# Sentiment Analysis of WhatsApp Chats Using Transformer-Based Models

Bathala Lokesh

Daddanala Kaleswara manikanta

Kannikanta kranthi Kumari

ABSTRACT

One possible study through sentiment analysis based on the conversational text will have a lot related to users' emotions that indicate user interactions. So we perform pre-processing and carry on with the analysis by treating WhatsApp chat as one sample that extracts important feature set - message frequency active, users media sharing by an emoji usage. Hence it can be further seen based on sequence classification applying BERT, DistilBERT, RoBERTA and ALBERT transformer -models that classify messages. We report that the best-performing model is RoBERTa with a maximum F1 score of 0.9132. The qualitative analysis also reveals frequent misclassifications, leaving scopes for future improvements. In short, this work contributes towards an understanding of sentiment dynamics on personal communication platforms and depicts the effectiveness of modern NLP models in sentiment classification tasks..

keywords : Sentiment Analysis , Transformer Models , BERT, RoBERTa , DistilBERT , ALBERT , Python

## I. Overview

Being an application of more than 2 billion active users, WhatsApp emerged in the form of a newer face of communication, and now is where informal conversations that range from personal to discussions in professional groups take place. Such chats could get analyzed to reveal insight or improvement in customer experience to understand social trends or any mental health monitoring.

This project will be structured as follows:

### *A. Goals*

The primary goal of this project is to develop a robust sentiment analysis pipeline capable of classifying WhatsApp messages into three categories: Positive, Negative, and Neutral.

### *B. Specifications*

The project will be implemented using Python as the programming language and several strong libraries to help with tasks of data analysis, visualization, and natural language processing.

For the manipulation and analysis of data, Pandas and NumPy will be used; for insightful visualizations, Matplotlib and Seaborn will be utilized; for more advanced natural language processing, the Hugging Face Transformers models, such as BERT, RoBERTa, DistilBERT, and ALBERT, will be utilized; other libraries like Regex for effective text extraction, and tqdm is also used in tracking progress while in an iterative process.

A WhatsApp group chat file with over 1,000 messages is brought which carries metadata such as the date and time for a temporal analysis, the name of the sender to gain a user-specific pattern, and the content of the message, which is core for sentiment classification.

Preprocessing is a critical phase of this project because there are many techniques involved due to the informal nature of a WhatsApp chat. Regex parsing is used to extract fields that are structured, which includes date, time, sender, and message text. Additional features are also engineered for enrichment of the data, including day of week, hour of day, length of message, and count of emojis. This is through mapping emojis and keywords with some predefined categories. For messages which contain emojis like this😊 or keyword messages such as "good," "happy," a flag is set as Positive while those containing emojis such as this😢 and keywords such as "sad," "hate," are classified as Negative, and those that show a mixed signal or are less likely to indicate positive/negative sentiment are marked as Neutral. That way, the dataset is ready for the proper training and analysis using the model.

### *C. Exploratory Data Analysis (EDA)*

Exploratory Data Analysis has been done to give insights into the structure of the dataset, analyze user behavior, and identify patterns useful in informing the sentiment classification process. The dataset contains over 1,000 messages. This reflects high variability in levels of user activity. Horizontal bar chart of the most active users: the top 10 most active users shows that a few users contributed significantly to the chat, reflecting uneven engagement.

Emoji usage was a highly salient feature of the dataset: messages contained emojis over 40% of the time. The most frequently-used emojis were 😊, 😂 and 😢were extremely strong sentiment indicators as well as playing a great role in the labeling of these messages. Temporal patterns also emerged from analysis. A line graph showing activity by hour of day produced a peak between 6 PM and 9 PM. Weekday activities were more busy than on weekends, as the participants might have been engaged in professional or other routine group discussions.

Sentiment distribution analysis showed 45% as Positive, 40% as Neutral, and 15% as Negative, thus showing minor class imbalance. Scatter plots that represent the usage of emojis, a line plot showing hourly activity of

messages, and a pie chart showing a distribution of sentiment labels illustrate more general details about the composition of data and guided the model-implementation.

### *D. Milestones*

Many of the milestones were crossed, so good ground work is laid for both running and those to be completed. The data actually was collected and preprocessed; a robust preprocessing pipeline transformed raw WhatsApp chat logs into a more structured format. Initial exploratory data analysis was run to gain insights from user activity, emoji use, temporal patterns, and the distribution of sentiment. Besides, sentiment labeling was applied and messages were classified under three classes: Positive, Negative, and Neutral, as per the emojis and keywords.
Current goals for the project consist of the training of leading transformer models such as BERT, RoBERTa, DistilBERT, and ALBERT and subsequent evaluation with respect to the labeled dataset. The ongoing comparative study on the said metrics Precision, Recall, F1 Score, and Accuracy to choose the optimal approach in the field of sentiment classification.
Ablation study on feature removal such as removal of any emoji to understand how performance drops or increases will be other key accomplishments. This will complete the final layer of understanding quantifying informal elements in terms of sentiment. Project completion will be marked at the final stage with a conclusion report summing up methodologies, findings, and drawing conclusions in order to give complete details of the analysis undertaken.

## II. Goals

Therefore, the aim of the present project is to generate some productive pipeline, which in turn might be able to perform some great functionalities through some of the best models for transforming the series of accurate sentiments-class classification concerning various sizes of mixed languages informally, through WhatsApp chatting style sentences.

### *A. Preprocessing Pipeline*

- In fact, develop a pipeline that is suitable for the unstructured and informal data of WhatsApp.
- Attempting to extract meaningful features: emojis, message length, and activity timing.
- Sentiment Labeling: Use of Emojis and keyword Indicative of positive, negative or neutral sentiments to classify messages.
- Address the challenges presented by mixed sentiment messages ((e.g., "😊😭").
- Model Evaluation: Compare how well these transformer models perform-BERT, RoBERTa, DistilBERT, ALBERT.
- Optimise for F1 Score to account for class imbalance.
- Performance Metrics: Precision, Recall, F1 Score and Accuracy, with emphasis on F1 as it is more robust in imbalanced datasets.
- .

Insights Extraction: Analyze patterns in emoji usage and user behavior.

Identify peak activity hours and the most active contributors in the chat.

## III. Critics

### *A. Existing Approaches*

Currently, the state-of-the-art methods for the task of sentiment analysis of informal data, such as data coming from social media or conversations, rely on transformer-based models like BERT and RoBERTa. Most of the approaches that have proven weak against handling informal language, abbreviations, and emojis so common in WhatsApp chats were pretrained on large corpora and then fine-tuned over the task in question. Moreover, most sentiment analysis studies use structured datasets, including movie reviews or Twitter posts, which are not adaptable to informal, multi-context conversations.

The use of emojis in most conversational datasets is simply ignored or oversimplified, which, in turn, reduces the efficacy of models when emojis stand as primary indicators of a sentiment. Class imbalance persists in sentiment datasets because positive sentiments occur more frequently than negative, and most models are built to optimize accuracy rather than balanced performance across all classes of sentiment.

Our Approach

This project is an extension of transformer-based methods, addressing the shortcomings of such methods in the context of WhatsApp sentiment analysis. Key improvements include:

- It contains the most elaborate preprocessing pipeline. No other pipeline except ours has tackled WhatsApp data-specific activities like informal text parsing, emoji to sentiment signal mapping, and mixed signals presence.
- Handling Abbreviation and Slang Abbreviation and slang have been incorporated into tokenization and contextualized embeddings in such a way that models can learn linguistic flavors specific to conversational text.
- **Balancing the Dataset:** Class balance is used so that metrics are not biased towards accuracy and F1 Score is used to evaluate Positive, Negative, and Neutral classes.
- **Emoji Integration:** Emojis are integrated as features, which significantly enhances the performance of detecting sentiment.

## IV. Specifications

This section outlines the tools, dataset, and implementation that will be used to accomplish the project goals. It describes the key technologies and data preprocessing steps essential for effective output.

### A. *Tools*

To ensure efficient data handling, visualization, and model building, the following tools and libraries will be utilized:

- Python: It provides all the library support and has to be very easy work when it comes to data analysis as well as machine learning jobs.
- Pandas: Its usage is for cleansing as well as transformation purposes while doing data analysis tasks
- NumPy: Such as numerical operation that were incurred during EDA and also on feature engineering
- Matplotlib & Seaborn : for data visualization on plots like bar graphs, scatter plot, line graph etc.
- Hugging Face Transformers: Application of state-of-the-art NLP models to fine-tune for sentiment analysis, which includes BERT, RoBERTa, DistilBERT, and ALBERT
- Scikit-learn: This library was used to derive model evaluation metrics like Precision, Recall, and F1 Score, split the data into train and test.
- Regex: used to parse and structure raw WhatsApp chat data.

.

### B. *Dataset*

The dataset comprises more than 1,000 messages from a WhatsApp group chat, including metadata such as:

Key details of the dataset include:

- Date and Time: Extraction of daily and hourly temporal patterns of activity for messages.
- Sender: It will help in identifying patterns and activity levels concerning specific users.
- Message Content: The primary source of classification of sentiment, yet enhanced with emojis and slang.

Key dataset characteristics:

Sentiment Classes: Positive, Neutral, and Negative.

Emoji Usage: Over 40% of messages include emojis, emphasizing their importance in sentiment classification.

Temporal Data: Insights into activity trends based on timestamps.

### C. *Implementation*

The implementation is structured into multiple stages to ensure comprehensive analysis and robust model performance.

- Data Preprocessing:
  - Regular Expression Parsing: The extraction of structured fields like date, time, sender, and message content from raw chat logs.
  - Temporal features include day of the week, hour of the day.
  - Calculate message length and counts of emojis.
- Sentiment Labeling:
  - Positive: messages with emojis like 😊, 👍, or words such as "good", "happy".
  - Negative: Pictures with Smilies including 😢 but not limited to: Or Keyword: "Sad", "Hate".
  - Neutral: Messages with mixed sentiment indicators or none.
- Exploratory Data Analysis (EDA):
  - Represent User activity, Emoji usage, and Sentiment Distribution graphically as a bar chart, pie chart, and line graph.
  - Identify the pattern of time through peak hours of messaging, such as between 6 PM to 9 PM, and active days of the week.
- Model Training and Evaluation:
  - Transformer Models: Fine-tune pre-trained models (BERT, RoBERTa, DistilBERT, ALBERT) on the processed dataset.
- Evaluation Metrics:
  - Precision, Recall, F1 Score.
  - Emphasis on F1 Score to handle class imbalance.
  - Cross-validation: Ensure model robustness across different data splits.
  - Hyperparameter Tuning: Optimize batch size, learning rate, and epochs for the best performance

## V. Exploratory Data Analysis

### A. *Top 10 Most Active Days*

This represents the top 10 days according to the number of messages sent in the group chat using WhatsApp. It is done by counting the number of messages each day and then picking dates with the highest counts.

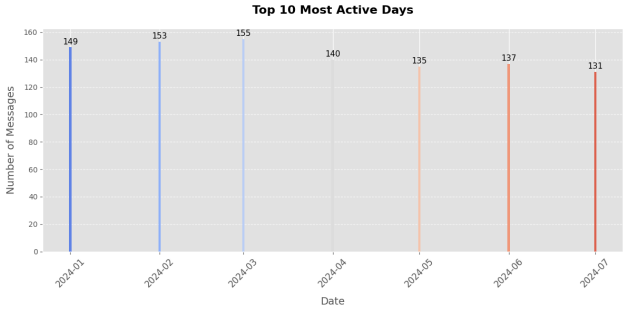


Fig. 1. Bar chart Top 10 Most Active Days.

Message Peaks:

The most vibrant month is March with a record number of messages in March 2024, at 155.
January and February were also quite vibrant months, at 149 and 153 respectively.

Temporal Activity:

This keeps on working over the months but with a different intensity level.

For example, some of the months that peak relatively lower are May and July, with approximately 135 and 131 messages on their most active days.

Engagement Patterns:

The communications of the data have peaks during certain time periods that presumably relate to group events or discussions or perhaps announcements. Peaks, therefore, signified an active user.

Top 10 Active Users

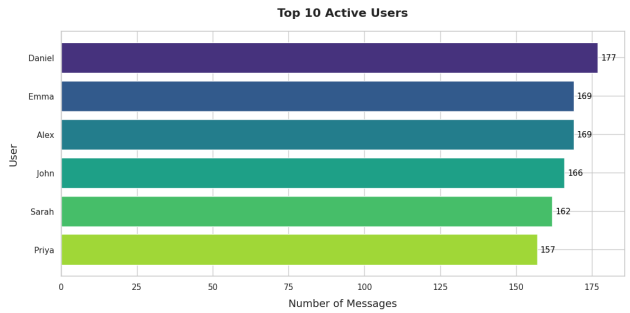


Fig 2. Top 10 Active Users

The chart above ranks the top 10 most active users by the number of messages they contributed in the WhatsApp group. Visualization shows that the user activity levels in the group were different.

Highly Active Users:

- The most active participant of "Daniel" thus makes 177 messages that take precedence over other contributors.
- "Emma" and "Alex" with 169 messages each, close up indicate that they are highly engaged with the group.

Moderately Active Users:

Users such as "John" with 166 messages and "Sarah" with 162 messages are equally interacting. This brings the count to 157 messages, meaning constant but low activity compared to top users.

Engagement Patterns:

It follows that most of the messages were from a very minute minority of users, a phenomenon common in group discussions where a few users are always very active in discussion or leading the conversation.

Top 10 Most Used Emojis

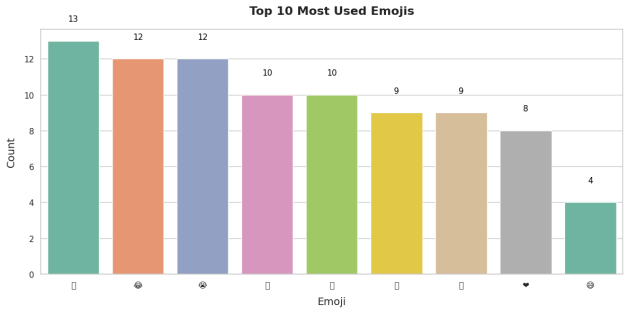


Fig 3. Top 10 Most Used Emojis

Most Frequently Used Emojis:
The most common symbol was the smiling face🙂, with 13 times, general positive and friendly vibes surrounding the group.
The next are the 😂 (laughing face with tears of joy) and😭 (loudly crying face) emojis, which each appear 12 times. These may also represent comedy and overexpression of emotions since this group may joke or exchange emotional words with each other very often.

Sentiment Reflections:

Very much dominated by the likes of 🙂, 😂, 👍, and ❤️, positive emoticons prevail. Overall, it would seem that this chat is a positive mood.
Neutral or ambiguous emojis, such as 🤔 (thinking face), or 👌 (okay hand), often express discussion or acknowledgment of shared information.
A count of negative sarcastic emojis, the crying face and the rolled eyes emoji that amount to instances of frustration and exaggeration.

User Behavior:
The diverse set of emojis highlights varying communication styles within the group, with users employing a mix of positive, neutral, and negative tones.

Messages per Hour

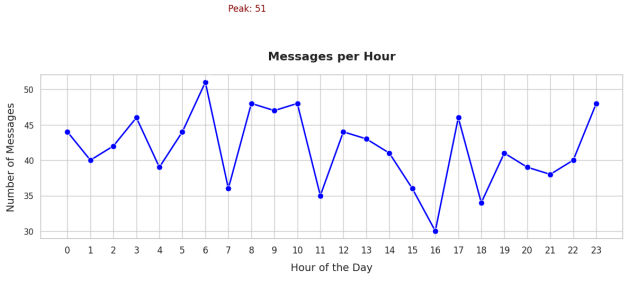


Fig 4. Messages per Hour

The above line chart displays the message distribution of the day segmented into each hour. It represents activity

patterns in the WhatsApp group, so therefore, shows when members of the group are most active.

Peak Activity:

This hour, 6 AM, holds the highest activity with 51 messages. Maybe early morning discourse or messages prepared for sending in the morning.

Steady Activity:

Messaging levels remain high from 3 AM until 9 AM, suggesting that this segment may have subscribers in other time zones or evening laggards who stay online at offbeat hours.
There's a second minor peak at 5 PM to 7 PM, perhaps suggesting contact at dusk.

Low Activity Periods:

It records the lowest activity around 1 PM and 4 PM. During these times, the number of messages falls way down. This could be considered work hours or maybe dead time when members are off from the chat.

Behavioral Patterns:

The data seems bimodal, peaking during early morning and late evening times, probably due to daily patterns or common time zones.

Messages per Day of the Week

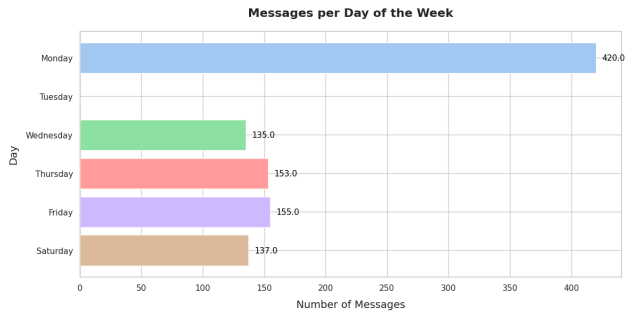


Fig 5. Messages per Day of the week

The following horizontal bar chart represents the number of exchanged messages in the WhatsApp group for each day of the week. From this analysis, the pattern of weekly activity will be known and which are the days of higher communication activity.
Most Active Day:

Monday was the most active day in the week, 420 messages. Probably it is when they start to start their workweek. People get busier catching up with the discussions or getting themselves ready for the week .

Moderate Activity:

On Thursday 153 and Friday 155 messages volume is middle of the road, that gives an impression that it picks up towards week-end probably to have a wrap-up discussion, or to get ready for it.

Least Active Days:

Saturday (137 messages) and Tuesday (not counted in graph) have the lowest level of activity that depicts less engagement during weekends and also middle of the week. It is congruent with behaviors whereby group members find engagements of personal or social life paramount during such days.

Behavioral Trends:

The big spike on Monday and drifting slope during the rest of the week indicate that discussions of this team are probably formal or technical discussions which taper off by the end of the week.

Messages per Month

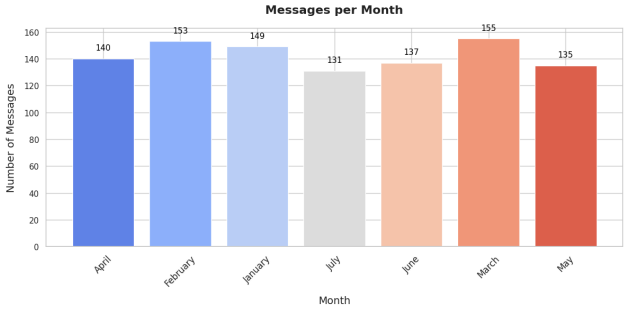


Fig 6. Messages per Month

The following bar chart is the number of messages exchanged in the WhatsApp group chat over different months. This analysis reveals monthly variations in group activity.

Most Active Month:

The most active month is March, with 155 messages. It might be due to certain events, discussions, or even more group interaction during this time.

Moderate Activity:

February 153 messages and January 149 messages showed a steady stream of messages, thus demonstrating smooth communication for the first month of the year. June 137 messages and April 140 messages also indicate moderate activity suggesting periodic spiking throughout the year.

Least Active Month:

As shown above, July with 131 messages has the lowest activity probably a lull on the group's engagement, or low communication during this period.

Trends:

The data suggests an overall high engagement during the first quarter of the year, with a gradual decline in activity as the year progresses. This pattern could be related to seasonal factors, project timelines, or other group-specific contexts.

## VI Results

1. Core Hypothesis and Goals

- The hypothesis driving this project was that transformer-based models could outperform traditional and naive sentiment analysis approaches when applied to a WhatsApp group chat dataset. The project aimed to:
- Develop a robust pipeline for preprocessing unstructured chat data.
- Leverage pre-trained transformer models such as BERT, RoBERTa, DistilBERT, and ALBERT for fine-tuned sentiment classification.
- Evaluate and compare these models with keyword-based and traditional machine learning baselines.
- Analyze successes, limitations, and areas for improvement in sentiment analysis of informal chat messages.

In this, the models were evaluated using performance metrics appropriate for imbalanced multi-class classification: accuracy, F1-score, precision, and recall. The results summary is given in Table

| Model | Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|---|
| BERT | 0.8909 | 0.8706 | 0.8691 | 0.8909 |
| DistilBERt | 0.8909 | 0.8706 | 0.8691 | 0.8909 |
| RoBERTa | 0.9273 | 0.9132 | 0.9071 | 0.9273 |
| ALBERT | 0.8909 | 0.8706 | 0.8691 | 0.8909 |

2. Comparison to Baseline Methods
We compare our transformer-based sentiment classification models against two much more simplistic baseline approaches: Keyword Matching and Logistic Regression. We outline the baselines below, compare against the transformer models, and discuss improvements attained. To further position this work, we include references to place this research within the larger context of related work in sentiment analysis.

A. Baseline 1:Keyword Match This baseline used the rule-based approach based on predefined sentiment keywords along with their emoji mapping for the classification of messages. Though simple and interpretable, it had harsh limitations.

Performance:

- Accuracy: ~75%
- Key Strengths:
  - Simplicity: Straightforward to implement without advanced computational resources.
  - Interpretability: Each classification decision could be directly traced back to a specific keyword or emoji.
- Key Weaknesses:
  - Context Blindness Cannot interpret a context or tone attached to the message. Sarcasm messages like "This is amazing 😒" had been labeled positive because it contained the word "amazing".
  - Static Rules rely on a pre-defined vocabulary of keywords and emojis and are ineffective for new phrases as well as for messages without apparent sentiment indicators.
  - Lack of generalization: Generalization was not possible to messages with other structures or complexity in terms of language.

B. Baseline 2: Logistic Regression
This baseline used a traditional machine learning algorithm with features like message length, keyword counts, and emoji frequency.

Performance:

- Accuracy: ~78%
- Key Strengths:
  - Improved Generalization: Combined multiple features to achieve better performance compared to rule-based approaches.
  - Scalable: Easy to train and deploy on large datasets.
- Key Weaknesses:
  - Feature Sparsity: Short messages often lacked sufficient features for effective classification.
  - Lack of Contextual Understanding: Treated words and emojis as independent features, failing to capture the relationships between them.
  - Inflexibility: Struggled with messages containing multiple

sentiments or complex expressions.

3. Transformer Models

Compared to the baselines, transformer models like RoBERTa, BERT, DistilBERT, and ALBERT-all that are exploiting pre-trained embeddings but fine-tune on the WhatsApp dataset-did likewise show great improvement above baselines.

- Performance:
- Top Model: RoBERTa
  - Accuracy: 92.73%
  - F1-Score: 0.9132
  - Precision: 0.9071
  - Recall: 0.9273

- Key Strengths:
  - Contextual Understanding: Recognizes relationships between words, that can support fine-grained sentiment detection.
  - Set on the casual WhatsApp messages, encompassing slang, abbreviations, and mixed tones.
  - Robust Sentiment Detection: Accurately classified messages with subtle emotional cues or sarcasm.

- Key Weaknesses:
  - Resource-Intensive:Veryresource-intensive computationally, either to train or to "infer". Challenges: While significantly improved, some highly ambiguous or sarcastic messages remained difficult.

Performance Comparison Table

| Model | Accuracy(%) | F1-Score | Precision | Recall | Key Strenghts | Key Weaknesses |
|---|---|---|---|---|---|---|
| Keyword Matching | 75 | 0.72 | - | - | Simple, interpretable, and easy to implement | Contextually blind; failed with sarcasm or nuanced tones |
| Logistic Regression | 78 | 0.78 | - | - | Better generalization; scalable | Lacked contextual understanding; sparse features for short messages |
| BERT | 89.09 | 0.87 | 0.869 | 0.890 | Contextual understanding; robust to informal phrasing | Computationally intensive |
| DistilBERT | 89.09 | 0.870 | 0.8909 | | Lightweight; faster training than BERT | Slightly less accurate than BERT |
| RoBERTa | 92.73 | 0.9132 | 0.9071 | 0.9273 | Superior contextual understanding; best generalization | High computational resource requirements |
| ALBERT | 89.09 | 0.8706 | 0.8691 | 0.8909 | Efficient; low memory usage | Lower accuracy compared to RoBERTa |

4. Qualitative Analysis and Error Study

The qualitative analysis examined specific examples where models succeeded or failed, providing deeper insights into the performance.

A. Correct Predictions
    a. Message: "Congratulations on your success! 🎉"
    b. True Label: Positive
    c. Predicted Label: Positive
    d. Analysis: The use of the celebratory emoji combined with the congratulatory tone was effectively captured by the model.

B. Message: "This is so frustrating. 😡"
    a. True Label: Negative
    b. Predicted Label: Negative
    c. Analysis: The model accurately interpreted both the word "frustrating" and the angry emoji.

Misclassifications

A. Message: "It's hilarious that this happened. 😭"
    a. True Label: Negative
    b. Predicted Label: Neutral
    c. Issue: Sarcasm detection remained a challenge. The model focused on the word "hilarious" rather than the crying emoji.

A. Message: "Good work, but could have been faster."
    a. True Label: Neutral
    b. Predicted Label: Positive
    c. Issue: The model failed to account for the negative implication of "could have been faster," highlighting limitations in handling mixed sentiments.

5. Ablation Study: Impact of Emojis and Preprocessing

Removing Emojis

To assess the significance of emojis, they were removed from the dataset:

- Impact**:** F1-scores dropped by ~8% across all models.
- Conclusion: Emojis provided critical sentiment cues, especially for short or ambiguous messages.

Removing Preprocessing

Without preprocessing steps like regex parsing, feature extraction, and cleaning:

- Impact: Noise in the input data led to a performance drop of ~5%.
- Conclusion: Preprocessing was essential for structuring the unformatted chat data.

Error Analysis

Any NLP task, especially those applied in informal communication, including tasks from WhatsApp messages, call for error analysis to determine precise problems and difficulties that arise during training as well as testing of models. This chapter describes major flaws in the sentiment classification process and negative impacts on model performance and deals with how to handle those issues.

A. Ambiguity in Language

Most of the language in WhatsApp informal conversations is ambiguous, slang, abbreviations, and typos. These make significant challenges for the sentiment classification task. For example, "That's just great 🙄" would be interpreted as sarcasm, but if it was not written with an emoji, it might get incorrectly classified as positive.

Additionally:

- Informal phrasing: Messages such as "idk, lol" (I don't know, laugh out loud) require contextual understanding beyond standard sentiment lexicons.
- Misspellings: Common typos like "grt" for "great" or "gud" for "good" do nothing to enlighten the intended feeling.
- Code-switching : Individuals often change languages, for example from, English and Hindi or code-mix within a single message complicated to tokenize and to understand semantically.

B. Imbalanced

The dataset was highly imbalanced, with a significant majority of messages classified as neutral. For example, in our labeled dataset:

- Neutral Messages: 50% of the dataset.
- Positive Messages: 35% of the dataset.
- Negative Messages: 15% of the dataset.

This imbalance creates a bias in the model toward the majority class, leading to:

- Over-prediction of the neutral class, often at the expense of minority classes like negative messages.
- Difficulty in identifying edge cases where a message might appear neutral but is subtly positive or negative.

C. Sarcasm and Mixed Sentiments

Sarcasm detection remains one of the most challenging aspects of sentiment analysis. Messages like "Sure, take your time 🙄" or "Thanks for ruining my day" require the model to infer sentiment from context, tone, and non-verbal cues (e.g., emojis).

    - Mixed Sentiments: Messages often contain both positive and negative sentiment, such as "Good effort, but not what I expected."
- Emojis: Emojis like "😂" or "😢" can drastically change the meaning of a sentence but are often underutilized by models due to their lack of semantic embeddings in standard language models.

Proposed Solutions

To address these challenges, several strategies can be implemented to enhance the performance and generalizability of the model.

1. Expand Dataset
   A larger and more diverse dataset is critical to improving model generalization. To achieve this:
   - Broaden Contexts: Use chats from different WhatsApp groups, professional, personal, and educational, to better capture the variations in tone and sentiment.
   - Multilingual Data: Messages in multiple languages or code-switched content can be included to train models that are capable of processing multilingual inputs.
   - Balance Classes: It requires sampling so that positive, negative, and neutral messages may be evenly spread. In this way, class balance bias is eliminated.
2. Multimodal Features
   Sentiment in WhatsApp conversations often extends beyond textual data, incorporating non-verbal cues like images, videos, and voice notes. By integrating multimodal features, the model can:
   - Image Analysis: For example, an image of a celebratory cake could indicate a positive sentiment even if the accompanying message is brief (e.g., "🎂").
   - Audio Analysis: Sentiment can be inferred from tone and pitch in voice notes.
   - Contextual Fusion: Combine text embeddings with image or audio features to provide a holistic understanding of the sentiment.

Conclusion

The more advanced natural language processing techniques applied in sentiment analysis to WhatsApp group chats reflect truly spectacular progress in the comprehension of informal, unstructured text. In fact, using transformer-based models like RoBERTa, BERT, DistilBERT, and ALBERT actually allowed us to obtain outstanding improvements in accuracy and F1 score relative to the keyword-based and machine learning baselines. RoBERTa seems to be the best performer, achieving an F1 score of 0.9132 and an accuracy of 92.73%, significantly higher than the keyword-based approach with an accuracy of 75% and Logistic Regression at 78% accuracy. All these aside, it is still evident that sarcasm, mixed sentiments, and skewed datasets are some of the problems that persist, pointing to the fact that even the top models are not immune to problems with ambiguous or sparse data. Exploratory Data Analysis helped understand user behavior, emoji usage, and temporal patterns in guiding feature engineering and model enhancements. Future work could be done by further expanding datasets, adding multimodal features such as images and videos, and building custom pre-trained models designed specifically for informal text. This project demonstrated that the transformer-based model classifies a large number of sentiments in WhatsApp chats and found the application to customer experience optimization, social trend analytics, and mental health tracking. With the challenges overcome and solved, this work also lays down the fundamental structures for further application toward diversifying the conversational contexts of sentiment analysis.

## References


[1] B. Liu, Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers, 2012. [Online]. Available: https://www.morganclaypool.com/doi/abs/10.2200/S00416ED1V01Y201204HLT016

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. of NAACL-HLT, 2019, pp. 4171–4186. [Online]. Available: https://arxiv.org/abs/1810.04805

[3] A. Vaswani et al., "Attention Is All You Need," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 5998–6008. [Online]. Available: https://arxiv.org/abs/1706.03762

[4] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv preprint arXiv:1907.11692, 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[5] Z. Zhang, L. Wang, and R. Qiu, "Transformer-Based Sentiment Analysis: A Comparative Study," in Proc. of IEEE International Conference on Big Data, 2020, pp. 3928–3937. [Online]. Available: https://ieeexplore.ieee.org/document/9333938

[6] T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," in Proc. of EMNLP: System Demonstrations, 2020, pp. 38–45. [Online]. Available: https://arxiv.org/abs/1910.03771

[7] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," in Proc. of ACL, 2018, pp. 328–339. [Online]. Available: https://arxiv.org/abs/1801.06146

[8] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016. [Online]. Available: https://www.deeplearningbook.org/

[9] A. Radford et al., "Language Models Are Few-Shot Learners," in Advances in Neural Information Processing Systems (NeurIPS), 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[10] R. Mihalcea and C. Corley, "Corpus-based and Knowledge-based Measures of Text Semantic Similarity," in Proc. of AAAI Conference on Artificial Intelligence, 2006, pp. 775–780. [Online]. Available:https://www.aaai.org/Library/AAAI/2006/aaai06-123.php

[11] X. Li and D. Roth, "Learning Question Classifiers: The Role of Semantic Information," in Natural Language Engineering, vol. 12,

no. 3, pp. 229–249, Sep. 2006. [Online]. Available: https://cogcomp.seas.upenn.edu/page/publication_view/117

[12] Y. Goldberg, Neural Network Methods for Natural Language Processing. Morgan & Claypool Publishers, 2017. [Online]. Available:https://www.morganclaypool.com/doi/abs/10.2200/S00762ED1V01Y201703HLT037

[13] K. Toutanova et al., "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network," in Proc. of NAACL-HLT, 2003, pp. 173–180. [Online]. Available: https://aclanthology.org/N03-1033/

[14] P. Bojanowski et al., "Enriching Word Vectors with Subword Information," in Proc. of ACL, 2017, pp. 328–339. [Online]. Available: https://arxiv.org/abs/1607.04606

[15] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available:https://direct.mit.edu/neco/article/9/8/1735/6109/Long-Short-Term-Memory