Zomato Gen AI Internship Assignment

Restaurant Data Scraper & RAG-based Chatbot

Overview

In this assignment, you will build an end-to-end Generative AI solution combining web scraping with a Retrieval Augmented Generation (RAG) chatbot. This project simulates a real-world application that could enhance Zomato's user experience by allowing customers to ask natural language questions about restaurants and receive accurate, contextual responses based on up-to-date information.

Problem Statement

Users often have specific questions about restaurants that aren't easily answerable through traditional search. For example:

- "Which restaurant has the best vegetarian options in their menu?"
- "Does ABC restaurant have any gluten-free appetizers?"
- "What's the price range for XYZ restaurant's dessert menu?"
- "Compare the spice levels mentioned in the menus of restaurants A and B"

Your task is to create a solution that can answer these types of questions by:

1. Collecting real restaurant data through web scraping
2. Processing and storing this information appropriately
3. Building a RAG-based chatbot that retrieves relevant information and generates helpful responses

Assignment Requirements

1. Web Scraper Component

- Create a web scraper to extract information from 5-10 restaurant websites of your choice
- The scraped data should include at minimum:
    - Restaurant name and location
    - Menu items with descriptions and prices (if available)
    - Special features (e.g., vegetarian options, spice levels, allergen information)
    - Operating hours and contact information
- Organize the scraped data into a structured format suitable for retrieval
- Implement proper error handling and respect website robots.txt policies
- Document any assumptions or limitations in your scraping approach

2. Knowledge Base Creation

- Transform the scraped data into a well-structured knowledge base
- Implement appropriate text preprocessing (cleaning, normalization)
- Create an efficient indexing system for information retrieval
- Ensure the knowledge base is organized to support different types of queries

3. RAG-based Chatbot

- Utilize Hugging Face's freely available models and APIs
- Implement the RAG architecture with:
  - A retrieval component that finds relevant information from your knowledge base
  - A generation component that produces natural, helpful responses
- Ensure the chatbot can handle at least these query types:
  - Menu item availability and details
  - Restaurant feature comparisons
  - Price range inquiries
  - Dietary restriction questions
- Implement basic conversation history management
- Handle edge cases (e.g., out-of-scope questions, ambiguous queries)

4. User Interface

- Create a simple interface for interacting with your chatbot
- This can be a command-line interface, Streamlit app, Gradio interface, or similar
- Include clear instructions for running and using your application

Technical Constraints

- Use Python as your primary programming language
- For web scraping: Use libraries like BeautifulSoup, Scrapy, or similar
- For the RAG implementation: Use Hugging Face's models accessible via their free tier
- Host your code on GitHub with clear documentation
- Do not use any paid APIs or services that require credit card information
- Your solution should be reproducible on a standard computer without specialized hardware

Deliverables

1. **Code Repository**
   - Complete, well-documented source code for all components
   - Clear setup and running instructions

2. **Scraped Dataset**
    - The structured data collected from restaurant websites
    - Documentation of the data schema and collection methodology
3. **Technical Documentation (PDF or Markdown)**
    - System architecture explanation
    - Implementation details and design decisions
    - Challenges faced and solutions implemented
    - Future improvement opportunities
4. **Demo Video (3 minutes max)**
    - Brief walkthrough of your implementation
    - Demonstration of at least 3 sample interactions showing different query types
    - Explanation of how your solution could scale or be improved

Evaluation Criteria

1. Web Scraping Component (25%)

- **Data Comprehensiveness (10%)**: Quality and quantity of relevant restaurant information collected
- **Data Organization (10%)**: Structure, cleanliness, and usability of the scraped data
- **Code Quality (5%)**: Efficiency, robustness, and ethics of the scraping implementation

2. RAG Implementation (30%)

- **Architecture Design (10%)**: Appropriate implementation of RAG methodology
- **Retrieval Effectiveness (10%)**: Ability to find relevant information for different query types
- **Response Generation (10%)**: Quality and accuracy of the generated responses

3. Response Quality (25%)

- **Accuracy (10%)**: Correctness of information provided
- **Relevance (5%)**: Appropriateness of responses to queries
- **Helpfulness (5%)**: Practical utility of the responses
- **Handling Edge Cases (5%)**: Graceful management of difficult or out-of-scope questions

4. Technical Excellence (20%)

- **Code Quality (5%)**: Readability, organization, and best practices
- **Documentation (5%)**: Clarity and completeness of documentation
- **Reproducibility (5%)**: Ease of setting up and running the solution
- **Innovation (5%)**: Creative approaches or additional features beyond the requirements

Submission Guidelines

- Submit your complete GitHub repository link
- Ensure all code is well-commented and follows best practices
- Include a README.md with clear setup and usage instructions
- Upload your demo video to a hosting platform and include the link in your submission
- Complete all deliverables by Saturday, 22nd April, 11:59 PM IST

Notes

- You may use publicly available libraries and resources, but ensure proper attribution
- You are encouraged to discuss approaches and share resources with fellow participants, but your implementation must be your own
- Questions about the assignment can be directed to garvit.bhardwaj@zomato.com
- This assignment is designed to test both your technical skills and your ability to build practical AI solutions relevant to Zomato's business

Good luck! We're excited to see what you create.