key function here is `compare_restaurants_with_user()`, which filters restaurants based on user criteria and provides the top recommendations.

4. **Recommendation Layer:** Presents the top restaurant recommendations to the user in a clear and organized manner, using the `recommendation_validation()` function to filter out poorly rated restaurants and then the `initialize_conv_reco()` function to structure the presentation.

5. **Dialogue Management System**: Orchestrates the entire conversation, managing user inputs, directing them through the different layers, and handling different conversation states. This layer utilizes functions from all previous layers in a loop to manage the ongoing interaction with the user.

- **Implementation Details**

The implementation leverages Python with libraries like Pandas for data manipulation, OpenAI for LLM interactions, and `tenacity` for retry mechanisms.

  - **Data:** The system requires a restaurant dataset in CSV format (the provided code snippet uses `df`).
  - **LLM Interaction**: OpenAI's API powers the core natural language understanding and generation capabilities.
  - **Dialogue Flow:** The `dialogue_mgmt_system` function manages the conversational flow and calls other functions appropriately.
  - **Error Handling:** Includes basic error handling and moderation to improve the system's robustness.

- **Challenges/Lessons Learnt**
  - Any small change in prompts change response.
  - Need to keep no. of tokens being passed in mind while designing chatbot like this.
  - Identifying right dataset is one of the challenges faced.
  - Chat completions API is so powerful and able to bring value with simple coding.
  - With this experience, got confidence that we can implement basic chatbot feature

- **Conclusion**

RestoFindAI provides a foundation for an intelligent restaurant recommendation system. While further development is needed, the prototype demonstrates the system's potential to effectively extract user requirements, match them to suitable restaurants, and engage users in a conversational manner. Future work will focus on improving the system's accuracy, robustness, and user experience.

Developer:

Somasekhar Gangarapu (saisomu@gmail.com)

Credits:

Dataset owner from Kaggle

Kshitij Jain who gave best sessions on GenAI

All professors, Mentors, buddies and staff at upGrad and IIITM