International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Analyzing WhatsApp Chat Data for User Interaction and Sentiment: A Comprehensive Approach

Mohammad Ali Siddiqui[1], Krishna Agarwal[2]

[1,2]Department of Information Technology, BBD Engineering College, India

Abstract

The proliferation of instant messaging applications like WhatsApp has revolutionized personal and professional communication. This research paper presents a detailed methodology for analyzing WhatsApp chat data to gain insights into user interaction patterns, message content, and sentiment. By employing various data preprocessing techniques and analytical methods, the project extracts meaningful statistics, generates word clouds, identifies frequently used words and emojis, analyzes user activity over time, and calculates response times. Additionally, sentiment analysis is performed to understand the emotional tone of conversations. This study provides a framework for leveraging chat data to inform social behavior analysis, digital communication studies, and personalized user experiences.

1. Introduction

Instant messaging platforms have become integral to daily communication, with WhatsApp being one of the most widely used applications globally. Understanding the dynamics of conversations on such platforms can offer valuable insights into social interactions, user behavior, and emotional well-being. This research aims to develop a comprehensive WhatsApp chat analyzer that can preprocess chat data, extract key metrics, and visualize interaction patterns and sentiments.

2. Literature Review

Prior research has explored various aspects of chat data analysis, including sentiment analysis (Medhat et al., 2014), user interaction patterns (Bamman et al., 2014), and the impact of digital communication on social behavior (Turkle, 2012). However, an integrated approach combining these elements in a user-friendly tool remains underdeveloped. This paper builds on existing methodologies and incorporates advanced data visualization techniques to enhance the analytical capabilities.

3. Methodology

3.1 Data Collection

WhatsApp chat data is typically exported as a text file, containing timestamps, user identifiers, and message content. For this study, chat data from multiple group and individual conversations were collected and anonymized to protect user privacy.

3.2 Data Preprocessing

Preprocessing involves parsing the text file to extract relevant fields such as date, time, user, and message content. Regular expressions were used to handle various date and time formats and to differentiate

between user messages and system notifications.

python

```
def preprocess(data):
pattern = '\d{1,2}/\d{1,2}/\d{2,4},\s\d{1,2}:\d{2}\s-\s'
messages = re.split(pattern, data)[1:]
dates = re.findall(pattern, data)
df = pd.DataFrame({'user_messages': messages, 'message_date': dates})
df['message_date'] = pd.to_datetime(df['message_date'].str.strip('- '), format='%m/%d/%y, %H:%M %p')
df.rename(columns={'message_date': 'Date'}, inplace=True)
return df
```

3.3 Statistical Analysis

Key metrics such as the number of messages, word count, media files shared, and links posted are calculated. Additionally, the busiest users are identified, and their activity levels are visualized.

python

```
def fetchstats(selected_user, df):
if selected_user != 'Overall':
df = df[df['User'] == selected_user]
num_messages = df.shape[0]
words = [word for message in df['Message'] for word in message.split()]
media_omitted = df[df['Message'] == '<Media omitted>'].shape[0]
links = [link for message in df['Message'] for link in extract.find_urls(message)]
return num_messages, len(words), media_omitted, len(links)
```

3.4 Common Words

The most common words are also identified and listed.

python

```
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

3.5 Emoji Analysis

Emojis play a crucial role in digital communication, expressing emotions succinctly. The frequency of different emojis is calculated to understand their usage patterns.

python

```
def getemojistats(selected_user, df):
if selected_user != 'Overall':
df = df[df['User'] == selected_user]
emojis = [char for message in df['Message'] for char in message if char in emoji.UNICODE_EMOJI['en']]
emoji_df = pd.DataFrame(Counter(emojis).most_common(), columns=['Emoji', 'Count'])
return emoji_df
```

3.6 Activity Analysis

User activity is analyzed on a daily and monthly basis to identify trends and peak interaction times. This is visualized using bar charts and line plots.

python

```
def weekactivitymap(selected_user, df):
if selected_user != 'Overall':
df = df[df['User'] == selected_user]
return df['Day_name'].value_counts()
```

3.7 Sentiment Analysis

Sentiment analysis is performed using the TextBlob library to gauge the emotional tone of the conversations. The sentiments are categorized as positive, negative, or neutral.

python

```
def fetch_sentiment_analysis(selected_user, df):
if selected_user != 'Overall':
df = df[df['User'] == selected_user]
df['Sentiment'] = df['Message'].apply(lambda msg: TextBlob(msg).sentiment.polarity)
sentiment_df = df['Sentiment'].apply(lambda polarity: 'Positive' if polarity > 0 else 'Negative' if polarity < 0 else 'Neutral').value_counts().reset_index()
sentiment_df.columns = ['Sentiment', 'Count']
return sentiment_df
```

3.8 Response Time Analysis

Response times between consecutive messages are calculated to understand the interaction dynamics and responsiveness of users.

python

Copy code

```
def calculate_response_times(df):
df['DateTime'] = pd.to_datetime(df['Date'])
df['ResponseTime'] = df['DateTime'].diff().shift(-1)
df['ResponseTime'] = df['ResponseTime'].dt.total_seconds().abs() / 60.0  # convert to minutes and make absolute
response_time_df = df.groupby('User')['ResponseTime'].mean().reset_index()
response_time_df.columns = ['User', 'AvgResponseTime']
response_time_df = response_time_df.dropna()
return response_time_df
```

4. Results and Discussion

The implemented chat analyzer successfully processes WhatsApp chat data to extract and visualize key metrics. The generated word clouds and common word lists reveal the prevalent topics and themes in the conversations. Emoji analysis highlights the most frequently used emojis, providing insights into the emotional tone. Activity maps show distinct patterns in user interaction over different times of the day and months of the year. Sentiment analysis categorizes messages into positive, negative, and neutral sentiments, offering a deeper understanding of the emotional dynamics. Response time analysis reveals the average responsiveness of users, highlighting interaction patterns.

4.1 Key Findings

1. **Statistical Analysis**: The number of messages, word counts, media files, and links shared were calculated for each user and overall.
2. **Word Cloud and Common Words**: The word cloud visualization highlighted the most frequently

used words, providing an overview of the conversation topics.

3. **Emoji Analysis**: Frequently used emojis were identified, illustrating the emotional expressions within the chat.
4. **Activity Analysis**: User activity trends were visualized, showing peak times of interaction on daily and monthly scales.
5. **Sentiment Analysis**: The sentiment of messages was categorized into positive, negative, and neutral, providing insights into the overall emotional tone.
6. **Response Time Analysis**: The average response times were calculated, revealing interaction patterns and user responsiveness.

5. Conclusion

This research demonstrates the potential of analyzing WhatsApp chat data to gain comprehensive insights into user interactions and sentiments. The developed analyzer provides a robust framework for preprocessing chat data, extracting meaningful statistics, and visualizing interaction patterns and sentiments. Future work can expand on this framework by incorporating advanced machine learning techniques for more nuanced sentiment analysis and user behavior prediction.

6. References

1. Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, 5(4), 1093-1113.
2. Bamman, D., Eisenstein, J., & Schnoebelen, T. (2014). Gender identity and lexical variation in social media. Journal of Sociolinguistics, 18(2), 135-160.
3. Turkle, S. (2012). Alone Together: Why We Expect More from Technology and Less from Each Other. Basic Books.