

PHOENIX CHATBOT

Information Retrieval

1. Introduction

Meet the PHOENIX CHATBOT, your versatile conversational companion. Unlike other chatbots, it can chat about ten different wikipedia related topics and even switch between topics in a single conversation. Whether you're curious about technology, health, politics or any other subject, this chatbot is here to chat, inform, and make your conversations engaging. Just say hello to the chatbot and get your response which is both informative and flexible.

With its wide range of topics and smooth transitions between them, the PHOENIX CHATBOT ensures you can explore multiple interests in a single chat. No more rigid, one-topic conversations – this chatbot is all about dynamic and meaningful dialogues, allowing multi-topic chats with users as well. Experience the endless possibilities of chatting with an intelligent information retrieval system like never before!

2. Methodology

2.1 Back End:

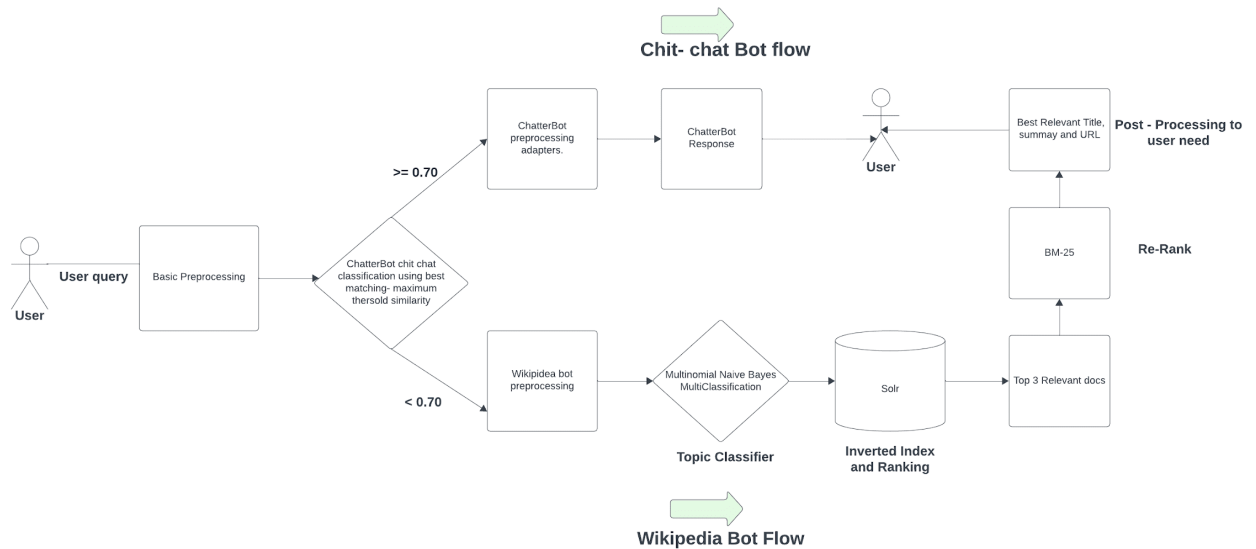
Firstly, the user query undergoes preprocessing to filter out nonsensical queries, such as those containing only white spaces or punctuations. Subsequently, it enters the chatterbot classification logic adapter to determine the best match. If the confidence or maximum similarity threshold is ≥ 0.70 , it's classified as a chit chat query; otherwise, it's classified as a Wikipedia query.

For chit-chat queries, the chatterbot utilizes preprocessing adapters and generates a response by examining the trained corpus, specifically the chatterbot English corpus, aiming to provide the most suitable and confidently matched result.

If the user query is classified as a Wikipedia query, the Wikipedia bot is activated. The query undergoes preprocessing steps, involving the removal of punctuations, stopwords, and lemmatization. If the user doesn't specify a topic, a multinomial naive Bayes classifier predicts the topic. Alternatively, if the user provides the topic, this prediction step is bypassed.

Next, the processed query and topic filters are passed to Solr. The Solr system has been already indexed with all the 50,000 scraped wiki documents (5000 docs per topic) using a PySolr script. The results are refined by considering tf-idf, phrase queries, boosting title and summary phrase weights, scoring the documents, extended dismax query parser, resulting in the retrieval of the top 3 highly relevant documents. These documents are then re-ranked using the BM25 algorithm.

Finally, the obtained documents undergo post-processing, and the coherent title, summary, and URL are presented to the user in the necessary format.



2.2 User Interface:

Chatbot Interface:

The chatbot, a central feature of your web application, is designed to provide users with an interactive and engaging experience. Embedded within a clean and user-friendly interface, the chatbot is represented by an iconic symbol that, upon clicking, expands to reveal a conversational window. Users can interact with the chatbot by selecting topics from a range of checkboxes, including Politics, Education, Health, and more, or by typing their queries directly into a text input field. The chatbot is programmed to respond with relevant information, including summaries, titles, and URLs, enhancing the user experience with informative and immediate feedback. The interface also includes options to close the chat window, ensuring a seamless and non-intrusive integration into the overall web page layout.

Visualization Features

The front-end UI boasts dynamic visualization features that provide real-time insights into user interactions with the chatbot. Two primary visualizations are included: a bar chart (interactionChart) and a pie chart (topic-distribution-chart). The bar chart dynamically displays the number of interactions per topic, offering a clear quantitative view of user engagement across different subjects. The pie chart, on the other hand, categorizes interactions into three segments - 'Chit-Chat', 'Error', and 'Wiki Bot', providing a quick visual representation of the chatbot's usage pattern. These visualizations update in real time as users interact with the

chatbot, giving immediate feedback on the topics that are most engaging or areas where users encounter issues.

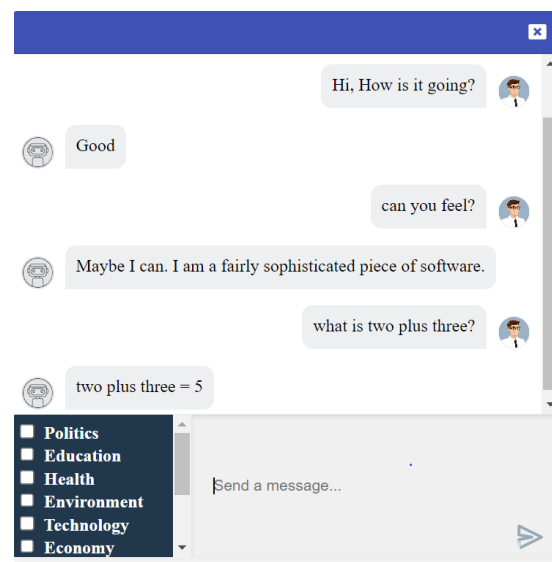
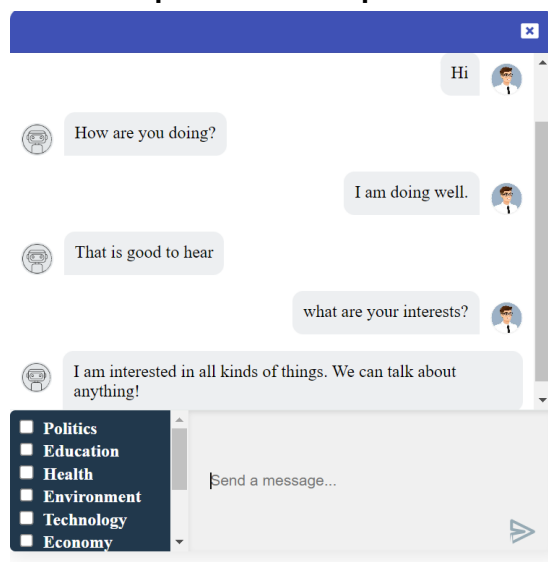
Technologies

The front-end UI is built using a combination of modern web technologies. HTML5 is used for structuring the content, ensuring a semantic and accessible markup. CSS3, particularly through a dedicated stylesheet (new.css), is employed to style the interface, offering a visually appealing and responsive design. JavaScript, with the help of jQuery, is used to add interactive elements and dynamically update content based on user interactions. The integration of Chart.js, a versatile and powerful charting library, allows for the creation and real-time update of the bar and pie charts, enriching the UI with meaningful and engaging data visualizations.

Overall, the front-end UI of your web application presents a harmonious blend of functionality and aesthetics, driven by user interaction and enhanced by real-time data visualization. The use of contemporary web technologies ensures a seamless and intuitive user experience, making the chatbot not only a source of information but also an engaging tool for user interaction and data analysis.

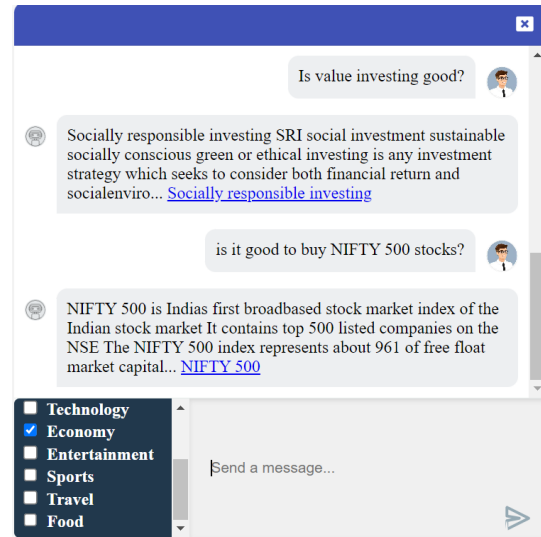
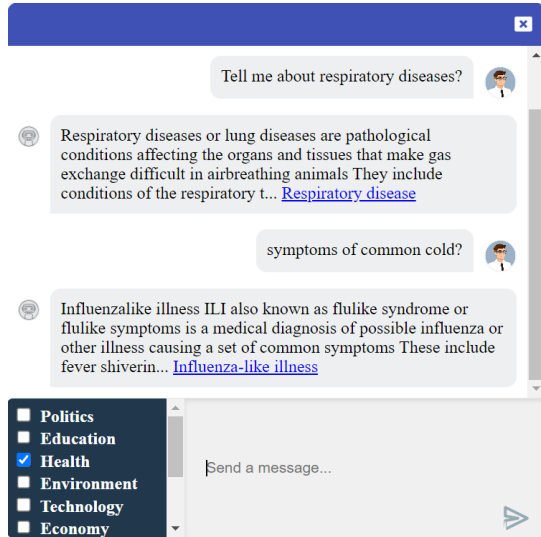
3. Sample UI screenshots

Chit Chat queries and response



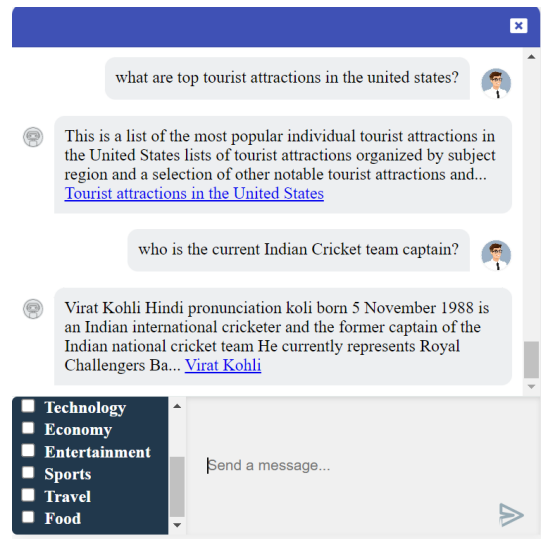
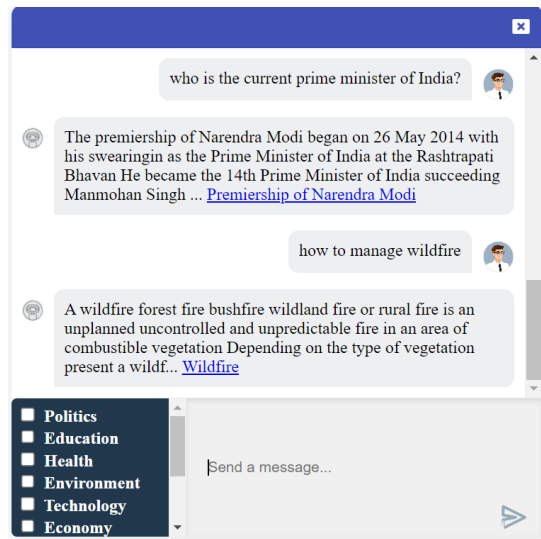
Wikipedia queries with topic selection

The response will be a coherent summary, title+link. Users can read more about the response by clicking on the link.



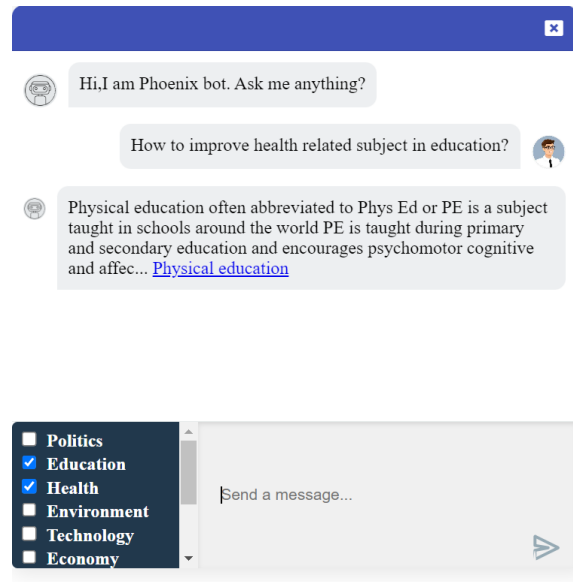
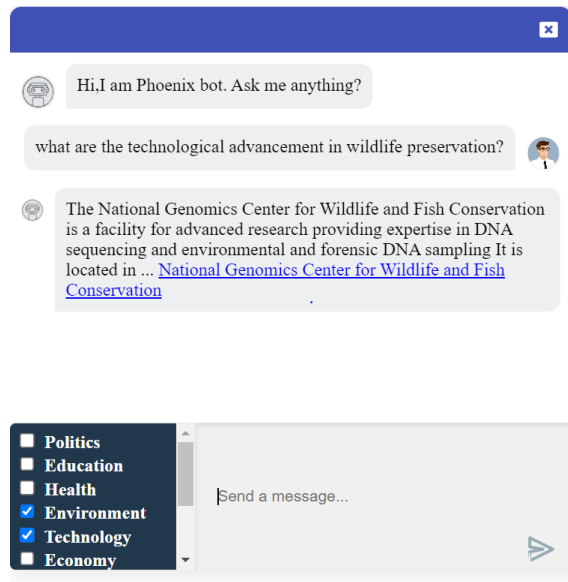
Wikipedia queries without topic selection

The backend topic classifier automatically predicts the topic and gives a relevant response to the user query.



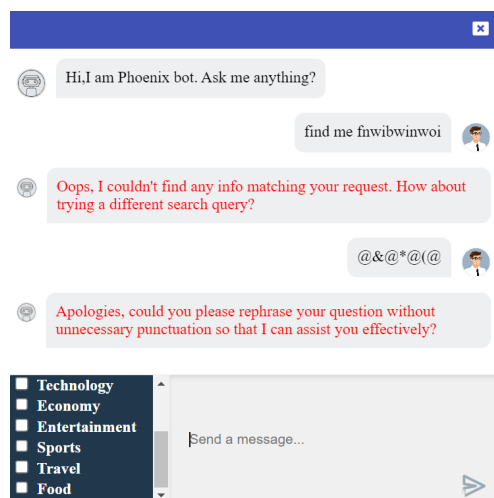
Multi topic Chat. (Bonus)

Users can select two or more topics so that we will fetch the documents only from that related topic filters. You can see the examples below.



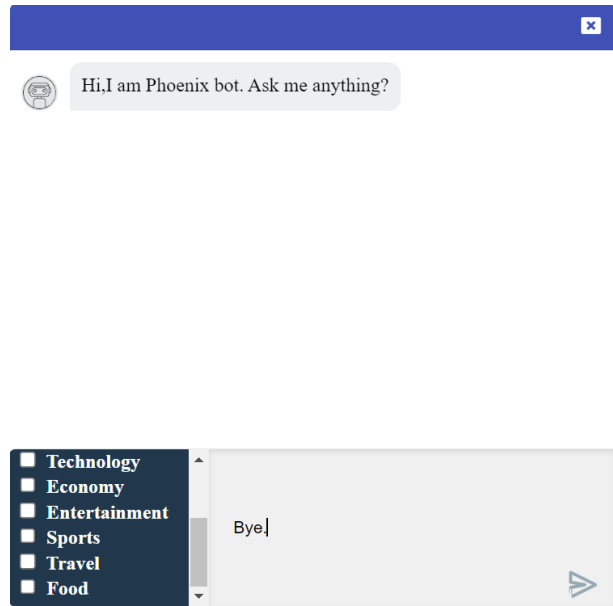
Sample Error Responses

If Enter queries that are wrong or they aren't matching with chit chat bot or wikibot then error responses will be displayed.



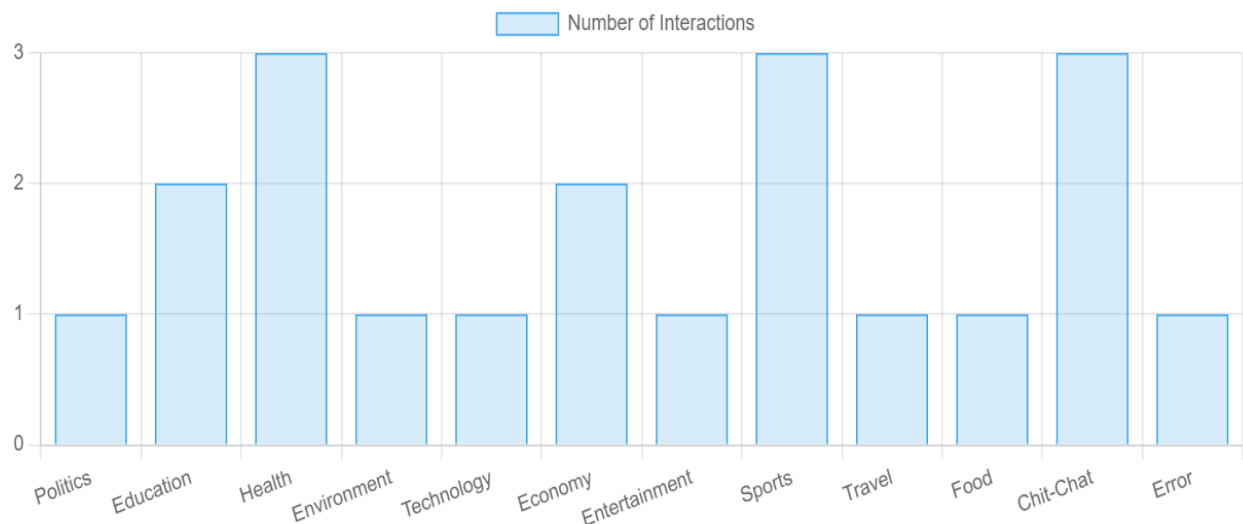
Sample Exit Queries.

We can use any exit queries like quit, exit, bye etc. so the chatbot window closes with goodbye only displaying the chat icon on your screen as shown in below figure or you can close manually by using the close button on top right.

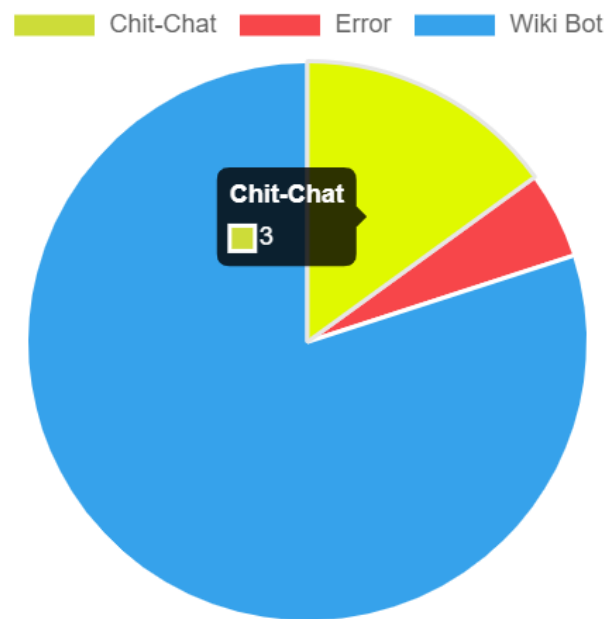


4. Sample Visualizations Screenshots.

Number of Interactions is a dynamically updated bar graph based on user interactions. It depicts the frequency of user interaction with the chat and different topics.



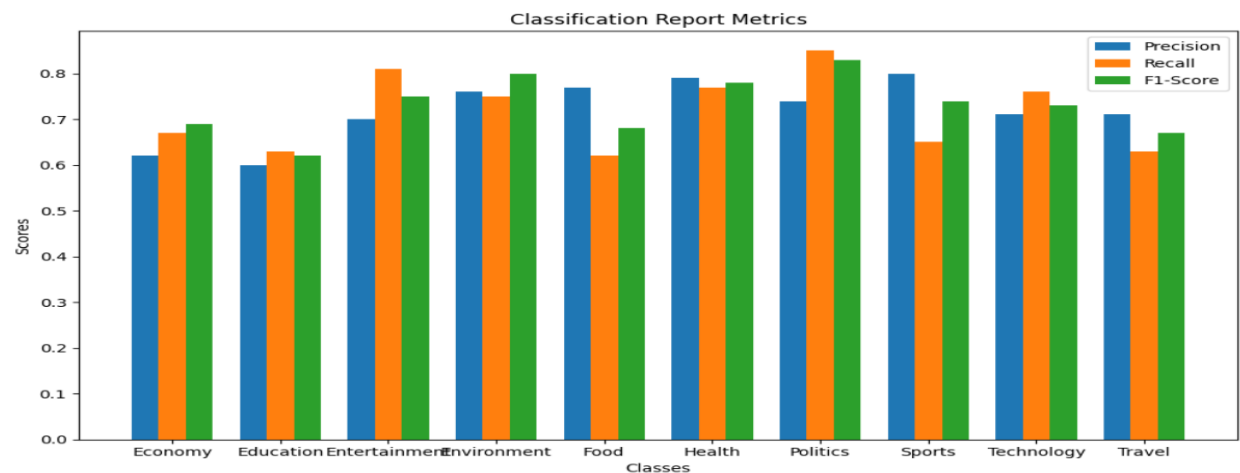
The following circular graph illustrates the user engagement metrics between Chit-Chat and Wiki Bot, alongside indicating the respective error rates associated with each interaction.



Along with the above visualizations, UI has 4 radio buttons where on selection of each button specifies one metric:

Visualization 1:

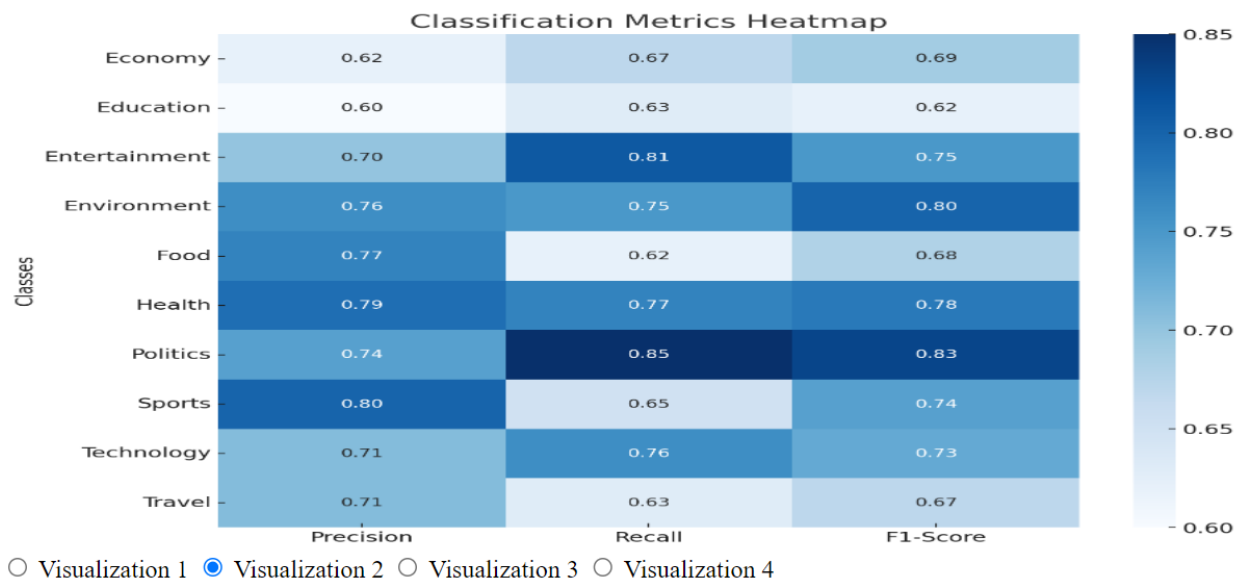
This visual representation displays the precision, recall, and F1-score metrics for individual topics, derived from the backend's predictions regarding those specific topics.



☒ Visualization 1 ☐ Visualization 2 ☐ Visualization 3 ☐ Visualization 4

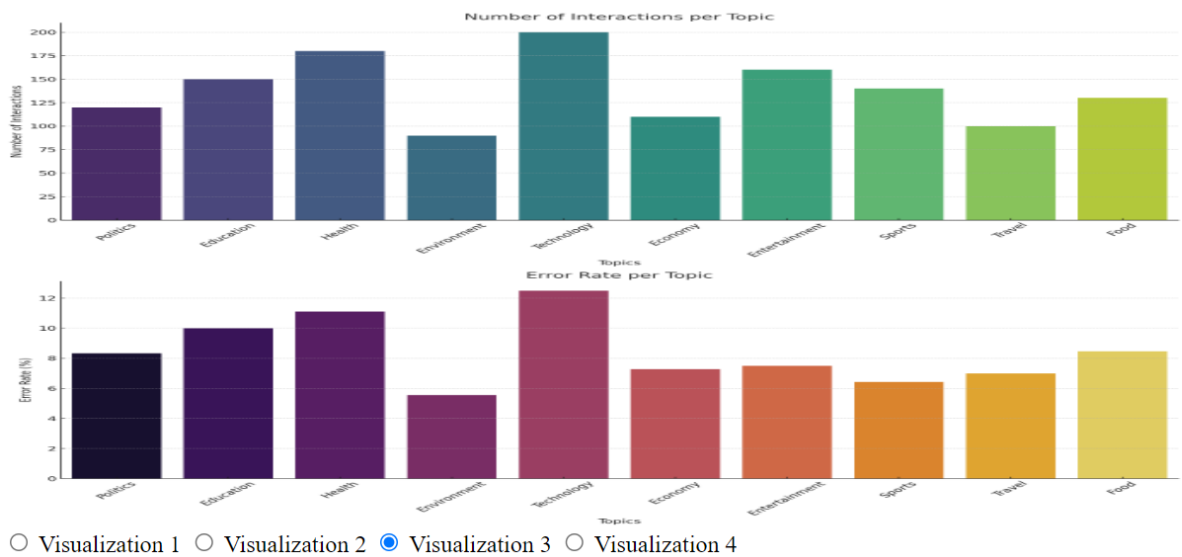
Visualization 2:

This visualization shows the precision, recall, and F1-score for each class (topic) with varying intensities of blue. Darker shades represent higher values, making it easier to identify which classes have higher metrics.



Visualization 3:

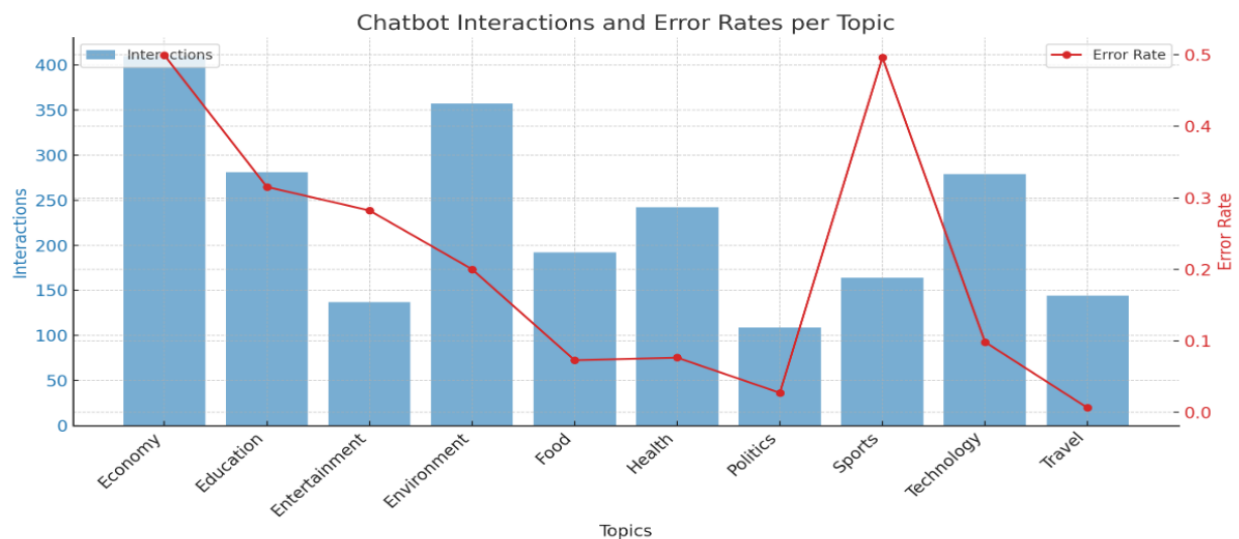
Visualization 3 is performed on sample data with 20k dataset records.



- The first bar chart displays the number of interactions per topic, which can give an insight into which topics the chatbot discusses more frequently.
- The second bar chart illustrates the error rate per topic, providing a clear view of where most of the errors are occurring in terms of percentage.

Visualization 4:

Visualization 4 is performed on sample 40k Wiki Dataset records. It also depicts the number of interactions per topic, which can give an insight into which topics the chatbot discusses more frequently and illustrates the error rate per topic, providing a clear view of where most of the errors are occurring in terms of percentage.



☐ Visualization 1
 ☐ Visualization 2
 ☐ Visualization 3
 ☒ Visualization 4

These visualizations can help you identify which topics may need further refinement in the chatbot's knowledge base or where to focus on improving error handling and responses.

5. Work breakdown by teammates (All work done for the project and member contributions)

Manikanta Kalyan Gokavarapu - 50465129

Responsible for the entire backend architecture, development and deployment (End to End backend development).

Tasks Performed:

Chit Chat Bot - Implemented the complete chit chat bot from scratch using chatter bot and chatterbot english training corpus.

Classifiers - Implemented various classifiers like Support Vector machine, Decision tree, BERT and Multinomial Naive Bayes for topic classification and finally choose the best performing classifier i.e Naive bayes for our bot.

Wikipedia Q/A Bot - Implemented complete Wikipedia Q/A bot from scratch by following the below steps. 1) Prepared a wiki scraping script and scrapped all the 50,000 topic related documents. 2) Using Solr indexed all the documents, wrote pysolr scripts to get the most relevant response for the userquery. 3) Used BM -25 for re-ranking and post processed all the documents to generate a coherent summary.

Exception Handling - Identified all the edge cases and handled most of the exceptions, so that the bot never crashes

Flask API - Built flask api by integrating and designing a flow for all the individual components and tested the flask API using postman.

Deployment - Installed all the required dependencies in the server and Deployed the flask api on the GCP server.

Report - Updated the Methodology, Flow diagrams and UI screenshots in the Report.

Yaswanth Reddy Thippaluri - 50479678

Front-End Web Development: I was responsible for the front-end development of the application, utilizing HTML, CSS, and JavaScript to craft an intuitive and visually appealing interface. My focus on user experience and design aesthetics ensured ease of navigation and interaction, making the application accessible and engaging for users. My efforts in UI/UX design were crucial in harmonizing the application's functionality with its visual appeal.

Overall, my contributions were pivotal in creating a comprehensive web application that seamlessly blends conversational AI with intuitive data visualization, offering a user-centric and analytically robust platform.

Dynamic Data Visualization: Recognizing the importance of data in understanding user interactions, I integrated real-time visualizations using Chart.js. These dynamic bar and pie charts update based on user interactions, providing critical insights into user preferences and chatbot efficacy. This implementation not only brought an analytical dimension to the application but also enriched the user experience through interactive and informative elements.

Gnana Abhinay Vadlamudi - 50496402

Instances Setup: My responsibilities included setting up the cloud instance according to specific requirements and configuring firewalls. Subsequently, I deployed the bot with an implementation that facilitated virtual access.

Scraped Data: I contributed to scraping several topics, formatting the data, and ensuring it was readily available for immediate utilization.

Data Visualization: I comprehended the flow of both UI and Backend systems, and assisted in designing visualizations that enhance comprehension of the chatbot's coherence.

Report: I meticulously organized the completed tasks within the report, ensuring it was presented in the required format and with clear formatting of details.

6. Conclusion

In conclusion, the PHOENIX CHATBOT redefines the way we interact with information retrieval systems. With its ability to seamlessly handle conversations on ten different Wikipedia-related topics and effortlessly switch between them, it offers users a unique and engaging experience. Whether you're inquiring about technology, health, politics, or any other subject, this chatbot is your versatile companion.

The methodology behind this chatbot is robust, involving precise preprocessing, intelligent classification, and advanced Solr-based information retrieval. It ensures that users receive accurate and relevant information presented in a comprehensible manner. The chatbot's dynamic approach enables multi-topic conversations, breaking free from the constraints of one-dimensional interactions.

From chit-chat queries to Wikipedia-related inquiries and even multi topic discussions, the PHOENIX CHATBOT offers a wide range of capabilities, providing informative responses tailored to your needs. With its extensive features and intelligent architecture, it opens the door to endless possibilities in the world of conversational information retrieval.