CSE 535 - Information Retrieval
Project 3 - Wikipedia Q/A System

Anirudh Anilkumar
MS, CSE
UB ID: 50544945

Chaitali Thakkar
MS, ESDS
UB ID: 50557808

I. Introduction

In this project, we successfully developed a Wiki Q/A Chatbot that integrates both interactive chit-chat and topic-specific information retrieval. By utilizing a dataset of over 50,000 unique Wikipedia documents, the chatbot covers diverse topics such as health, technology, and politics. It employs a non-rule-based chit-chat model for fluid conversations, while its topic classification system ensures that responses are accurate and relevant. The chatbot retrieves and summarizes information to provide concise answers to user queries. We also implemented exception handling for smooth interactions, and a web-based interface with data visualizations enables effective tracking and analysis of user engagement.

II. Methodology

A. Wiki Scraper

To develop the knowledge base for the Wiki Q/A Chatbot, we used the Wikipedia library to scrape data from Wikipedia. We focused on 10 major topics—Health, Environment, Technology, Economy, Entertainment, Sports, Politics, Education, Travel, and Food—and ensured we collected around 7000 unique documents for each topic. The scraping process involved extracting comprehensive articles related to each topic, such as health statistics, technological advancements, and political policies. We preprocessed the scraped data by cleaning irrelevant content and ensuring its consistency. This data was then stored in a structured format for easy retrieval, reducing processing time and improving the overall performance of the Q/A system.

B. Chit Chat

For the chit-chat component, we incorporated BlenderBot, a pre-trained conversational AI developed by Facebook. BlenderBot is capable of engaging in dynamic and natural conversations without being bound by a predefined set of rules. As a pre-trained model, it has been trained on a large dataset of diverse conversations, allowing it to handle a wide variety of topics. The bot was fine-tuned to ensure smooth and fluid interactions, enabling users to converse without interruptions unless explicitly desired. The goal of the chatbot was to maintain a friendly and engaging interaction, ensuring users felt comfortable in casual conversations.

C. Topic Analysis

To enhance the chatbot's ability to understand and classify user queries, we integrated zero-shot classification using the model 'MoritzLaurer/bge-m3-zeroshot-v2.0', a powerful natural language processing model. This model enables the chatbot to classify queries into specific topics without requiring additional training on domain-specific data. By using this model, the chatbot can effectively identify the core subject of a user's query and provide relevant responses. The system is also capable of handling multi-topic conversations, adapting the discussion as the user's interests shift from one topic to another. Leveraging this zero-shot classification approach ensures that the chatbot can maintain a relevant and topic-specific conversation, enhancing user satisfaction.

D. Wiki Q/A Bot

For the Wiki Q/A Bot, after classifying the user query, we implemented a document retrieval and summarization process to provide concise and accurate responses. We preprocessed the scraped data to reduce running time by cleaning and organizing the content, removing any irrelevant or redundant information, and structuring it for efficient retrieval. We used TF-IDF vectorization to represent the documents as numerical vectors, which allowed the system to retrieve the most relevant content based on the user's query. The retrieved documents were then summarized, and we utilized a generative model to combine the information and generate coherent answers. This approach ensured that the bot not only fetched relevant data but also provided users with an understandable and summarized response, minimizing information overload and improving response time.

E. Exception Handling

To ensure smooth interactions, we implemented extensive exception handling throughout the chatbot's processes. Since user queries and document retrieval can sometimes lead to errors or ambiguities, we focused on preventing the system from crashing under such circumstances. For example, if a query cannot be classified or a document fails to load, the bot

gracefully handles these exceptions by providing users with clear error messages or rephrasing prompts. This ensures that the chatbot remains operational, maintains user engagement, and avoids interruptions during conversations.

F. Visualization

In this project, we implemented three main visualizations to enhance the user experience and monitor the bot's performance:

1) **Topic Distribution**: This visualization shows the frequency of queries across different topics, helping to identify which areas users are most interested in.
2) **Response Time of Every Query**: This metric tracks how quickly the bot responds to each user query, providing insights into the system's performance and areas for improvement.
3) **Time Between Queries**: This visualization measures the average time between consecutive user queries, offering valuable insights into user engagement and interaction patterns.

These visualizations help monitor the bot's performance, gather insights on user behavior, and ensure that improvements can be made based on actual usage patterns.

G. Chat UI

The user interface (UI) for our Wiki Q/A Chatbot was developed as a web application, providing users with an intuitive way to interact with the bot. The interface includes several key features to enhance user experience. First, users can select from multiple topics, allowing them to narrow down the scope of the conversation and ensure the chatbot provides relevant responses based on their choice. Additionally, the chatbot includes a chit-chat option, enabling users to engage in casual conversations. This feature allows the bot to switch to a more informal mode, offering natural and fluid interactions. Lastly, the chat interface includes an end chat option, allowing users to gracefully conclude their conversation at any time. These features ensure that the bot provides both flexibility and a seamless experience, accommodating various user preferences throughout the interaction.

III. Work Breakdown

Team Member	Tasks
Anirudh	Wiki Scraper, Chit Chat, Topic Analysis, Frontend-Backend Integration, Visualizations
Chaitali	Wiki Q/A Chatbot, Exception Handling, Chat UI, Report

TABLE I
Work Breakdown among Team Members

IV. Sample Screenshots

We have inserted the screenshots of the visualizations and the chatbot below:

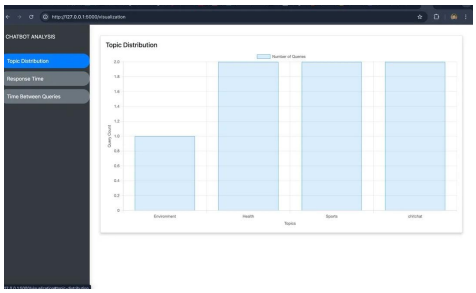

Fig. 1. Visualization - I

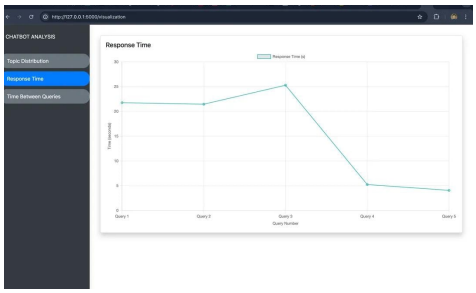

Fig. 2. Visualization - II

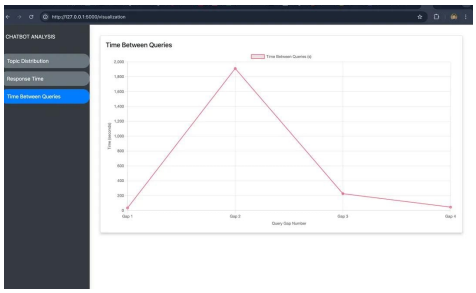

Fig. 3. Visualization - III

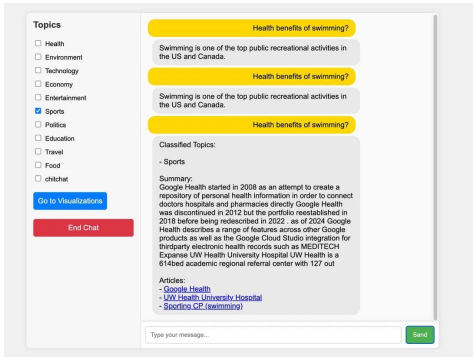

Fig. 4. Chat Bot - I

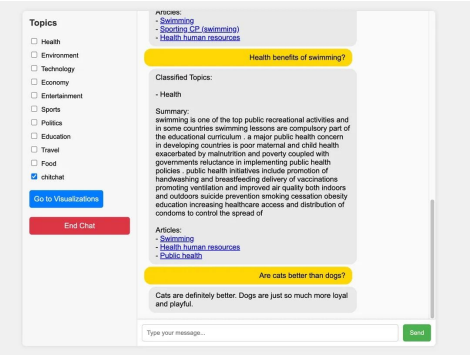

Fig. 5. Chat Bot - II

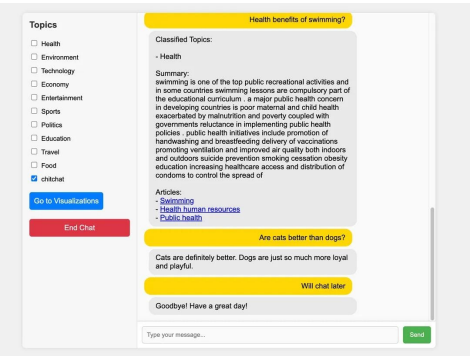

Fig. 6. Chat Bot - III

V. Conclusion

In this project, we successfully developed a Wiki Q/A Chatbot that combines dynamic chit-chat capabilities with topic-specific information retrieval. By leveraging pre-trained models such as BlenderBot for the chit-chat component and BERT for topic classification, we created a highly interactive and intelligent system capable of handling a wide variety of queries. Our approach to document retrieval using TF-IDF vectorization and subsequent summarization ensures that users receive concise and relevant information based on their queries. The integration of a web-based user interface with real-time feedback and analytical visualizations allowed for an engaging user experience and effective tracking of the bot's performance. Additionally, the implementation of exception handling ensured that the bot could gracefully handle errors and provide a smooth interaction, even in the case of unexpected inputs. Overall, this project demonstrates the potential of combining advanced machine learning models with interactive interfaces to create a powerful, user-friendly chatbot capable of providing both casual and informative conversations.