

Data Analyzer for WhatsApp Chats

Project Overview

Python in the Enterprise
30/11/2023

Antonio Murillo Sevillano
Enmanuel De Abreu Gil
Jorge Álvarez Fernández
Julio Ribas de Novales

Abstract

This application will be useful to collect data from exported WhatsApp chats, and its subsequent analysis and processing. It will retrieve specific data about the topics discussed in the conversations, as well as any relevant statistics that can be derived from them.

In this document, we'll analyze the user requirements to define the tasks that must be carried by the system.

1. System Description

1.1. Description and Motivation

The system will facilitate the organization and categorization of messages, allowing the identification of patterns, trends, or key information in interactions. Through natural language processing techniques, it will seek to understand the context and tone of conversations to provide a more comprehensive view of the topics addressed.

Additionally, the application aims to provide quantitative metrics, such as the frequency of certain terms, the average length of messages, or the temporal activity of conversations. This will contribute to generating analytical reports that can be used for strategic decision-making or improving communication in various contexts, whether at the business, social, or investigative level.

1.2. System Goals

The objectives of the application are:

- Conduct a detailed analysis of downloaded messages to obtain specific information about the topics discussed in conversations.
- Process collected data using some sort of language processing technique to understand the context and tone of interactions.
- Organize and categorize messages to identify patterns, trends, and key information in conversations.
- Generate relevant statistics, such as term frequency, average message length, and temporal activity, providing a quantitative view of communication dynamics.
- Facilitate the generation of analytical reports that can be used for strategic decision-making or to improve communication in various contexts.
- Offer an intuitive and accessible user interface that allows users to easily interact with the functions of the system.

2. System Requirements

Functional Requirements:

1. Content Analysis:

It should conduct a detailed analysis of message content to identify themes, linguistic patterns, and conversational tone.

2. Organization and Categorization:

Messages should be effectively organized and categorized to facilitate trend identification and extraction of key information.

3. Statistics Generation:

The system must generate quantitative statistics, such as term frequency, average message length, and temporal activity in conversations.

4. Intuitive User Interface:

The user interface should be intuitive and user-friendly, enabling users to interact efficiently with the system's functions.

5. Real-Time Notifications:

Capability to provide real-time notifications to alert users about important events or completed analyses.

6. Multilingual Support:

The system should be capable of analyzing messages in multiple languages to accommodate user diversity.

Non-Functional Requirements:

1. Data Security:

A high level of security is required to protect the privacy and confidentiality of extracted messages and analyzed data.

2. Efficient Performance:

The system must be efficient in terms of performance, ensuring fast response times and agile processing capacity.

3. Multi-Platform Compatibility:

It should be compatible with various platforms to ensure accessibility from different devices and operating systems.

4. Scalability:

The system must be scalable, capable of handling an increase in message volume and users without significant performance degradation.

5. Comprehensive Documentation:

Detailed and clear documentation must be provided, describing the installation, configuration, and use of the system, facilitating user understanding and adoption.

3. Technologies Used

The project will primarily utilize Python as the programming language, making use of the multiple libraries made available by the Python community. The system will use a hand-programmed *parser* to crop the exported .txt file from WhatsApp and fill the proper data structures, making the data easily accessible for the future. For the statistics calculation, no more than a couple mathematical functions will be needed. We will use the native Python plotter for the graphical display of statistics and a library (haven't decided which one yet) to program the GUI. Lastly, we will need an AI analyzer that'll read the chat, retrieve information about the topics discussed and the main outcomes of the conversation. For this, we will most likely use the ChatGPT Premium API.

4. Project Timeline (ideally)

Week 1: Have a look at the format of the exported chats. Write a parser according to it and proceed with a couple of tests.

Week 2: Write the code in charge of data management and recognition, as well as the methods that'll generate the plots and graphics.

Week 3: Choose the technical tools that we'll use for the GUI implementation. We don't want it to be too complicated, so the implementation can start here.

Week 4: Finish the GUI implementation and thoroughly test the functionality developed until this moment. If possible, start to study the ChatGPT API.

Week 5: Develop the AI analyzer and implement it in the existing application. Improve the GUI with this addition.

Week 6: Final tests.