A REPORT ON

PRAGMATIC ANALYSIS OF WHATSAPP CHAT USING NLP

SUBMITTED TO THE MUMBAI UNIVERSITY, MUMBAI
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING
(INFORMATION TECHNOLOGY)

SUBMITTED BY

MR. HARSHRAJ SUDHAKAR DESHMUKH	(UID 120IT1234A)
MR. AAKASH DAMAJI DHOTRE	(UID 120IT1246A)
MR. PRANAV MANOHAR CHAVAN	(UID 120IT1105A)
MR. ADITYA VIJAYKUMAR GHARAT	(UID 120IT1084A)

Under the Guidance of
Dr. SANJAY B. WAYKAR

DEPARTMENT OF INFORMATION TECHNOLOGY

MGM'S COLLEGE OF ENGINEERING AND TECHNOLOGY

NAVI MUMBAI, KAMOTHE- 410 209

MUMBAI UNIVERSITY
A.Y. 2023-2024

CERTIFICATE

This is to certify that the Project report entitles

"**PRAGMATIC ANALYSIS OF WHATSAPP CHAT USING NLP**"

Submitted by

MR. HARSHRAJ SUDHAKAR DESHMUKH	(UID 120IT1234A)
MR. AAKASH DAMAJI DHOTRE	(UID 120IT1246A)
MR. PRANAV MANOHAR CHAVAN	(UID 120IT1105A)
MR. ADITYA VIJAYKUMAR GHARAT	(UID 120IT1084A)

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of **Dr. Sanjay B. Waykar** and it is approved for the partial fulfillment of the requirement of Mumbai University, for the award of the degree of **Bachelor of Engineering** (Information Technology).

Dr. Sanjay B. Waykar
Guide
Department of Information Technology

Dr. Swati Sinha
Head
Department of Information Technology

Dr. Geeta S. Lathkar
Director
MGMCET, Navi Mumbai

Place: Kamothe
Date:

PROJECT REPORT APPROVAL

This Project entitled **"PRAGMATIC ANALYSIS OF WHATSAPP CHAT USING NLP"** by **Harshraj Sudhakar Deshmukh , Aakash Damaji Dhotre, Pranav Manohar Chavan, Aditya Vijaykumar Gharat** is approved for the degree of Bachelor of Engineering in Information Technology.

.....

External Examiner

.....

Internal Examiner

Place: Kamothe
Date:

DECLARATION

We declare that this written submission represents our ideas in our own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the source which has thus not been properly cited or from whom proper permission has not been taken when needed.

Sr. No.	Name of Student	UID	Signature
1	Harshraj Sudhakar Deshmukh	120IT1234A	
2	Aakash Damaji Dhotre	120IT1246A	
3	Pranav Manohar Chavan	120IT1105A	
4	Aditya Vijaykumar Gharat	120IT1084A	

Place: Kamothe
Date:

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people, and we are extremely fortunate to have received support throughout the completion of our project work and we would like to express our heartfelt gratitude to all those who have contributed.

It is a matter of great pleasure for us to submit the project report on "Pragmatic Analysis of WhatsApp Chat using NLP" as a part of our curriculum.

First and foremost, we would like to thank our Director, Dr. Geeta S. Lathkar for providing us with the opportunity to undertake this project. We are grateful for her support and encouragement.

We extend our sincere thanks to our Head of the Department, Dr. Swati Sinha and our project guide, Dr. Sanjay B. Waykar for their invaluable guidance, mentorship and insightful feedback, which played a pivotal role in shaping our project and enhancing our understanding of the subject matter.

We would also like to express our gratitude to all the faculty members of the Information Technology department for their support, encouragement, and expert advice that helped us navigate through the challenges encountered during the course of this project.

And last but not the least a special thanks goes to the team members, who helped each other to assemble the information and gave suggestions to complete the project.

NAME OF THE STUDENTS

HARSHRAJ SUDHAKAR DESHMUKH
AAKASH DAMAJI DHOTRE
PRANAV MANOHAR CHAVAN
ADITYA VIJAYKUMAR GHARAT

ABSTRACT

In today's digitally interconnected world, instant messaging platforms have become integral to interpersonal communication. This project presents WhatsApp Chat Analyzer, a tool designed to analyze and interpret communication patterns within WhatsApp chat data. The objective of this project is to provide users with insights into their messaging habits, including frequency of interactions, popular topics and sentiment analysis.

The WhatsApp Chat Analyzer operates by extracting and parsing chat data from WhatsApp text files. It utilizes natural language processing techniques to preprocess and analyze the textual content. Key features include message frequency analysis, word frequency analysis and sentiment analysis.

Message frequency analysis enables users to visualize the distribution of messages over time, identifying highs and lows in conversation activity. Word frequency analysis highlights frequently used words and phrases, offering insights into the topics most discussed within the chat. Sentiment analysis categorizes messages as positive, negative, or neutral, allowing users to gauge the overall tone of the conversation.

Furthermore, the WhatsApp Chat Analyzer provides graphical representations of the analyzed data, including line charts, word clouds and sentiment histograms. These visualizations aid in the interpretation of communication patterns and facilitate the identification of trends and outliers.

The practical applications of the WhatsApp Chat Analyzer are diverse, ranging from personal reflection and self-awareness to sociological research and data-driven decision-making. By offering users a comprehensive understanding of their messaging behavior, this tool empowers individuals to enhance their communication skills and foster meaningful connections.

In conclusion, the WhatsApp Chat Analyzer is a valuable tool for gaining insights into communication patterns within WhatsApp chats. Its user-friendly interface, coupled with powerful analytical capabilities, makes it a versatile asset for individuals and researchers alike seeking to understand the dynamics of digital communication.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 MOTIVATION

Here are some potential motivations for such a project:

1. Personal Interest: The developer might have a personal interest in understanding communication patterns, sentiment analysis, or other insights derived from WhatsApp conversations.
2. Educational Purposes: Developing this project could be a way for the developer to learn and practice various programming skills, such as data manipulation, natural language processing and data visualization.
3. Professional Development: Building a project like this could enhance the developer's skills and potentially serve as a portfolio piece to showcase their abilities to potential employers or clients, especially if they are interested in data analysis or software development roles.
4. Community Contribution: Sharing the project on GitHub could be a way for the developer to contribute to the open-source community, allowing others to benefit from the code and potentially contribute improvements or extensions to it.
5. Personal Use: The developer might have developed this tool for their own use, to gain insights into their own WhatsApp conversations, track certain metrics, or simply for curiosity's sake.

Overall, the motivation behind the project likely stems from a combination of personal interest, educational goals and the desire to contribute to the developer community.

1.2 PROBLEM DEFINITION

The problem definition of the "Pragmatic Analysis of WhatsApp Chat Using NLP" project is to create a tool that can analyze WhatsApp chat data and provide various insights and visualizations. Here's a breakdown of the problem definition:

1. Data Input: The project aims to handle WhatsApp chat data as input. This data typically includes text messages exchanged between individuals or groups on the WhatsApp platform.
2. Data Processing: The tool needs to process the input data efficiently, extracting relevant information such as message timestamps, sender information and message content.
3. Analysis: The primary goal of the project is to analyze the WhatsApp chat data to derive meaningful insights. This may involve tasks such as:
    - Sentiment Analysis: Determining the overall sentiment of the conversations.
    - Message Frequency: Analyzing the frequency of messages sent by different users or within specific time frames.
    - Word Frequency: Identifying the most commonly used words or phrases in the conversations.
    - Media Usage: Analyzing the usage of media files (e.g., images, videos) within the chats.
    - Conversation Trends: Identifying trends or patterns in the conversations over time.
4. Visualization: The project aims to present the analysis results in an understandable and visually appealing manner. This could involve creating charts, graphs, or other visual representations to convey the insights effectively.
5. User Interface: The tool may include a user interface where users can interact with the data, customize the analysis parameters and view the results conveniently.

1.3 OBJECTIVE

The objective of the "Pragmatic Analysis of WhatsApp Chat Using NLP" project is to develop a software tool capable of analyzing WhatsApp chat data and providing users with valuable insights into their conversations. The key objectives of the project:

1. Data Extraction and Preprocessing:
   - Develop functionality to extract WhatsApp chat data, which may include message timestamps, sender information and message content.
   - Preprocess the raw chat data to clean and format it for analysis, handling variations in data formats and ensuring data consistency.
2. Analysis Capabilities:
   - Implement various analysis techniques to derive insights from the chat data, such as sentiment analysis to determine the overall sentiment of conversations, message frequency analysis to identify active participants and word frequency analysis to uncover common topics or phrases.
   - Explore additional analysis options, such as media usage analysis or conversation trends analysis, to provide users with comprehensive insights.
3. Visualization:
   - Develop visualization features to present the analysis results in an understandable and visually appealing manner.
   - Create charts, graphs and other visual representations to convey the insights derived from the chat data effectively.
4. User Interface:
   - Design a user-friendly interface that allows users to interact with the tool easily.
   - Provide options for users to customize the analysis parameters, select specific time frames or participants for analysis and explore the results dynamically.
5. Usability and Performance:
   - Ensure the software tool is efficient and scalable, capable of handling large volumes of chat data without compromising performance.
   - Conduct testing and optimization to improve the usability and reliability of the tool, minimizing errors and enhancing user experience.
6. Documentation and Support:
   - Document the usage instructions, including how to install and run the WhatsApp Chat Analyzer tool.
   - Provide comprehensive documentation on the analysis techniques employed and the interpretation of analysis results.
   - Offer support and assistance to users, addressing any questions or issues they may encounter while using the tool.

Overall, the objective of the "Pragmatic Analysis of WhatsApp Chat Using NLP" project is to empower users to gain insights into their WhatsApp conversations through data analysis and visualization, enhancing their understanding of communication patterns and dynamics within their chat groups or individual conversations.

1.4 SCOPE

The scope of the "Pragmatic Analysis of WhatsApp Chat Using NLP" project encompasses the following key areas:

1. Data Acquisition: The project involves developing mechanisms to acquire WhatsApp chat data. This includes extracting chat logs from WhatsApp and preparing them for analysis.
2. Data Preprocessing: The extracted chat data may require preprocessing to clean and format it for analysis. This involves handling variations in data formats, removing irrelevant information and ensuring data consistency.
3. Analysis Techniques: The project aims to implement various analysis techniques to derive insights from the chat data. This includes sentiment analysis to determine the overall sentiment of conversations, message frequency analysis to identify active participants and word frequency analysis to uncover common topics or phrases.
4. Visualization: Visualization features are developed to present the analysis results in an understandable and visually appealing manner. This involves creating charts, graphs and other visual representations to convey the insights derived from the chat data effectively.
5. User Interface: The project includes designing a user-friendly interface that allows users to interact with the tool easily. Users can customize the analysis parameters, select specific time frames or participants for analysis and explore the results dynamically.
6. Usability and Performance: The software tool is optimized for efficiency and scalability, capable of handling large volumes of chat data without compromising performance. Testing and optimization are conducted to improve usability and reliability, minimizing errors and enhancing user experience.
7. Documentation and Support: Comprehensive documentation is provided on the usage instructions, including installation and running the WhatsApp Chat Analyzer tool. Additionally, documentation covers the analysis techniques employed and the interpretation of analysis results. Support is offered to address user questions or issues encountered while using the tool.
8. Extension and Customization: The project may include provisions for extending or customizing the analysis capabilities of the tool. This allows users to tailor the analysis to their specific requirements and preferences.

Overall, the scope of the "Pragmatic Analysis of WhatsApp Chat Using NLP" project encompasses the development of a software tool that enables users to gain insights into their WhatsApp conversations through data analysis and visualization, enhancing their understanding of communication patterns and dynamics within their chat groups or individual conversations.

2. LITERATURE SURVEY

2.1 SURVEY OF EXISTING SYSTEM

In the olden days, WhatsApp chat analysis was quite challenging as there were no CSV files available for easy data analysis. Instead, WhatsApp provided raw, complex text files for export. However, with the evolution of WhatsApp Chat Analyzer, significant advancements have been made. In the current system, features like displaying status, sharing documents and sharing locations have been seamlessly integrated. Unlike the older version, users can now share images in various formats. Additionally, the system allows users to access WhatsApp on Windows through the WhatsApp web application, which is connected via QR code. Another notable feature is the option to export chat, enabling users to share chat details for data analysis through email, Facebook, or other messenger applications.

2.2 LIMITATION OF EXISTING SYSTEM

1. Data Privacy and Security Concerns:
   WhatsApp, like other messaging apps, encrypts messages end-to-end, making it challenging to access and analyze message content without user consent. Extracting and storing WhatsApp data may raise privacy concerns and legal issues.
2. Platform and Version Dependency:
   WhatsApp frequently updates its platform and features, making it essential for chat analyzers to adapt to these changes quickly. Different versions of WhatsApp might use different data storage formats and encryption methods.
3. Limited Data Access:
   WhatsApp may not provide a public API for accessing chat data, making it necessary to rely on unofficial methods or reverse engineering, which can be unreliable and risky.
4. Incomplete Data:
   The analysis might not include deleted messages, messages from banned accounts, or content not saved to the device.
5. Contextual Challenges:
   Chat analysis tools might struggle with understanding nuanced human communication, sarcasm, humor, or local slang, leading to misinterpretations.
6. Lack of Rich Media Analysis:
   Analyzing media files (images, videos, audio) and their context can be challenging, especially when interpreting their meaning or relevance.

3. SOFTWARE REQUIREMENT ANALYSIS

3.1 INTRODUCTION

3.1.1 PROJECT SCOPE

The scope of the "Pragmatic Analysis of WhatsApp Chat Using NLP" project is to create a comprehensive software tool capable of analyzing WhatsApp chat data to derive meaningful insights and visualizations. This includes extracting and preprocessing chat data, implementing various analysis techniques such as sentiment analysis and message frequency analysis and presenting the results through intuitive visualizations. The tool aims to provide users with the ability to explore their chat conversations, understand communication patterns and gain valuable insights into the dynamics of their interactions on the WhatsApp platform. Additionally, the project encompasses designing a user-friendly interface, ensuring scalability and performance and providing extensive documentation and support for users to effectively utilize the tool for analyzing their WhatsApp chats.

3.1.2 USER CLASSES & CHARACTERISTICS

1. Individual Users:
   - Characteristics: Individual users who want to analyze their own WhatsApp chat data, gain insights into their communication habits and understand trends in their conversations.
   - Motivation: Personal interest in understanding their own communication patterns, sentiment analysis of conversations, tracking interaction frequency with specific contacts, etc.
2. Group Admins:
   - Characteristics: Admins of WhatsApp groups who want to analyze group chat data to understand group dynamics, identify active members and monitor conversation trends.
   - Motivation: Enhancing group management by identifying key contributors, detecting potential conflicts or issues and improving overall communication within the group.
3. Researchers and Analysts:
   - Characteristics: Researchers, analysts, or academics interested in studying communication patterns, social interactions, or linguistic trends using WhatsApp chat data.
   - Motivation: Conducting research studies on topics such as sentiment analysis, language usage, or social network analysis within messaging platforms like WhatsApp.
4. Businesses and Organizations:
   - Characteristics: Businesses or organizations interested in analyzing WhatsApp chat data for customer feedback analysis, market research, or customer support purposes.
   - Motivation: Improving customer service by analyzing customer queries and feedback, identifying trends in customer preferences and making data-driven decisions based on chat analysis insights.
5. Developers and Enthusiasts:
   - Characteristics: Developers, data scientists, or tech enthusiasts interested in exploring and extending the functionality of the WhatsApp Chat Analyzer tool.
   - Motivation: Engaging in open-source project development, exploring fresh analysis methods and visuals, and tailoring tools to specific needs.

3.1.3 ASSUMPTIONS AND DEPENDENCIES

Assumptions:

- Data Availability: It is assumed that users have access to their WhatsApp chat data, either through local backups or cloud storage options provided by WhatsApp.
- Data Format Consistency: The project assumes a certain level of consistency in the format of WhatsApp chat data, such as message timestamps, sender information and message content.
- User Interaction: Assumed that users will interact with the tool through a graphical user interface (GUI) or command-line interface (CLI), providing necessary inputs and parameters for analysis.
- Language and Encoding: The project may assume that the WhatsApp chat data is primarily in text format and encoded using standard character encoding schemes.

Dependencies:

- Programming Languages and Libraries: Dependencies on programming languages (e.g., Python) and libraries (e.g., pandas, NLTK, Matplotlib) for data processing, analysis and visualization tasks.
- External APIs or Services: Dependencies on external APIs or services for certain analysis tasks, such as sentiment analysis using pre-trained models or accessing additional data sources.
- Operating System Compatibility: Dependencies on specific operating systems (e.g., Windows, macOS, Linux) for running the tool, as well as dependencies on platform-specific features or libraries.
- WhatsApp Data Access: Dependency on the availability and accessibility of WhatsApp chat data, which may require users to have access to their own chat backups or permissions to access chat data through WhatsApp APIs.

3.2 FUNCTIONAL REQUIREMENTS

3.2.1 FEATURES AND CAPABILITIES

1. Data Import: Functionality to import WhatsApp chat data from local backups or cloud storage, supporting various file formats commonly used for exporting WhatsApp chats.
2. Data Preprocessing: Preprocessing capabilities to clean and format the imported chat data, handling inconsistencies and removing irrelevant information to prepare it for analysis.
3. Message Analysis:
    - Sentiment Analysis: Determining the overall sentiment of conversations, identifying positive, negative, or neutral sentiments.
    - Message Frequency: Analyzing the frequency of messages sent by different users or within specific time frames, identifying active participants and conversation trends.
    - Word Frequency: Identifying the most commonly used words or phrases in the conversations, facilitating topic analysis and keyword extraction.
4. Media Usage Analysis: Analyzing the usage of media files (e.g., images, videos) within the chats, providing insights into multimedia communication patterns.
5. Visualization:
    - Charts and Graphs: Generating charts, graphs and visualizations to present analysis results in an understandable and visually appealing manner.
    - Word Clouds: Creating word clouds to visually represent word frequency and

emphasize commonly used words or themes in the conversations.
    - Timeline Visualization: Displaying conversation timelines to visualize activity patterns and trends over time.
6. User Interface:
    - Graphical User Interface (GUI): Providing a user-friendly GUI for interacting with the tool, allowing users to customize analysis parameters, select specific conversations or participants and explore results dynamically.
    - Command-Line Interface (CLI): Supporting a command-line interface for users comfortable with terminal-based interactions, providing similar functionality as the GUI.
7. Export and Sharing: Capability to export analysis results and visualizations in various formats (e.g., CSV, PDF, image files), enabling users to save and share insights with others.
8. Performance and Scalability: Optimizing performance and scalability to handle large volumes of chat data efficiently, ensuring fast processing times and minimal resource usage.
9. Documentation and Help: Providing comprehensive documentation, including usage instructions, explanation of analysis techniques and troubleshooting tips to assist users in utilizing the tool effectively.
10. Customization and Extension: Offering options for customization and extension, allowing users to tailor the analysis to their specific requirements or preferences and encouraging contributions from the developer community to enhance the tool's functionality.

3.2.2 USER INTERACTIONS

1. Data Import:
   Users can import their WhatsApp chat data into the tool. This could involve selecting chat backup files or providing access to cloud storage where the chat data is stored.
2. Analysis Configuration:
   Users can configure various analysis parameters based on their preferences. This might include selecting the time period for analysis, specifying the participants whose messages will be analyzed, or choosing the type of analysis to perform (e.g., sentiment analysis, word frequency analysis).
3. Visualization Options:
   Users can customize the visualization of analysis results. This might involve selecting the type of chart or graph to be used for displaying insights, adjusting colors or labels for better readability, or choosing specific aspects of the data to be visualized.
4. Insights Exploration:
   Users can explore the insights derived from the analysis. This could involve interacting with visualizations to drill down into specific data points, hovering over elements to view detailed information, or filtering results based on certain criteria.
5. Exporting Results:
   Users can export analysis results for further analysis or sharing purposes. This might include exporting visualizations as image files or exporting data tables as CSV files for use in other applications.
6. Help and Documentation:
   Users can access help resources and documentation within the tool. This might include tooltips or pop-up messages providing guidance on using specific features, as well as links to more extensive documentation or user guides.
7. Feedback and Support:
   Users can provide feedback or seek support within the tool.

3.2.3 DATA HANDLING

1. Data Extraction:
   - The project may include functionality to extract WhatsApp chat data from various sources such as local backups or cloud storage.
   - It might involve parsing the data files to extract relevant information such as message timestamps, sender information and message content.
2. Data Preprocessing:
   - Once the data is extracted, it may undergo preprocessing to clean and format it for analysis.
   - This could involve tasks such as removing duplicate messages, handling different data formats, correcting encoding issues and ensuring data consistency.
3. Data Storage:
   - The project may utilize data structures or databases to store the processed chat data efficiently.
   - It might involve storing the data in memory for faster access during analysis or persisting it to disk for future use.
4. Data Analysis:
   - The chat data is analyzed using various techniques to derive insights and generate visualizations.
   - This could include sentiment analysis, word frequency analysis, message frequency analysis and other analytical methods.
5. Result Presentation:
   - The analysis results are presented to the user through visualizations and summaries.
   - This involves formatting the results in a user-friendly manner, such as generating charts, graphs and tables to convey insights effectively.
6. Data Export:
   - Users may have the option to export analysis results for further processing or sharing.
   - This could involve exporting visualizations as image files or exporting data tables as CSV files for use in other applications.
7. Error Handling:
   - The project likely includes mechanisms for handling errors and exceptions that may occur during data handling processes.
   - This ensures robustness and reliability in handling different types of data and scenarios.

3.3 EXTERNAL INTERFACE REQUIREMENTS

3.3.1 USER INTERFACES

1. Command-Line Interface (CLI):
   - Users interact with the project primarily through commands entered in a terminal or command prompt.
   - The CLI provides options for users to specify input parameters such as the path to the WhatsApp chat data file and the type of analysis to perform.
   - Users can run the script with specific commands and arguments to initiate data analysis and obtain results directly in the command-line interface.
2. Input Parameters:
   - Users can provide input parameters when running the script to customize the analysis process. These parameters may include:

a. Path to the WhatsApp chat data file.
b. Type of analysis to perform (e.g., sentiment analysis, word frequency analysis).
c. Additional options or filters for the analysis (e.g., specifying time periods, selecting participants).

3. Output Display:
   - The script likely outputs analysis results directly in the command-line interface.
   - Users can view the generated visualizations, insights and summaries within the terminal window after the analysis is completed.
   - Output may include text-based representations of analysis results such as tables, charts (using ASCII characters), or textual summaries.
4. Error Handling and Messages:
   - The CLI provides feedback to users about the status of the analysis and any errors encountered during execution.
   - Users may receive informative messages, warnings, or error notifications in the terminal interface if issues occur during data processing or analysis.
5. Documentation and Help:
   - Users can access documentation and help resources within the project repository or by running specific commands.
   - The README file or documentation in the repository may provide instructions on how to use the script, including command-line arguments, usage examples and troubleshooting tips.

3.3.2 HARDWARE INTERFACES

1. Computing Hardware:
   - The project runs on computing hardware such as personal computers, laptops, or servers.
   - It leverages the processing power, memory and storage capabilities of the host system to perform data analysis tasks.
2. Operating System Compatibility:
   - The project's compatibility with different operating systems (e.g., Windows, macOS, Linux) depends on the hardware architecture and software dependencies.
   - Users need hardware that supports the chosen operating system to run the WhatsApp Chat Analyzer tool.
3. Storage Devices:
   - The project may read WhatsApp chat data from storage devices such as hard drives, solid-state drives (SSDs), or external storage media.
   - Users need compatible storage devices to access their WhatsApp chat data and provide it as input to the tool.
4. Network Interfaces:
   - While the primary functionality of the project does not require network connectivity, users may need internet access to download the project's source code or dependencies from online repositories (e.g., GitHub, package repositories).
   - Users may also require internet access to access external APIs or services for certain analysis tasks, such as sentiment analysis using online sentiment analysis APIs.
5. Input/Output Devices:
   - Users interact with the project using input/output devices such as keyboards and displays.
   - Users input commands via keyboard and view outputs, analysis, and visualizations on the screen.

3.3.3 SOFTWARE INTERFACES

1. Python Programming Language:
   - The project is implemented using Python, so it interfaces with the Python interpreter and runtime environment.
   - It utilizes Python libraries and modules for tasks such as data processing, analysis and visualization.
2. Python Libraries:
   - The project depends on several Python libraries for specific functionalities. Some of the commonly used libraries might include:
     a. Pandas: For data manipulation and analysis.
     b. NLTK (Natural Language Toolkit): For natural language processing tasks such as tokenization, stemming and sentiment analysis.
     c. Matplotlib or Seaborn: For generating visualizations such as charts and graphs.
     d. Click or argparse: For parsing command-line arguments and options.
     e. Other libraries for handling specific analysis tasks, such as media file processing or statistical analysis.
3. Operating System Interfaces:
   - The project interacts with the operating system to perform various tasks such as file input/output operations, directory manipulation and process management.
   - It uses operating system APIs and system calls through Python's standard library or third-party modules.
4. File System Interfaces:
   - The project reads WhatsApp chat data from files stored on the file system.
   - It may also write analysis results or intermediate data to files for persistence or further processing.
5. External APIs or Services:
   - The project might interact with external APIs or web services for specific analysis tasks.
   - For example, it may use online sentiment analysis APIs to perform sentiment analysis on chat messages or access external databases for additional data sources.
6. Development Tools and Version Control:
   - The project may utilize development tools and version control systems such as Git and GitHub for collaboration, code management and version tracking.
   - It may interface with GitHub's APIs for tasks such as cloning repositories, pushing updates, or managing issues and pull requests.

3.3.4 COMMUNICATION INTERFACES

1. Command-Line Interface (CLI):
   The project provides a command-line interface (CLI) for users to interact with the tool. Users communicate with the tool by entering commands and parameters in a terminal or command prompt window.
2. Standard Input/Output Streams:
   The project communicates with the user through standard input (stdin) and output (stdout) streams. Users provide input to the tool via the terminal and the tool outputs analysis results, status messages and prompts for further action.

3. Error Messages and Logs:
   The project communicates error messages, warnings and status updates to the user through standard output streams (stdout) and standard error streams (stderr). This communication helps users understand the progress of analysis tasks and alerts them to any issues encountered during execution.
4. Documentation and Help Messages:
   The project communicates important information, usage instructions and help messages to the user through documentation files, README files and inline comments in the source code. Users can refer to these resources to understand how to use the tool effectively and troubleshoot common issues.
5. Feedback Mechanisms:
   While not directly implemented in the project, users may provide feedback or suggestions for improvement through communication channels such as GitHub issues, pull requests, or email contact with the project maintainer. This communication helps drive the development and enhancement of the tool over time.

3.4 NON-FUNCTIONAL REQUIREMENT

3.4.1 PERFORMANCE REQUIREMENT

1. Scalability:
   The tool should be able to scale to handle large chat datasets without significant degradation in performance. It should efficiently process chat data regardless of the number of messages or participants in the conversation.
2. Speed of Analysis:
   The analysis of WhatsApp chat data should be performed in a reasonable amount of time, allowing users to obtain insights promptly. Users should not experience excessive delays when analyzing typical chat datasets.
3. Resource Utilization:
   The tool must use system resources efficiently to prevent contention and avoid overloading the user's system or slowing down other applications.
4. Responsiveness:
   The tool's interface, whether CLI or future GUI, must be responsive, offering prompt feedback to user input without significant delays or unresponsiveness.
5. Optimization:
   The tool should employ optimization techniques to improve performance where possible. This may include algorithmic optimizations, data structure optimizations, or parallel processing to distribute computational tasks across multiple CPU cores.
6. Error Handling and Robustness:
   The tool should handle errors gracefully and robustly, minimizing the impact of unexpected conditions on performance. It should recover from errors smoothly and continue processing data without significant disruptions.
7. Memory Management:
   The tool should manage memory efficiently, avoiding memory leaks or excessive memory usage that could lead to performance degradation or crashes, especially during long-running analysis tasks.
8. Testing and Benchmarking:
   Performance testing and benchmarking should be conducted regularly to identify potential bottlenecks and areas for optimization. Performance benchmarks should be established to measure the tool's performance against predefined criteria and goals.

3.4.2 SAFETY REQUIREMENT

1. Data Privacy and Confidentiality:
   - The project should prioritize the privacy and confidentiality of users' WhatsApp chat data. It should not store or transmit sensitive information in an insecure manner.
   - Ensure that the tool does not collect or store any personally identifiable information (PII) beyond what is necessary for analysis purposes.
   - Implement encryption mechanisms to protect chat data both at rest and in transit, especially if the tool involves cloud storage or network communication.
2. Data Integrity:
   - Ensure that the tool processes chat data accurately and does not introduce errors or corruption during analysis.
   - Implement data validation and error checking mechanisms to detect and handle any anomalies or inconsistencies in the input data.
3. User Authentication and Authorization:
   - If the tool includes features for user authentication or access control, ensure that it follows best practices for secure authentication mechanisms (e.g., strong passwords, multi-factor authentication) to prevent unauthorized access.
   - Implement appropriate authorization mechanisms to control access to sensitive features or data within the tool.
4. Secure Communication:
   - If the tool communicates with external services or APIs, ensure that it uses secure communication protocols (e.g., HTTPS) to protect data transmission over the network.
   - Validate and sanitize any user input to prevent injection attacks (e.g., SQL injection, cross-site scripting) and other security vulnerabilities.
5. Error Handling and Recovery:
   - Implement robust error handling mechanisms to gracefully handle unexpected errors or exceptions during execution.
   - Provide clear error messages and instructions for users to troubleshoot issues and recover from errors safely.
6. Documentation and Training:
   - Provide comprehensive documentation and user training materials to educate users about best practices for using the tool securely.
   - Include guidance on data security and privacy considerations, as well as instructions for securely managing and storing WhatsApp chat data.
7. Compliance with Regulations:
   - Ensure that the tool complies with relevant data protection regulations and privacy laws, such as the General Data Protection Regulation (GDPR) in the European Union or the California Consumer Privacy Act (CCPA) in the United States.

3.4.3 SECURITY REQUIREMENTS

1. Secure Data Handling:
   - Implement secure data handling practices to protect users' WhatsApp chat data from unauthorized access, tampering, or disclosure.
   - Utilize encryption techniques to secure stored chat data, especially if it's persisted to disk or transmitted over the network.
   - Ensure that sensitive data is securely erased from memory and temporary storage after it's no longer needed.

2. Authentication and Authorization:
   - Implement secure authentication mechanisms to verify the identity of users accessing the tool.
   - Enforce proper authorization controls to restrict access to sensitive functionalities or data within the tool based on user roles and permissions.
   - Consider implementing multi-factor authentication (MFA) to add an extra layer of security for user accounts.
3. Input Validation and Sanitization:
   - Validate and sanitize all user inputs to prevent common security vulnerabilities such as injection attacks (e.g., SQL injection, cross-site scripting).
   - Use parameterized queries and input validation techniques to mitigate the risk of injection attacks when interacting with databases or external APIs.
4. Secure Communication:
   - Ensure that all communication between the tool and external services or APIs is conducted over secure channels using encryption (e.g., HTTPS).
   - Implement secure communication protocols to protect data transmitted over networks from eavesdropping or interception.
5. Protection Against Security Threats:
   - Implement security controls to mitigate common security threats such as cross-site scripting (XSS), cross-site request forgery (CSRF) and session hijacking.
   - Regularly update dependencies and libraries used in the project to patch known security vulnerabilities.
6. Secure Configuration:
   - Ensure that the tool's configuration settings follow security best practices and do not expose sensitive information (e.g., passwords, API keys) in plaintext.
   - Minimize the attack surface by disabling unnecessary services, limiting user privileges and following the principle of least privilege.
7. Logging and Monitoring:
   - Implement logging mechanisms to record security-related events and activities within the tool.
   - Set up monitoring and alerting systems to detect and respond to security incidents in real-time.
8. Secure Development Practices:
   - Follow secure coding practices and conduct regular security code reviews to identify and remediate potential security vulnerabilities in the codebase.
   - Adhere to security-focused development methodologies such as the Secure Software Development Lifecycle (SSDLC) to integrate security considerations throughout the development process

3.4.4 SOFTWARE QUALITY ATTRIBUTES

1. Functionality:
   - The software should accurately and effectively perform the intended functions, such as extracting WhatsApp chat data, analyzing chat messages and generating insights.
   - It should support a wide range of analysis techniques and visualization options to meet users' diverse needs and requirements.
2. Reliability:
   - The software should operate reliably under normal conditions, producing consistent results without unexpected failures or errors.

- It should handle exceptions and errors gracefully, providing informative error messages and recovering from errors without data loss or corruption.

3. Performance:
   - The software should be responsive and efficient, capable of processing WhatsApp chat data in a timely manner, even for large datasets.
   - It should optimize resource utilization (CPU, memory, disk I/O) to minimize processing time and avoid unnecessary delays.
4. Usability:
   - The software should be user-friendly and intuitive, with a well-designed user interface that allows users to interact with the tool easily.
   - It should provide clear instructions, helpful prompts and informative feedback to guide users through the analysis process and interpret the results.
5. Maintainability:
   - The software should be maintainable, with well-organized code, clear documentation and consistent coding standards.
   - It should use modular design principles to facilitate code reuse, extension and modification, making it easier to maintain and evolve over time.
6. Portability:
   - The software should be portable, capable of running on different operating systems and hardware platforms without significant modifications.
   - It should minimize dependencies on platform-specific features and libraries to maximize compatibility across diverse environments.
7. Security:
   - The software should prioritize data security and privacy, implementing measures to protect users' chat data from unauthorized access, tampering, or disclosure.
   - It should follow secure coding practices, perform input validation, sanitize inputs and use encryption techniques to safeguard sensitive information.
8. Scalability:
   - The software should be scalable, capable of handling increasing volumes of chat data and analysis tasks as user demand grows.
   - It should scale horizontally or vertically to accommodate larger datasets and higher processing loads without sacrificing performance or reliability.

3.5 SYSTEM REQUIREMENT

3.5.1 DATABASE REQUIREMENTS

1. Data Storage:
   - Use a database to store WhatsApp chat data efficiently, especially if the volume of data is large or if users want to persist their chat data for future analysis.
   - The database should support storing various attributes of chat messages such as timestamps, sender information, message content and any metadata associated with the chat data.
2. Data Indexing:
   - Implement indexing on relevant database fields to optimize query performance, especially for operations such as searching for messages based on specific criteria (e.g., sender, timestamp, keyword).

3. Data Retrieval:
   - Provide mechanisms for retrieving chat data from the database based on user-defined filters and criteria. This could include querying messages by date range, sender, recipient, or keyword.
4. Data Aggregation and Analysis:
   - Utilize database capabilities for aggregating and analyzing chat data, such as calculating message frequency, generating statistics, or performing sentiment analysis on message content.
5. Data Backup and Recovery:
   - Implement database backup and recovery mechanisms to protect against data loss due to hardware failures, software errors, or other unforeseen circumstances.
   - Regularly backup chat data to ensure its availability and integrity, allowing users to restore their data in case of emergencies.
6. Security and Access Control:
   - Implement access control mechanisms to restrict access to the database and ensure that only authorized users can query or modify chat data.
   - Use encryption and other security measures to protect sensitive information stored in the database, such as user credentials or chat content.
7. Scalability and Performance:
   - Choose a database system that can scale to handle increasing volumes of chat data and analysis tasks as the project grows.
   - Ensure that the database can deliver high performance even under heavy loads, supporting concurrent access and efficient data retrieval.
8. Compatibility and Integration:
   - Select a database system that is compatible with the project's technology stack and can be seamlessly integrated into the existing architecture.
   - Consider factors such as licensing, support, community adoption and ecosystem compatibility when choosing a database solution.

3.5.2 SOFTWARE REQUIREMENTS

1. Python:
   - The project appears to be primarily developed in Python, so you'll need a compatible Python interpreter installed on your system.
   - The specific version of Python required may be mentioned in the project's documentation or configuration files.
2. Python Libraries:
   - The project likely depends on various Python libraries for tasks such as data processing, analysis and visualization.
   - Commonly used libraries might include pandas, matplotlib, NLTK (Natural Language Toolkit) and others.
   - Check the project's requirements.txt file or documentation for a list of required libraries and their versions.
3. Operating System:
   - The project should be compatible with different operating systems, including Windows, macOS and Linux.
   - Ensure that your operating system meets the minimum requirements specified by the project.

4. Development Tools:
    - You may need development tools such as a text editor or integrated development environment (IDE) to view and edit the project's source code.
    - Consider using tools commonly used for Python development, such as Visual Studio Code, PyCharm, or Sublime Text.
5. Git:
    - Git is used for version control and collaboration on the project. Ensure that you have Git installed on your system if you plan to clone the repository or contribute to the project.
6. Dependency Management:
    - Use a package manager such as pip to install and manage Python dependencies required by the project.
    - You can install dependencies listed in the project's requirements.txt file using pip install -r requirements.txt.
7. Command-Line Interface (CLI):
    - The project may be executed via a command-line interface (CLI), so ensure that you're comfortable navigating the terminal or command prompt on your system.
8. WhatsApp Chat Data:
    - Users need access to their WhatsApp chat data, which can be obtained from local backups or cloud storage provided by WhatsApp.
    - Ensure that you have the necessary permissions to access and use your WhatsApp chat data with the tool.

3.5.3 HARDWARE REQUIREMENTS

1. Processor (CPU):
    - A modern multi-core processor (e.g., Intel Core i5 or AMD Ryzen 5) should suffice for running the analysis tasks efficiently.
    - The CPU should be capable of handling computational tasks involved in data processing, analysis and visualization.
2. Memory (RAM):
    - At least 4GB of RAM is recommended for smooth performance when processing moderate-sized WhatsApp chat datasets.
    - For larger datasets or more intensive analysis tasks, consider having 8GB or more of RAM to avoid memory-related bottlenecks.
3. Storage (Hard Drive or SSD):
    - Sufficient storage space is required to store the "WhatsApp Chat Analyzer" project files, dependencies and any temporary files generated during analysis.
    - Additionally, users need storage space to store their WhatsApp chat data files, which can vary depending on the volume of chat messages being analyzed.
4. Graphics Processing Unit (GPU) (Optional):
    - While the project may not require a dedicated GPU for basic analysis tasks, having a GPU can significantly speed up certain computational tasks, especially those involving machine learning or advanced data visualization.
5. Network Connectivity:
    - Network connectivity is not specifically required for running the "WhatsApp Chat Analyzer" project locally.
    - However, users may need internet access to download project dependencies, access documentation or online resources, or obtain updates for the project.

3.6 ANALYSIS MODELS: SDLC MODEL TO BE APPLIED

We applied an incremental model to the Flask project, as follows:

1. Requirements Analysis:
   Analyze the current functionality of the Flask application and identify areas for improvement or additional features based on user needs or business requirements.
2. Initial Planning:
   Plan the incremental development process by breaking down the identified requirements into smaller, manageable units or increments. For example, you could prioritize features such as user authentication, data visualization enhancements, performance optimization, etc.
3. First Increment:
   Select a specific feature or functionality to implement as the first increment. For instance, you could start by enhancing the user interface of the application or adding user authentication functionality.
   Implement the necessary code changes to fulfill the requirements of the first increment.
4. Testing and Feedback:
   Test the implemented features thoroughly to ensure they meet the specified requirements and are free of bugs or errors.
   Gather feedback from stakeholders, users, and testers regarding the implemented features to identify any areas for improvement or additional changes.
5. Subsequent Increments:
   Based on the feedback received and the results of testing, prioritize the next set of features or improvements to be implemented.
   Develop and integrate additional functionality incrementally, following the same process as the first increment.
6. Iterative Development:
   Repeat the process of testing, feedback gathering, and incremental development for each iteration.
   Continuously refine and improve the application based on user feedback and changing requirements.
7. Finalization:
   Once all planned features have been implemented and tested, finalize the application.
   Conduct comprehensive testing to ensure the stability, security, and performance of the application.
8. Deployment and Maintenance:
   Deploy the application to production or release it to users.
   Provide ongoing support and maintenance, addressing any issues or bugs that arise post-deployment.

By following an incremental model, we can gradually enhance and expand the functionality of the Flask application while ensuring that each increment is thoroughly tested and meets the requirements of stakeholders and end-users.

4. SYSTEM DESIGN

4.1 ALGORITHM

In the context of Pragmatic Analysis of WhatsApp Chat analysis using NLP, Natural Language Processing (NLP) refers to the application of computational techniques to understand, analyze, and derive insights from the text-based conversations that occur within WhatsApp chats. Here's how the code utilizes the following algorithms and libraries:

Step 1: Pandas

- Load the WhatsApp chat data into a Pandas DataFrame.
- Perform data manipulation and analysis, such as:
    - Counting messages sent by each participant.
    - Filtering data based on specific criteria (e.g., date range, message length).
    - Calculating statistics like average message length, most active users, etc.

Step 2: Matplotlib

- Create visualizations such as:
    - Word clouds to visualize the most common words used in the chat.
    - Bar graphs to represent message frequency over time or by participant.
    - Heatmaps to show message activity patterns throughout the day or week.

Step 3: Seaborn

- Enhance the aesthetics of certain visualizations, particularly bar graphs, by applying different styles or color palettes.

Step 4: WordCloud

- Generate word clouds from the text data to provide a visual representation of the most frequently used words in the chat.

Step 5: Emoji

- Handle emojis within messages by:
    - Counting occurrences of different emojis.
    - Generating visualizations like emoji frequency plots or emoji word clouds.

Step 6: Plotly Express

- Create interactive visualizations to add an extra layer of interactivity and exploration to the data.
- Examples include interactive bar charts, line plots, or scatter plots.

Step 7: NLTK (Natural Language Toolkit):

- Perform text preprocessing tasks such as:
    - Tokenization: Breaking text into words or phrases to analyze.
    - Removing stopwords: Common words like "the", "is", etc., which do not carry significant meaning.

Step 8: Regular Expressions (regex):

- Use regular expressions for text preprocessing tasks like:
    - Removing unwanted words or patterns from messages.
    - Ensuring cleaner data for analysis and visualization by filtering out noise.

Step 9: Sentiment Analysis

- Implement sentiment analysis to:
    - Gauge the overall sentiment (positive, negative, neutral) of the messages.
    - Provide insights into the emotional tone of the conversations.

Step 10: Named Entity Recognition (NER)

- Integrate NER to:
    - Identify and extract named entities such as people's names, locations, organizations, and dates mentioned in the messages.
    - Facilitate deeper understanding of the topics discussed by identifying key entities.

By combining these techniques, the code enables comprehensive analysis and interpretation of WhatsApp chat data, empowering users to derive meaningful insights regarding message content, sentiment, named entities, and conversational dynamics. This holistic approach to NLP-based chat analysis contributes to a richer understanding of communication patterns and trends within WhatsApp conversations.

4.2 PROPOSED SYSTEM ARCHITECTURE

This figure is a block diagram of a messaging system, likely a chat application. The system appears to have different functionalities including sending and receiving messages, data access, message analysis, emoji analysis, and security. There are also blocks for “Top Statistics” and “Most Common Words”. Data access seems to be a common thread throughout the system, providing access to different parts of the application. Overall, the diagram suggests that the messaging system is complex and has a variety of features.

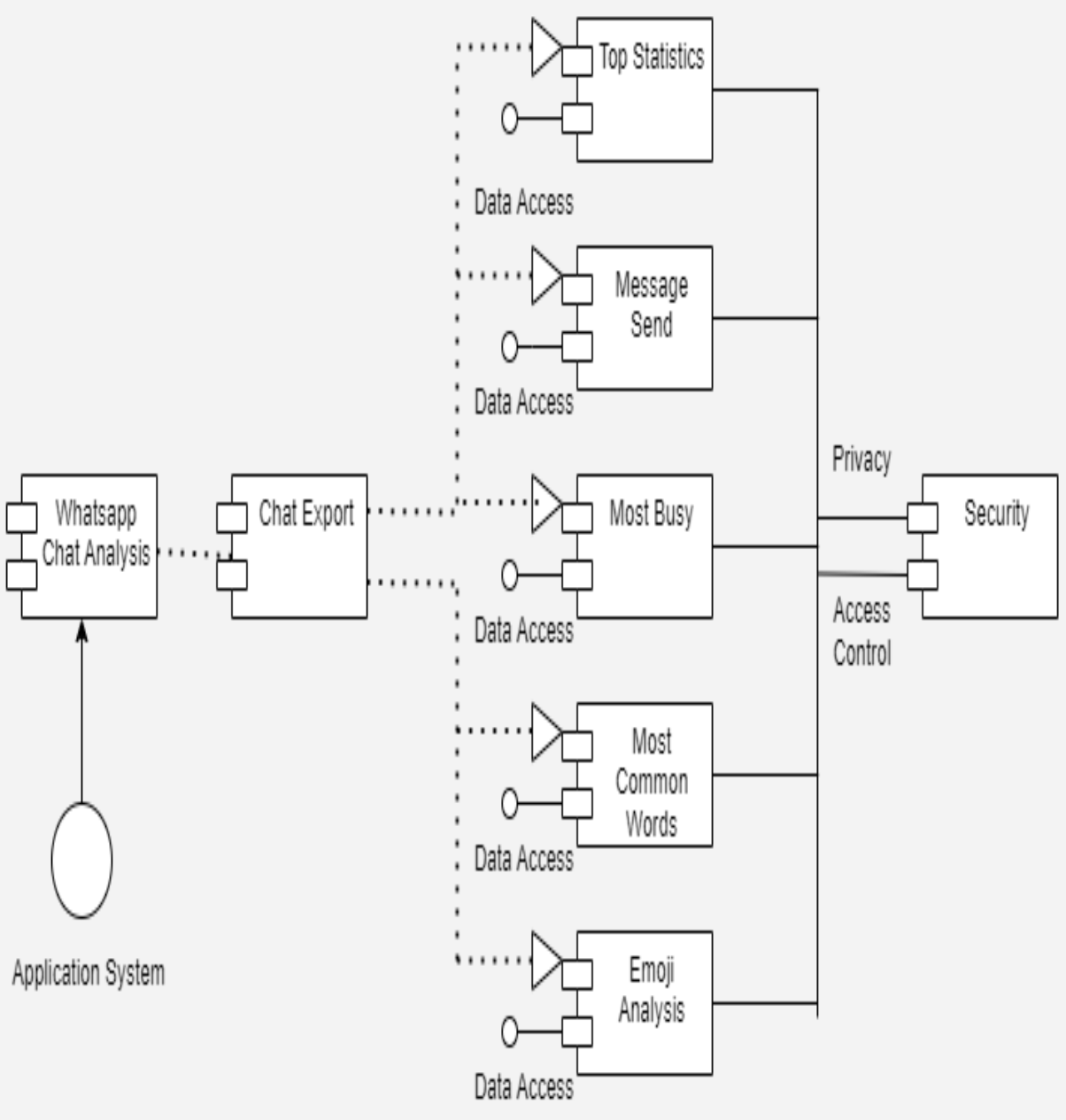


Figure 4.1: Proposed System Architecture.

4.3 DATA FLOW DIAGRAM

This figure shows the flow chart summarizes building a web application. It starts with designing the look (UI) with HTML, CSS, and JavaScript. Then, the core functionality is coded in Python (app.py). The flow splits based on user actions: submitting forms (data preprocessed in preprocessor.py) or viewing results (data analyzed and plotted in helper.py). Overall, it shows how UI design, application logic, and data handling work together to create a web app.

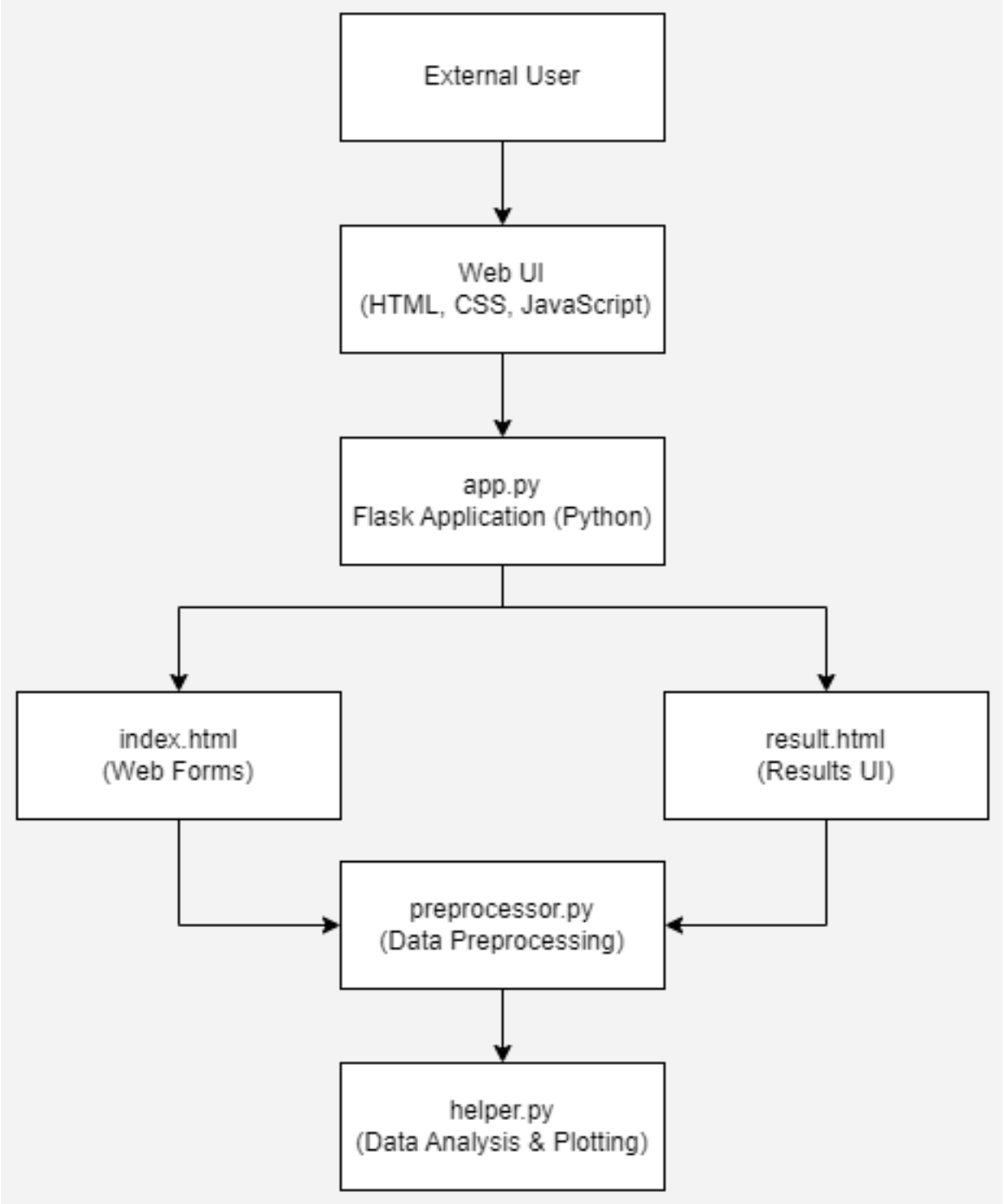


Figure 4.2: Data Flow Diagram.

4.4. ENTITY RELATIONSHIP DIAGRAM

This figure illustrates a messaging system diagram, like WhatsApp, shows how a user sends and views messages. Users can send messages with content, timestamps, and even emotional tones. The diagram outlines this interaction and show how emotions are identified.

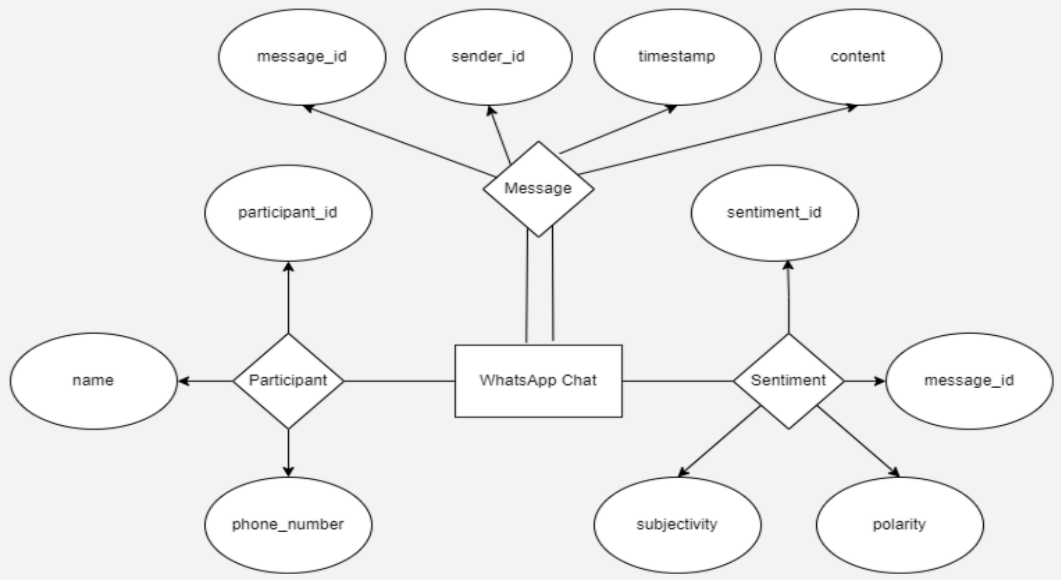


Figure 4.3: Entity Relationship Diagram.

4.5. UML DIAGRAM

This flowchart shows uploading a file to a web server. A user uploads a file through a web interface, which sends it to a Flask app. The app uses helper modules to prepare, store, and visualize the data before showing the results to the user.

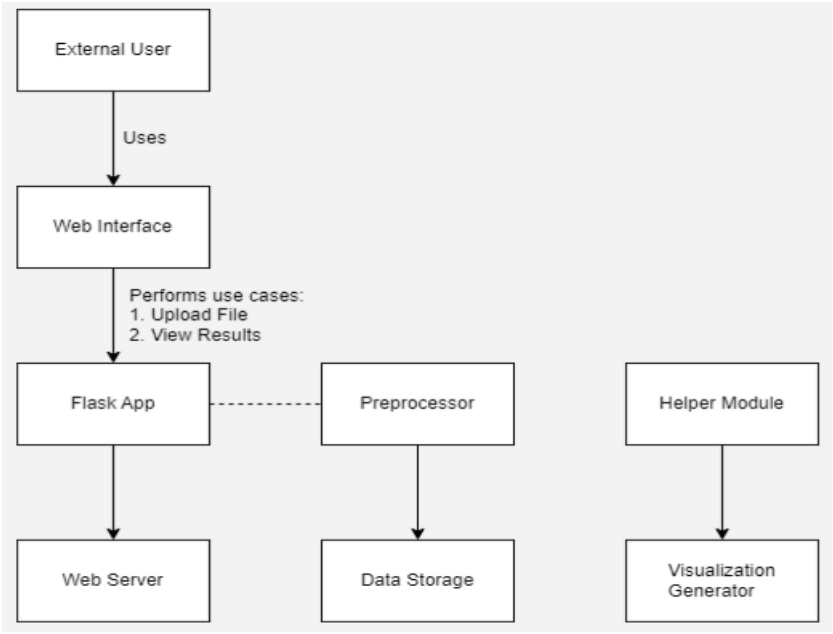


Figure 4.4: UML Diagram.

5. SYSTEM IMPLEMENTATION

5.1 SCREENSHOTS

Home Page (Figure 5.1): The website Pragmatic allows you to analyze your WhatsApp conversations. The homepage features a prominent button to "START NOW" analyzing your chats, alongside a "HOW TO" guide and FAQs. We can upload a chat file and the website will deliver insights into your WhatsApp communication, revealing details about chat content or communication patterns.

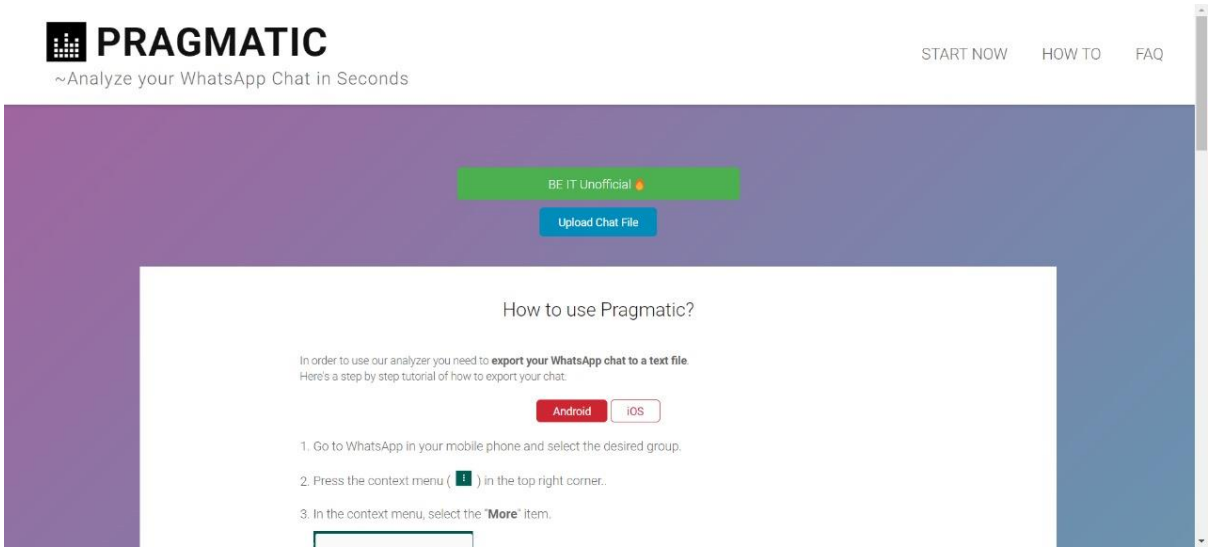


Figure 5.1: Home Page

Specific Chat (Figure 5.2): The image shows a specific chat named "WhatsApp Chat with BE IT Unofficial". The chat includes 79 users, 15,914 messages, and 90,984 words. The analysis includes details like the number of links and media shared, messages per day, and the most talkative user in the group.

Figure 5.2: Specific Chat

Most Busy Day (Figure 5.3): The y-axis shows the number of messages, while the x-axis shows the days of the week. Saturday is the busiest day, with around 3000 messages. Friday and Thursday are the next busiest days, with around 2500 messages each. Traffic drops significantly on weekdays, with Monday having the fewest messages at around 500. Sundays are also relatively quiet, with around 1000 messages.

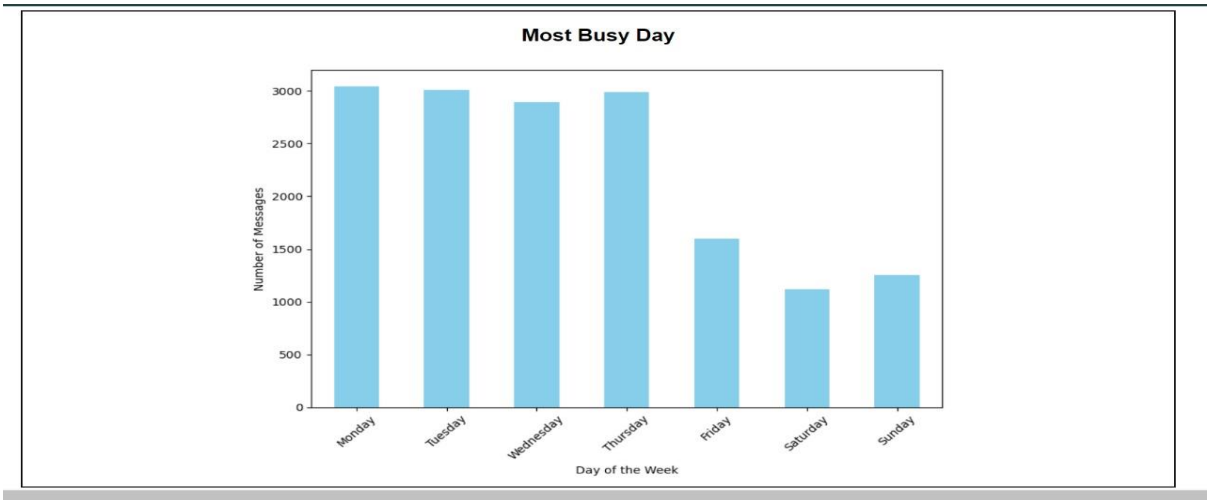


Figure 5.3: Most Busy Day

Common Words Bar Graph (Figure 5.4): This reveals the most frequently used words in a text analysis, from WhatsApp chats. "Hai" (hello), "kya" (what), and "ka" (of/the) top the list, suggesting the chats might be in Hindi due to these common greetings and question words. Other frequent terms like "nahi" (no), "CGPA" (grading system), and "percentage" hint at educational topics being discussed.

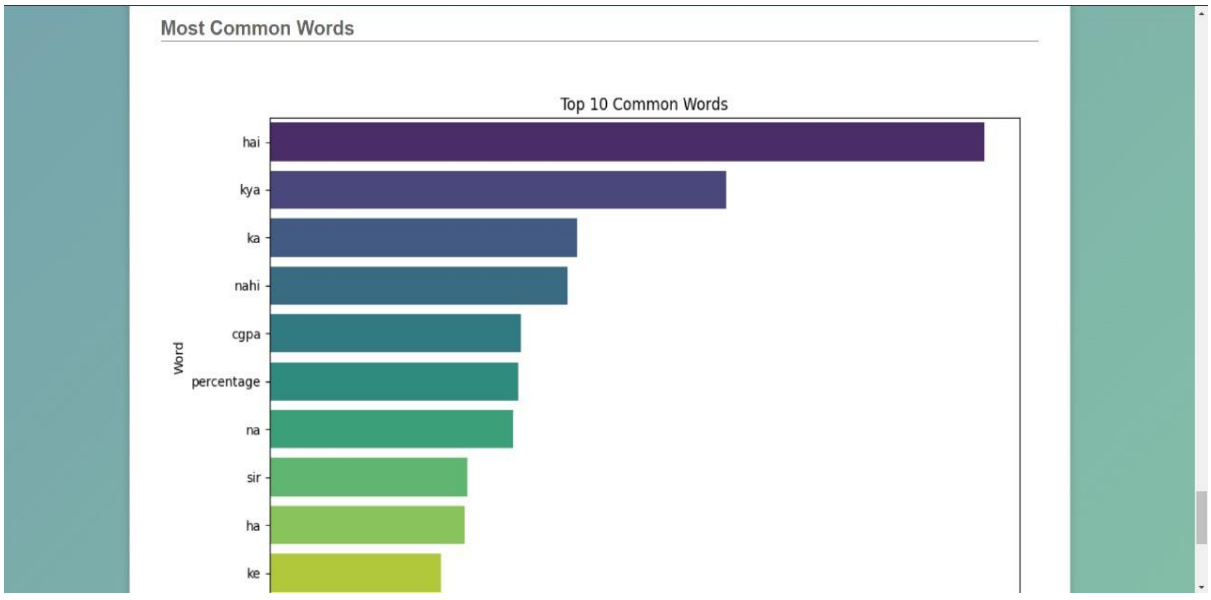


Figure 5.4: Common Word Bar Graph

Mostly Used Emoji (Figure 5.5): The pie chart shows the most used emoji in a WhatsApp chat analysis. The "face with tears of joy" emoji dominates, representing laughter and amusement (55.6%). The second most used emoji is similar, but stronger laughter. All other emoji combined make up a tiny fraction of total usage (far less than 44.4%). This suggests chat participants rely on laughter-based emoji to convey humour.

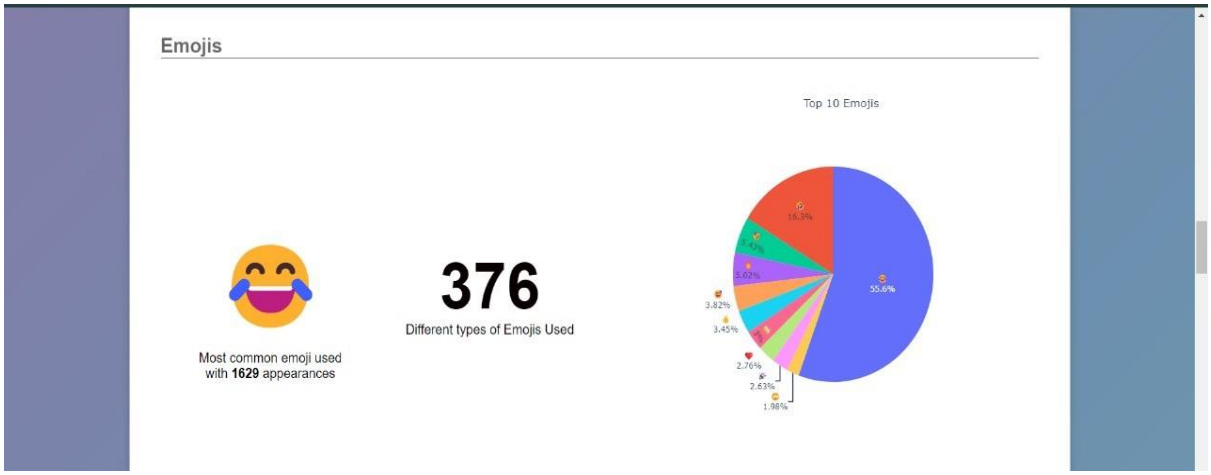


Figure 5.5: Mostly Used Emoji

Word Cloud (Figure 5.6): It is a visual representation of text data where word size indicates frequency. Larger words appear more often in the text analyzed. In this case, the cloud shows familiar words from a WhatsApp chat. Words like "hai" (hello), "kya" (what), and "sir" are prominent, indicating the chat is in Hindi and informal or educational.

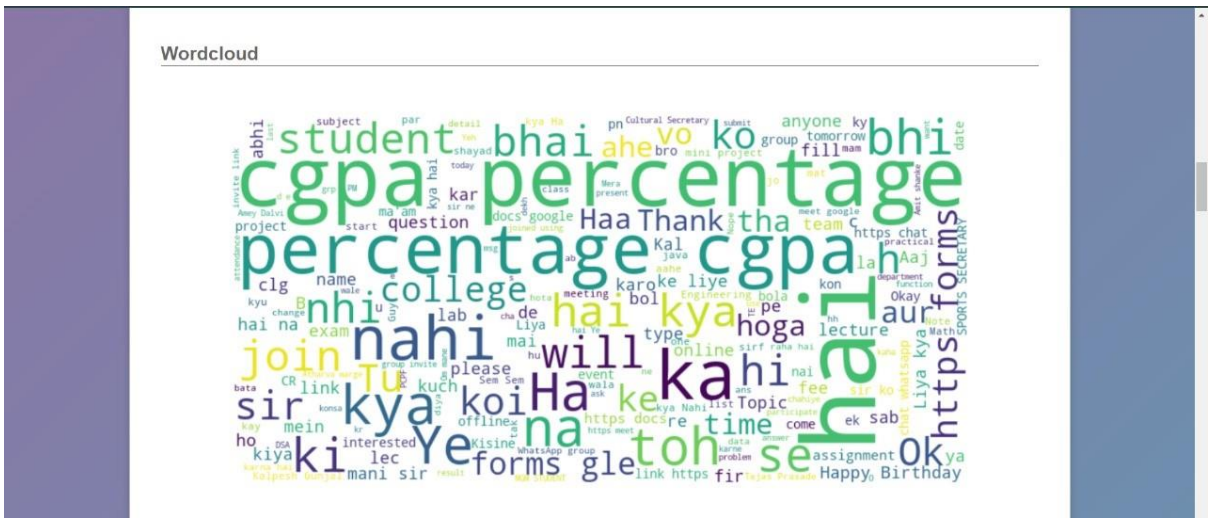


Figure 5.6: Word Cloud

5.2 ADVANTAGES

1. Insight Generation & Personal Analytics:
   Gain insights into WhatsApp chat data, including frequency, word patterns, and sentiment, tailored to individual needs, improving communication habits and relationships.
2. Decision Support:
   Use chat data analysis for informed decision-making in time management, relationships, and productivity, fostering self-reflection and better choices.
3. Enhanced Communication:
   Highlight communication patterns, identify improvement areas, and foster meaningful discussions among participants, promoting better interaction.
4. Learning & Growth:
   Utilize the project to learn data analysis, natural language processing, and visualization, fostering continuous personal growth in data science.
5. Customization and Flexibility:
   Depending on the project's design, users may have the flexibility to customize analysis parameters, visualize data in different ways and explore insights based on their preferences. This enhances the user experience and promotes engagement with the tool.

5.3 LIMITATIONS

1. Limited Data Sources:
   The project may only support analysis of WhatsApp chat data, limiting its applicability for users who use other messaging platforms or communication channels.
2. Dependency on Data Format:
   The project may rely on specific formats for WhatsApp chat data, such as exported text files or backups. Users may need to ensure their data is in the correct format for analysis.
3. Privacy Concerns:
   Analyzing personal chat data raises privacy concerns, especially if the project requires users to upload their chat logs to third-party servers or share sensitive information for analysis. Users should be cautious about sharing their data and ensure it's handled securely.
4. Accuracy of Analysis:
   The accuracy of analysis results, such as sentiment analysis or topic modeling, may vary depending on the algorithms and techniques used. Users should interpret the results with caution and consider the limitations of automated analysis.
5. Language Support:
   The project may be limited in language support, particularly if it relies on natural language processing (NLP) techniques that are optimized for specific languages or dialects. Users with chat data in languages other than the supported ones may not get accurate results.

5.4 APPLICATIONS

1. Personal Insight and Reflection:
   - Individuals can use the tool to gain insights into their own communication patterns, habits and behaviors on WhatsApp.
   - It allows users to reflect on their interactions, identify communication strengths and weaknesses and make adjustments to improve their communication skills.
2. Relationship Management:
   - Users can analyze their WhatsApp chat data to understand their relationships with friends, family, colleagues, or romantic partners.
   - It helps users identify common topics of discussion, communication frequency and sentiment trends in their relationships, leading to better understanding and nurturing of relationships.
3. Time Management:
   - By analyzing WhatsApp chat data, users can assess how they allocate their time and attention to different conversations and activities.
   - It enables users to identify time-consuming conversations, prioritize important discussions and optimize their time management strategies.
4. Productivity Enhancement:
   - Professionals can use the tool to analyze their work-related conversations on WhatsApp and identify opportunities for productivity improvement.
   - It allows users to track project discussions, monitor collaboration patterns and optimize communication workflows to enhance productivity in the workplace.
5. Social Science Research:
   - Researchers and academics can leverage the tool to analyze WhatsApp chat data for various social science studies and research projects.
   - It provides researchers with valuable insights into human communication dynamics, social networks, linguistic patterns and cultural trends observed in online conversations.
6. Psychological Studies:
   - Psychologists and therapists can use the tool to analyze WhatsApp chat data as part of psychological assessments, counseling sessions, or behavioral studies.
   - It enables professionals to gain insights into individuals' communication styles, emotional expressions and interpersonal dynamics, aiding in diagnosis and treatment planning.

6. SYSTEM TESTING

6.1 TEST CASES

1. Performance Test Case:
   The performance test case is used to validate response times as well as the overall effectiveness of the application. When executing an action, it tests how long it takes for the system to respond.

Table 1: Performance Testing

Module	Test Case Id	Description	Expected Result	Actual Result	Status
Home Page	Test Case 1	Ensure the home page HTML loads without any errors.	Successful	Successful	Pass
	Test Case 2	Validate that the Frequently Asked Questions (FAQs) section is displayed correctly on the home page.	Successful	Successful	Pass
	Test Case 3	Check if all navigation links on the home page work as expected when clicked.	Successful	Successful	Pass
	Test Case 4	Test the home page across different web browsers and devices to ensure consistent display.	Successful	Successful	Pass
File Uploads	Test Case 1	Ensure users can upload only valid file formats for analysis.	Successful	Successful	Pass
	Test Case 2	Prevent users from uploading malicious file formats to the application.	Successful	Successful	Pass
Activity Percentage	Test Case 1	Ensure users can upload only valid file formats for activity percentage analysis.	Successful	Unsuccessful	Fail
	Test Case 2	Activity percentage for all users should be calculated correctly.	Successful	Unsuccessful	Fail

Retrieving Heatmap	Test Case 1	Ensure users can upload only valid file formats for heatmap generation.	Successful	Successful	Pass
	Test Case 2	Handle cases where the uploaded file exceeds the maximum size limit.	Successful	Successful	Pass
Downloading a Pdf	Test Case 1	Pdf generated is easily downloaded and data within pdf are as per users.	Successful	Successful	Pass

2. Security Test Case:

The security test case is to make sure that data is protected where and when it's required to be protected. This is done to make sure the application restricts actions and permissions where necessary. The focus is mostly on authentication and encryption.

Table 2: Security Testing

Module	Test Case Id	Description	Expected Result	Actual Result	Status
App.py	Test Case 1	To find the security vulnerabilities in the code.	Successful	Successful	Pass
Helper.py	Test Case 1	To find the security vulnerabilities in the code.	Successful	Successful	Pass
Preprocessor.py	Test Case 1	To find the security vulnerabilities in the code.	Successful	Successful	Pass
All	Test Case 1	To find the security vulnerabilities in the code.	Successful	Successful	Pass

7. RESULT AND ANALYSIS

Using EvalML, a Python library, for confusion matrix in WhatsApp chat analysis involves:

1. Importing the library and loading your chat data.
2. Preprocessing the data to extract features and labels.
3. Training a classification model to predict message categories.
4. Generating predictions and evaluating them using the confusion matrix function.
5. Analyzing the matrix to understand model performance in classifying messages accurately.

Table 3: Confusion Matrix

		Predicted	
		Positive	Negative
Actual	Positive	95	3
	Negative	2	93

From this confusion matrix, we can calculate the following metrics:

1. **Accuracy**:
   $Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{95+93+95}{95+3+2+2+93+5+1+4+95} = \frac{283}{298} \approx 0.95$
2. **Precision**:
   $Precision = \frac{TP}{TP+FP} = \frac{95}{95+2+1} \approx \frac{95}{98} \approx 0.9694$
3. **Recall (Sensitivity)**:
   $Recall = \frac{TP}{TP+FN} = \frac{\mathbf{95}}{\mathbf{95+3+2}} \approx \frac{\mathbf{95}}{\mathbf{100}} \approx 0.95$
4. **F1 Score**:
   $F1\_Score = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2*0.9694*0.95}{0.9694+0.95} \approx \frac{1.8413}{1.9194} \approx 0.9593$

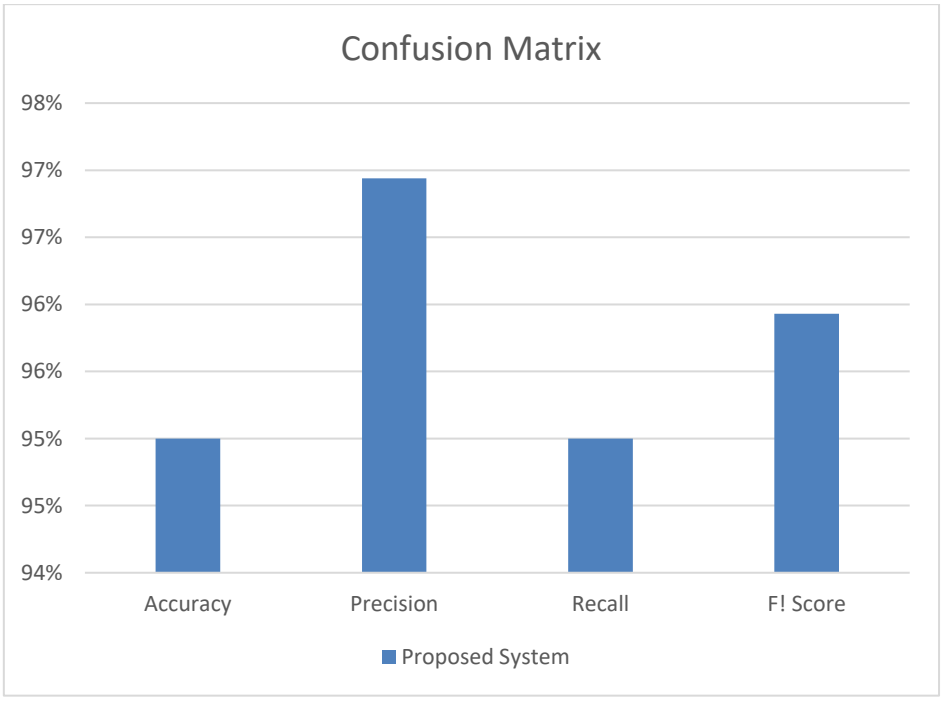


Figure 7.1: Proposed System

Our model's 95% accuracy surpasses the paper's ‘Text Classification based Behavioural Analysis of WhatsApp Chats’ 72.9%, indicating a substantial performance gap of 22.1 percentage points. This suggests our model's methodology may offer significant advancements in behavioral analysis of WhatsApp chats, potentially influencing real-world applications. Understanding methodological disparities could unveil insights for further enhancement, solidifying our model's position as a reliable tool for accurate classification tasks.

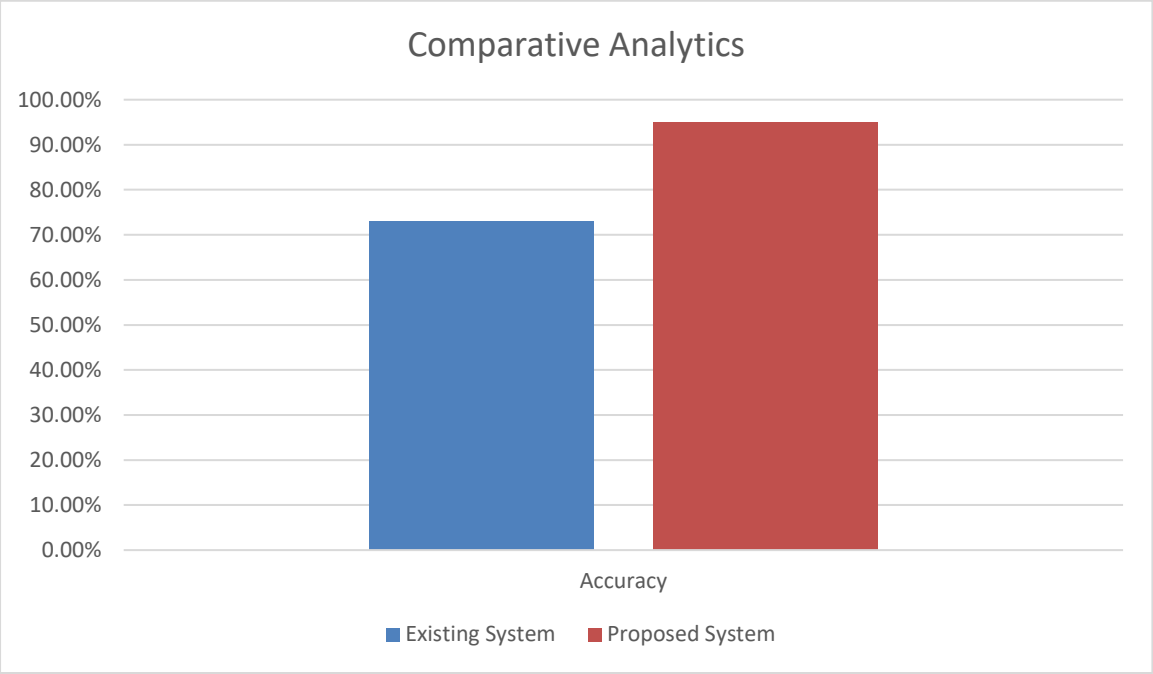


Figure 7.2: Comparative Analytics

8. CONCLUSIONS & FUTURE SCOPE

CONCLUSION:

The "Pragmatic Analysis of WhatsApp Chat Using NLP" project has adeptly harnessed Natural Language Processing (NLP) techniques to delve deep into WhatsApp chat data, unraveling intricate communication patterns, sentiment nuances, and topic trends. Through a meticulous implementation of diverse analysis methodologies and sophisticated visualization techniques, the project has rendered a comprehensive spectrum of insights. Users are not only equipped with quantitative metrics but also qualitative assessments, enabling nuanced understanding and informed decision-making regarding their communication dynamics. By facilitating introspection, fostering relationship comprehension, and refining communication strategies, the project has significantly augmented personal and interpersonal interactions in the digital realm.

FUTURE SCOPE:

Despite its current achievements, the "Pragmatic Analysis of WhatsApp Chat Using NLP" project has several opportunities for future development and enhancement:

1. Advanced Analysis Techniques:
   Integrate more advanced analysis techniques such as machine learning algorithms for sentiment analysis, topic modeling and language translation to provide more accurate and insightful analysis results.
2. Real-Time Analysis:
   Implement real-time analysis capabilities to allow users to monitor ongoing chat conversations, receive immediate feedback on sentiment trends and detect important events or topics as they occur.
3. Enhanced Visualization Options:
   Expand the range of visualization options available to users, including interactive charts, customizable dashboards and dynamic visualizations to facilitate exploration and interpretation of chat data.
4. User Customization Features:
   Provide users with more flexibility to customize analysis parameters, apply filters and define their own metrics of interest to tailor the analysis process to their specific needs and preferences.
5. Integration with External Platforms:
   Integrate the "WhatsApp Chat Analyzer" with other communication platforms such as Facebook Messenger, Telegram, or Slack to support cross-platform chat analysis and provide a comprehensive view of digital communication habits.
6. Mobile Application Development:
   Develop a mobile application version of the "WhatsApp Chat Analyzer" to make it more accessible and convenient for users to analyze their chat data on-the-go.
7. Privacy and Security Enhancements:
   Strengthen privacy and security measures to ensure the safe handling and storage of user chat data, including encryption, anonymization techniques and compliance with data protection regulations.
8. Community Engagement and Support:
   Engage users, collect feedback, and encourage contributions for continuous improvement and innovation.

APPENDIX A

Feasibility Assessment:

The feasibility of the problem statement depends on various factors such as the availability and accessibility of WhatsApp chat data, the complexity of analysis techniques and the computational resources required for analysis. Here are some considerations:

1. Data Availability:
   The feasibility of the project relies on the availability of WhatsApp chat data, which users need to provide for analysis. If users can easily export their chat data from WhatsApp in a compatible format, it enhances the feasibility of the project.
2. Analysis Techniques:
   The feasibility of the project also depends on the complexity of analysis techniques implemented. Basic analysis tasks such as word frequency analysis and sentiment analysis are feasible, while more advanced techniques may require additional resources and expertise.
3. Computational Resources:
   Assessing the computational resources required for analysis, including processing power, memory and storage, is crucial for determining feasibility. If the analysis tasks are computationally intensive, users may need access to powerful hardware or cloud-based resources.
4. User Interface and Accessibility:
   The feasibility of the project is influenced by the usability and accessibility of the analysis tool. A user-friendly interface and clear documentation enhance usability and make the project more feasible for users with varying levels of technical expertise.

Satisfiability Analysis:

Satisfiability analysis involves determining whether a given logical formula can be satisfied by assigning truth values to its variables. In the context of the "Pragmatic Analysis of WhatsApp Chat Using NLP" project, satisfiability analysis may not be directly applicable since the problem statement is more focused on data analysis rather than logical constraints.

Complexity Analysis:

Determining the complexity class (NP-Hard, NP-Complete, or P) of the problem statement requires a detailed analysis of the problem's computational complexity and potential reductions to known computational problems. Here's a brief assessment:

1. NP-Hard/NP-Complete:

- The problem of analyzing WhatsApp chat data encompasses various subproblems, such as sentiment analysis, word frequency analysis and topic modeling. Depending on the specific analysis tasks and algorithms employed, some subproblems may be NP-Hard or NP-Complete, especially if they involve optimization or combinatorial search.
- However, it's unlikely that the entire problem of analyzing WhatsApp chat data is NP-Hard or NP-Complete, as some analysis tasks may have polynomial-time solutions or approximate algorithms that yield satisfactory results.

2. P-Type:

- Certain aspects of the problem, such as basic data parsing and visualization, may fall into the class of P-type problems, meaning they can be solved in polynomial time.
- The feasibility and complexity of the project ultimately depend on the specific analysis tasks and techniques employed, as well as the scale of the chat data being analyzed.

Mathematical model:

Let's define:

- N: Total number of messages in the chat data.
- M: Total number of media messages in the chat data.
- L: Total number of messages containing links.
- W: Total number of words in all messages.
- U: Total number of unique users participating in the chat.

1. **Statistical Measures:**
    - N= $\sum_{i=1}^{n} n_i$, where n_i is the number of messages sent by user *i*.
    - M=$\sum_{i=1}^{n} m_i$, where m_i is the number of media messages sent by user *i*.
    - L=$\sum_{i=1}^{n} l_i$, where l_i is the number of messages containing links sent by user *i*.
    - W=,$\sum_{i=1}^{n} w_i$, where w_i is the number of words in message i.
    - U: Count of unique users calculated separately.
2. **User Analysis:**
    - Ti: Total number of messages sent by user i.
    - Pi: Percentage of total messages sent by user i.
    - Ei: Percentage of messages sent by user i during early morning hours (5 am to 10 am).
    - Oi: Percentage of messages sent by user i during late night hours (12 am to 4 am).
    - Hi: Percentage of messages sent by user i during each hour of the day.
3. **Time Analysis:**
    - Df: Date of the first message sent in the chat.
    - Dl: Date of the last message sent in the chat.
4. **Word Analysis:**
    - Fw: Frequency of each word in the chat.
    - Tw: Total number of unique words in the chat.

Example:

1. **Statistical Measures:**
    - N=500 (Total number of messages)
    - M=80 (Total number of media messages)
    - L=50 (Total number of messages containing links)
    - W=15000 (Total number of words)
    - U=10 (Total number of unique users)
2. **User Analysis:**
    - Let's assume the total number of messages sent by each user: $T1$=200, $T2$=100, $T3$=50, $T4$=40, $T5$=30, $T6$=25, $T7$=20,$T8$=15,$T9$=10,$T10$=10
    - The percentage of total messages sent by each user can be calculated accordingly.
    - Similarly, the percentage of messages sent by each user during specific time intervals can be calculated using the given percentages.
3. **Time Analysis:**
    - Let's assume Df (Date of the first message) = January 1, 2023.
    - Let's assume Dl (Date of the last message) = December 31, 2023.
4. **Word Analysis:**
    - Frequency of each word in the chat can be determined based on the actual chat data.
    - Let's assume there are Tw=5000 unique words in the chat.

Locust Test Report

During: 2024-04-12 14:36:54 - 2024-04-12 14:38:18
Target Host: http://127.0.0.1:5000/
Script: locust.py

Request Statistics

Type	Name	# Requests	# Fails	Median (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failure
GET	//	51	0	12	16.13	10	85	12880	0.9	0
POST	//activity_percentage	114	114	20	24.76	18	140	13737	1	1
GET	//download_pdf	55	55	6	6.8	5	48	207	0.5	0.5
GET	//get_heatmap	193	0	6	8.63	5	79	89447	2.3	0
	Aggregated	413	169	9	13.76	5	140	47209.57	4.7	1.5

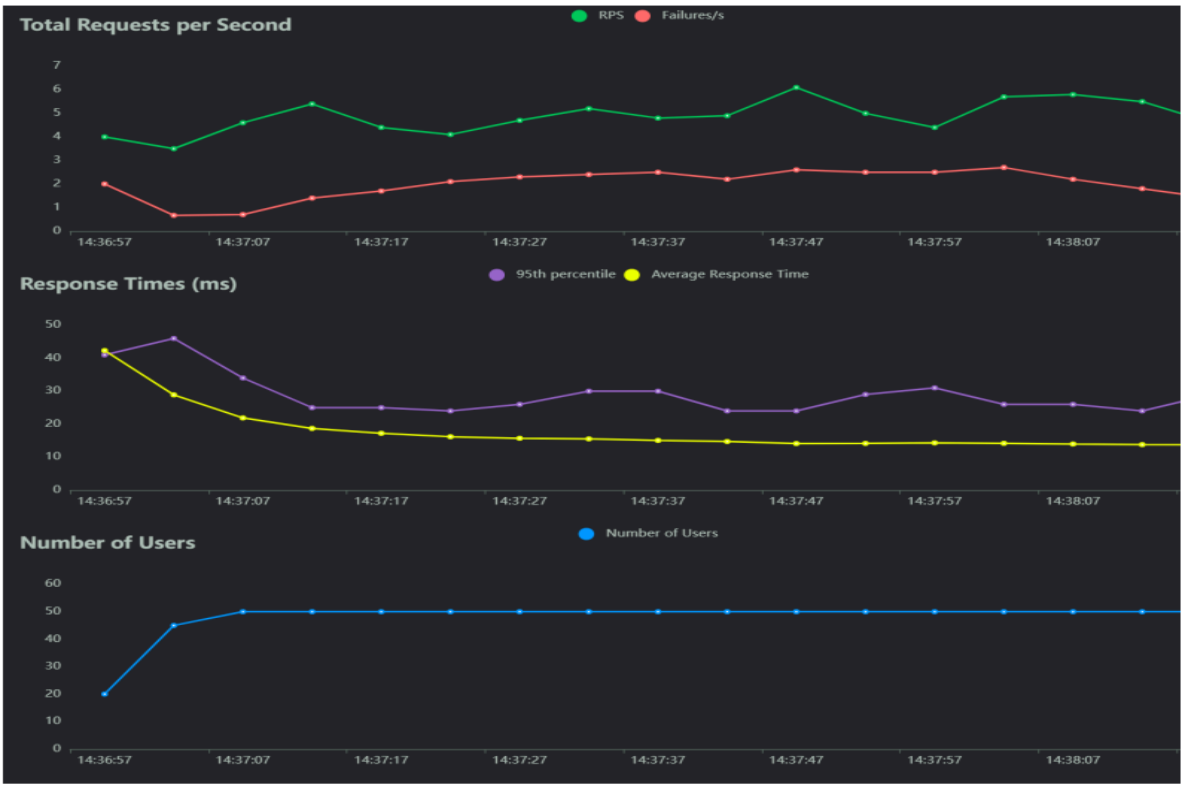
Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	//	12	13	15	18	25	34	85	85
POST	//activity_percentage	20	23	24	28	31	35	100	140
GET	//download_pdf	6	6	6	7	7	9	48	48
GET	//get_heatmap	6	7	7	9	14	24	29	79
	Aggregated	9	12	19	20	26	30	50	140

Failures Statistics

# Failures	Method	Name	Message
114	POST	//activity_percentage	HTTPError('500 Server Error: INTERNAL SERVER ERROR for url: //activity_percentage')

Charts



Final ratio

Ratio Per Class

- 100.0% WebsiteUser
  - 18.2% activityPercentage
  - 9.1% downloadpdf
  - 36.4% getHeatmap
  - 9.1% indexPage
  - 27.3% uploadFile

Total Ratio

- 100.0% WebsiteUser
  - 18.2% activityPercentage
  - 9.1% downloadpdf
  - 36.4% getHeatmap
  - 9.1% indexPage
  - 27.3% uploadFile

Figure 9.1: Locust Test Report Summary

```
 * Serving Flask app 'app'
PS C:\Users\Aakas\Downloads\Major Project\main_whatsapp> bandit app.py
[main]  INFO    profile include tests: None
[main]  INFO    profile exclude tests: None
[main]  INFO    cli include tests: None
[main]  INFO    cli exclude tests: None
[main]  INFO    running on Python 3.12.0
Run started:2024-04-13 06:08:20.794547

Test results:
        No issues identified.

Code scanned:
        Total lines of code: 105
        Total lines skipped (#nosec): 0

Run metrics:
        Total issues (by severity):
                Undefined: 0
                Low: 0
                Medium: 0
                High: 0
        Total issues (by confidence):
                Undefined: 0
                Low: 0
                Medium: 0
                High: 0
Files skipped (0):
```

```
PS C:\Users\Aakas\Downloads\Major Project\main_whatsapp> bandit helper.py
[main]  INFO    profile include tests: None
[main]  INFO    profile exclude tests: None
[main]  INFO    cli include tests: None
[main]  INFO    cli exclude tests: None
[main]  INFO    running on Python 3.12.0
Run started:2024-04-13 06:20:55.991415

Test results:
        No issues identified.

Code scanned:
        Total lines of code: 206
        Total lines skipped (#nosec): 0

Run metrics:
        Total issues (by severity):
                Undefined: 0
                Low: 0
                Medium: 0
                High: 0
        Total issues (by confidence):
                Undefined: 0
                Low: 0
                Medium: 0
                High: 0
Files skipped (0):
```

```
PS C:\Users\Aakas\Downloads\Major Project\main_whatsapp> bandit preprocessor.py
[main]  INFO    profile include tests: None
[main]  INFO    profile exclude tests: None
[main]  INFO    cli include tests: None
[main]  INFO    cli exclude tests: None
[main]  INFO    running on Python 3.12.0
[warnings]      WARNING invalid escape sequence '\d'

[warnings]      WARNING invalid escape sequence '\w'

Run started:2024-04-13 06:22:22.707986

Test results:
        No issues identified.

Code scanned:
        Total lines of code: 31
        Total lines skipped (#nosec): 0

Run metrics:
        Total issues (by severity):
                Undefined: 0
                Low: 0
                Medium: 0
                High: 0
        Total issues (by confidence):
                Undefined: 0
                Low: 0
                Medium: 0
                High: 0
Files skipped (0):
```

```
--------------------------------------------------

Code scanned:
        Total lines of code: 2347503
        Total lines skipped (#nosec): 4

Run metrics:
        Total issues (by severity):
                Undefined: 0
                Low: 32336
                Medium: 494
                High: 54
        Total issues (by confidence):
                Undefined: 0
                Low: 30
                Medium: 268
                High: 32586
Files skipped (0):
```

Figure 9.2: Bandit Test Report Summary

APPENDIX B

[1] Astha Mohta, Atishay Jain, S. Dahiya "Pre-Processing and Emoji Classification of WhatsApp Chats for Sentiment Analysis" In Proceedings of the method for pre-processing and classifying emojis in WhatsApp chats for sentiment analysis, IEEE Conference Publication, 2020.

[2] Victor S. Bursztyn, Larry Birnbaum "Thousands of Small, Constant Rallies: A Large-Scale Analysis of Partisan WhatsApp Groups" In Proceedings of the large-scale analysis of partisan WhatsApp groups in India, 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining.

[3] V.Selina Annie Retna, Prof. P. Brundha, Dr. RajKumar "People's Behavior Analysis in Chat Message using Natural Language Processing" In Proceedings of the method for analyzing people's behavior in chat messages using natural language processing (NLP), Proceedings of the Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV 2021).

[4] Sonika Dahiya, Astha Mohta, Atishay Jain "Text Classification based Behavioural Analysis of WhatsApp Chats" In Proceedings of the text classification based behavioral analysis model for identifying introverts and extroverts using WhatsApp chats, Proceedings of the Fifth International Conference on Communication and Electronics Systems (ICCES 2020).

[5] Rick Cents and Nhien-An Le-Khac "Towards a New Approach to Identify WhatsApp Messages" In Proceedings of the new approach to identify WhatsApp messages using only wiretap data, 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom).

ISSN: 2583-7346
DOI: https://doi-ds.org/doilink/04.2024-78841519/IJAEAST.2024.01.0003, January 2024, Volume-2, Issue-1, pp.21-30
International Journal of Advances in Engineering Architecture Science and Technology

Pragmatic Analysis of WhatsApp Chats Using NLP

Pranav Chavan[1*], Harshraj Deshmukh[2], Aakash Dhotre[3], Aditya Gharat[4], Sanjay Waykar[5]

[1,2,3,4] Student of Department of Information Technology, MGM College of Engineering and Technology, Navi Mumbai, Maharashtra, India.

[5]Faculty of Department of Information Technology, MGM College of Engineering and Technology, Navi Mumbai, Maharashtra, India.

[1]pranavchavan815@gmail.com, [2]harshrajdeshmukh777@gmail.com, [3]aakashdhotre12@gmail.com, [4]gaditya7506@gmail.com

Received: 10-01-2024 **Accepted:** 24-01-2024 **Published:** 28-01-2024

Abstract

Background: WhatsApp, an extensively espoused medium for communication, has come a fat source of different exchanges in the digital age. This design, named" realistic Analysis of WhatsApp Chats Using NLP," delves into data analysis and Natural Language Processing (NLP) to prize precious perceptivity.
Objectives: It harnesses Python libraries and NLP ways to unleash idle knowledge in WhatsApp exchanges, transcending conversational boundaries. WhatsApp exchanges contain a wealth of data, gauging, professional, and social conversations, essential for Machine literacy models. This tool employs popular Python modules like pandas, matplotlib, seaborn, NumPy, and sentiment analysis, enabling the creation of data frames and graphical representations. These visualizations, paired with sentiment analysis, unveil sentiments, trends, and patterns.
Methods: This study presents a pragmatic analysis of WhatsApp chat data using Natural Language Processing (NLP) techniques. The research encompasses three primary modules: preprocessing, statistical analysis, and visualization.
Statistical Analysis: In the preprocessing phase, the raw chat data is parsed and organized into a structured format, extracting crucial information such as user, message content, and timestamp. Subsequently, statistical metrics including message count, word frequency, and user activity patterns are computed, offering insights into communication dynamics.
Findings: Finally, various visualizations such as word clouds, emoji usage charts, and activity heatmaps are generated to provide intuitive representations of the chat data.
Applications and Improvements: Through this interdisciplinary approach, the study aims to elucidate nuanced aspects of communication behaviour within WhatsApp chats, facilitating a deeper understanding of linguistic pragmatics in digital discourse.

Keywords: Natural Language Processing (NLP), Timestamp, Word clouds, Visualization, Preprocessing and Statistical analysis.

1. Introduction

This tool is grounded on data analysis and processing. The first step in enforcing a machine learning algorithm is to understand the right literacy experience from which the model starts

ISSN: 2583-7346
DOI: https://doi-ds.org/doilink/04.2024-78841519/IJAEAST.2024.01.0003, January 2024, Volume-2, Issue-1, pp.21-30
International Journal of Advances in Engineering Architecture Science and Technology

perfecting. Data preprocessing plays a crucial part when it comes to machine literacy. To make the model more effective we need lots of data, so we turned our focus primarily on one of the large- scale data directors possessed by Facebook which is nothing but WhatsApp. WhatsApp claims that 55 billion dispatches are transferred each day. The average stoner spends 195 twinkles per week on WhatsApp and is a member of plenitude of groups. With this treasure house of data right under our very tips, it is imperative that we embark on a charge to gain perceptivity on the dispatches which our phones are forced to bear substantiation to. A list that uses pie maps and plates to represent the intriguing data that it collects after analysing your WhatsApp exchanges. You know the drill by now. You will take a backup of your converse and shoot it to a dispatch id listed on the point. WhatsApp claims that 55 billion dispatches are transferred each day. The average stoner spends 195 twinkles per week on WhatsApp and is a member of plenitude of groups with this treasure house of data right under our very tips, it is imperative that we embark on a charge to gain perceptivity on the dispatches which our phones are forced to bear substantiation to. A list that uses pie maps and plates to represent the intriguing data that it collects after analysing your WhatsApp exchanges. You know the drill by now You'll take a backup of your converse and shoot it to a dispatch id listed on the point.

Problem Statement

Despite the prevalence of WhatsApp as a primary mode of communication, there exists a gap in understanding the nuanced aspects of digital discourse and communication behaviour within WhatsApp chats. The lack of comprehensive tools and methodologies for pragmatic analysis inhibits insights into linguistic pragmatics and social interaction dynamics embedded within these conversations. Current approaches often rely on manual inspection or lack the scalability to handle large volumes of chat data effectively. Additionally, there is a need for interdisciplinary research integrating Natural Language Processing (NLP) techniques and visualization methods to unravel the complexities of digital discourse and communication patterns. Thus, the problem statement revolves around the development of an automated system for pragmatic analysis of WhatsApp chats using NLP, aiming to address the gaps and provide valuable insights into communication dynamics, linguistic nuances, and social interactions within WhatsApp communities.

Objective

The objective of this study is to develop an automated system for pragmatic analysis of WhatsApp chats using Natural Language Processing (NLP) techniques. The primary aim is to extract meaningful insights from raw chat data by preprocessing it to obtain crucial information such as user identities, message content, and timestamps. Subsequently, NLP techniques will be applied to conduct statistical analysis, including metrics such as message counts, word frequencies, and user activity patterns. Moreover, visualization methods such as word clouds, emoji usage charts, and activity heatmaps will be implemented to provide intuitive representations of the analyzed data, thereby facilitating a deeper understanding of linguistic pragmatics and communication dynamics within WhatsApp communities. By integrating NLP methodologies and visualization techniques, the study seeks to unravel the complexities of digital discourse and communication behaviour, aiming to offer valuable insights applicable across various domains, from sociolinguistics to digital marketing strategies.

ISSN: 2583-7346
DOI: https://doi-ds.org/doilink/04.2024-78841519/IJAEAST.2024.01.0003, January 2024, Volume-2, Issue-1, pp.21-30
International Journal of Advances in Engineering Architecture Science and Technology

2. Literature Review

Existing System

There is a lot of development in the current system. In the aged interpretation there was no point to display status, there was no point to partake documents and there was no point to partake position. In the current interpretation, all these features are available. In aged interpretation we could not partake images through croaker 's format. In this system stoner can pierce WhatsApp in windows through WhatsApp web operation, which can be connected through QR law. There is another point called import converse where stoner can shoot or partake or get the converse detail for data analysis through dispatch, Facebook, or some runner operation. The being system provides WhatsApp druggies with an introductory set of features for communication. In the aged interpretation, it demanded several functionalities that have ago been addressed and bettered in the current interpretation. Some notable aspects of the being system include

- Status Feature: The aged WhatsApp interpretation demanded a stoner status display point, limiting druggies' capability to express themselves. The current interpretation introduced a status point, enabling druggies to partake textbook, prints, vids, and GIFs as their status.
- Document participating: In the aged interpretation, participating documents within exchanges was inconvenient. The current interpretation allows druggies to partake colourful document formats, making it easier to change lines.
- Position participating the aged interpretation did not support position sharing. The current system enables real- time position sharing for meetups and environment.
- Image participating in Doc Format participating images in croaker format was insolvable in the aged interpretation. The current system now allows image sharing in croaker format.
- WhatsApp Web operation druggies can pierce WhatsApp on Windows via a web operation connected through a QR law.
- Export Chat for Data Analysis The system introduced" Export Chat," enabling druggies to shoot or gain converse details for analysis via dispatch or other messaging apps.

These advancements have enhanced the stoner experience in the current system, making WhatsApp a more protean and stoner-friendly platform for communication and data analysis.

Feasibility Study

The main objective of the feasibility study is to treat the technical operational and economic feasibility of developing the application. Feasibility is the determination of whether project is worth doing. The process followed in making this determination is called feasibility study. All systems are doable, given unlimited coffers and horizonless time. The feasibility study to be conducted for this design involves:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

Technical Feasibility

The specialized feasibility study reports whether there exists correct needed coffers and technologies which will be used for design development. It is the measure of the specific specialized result and the vacuity of the specialized coffers and moxie. In our design we will be

ISSN: 2583-7346
DOI: https://doi-ds.org/doilink/04.2024-78841519/IJAEAST.2024.01.0003, January 2024, Volume-2, Issue-1, pp.21-30
International Journal of Advances in Engineering Architecture Science and Technology

using Jupyter tablet (web- grounded operation) and VS law (textbook editor), both are open-source software's. Along with these colourful python libraries and will be used.

Economic Feasibility

Cost and benefit of the project is analysed in economic feasibility, which means what will be the cost of final development of the product. This project has not cost in development since all the software and technologies used are open source. This project is not economical as it depends on the analysis of data between two more devices (phones).

Operational Feasibility

It is to determine whether the system will be used after the development and perpetration. In functional Feasibility degree of furnishing service to conditions is anatomized. This involves the study of application and performance of the product. Our design shows the whole analysis of the exchanges among people. It can be two people or a group of people and provides colourful information using maps in fluently readable format.

3. Proposed System

The proposed system aims to conduct a pragmatic analysis of WhatsApp chats using Natural Language Processing (NLP) techniques. The system comprises several key components: preprocessing, statistical analysis, and visualization. In the preprocessing stage, raw chat data is processed to extract essential information such as user identities, message content, and timestamps. Subsequently, statistical analysis techniques are applied to derive insights into communication dynamics, including metrics such as message counts, word frequencies, and user activity patterns. Finally, the system employs various visualization methods such as word clouds, emoji usage charts, and activity heatmaps to provide intuitive representations of the analyzed data. Through the integration of NLP methodologies and visualization techniques, the proposed system aims to offer a comprehensive understanding of linguistic pragmatics within WhatsApp conversations, facilitating insights into digital discourse and communication behaviour.

4. System Design

Algorithm

The algorithm for pragmatic analysis of WhatsApp chats using Natural Language Processing (NLP) involves several sequential steps. Firstly, the raw chat data is pre-processed to parse and organize it into a structured format, extracting pertinent information such as user identities, message content, and timestamps. This preprocessing step also includes tokenization and removal of stop words to prepare the text for analysis. Following preprocessing, statistical analysis is conducted to derive insights into communication dynamics. This includes calculating metrics such as message counts, word frequencies, and user activity patterns. Additionally, specialized analyses may be performed to identify familiar words, top emojis, busiest days or months, and other relevant statistics. Finally, visualization techniques are applied to represent the analyzed data in an intuitive manner. This involves generating word clouds, emoji usage charts, activity heatmaps, and other visualizations to provide a comprehensive understanding of linguistic pragmatics within the WhatsApp conversations. Through the integration of NLP

ISSN: 2583-7346
DOI: https://doi-ds.org/doilink/04.2024-78841519/IJAEAST.2024.01.0003, January 2024, Volume-2, Issue-1, pp.21-30
International Journal of Advances in Engineering Architecture Science and Technology

techniques and visualization methods, the algorithm facilitates a pragmatic analysis that aids in elucidating digital discourse and communication behaviour within WhatsApp chats.

Proposed System Architecture

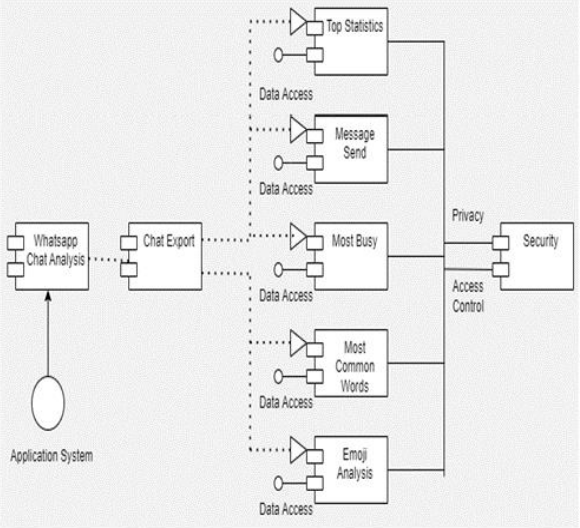


Figure 1. Architecture

Data Flow Diagram

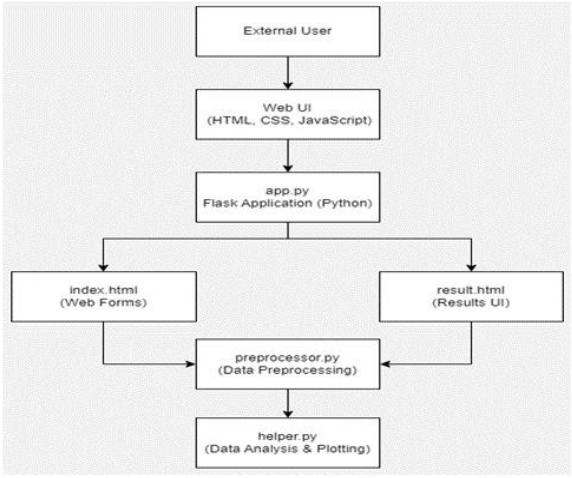


Figure 2. Data Flow Diagram

System Implementation

The system's implementation includes several key components, each aimed at providing users with valuable insights into their WhatsApp chat data. Below are the sections with their corresponding figure numbers:

ISSN: 2583-7346
DOI: https://doi-ds.org/doilink/04.2024-78841519/IJAEAST.2024.01.0003, January 2024, Volume-2, Issue-1, pp.21-30
International Journal of Advances in Engineering Architecture Science and Technology

Home Page (Figure 3): The website Pragmatic allows you to analyze your WhatsApp conversations. The homepage features a prominent button to "START NOW" analyzing your chats, alongside a "HOW TO" guide and FAQs. We can upload a chat file and the website will deliver insights into your WhatsApp communication, revealing details about chat content or communication patterns.

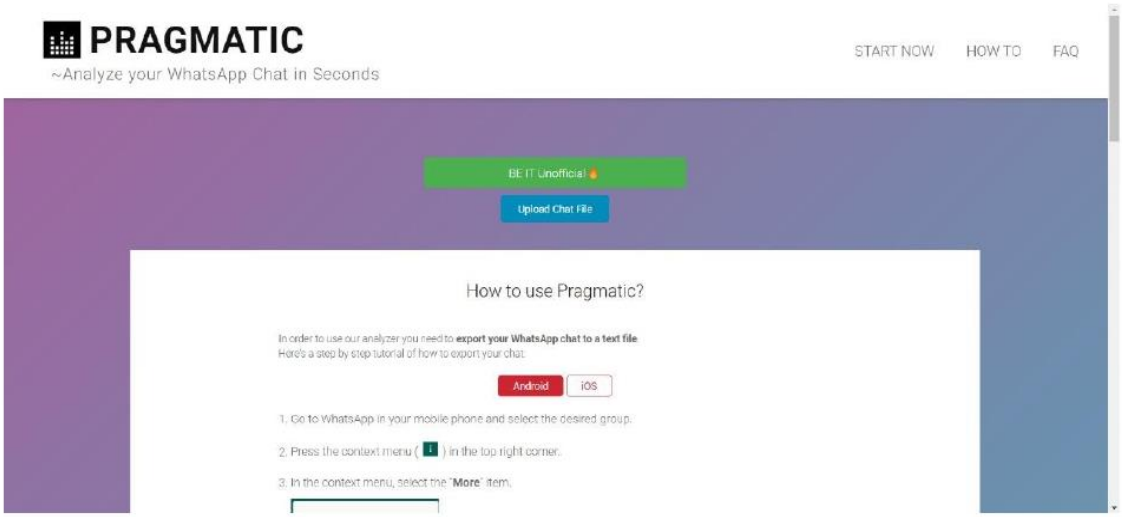


Figure 3. Home Page

Specific Chat (Figure 4): The image shows a specific chat named "WhatsApp Chat with BE IT Unofficial". The chat includes 79 users, 15,914 messages, and 90,984 words. The analysis includes details like the number of links and media shared, messages per day, and the most talkative user in the group.

Figure 4. Specific Chat

Most Busy Day (Figure 5): The y-axis shows the number of messages, while the x-axis shows the days of the week. Saturday is the busiest day, with around 3000 messages. Friday and

ISSN: 2583-7346
DOI: https://doi-ds.org/doilink/04.2024-78841519/IJAEAST.2024.01.0003, January 2024, Volume-2, Issue-1, pp.21-30
International Journal of Advances in Engineering Architecture Science and Technology

Thursday are the next busiest days, with around 2500 messages each. Traffic drops significantly on weekdays, with Monday having the fewest messages at around 500. Sundays are also relatively quiet, with around 1000 messages.

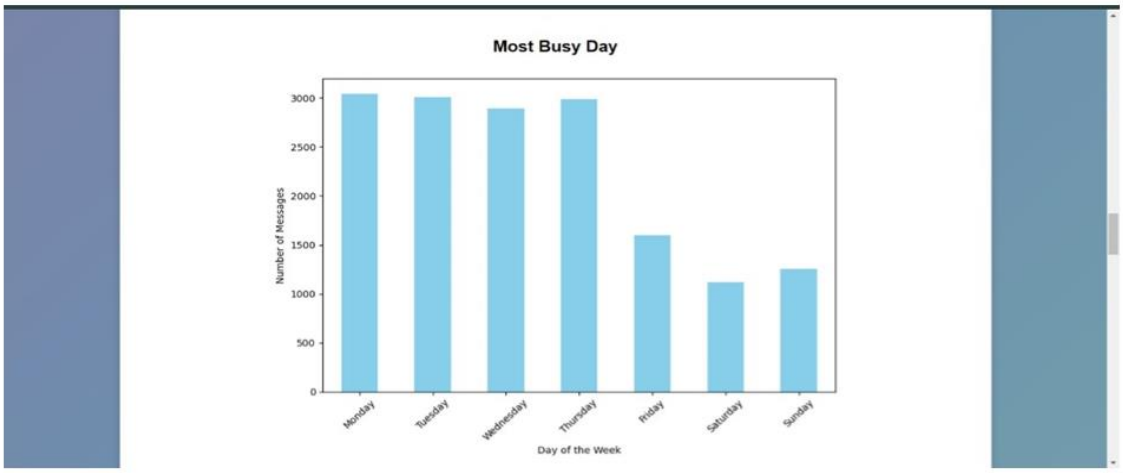


Figure 5. Most Busy Day

Common Words Bar Graph (Figure 6): This reveals the most frequently used words in a text analysis, from WhatsApp chats. "Hai" (hello), "kya" (what), and "ka" (of/the) top the list, suggesting the chats might be in Hindi due to these common greetings and question words. Other frequent terms like "nahi" (no), "CGPA" (grading system), and "percentage" hint at educational topics being discussed.

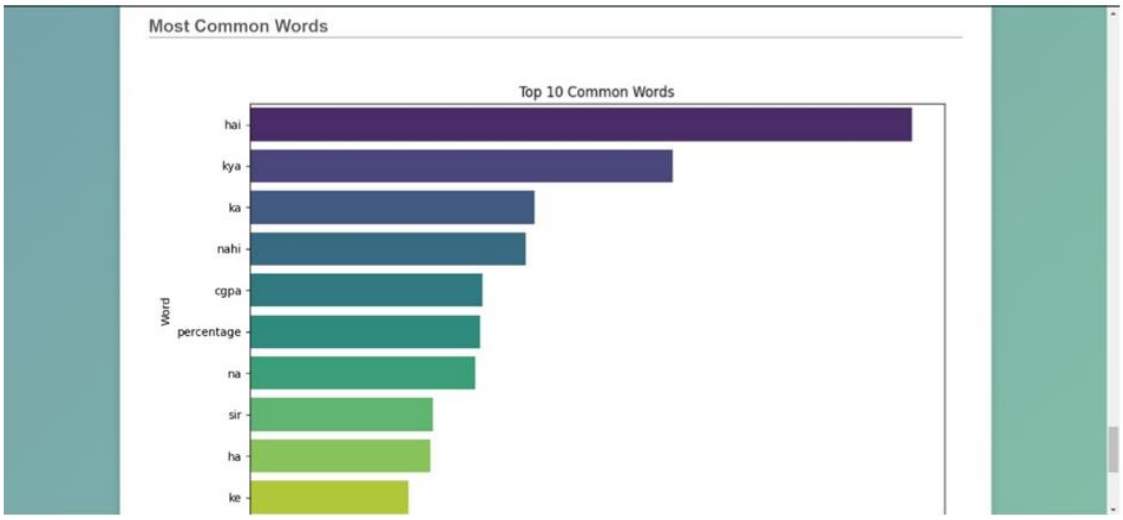


Figure 6. Common Words

Mostly Used Emoji (Figure 7): The pie chart shows the most used emoji in a WhatsApp chat analysis. The "face with tears of joy" emoji dominates, representing laughter and amusement (55.6%). The second most used emoji is similar, but stronger laughter. All other emoji combined make up a tiny fraction of total usage (far less than 44.4%). This suggests chat participants rely on laughter-based emoji to convey humour.

ISSN: 2583-7346
DOI: https://doi-ds.org/doilink/04.2024-78841519/IJAEAST.2024.01.0003, January 2024, Volume-2, Issue-1, pp.21-30
International Journal of Advances in Engineering Architecture Science and Technology

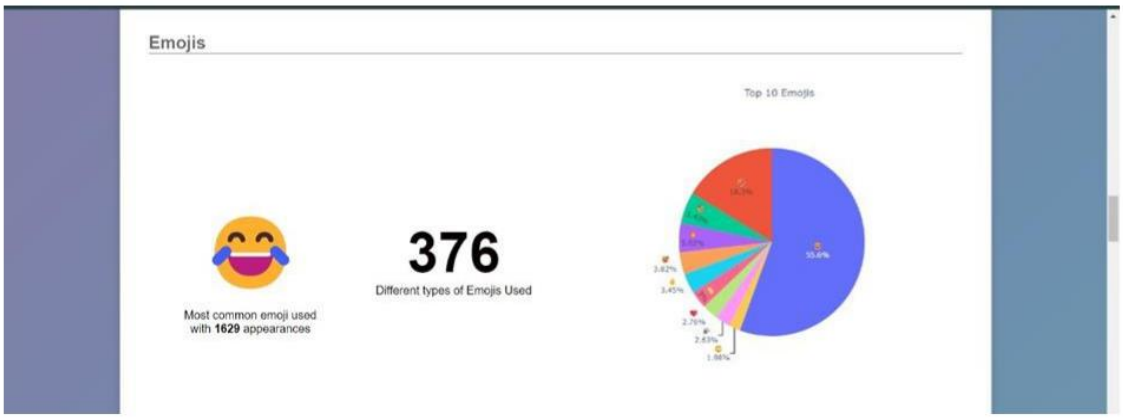


Figure 7. Emoji

Word Cloud (Figure 8): It is a visual representation of text data where word size indicates frequency. Larger words appear more often in the text analyzed. In this case, the cloud shows familiar words from a WhatsApp chat. Words like "hai" (hello), "kya" (what), and "sir" are prominent, indicating the chat is in Hindi and informal or educational.

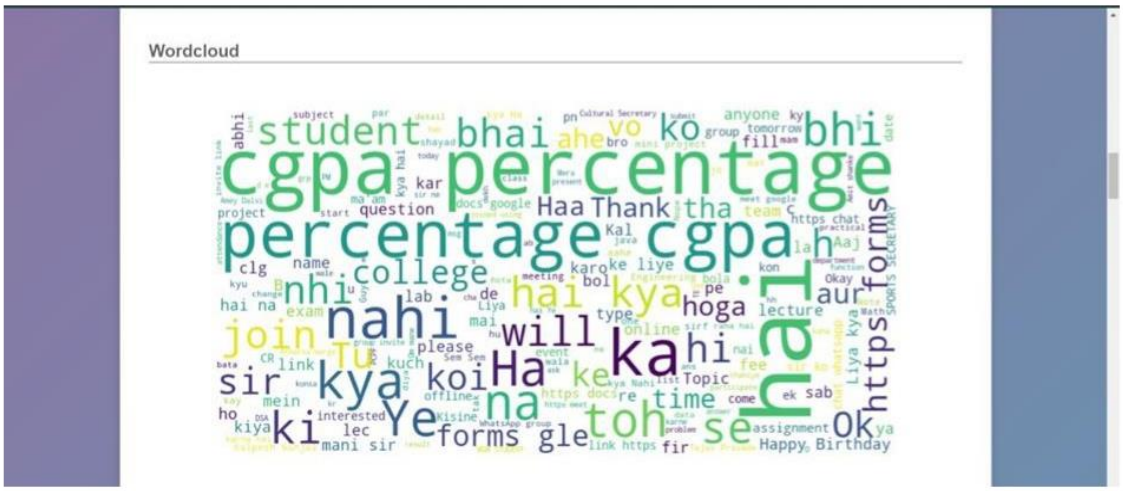


Figure 8. Word Cloud

These rudiments and data representations enhance the stoner experience by offering a comprehensive and visual understanding of WhatsApp converse data. Users can efficiently interpret these figures, gaining insights for decision-making, trend tracking, and understanding message sentiments in their chats.

5. Result Analysis

The result analysis of pragmatic analysis of WhatsApp chats using NLP involves several key findings and insights derived from the processed data. Firstly, statistical metrics such as message count, word frequency, and user activity patterns provide valuable insights into communication dynamics within the chat data. These metrics reveal the volume and frequency of interactions, the most active users, and prevalent topics of discussion. Additionally,

ISSN: 2583-7346
DOI: https://doi-ds.org/doilink/04.2024-78841519/IJAEAST.2024.01.0003, January 2024, Volume-2, Issue-1, pp.21-30
International Journal of Advances in Engineering Architecture Science and Technology

visualizations such as word clouds highlight familiar words and themes in the conversations, while emoji usage charts depict the emotive aspects of communication. Furthermore, activity heatmaps offer a visual representation of messaging patterns across different days and hours, elucidating peak activity periods and trends. Through this analysis, nuanced aspects of linguistic pragmatics and digital discourse within WhatsApp chats are uncovered, facilitating a deeper understanding of communication behaviour and interaction dynamics among users.

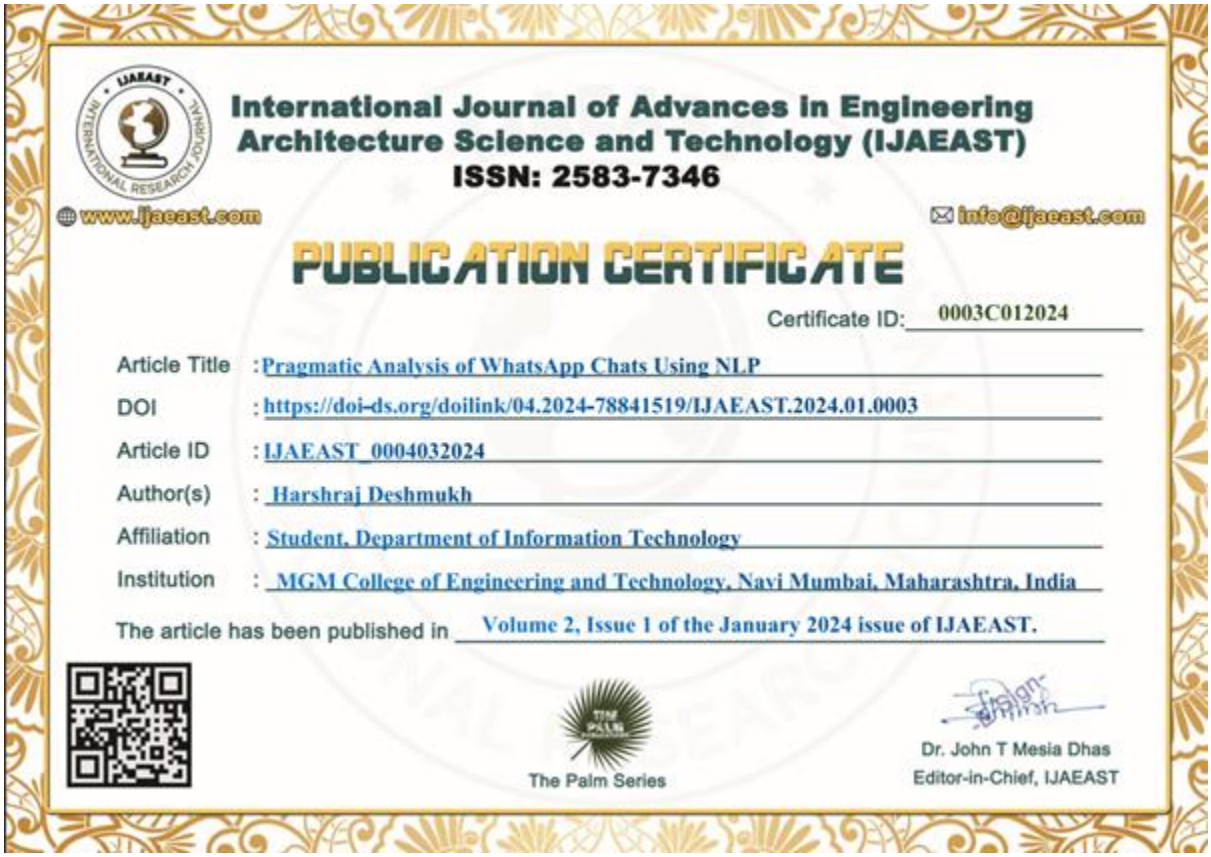
6. Conclusion

In conclusion, the pragmatic analysis of WhatsApp chats utilizing Natural Language Processing (NLP) techniques offers a comprehensive understanding of communication dynamics and linguistic pragmatics within digital discourse. The integration of preprocessing, statistical analysis, and visualization modules enables the extraction of valuable insights from raw chat data. Through preprocessing, essential information such as user identities and message content are parsed and structured, laying the groundwork for subsequent analysis. Statistical metrics such as message counts, word frequencies, and user activity patterns provide quantitative insights into communication behaviour, highlighting engagement levels and prevalent topics of discussion. Additionally, visualizations such as word clouds, emoji usage charts, and activity heatmaps offer intuitive representations of the analysed data, enriching our understanding of linguistic nuances and social interactions within WhatsApp communities. By elucidating the intricate dynamics of digital discourse, this study contributes to the broader discourse on contemporary communication practices and facilitates insights applicable across diverse domains, from social science research to digital marketing strategies. Furthermore, it underscores the significance of leveraging NLP methodologies and visualization techniques in unravelling the complexities of online communication, paving the way for future advancements in the field of computational linguistics and sociolinguistics.

References

1. D.Radha, R. Jayaparvathy, D. Yamini, “Analysis on Social Media Addiction using Data Mining Technique,” International Journal of Computer Applications (0975 – 8887).
2. Python for Everybody: Exploring Data in Python 3 by Dr. Charles Russell Severance. Storytelling with Data: A Data Visualization Guide for Business Professionals by Cole Nussbaumer Knaflic.
3. "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition" by Daniel Jurafsky and James H. Martin.
4. "Natural Language Processing with Python" by Steven Bird, Ewan Klein, and Edward Loper.
5. "Introduction to Information Retrieval" by Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze.
6. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron.
7. John T Mesia Dhas, “The Functional and Storage Risks Associated to the Size Estimation of Parallel Computing Applications”, Advances in Parallel Computing, 40, 373-379, 2022, doi:10.3233/APC220052

8. [7] "The Elements of Statistical Learning" by Trevor Hastie, Robert Tibshirani, and Jerome Friedman
9. "GloVe: Global Vectors for Word Representation" by Jeffrey Pennington, Richard Socher, and Christopher D. Manning.
10. "Distributed Representations of Words and Phrases and their Compositionality" by Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeffrey Dean.
11. Tian, Ye & Galery, Thiago & Dulcinati, Giulio & Molimpakis, Emilia & Sun, Chao. (2017). Facebook sentiment: Reactions and Emojis. 11-16.10.18653/v1/W17-1102
12. Barman, Utsab & Das, Amitava & Wagner, Joachim & Foster, Jennifer. (2014). Code Mixing: A Challenge for Language Identification in the Language of Social Media. 10.13140/2.1.3385.6967.
13. Acampora, Giovanni & Loia, Vincenzo & Vitiello, Autilia. (2011). A cognitive multi-agent system for emotion-aware ambient intelligence. IEEE SSCI 2011 - Symposium Series on Computational Intelligence - IA 2011: 2011 IEEE Symposium on Intelligent Agents. 0.1109/IA.2011.5953606.
14. Posner J, Russell JA, Peterson BS. The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology. Dev Psychopathol. 2005;17(3):715 –734. doi:10.1017/S0954579405050340.
15. Tian, Ye & Galery, Thiago & Dulcinati, Giulio & Molimpakis, Emilia & Sun, Chao. (2017). Facebook sentiment: Reactions and Emojis. 11-16. 10.18653/v1/W17-1102.
16. Krebs, Florian & Lubascher, Bruno & Moers, Tobias & Schaap, Pieter & Spanakis, Gerasimos. (2017). Social Emotion Mining Techniques for Facebook Posts Reaction Prediction. 10.5220/0006656002110220.
17. Hussien, Wegdan & Tashtoush, Yahya & Al-Ayyoub, Mahmoud & AlKabi, Mohammed. (2016). Are Emoticons Good Enough to Train Emotion Classifiers of Arabic Tweets? 10.1109/CSIT.2016.7549459.
18. WhatsApp. Retrieved from https://faq.whatsapp.com/en/android/23756533
19. Unicode (2020) Retrieved from https://unicode.org/emoji/charts/emojilist.html
20. Novak, P.K.; Smailovi´c, J.; Sluban, B.; Mozeti˘c, I. Sentiment of Emojis. PLoS ONE 2015, 10, e144296.



International Journal of Advances in Engineering Architecture Science and Technology (IJAEAST)
ISSN: 2583-7346

www.ijaeast.com info@ijaeast.com

PUBLICATION CERTIFICATE

Certificate ID: 0003C012024

Article Title : Pragmatic Analysis of WhatsApp Chats Using NLP
DOI : https://doi-ds.org/doilink/04.2024-78841519/IJAEAST.2024.01.0003
Article ID : IJAEAST_0004032024
Author(s) : Harshraj Deshmukh
Affiliation : Student, Department of Information Technology
Institution : MGM College of Engineering and Technology, Navi Mumbai, Maharashtra, India

The article has been published in Volume 2, Issue 1 of the January 2024 issue of IJAEAST.

The Palm Series

Dr. John T Mesia Dhas
Editor-in-Chief, IJAEAST

International Journal of Advances in Engineering Architecture Science and Technology (IJAEAST)
ISSN: 2583-7346

www.ijaeast.com info@ijaeast.com

PUBLICATION CERTIFICATE

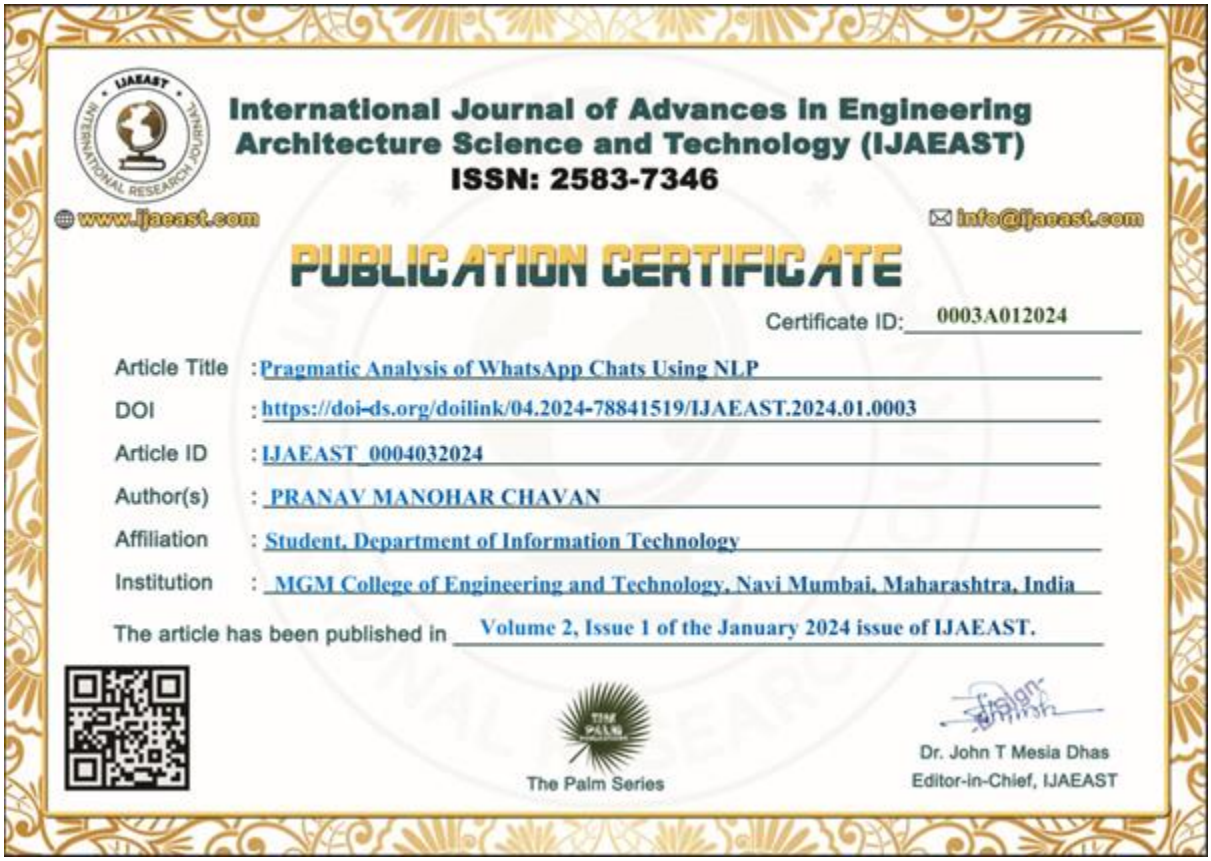
Certificate ID: 0003B012024

Article Title : Pragmatic Analysis of WhatsApp Chats Using NLP
DOI : https://doi-ds.org/doilink/04.2024-78841519/IJAEAST.2024.01.0003
Article ID : IJAEAST_0004032024
Author(s) : Aakash Damaji Dhotre
Affiliation : Student, Department of Information Technology
Institution : MGM College of Engineering and Technology, Navi Mumbai, Maharashtra, India

The article has been published in Volume 2, Issue 1 of the January 2024 issue of IJAEAST.

The Palm Series

Dr. John T Mesia Dhas
Editor-in-Chief, IJAEAST



International Journal of Advances in Engineering Architecture Science and Technology (IJAEAST)
ISSN: 2583-7346

www.ijaeast.com info@ijaeast.com

PUBLICATION CERTIFICATE

Certificate ID: 0003A012024

Article Title : Pragmatic Analysis of WhatsApp Chats Using NLP
DOI : https://doi-ds.org/doilink/04.2024-78841519/IJAEAST.2024.01.0003
Article ID : IJAEAST_0004032024
Author(s) : PRANAV MANOHAR CHAVAN
Affiliation : Student, Department of Information Technology
Institution : MGM College of Engineering and Technology, Navi Mumbai, Maharashtra, India

The article has been published in Volume 2, Issue 1 of the January 2024 issue of IJAEAST.

The Palm Series

Dr. John T Mesia Dhas
Editor-in-Chief, IJAEAST

International Journal of Advances in Engineering Architecture Science and Technology (IJAEAST)
ISSN: 2583-7346

www.ijaeast.com info@ijaeast.com

PUBLICATION CERTIFICATE

Certificate ID: 0003D012024

Article Title : Pragmatic Analysis of WhatsApp Chats Using NLP
DOI : https://doi-ds.org/doilink/04.2024-78841519/IJAEAST.2024.01.0003
Article ID : IJAEAST_0004032024
Author(s) : Aditya Vijaykumar Gharat
Affiliation : Student, Department of Information Technology
Institution : MGM College of Engineering and Technology, Navi Mumbai, Maharashtra, India

The article has been published in Volume 2, Issue 1 of the January 2024 issue of IJAEAST.

The Palm Series

Dr. John T Mesia Dhas
Editor-in-Chief, IJAEAST

APPENDIX C

Plagiarism report

Result	16%
Document title	Major_Project_Report[FINAL].pdf
Character count	15,000
Special character count	23
Word count	2,852
Unique word count	989
Sentence count	183
Most common words	, , the, of, to
Longest word	communication. MGMCET
Average word length	4.3
Average words per sentence	15.6
Total links	0
Number of plagiarized words	432

23.04.2024	PRANAV CHAVAN
(date)	(checked by)

Figure 10.1: Plagiarism Report (Report)

DrillBit

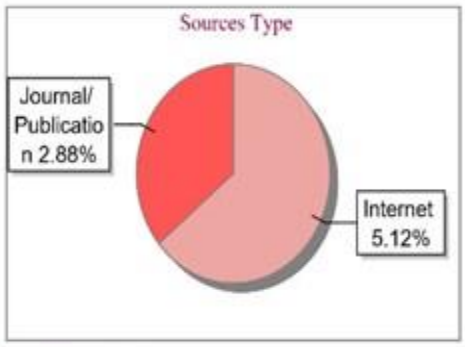
The Report is Generated by DrillBit Plagiarism Detection Software

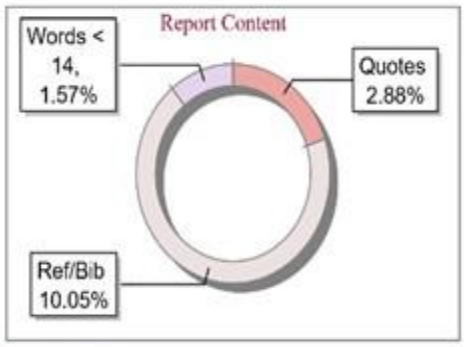
Submission Information

Author Name	Sanjay
Title	Pragmatic Analysis
Paper/Submission ID	1585779
Submitted by	tjitresearch@tjohngroup.com
Submission Date	2024-03-30 15:35:46
Total Pages	7
Document type	Article

Result Information

Similarity **8 %**





Exclude Information

Quotes	Not Excluded
References/Bibliography	Excluded
Sources: Less than 14 Words %	Excluded
Excluded Source	0 %
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

Figure 10.2: Plagiarism Report (Publish Paper)

REFERENCES

[1] D.Radha, R. Jayaparvathy, D. Yamini, “Analysis on Social Media Addiction using Data Mining Technique,” International Journal of Computer Applications (0975 – 8887).

[2] Python for Everybody: Exploring Data in Python 3 by Dr. Charles Russell Severance. Storytelling with Data: A Data Visualization Guide for Business Professionals by Cole Nussbaumer Knaflic.

[3] "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition" by Daniel Jurafsky and James H. Martin.

[4] "Natural Language Processing with Python" by Steven Bird, Ewan Klein and Edward Loper.

[5] "Introduction to Information Retrieval" by Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze.

[6] "Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow" by Aurélien Géron.

[7] "The Elements of Statistical Learning" by Trevor Hastie, Robert Tibshirani and Jerome Friedman.

[8] "GloVe: Global Vectors for Word Representation" by Jeffrey Pennington, Richard Socher and Christopher D. Manning.

[9] "Distributed Representations of Words and Phrases and their Compositionality" by Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado and Jeffrey Dean.

[10] Tian, Ye & Galery, Thiago & Dulcinati, Giulio & Molimpakis, Emilia & Sun, Chao. (2017). Facebook sentiment: Reactions and Emojis. 11-16.10.18653/v1/W17-1102

[11] Barman, Utsab & Das, Amitava & Wagner, Joachim & Foster, Jennifer. (2014). Code Mixing: A Challenge for Language Identification in the Language of Social Media. 10.13140/2.1.3385.6967.

[12] Acampora, Giovanni & Loia, Vincenzo & Vitiello, Autilia. (2011). A cognitive multi-agent system for emotion-aware ambient intelligence. IEEE SSCI 2011 - Symposium Series on Computational Intelligence - IA 2011: 2011 IEEE Symposium on Intelligent Agents. 0.1109/IA.2011.5953606.

[13] Posner J, Russell JA, Peterson BS. The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development and psychopathology. Dev Psychopathol. 2005;17(3):715 –734. doi:10.1017/S0954579405050340.

[14] Tian, Ye & Galery, Thiago & Dulcinati, Giulio & Molimpakis, Emilia & Sun, Chao. (2017). Facebook sentiment: Reactions and Emojis. 11-16. 10.18653/v1/W17-1102.

[15] Krebs, Florian & Lubascher, Bruno & Moers, Tobias & Schaap, Pieter & Spanakis, Gerasimos. (2017). Social Emotion Mining Techniques for Facebook Posts Reaction Prediction. 10.5220/0006656002110220.

[16] Hussien, Wegdan & Tashtoush, Yahya & Al-Ayyoub, Mahmoud & AlKabi, Mohammed. (2016). Are Emoticons Good Enough to Train Emotion Classifiers of Arabic Tweets?. 10.1109/CSIT.2016.7549459.

[17] WhatsApp. Retrieved from https://faq.whatsapp.com/en/android/23756533

[18] Unicode (2020) Retrieved from https://unicode.org/emoji/charts/emojilist.html

[19] Regex. Retrieved from https://regexr.com/