WENSA: Whatsapp Emotion and Sentiment Analyzer Using LSTM and BI-LSTM

R V Sanjay[a], Abhinav Kumar[a]

[a]*National Institute of Technology, Puducherry, , Karaikal, 609609, Puducherry, India*

Abstract

In recent times, messages are the primary mode of communication. In today's world, the most popular chat application for communication is WhatsApp. It has been widely used by all, especially among the business people and youngsters. WhatsApp group chats can be analyzed using several analyzing tools. Authentically, users wish to analyze their chat for several purposes. This study's primary goal is to examine WhatsApp group discussions in order to gauge how involved and active each member is. The most active day in the group, the quantity of messages sent on that date, the most active user overall, the list of active group members, the total number of users, the quantity of posts made by each member on the group, and the most frequently used phrase on the platform are all analyzed. Additionally, a daily, monthly, and annual study of the top 15 members' messages is conducted. The other objective is to predict the emotion of text data in terms of joy, sadness, anger, etc. In this study, LSTM (a deep learning algorithm) network architecture has been used to analyze emotions for the given text. The experimental evaluations have been performed on emotion data-set from Kaggle, which contains 16,000 records in the English language with different emotions. Validation Accuracy is the evaluation metric used to analyze emotions in this study.

Keywords: WhatsApp, Sentiment Analysis, LSTM (Long Short-Term Memory), Group Chats, Communication

1. Introduction

Curiosity serves as a powerful motivator that has driven many individuals to accomplish remarkable feats. People are naturally curious about how they are perceived by others based on their conversations, whether one-on-one or in group settings. To enhance the efficiency of our model, we require a substantial amount of data. For this purpose, we've turned to one of Meta's data-rich assets – WhatsApp. WhatsApp boasts an impressive statistic of nearly 85 billion messages exchanged daily, with the average user dedicating over four hours a week to the platform and participating in numerous group chats.
The solution presented by this study involves applying exploratory data analysis to messages extracted from WhatsApp chats. These messages are obtained through WhatsApp's built-in export feature and require preprocessing to optimize computation. Emotion detection, or sentiment analysis, refers to the techniques, methods, and tools used for identifying and extracting subjective information, including attitudes and opinions, from text. Historically, the main focus of sentiment analysis has been to classify text into positive, negative, or neutral[2]. However, these three classifications might not adequately convey the complex implications of the underlying sentiment in the given text.

2. Literature Survey

Sentiment analysis, often known as opinion mining, is the field of research that focuses on the analysis of people's opinions, feelings, assessments, judgments, attitudes, and emotions about different entities. These entities cover a broad range of things, such as goods, services, groups, people, problems, occasions, subjects, and the qualities that go along with them. [3] This field encompasses a broad and multifaceted problem domain, with numerous names and related tasks, such as sentiment analysis, opinion mining, opinion extraction, subjectivity analysis, affect analysis, emotion analysis, and review mining. While the industry predominantly employs the term "sentiment analysis," in academic circles, both "sentiment analysis" and "opinion mining" are frequently utilized. Before the turn of the millennium, linguistic and natural language processing (NLP) had a well-established history, but there was a noticeable scarcity of research dedicated to understanding people's opinions and sentiments. Subsequently, the field of sentiment analysis has evolved into a highly active research domain. Several factors have contributed to this evolution. Firstly, sentiment analysis boasts a wide array of applications spanning almost every domain. The industry, driven by the proliferation of commercial applications, has flourished, serving as a compelling incentive for research endeavors. Secondly, sentiment analysis presents an array of complex research challenges that had remained largely unexplored until recently. Figure 1 shows that consequently, research in sentiment analysis not only exerts a significant influence on the field of NLP but also holds the potential to deeply impact disciplines such as management sciences, political science, economics, and social sciences, given their inter connectedness with public opinion.

2.1. Analysis of Usage and Impact of WhatsApp Using Text Mining

A survey was conducted by RaviShankara.[1] to investigate WhatsApp usage patterns and associated impacts. The same

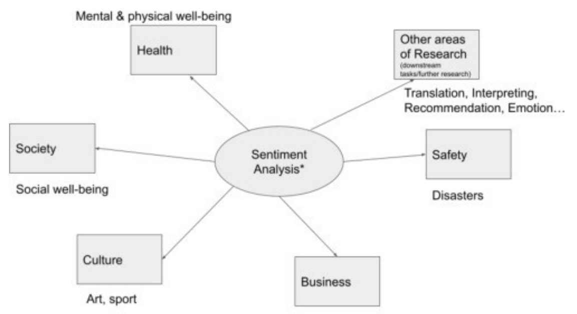
Mental & physical well-being
Health
Other areas of Research
(downstream tasks/further research)
Translation, Interpreting, Recommendation, Emotion…
Society
Sentiment Analysis*
Safety
Social well-being
Disasters
Culture
Business
Art, sport

Figure 1: Emotional Analysis [5]

topics have been examined in numerous other studies. While some studies examined people in particular areas, others concentrated on students. In a study conducted in southern India, participants ages 18 to 23 were asked to rate the significance of WhatsApp to them. It was discovered that the students use WhatsApp for roughly eight hours every day and are online for nearly sixteen hours. Additionally, they discovered that WhatsApp is mostly used by young people when they are using smartphones.

2.2. *Statistical Analysis on Group chats of WhatsApp Using R*

Sunil Joshi [4] introduced a statistical analysis approach to delve into WhatsApp group chats, aiming to categorize these conversations based on various key parameters. The dataset under scrutiny encompasses WhatsApp group chats spanning a one-year period, from May 2019 to May 2020, comprising a total of 55,563 records. This dataset includes distinctive attributes that shed light on the extent to which individuals engage with WhatsApp group chats.These characteristics include factors such as the duration of usage on a daily, weekly, monthly, or annual basis, response patterns, message formats (including emojis, text, and multimedia), the age groups of active participants, and more.

The primary attributes examined in this analysis involve the nature of the messages exchanged, the frequency of usage across different timeframes, timestamps (AM/PM), the age groups and gender identities of chat participants. RStudio, a widely favored open-source integrated development environment for the R programming language, was the tool of choice for conducting both exploratory data analysis and data visualization. The selection of RStudio was primarily motivated by its open-source and accessible nature.

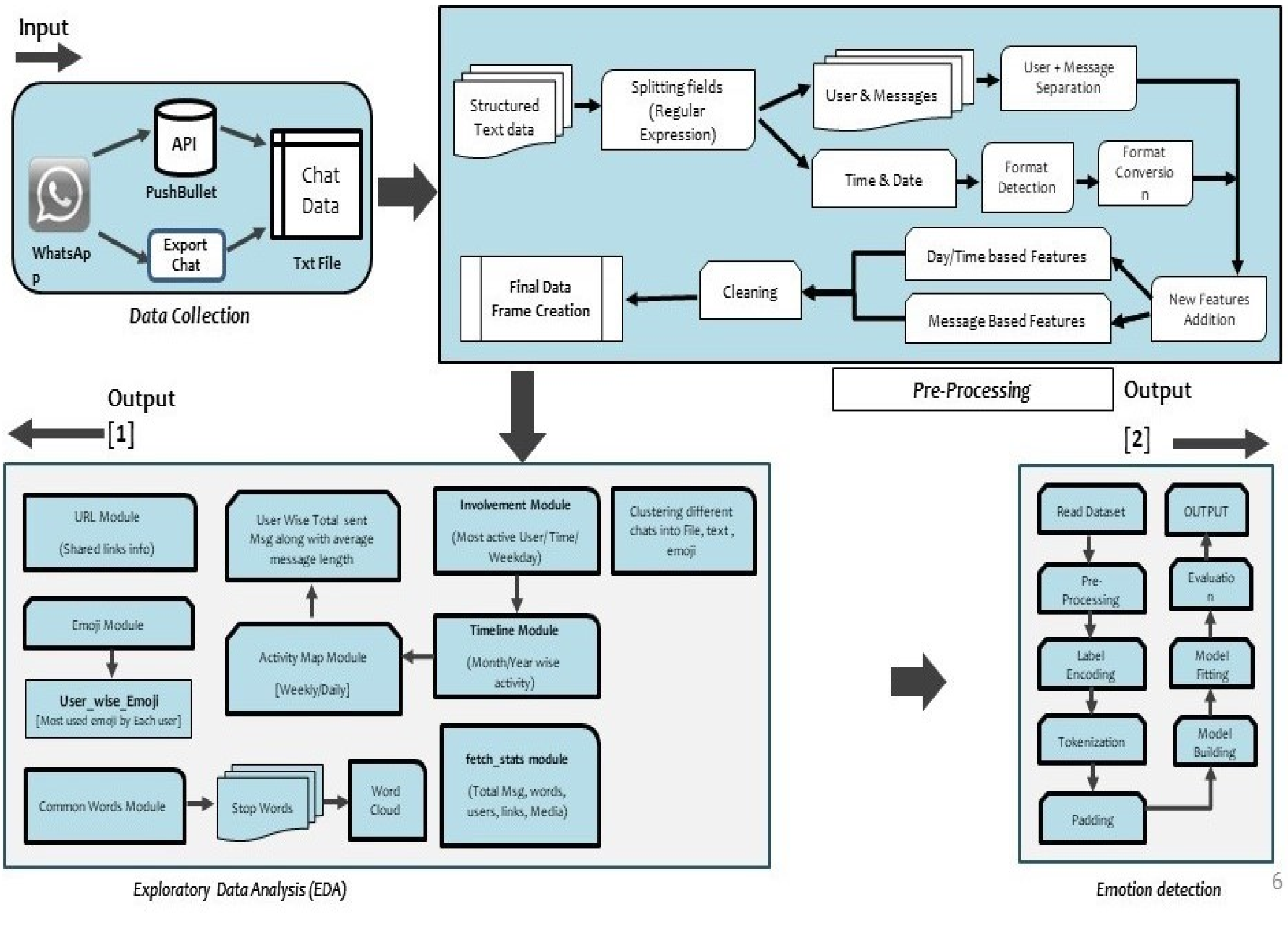
Input
API
PushBullet
Chat Data
WhatsApp
Export Chat
Txt File
Data Collection
Structured Text data
Splitting fields (Regular Expression)
User & Messages
User + Message Separation
Time & Date
Format Detection
Format Conversion
Day/Time based Features
Message Based Features
New Features Addition
Cleaning
Final Data Frame Creation
Pre-Processing
Output [1]
Output [2]
URL Module (Shared links info)
User Wise Total sent Msg along with average message length
Involvement Module (Most active User/Time/Weekday)
Clustering different chats into File, text, emoji
Emoji Module
User_wise_Emoji [Most used emoji by Each user]
Activity Map Module [Weekly/Daily]
Timeline Module (Month/Year wise activity)
Common Words Module
Stop Words
Word Cloud
fetch_stats module (Total Msg, words, users, links, Media)
Exploratory Data Analysis (EDA)
Read Dataset
Pre-Processing
Label Encoding
Tokenization
Padding
Model Building
Model Fitting
Evaluation
OUTPUT
Emotion detection
6

Figure 2: Proposed FrameWork .

3. Methodology

3.1. Proposed FrameWork

Figure 2 depicts the architectural model which is used to perform EDA on chat data and analyze emotion from any text. A WhatsApp group discussion was used as a baseline for exploratory data analysis, and a dataset with four different emotions was used for emotion detection. The LSTM and BI-LSTM network architectures are used to implement the emotion detection model. We provide a platform with several levels for sentiment analysis (Fig.2).The embedding layer uses vectors of length 200 for representation of each word, the convolution layer is responsible for feature extraction with respect to filters and the LSTM layer has 128 memory units(smart neurons) that represent the dimensionality of outer space, mainly used for classifying, processing and predicting time series data.

3.2. Module Description

1. Data Collection
2. Preprocessing
3. Exploratory Data Analysis
4. FastText Word Vector Model
5. LSTM
6. BI-LSTM

3.2.1. Data Collection

When conducting Exploratory Data Analysis (EDA), any WhatsApp chat that has been exported can serve as a suitable dataset. WhatsApp typically exports these chats in the form of text/zip files. These chat logs consistently adhere to a set pattern, structured as [Timestamp-Username-Messages], presented as a sequential string. To extract information from these logs, regular expressions are applied. Users have the option to decide whether to include media files (such as images and other attachments) or exclude them. It's worth noting that including media can significantly increase the file size, which is why many prefer to export chats without media files. Once the chats are exported, they are uploaded as raw datasets ready for analysis.

3.2.2. Preprocessing

Once the exported chat is loaded into the application, the initial step is to extract key information such as Date, Time, Day, User, Message, URL, and more. Following this, a series of date-related manipulations and cleaning processes are carried out on the messages. This involves the removal of stop-words and punctuation marks. The pre-processing stage can be further broken down into six major sub-modules.

1. Splitting fields
2. Data Frame creation
3. Data format conversion
4. Time format conversion
5. Feature addition
6. Cleaning

3.2.3. Exploratory Data Analysis

The Streamlit framework is used to build the interface. During EDA, various libraries like Numpy, Pandas, Seaborn, Matplotlib, Emoji, and nltk have been used. Exploratory analysis on WhatsApp comes under the category of text mining in Natural Language Processing.
Tasks like feature extraction, regular expressions operations, clustering, and classification have been used on pre-processed WhatsApp data. Various kinds of charts such as bar charts, pie charts, column charts, line charts, and scatter plots come under the Matplotlib library. Pandas and Numpy libraries have been used as they contain a lot of methods to support data frame-related stuff. The final output is generated as various kinds of charts and graphs in the form of visualization. This module is responsible for data visualisation and observation. It mainly includes modules like
1.**Fetch_stats** This module is used to filter people, URLs, messages, and words, as well as their counts, by going through all communications. The URLSurf library and regular expressions were used to identify a URL. In Figure 3, statistical information from chats has been calculated.

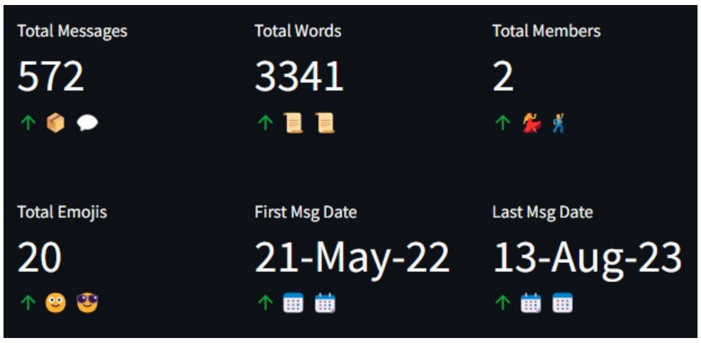
Total Messages
572
Total Words
3341
Total Members
2
Total Emojis
20
First Msg Date
21-May-22
Last Msg Date
13-Aug-23

Figure 3: Analysis of a chat

2.**Timeline** This module is responsible for analyzing the time in a day or weekday during which most conversations have occurred. It also helps in predicting which time is best to send a message so that we can get a response.
In Figure 4, day-wise message activity throughout the year has been displayed, and higher raised lines state that most conversations occur in chats on that day. The intuition behind it is to count the total number of messages day-wise and sort them in descending order and plot it on a graph.

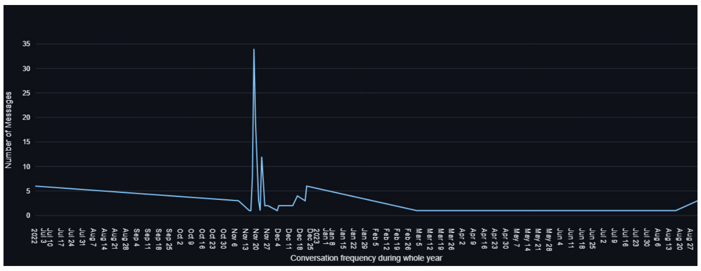

Figure 4: Day wise message activity throughout the year

3.**Top Users and Message Length** This module is responsible for performing the text mining techniques like classification on WhatsApp chats. After performing these instructions, a 2D dataframe with relevant records will be evaluated. Mainly two different functionalities have been performed here, which are:

- Separating user message and applying count to identify busiest user.
- Fetching top users from chat by most number of messages and average length of messages.

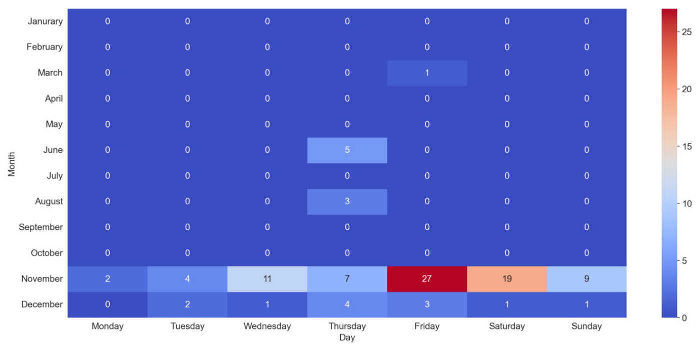

Figure 5: Monthly and weekly Active Time Duration

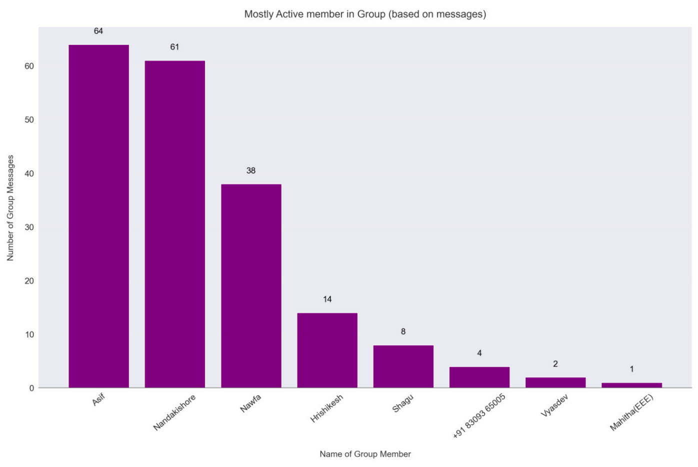

Figure 6: Most Active Days During the Chat

Figure 5 shows the user activity throughout all the days in a week for each month. Top 10 members with the most sent messages have been displayed in Figure 6, and average word length has been displayed here to detect who sends long messages normally. Here, matplotlib bar chart and line chart have been used for visualization.

4.**Word Cloud** WordCloud is a powerful visual representation object for text processing, which shows the most frequent words with bigger and bolder letters, and with different colors. The smaller the size of the word, the lesser it's important.

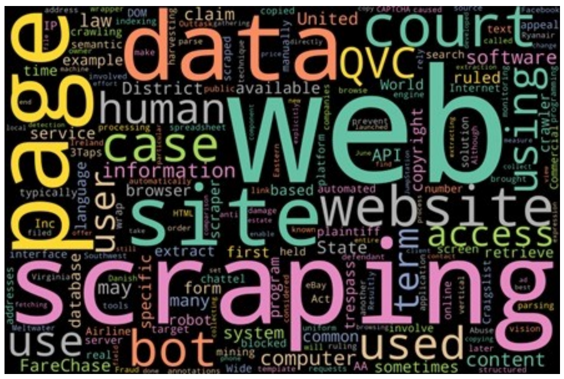

Figure 7: WordCloud

Figure 7 states the most used words which have been filtered after removing punctuation and stopwords.

3.2.4. FastText Word Vector Model

Word embedding techniques are widely used in sentiment analysis, and the FastText word embedding technique is used in this study. FastText is a Word2Vec model extension that uses word embedding. Instead of using words to create word embeddings, FastText goes a step further. This deeper level is made up of word and character fragments. For example, the FastText representation of the word "artificial" with n=5 is [artif, rtifi, tific, ifici, ficia, icial] where the angular brackets indicate the beginning and end of the word.

This approach has two major advantages. To begin, generalization is possible as long as new words share the same characteristics as existing ones. Second, because more information can be extracted from each piece of text, less training data is required.

3.2.5. LSTM based classification

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network that has been modified to make it easier to remember and recall past data. They successfully address the vanishing gradient issue that plagues traditional recurrent neural networks. LSTMs excel at tasks like classifying, processing, and predicting time series data with unknown inter-event durations. Back-propagation is used to train these neural networks. At the core of an LSTM network is the cell state, which serves as a kind of memory that enables the LSTM to remember previous information. LSTMs feature gates with sigmoid activation functions, which produce values between 0 and 1, often becoming either 0 or 1 in practice. In an LSTM network, three gates are incorporated to manage and control the flow of information.LSTM is made up of gates, in LSTM we have 3 gates:[6]

Formula for calculating Input Gate

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, xt] + b_i\right) \tag{1}$$

$$\tilde{C}_t = \tanh\left(W_c\right).[h_{t-1}, x_t] + b_c \tag{2}$$

Formula for calculating Forget Gate

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{3}$$

Formula for calculating Forget Gate

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{4}$$

$$h_t = O_t * \tanh(C_t) \tag{5}$$

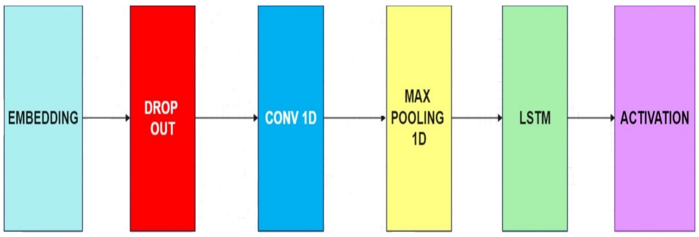

Figure 8: LSTM Architecture

Proposed algorithm for EDA

```
1: Start
2: Import numpy,pandas,matplotlib etc.
3: Read Dataset()
4: Split the dataset into timestamp and message
5: Split the timestamp into time and date
6: Split the message into user and the actual message(msg)
7: pre data ← This is the modified dataset after Pre-Processing
8: Check for any url's in msg
9: Check for any emoji's in msg
10: Analyse the sentiment of msg
11: Time Series of msg
12: Activity Map per day/week
13: Statistic Calculation of msg
14: Scatter plot of user activity
15: Word cloud(most common word)
16: End
```

3.3. Emotion Detection Using LSTM

Long Short-Term Memory Networks (LSTMs) are a type of RNN capable of learning long-term dependencies. LSTMs are explicitly designed to avoid the problem of long-term dependency. It is practically their default behavior to remember information for long periods of time. This repeating module in standard RNNs will have a very simple structure, such as a single tanh layer. A memory cell known as a 'cell state' that maintains its state over time plays a central role in an LSTM model. The horizontal line that runs through the top of the diagram below represents the cell state. In LSTM, information can be added to or removed from the cell state, which is controlled by gates. These gates allow information to flow into and out of the cell. It includes a pointwise multiplication operation as well as a sigmoid neural net layer to help the mechanism.

Proposed algorithm for LSTM

```
1: Start
2: Import numpy, pandas, LabelEncoder, train_test_split,
Tokenizer
3: From keras.layers import Dense, LSTM, Embedding,
Dropout, Activation, Conv1D, MaxPooling1D
4: Read the csv file "train.txt" into df_train
5: Apply Label Encoding on y_train
6: Initialise Tokenizer() as tokenizer
7: Fit tokenizer on X_train & X_test on axis=0
8: Apply padding to X_train
9: Initialise Sequential() as model
10: Add Embedding Layer to the model
11: Add Dropuot Layer to the model
12: Add one dimensional Convolutional Layer to the model
13: Add Max Pooling Layer to the model
14: Add LSTM Layer to the model
15: Add more hidden layers to the model
16: Add softmax Activation Layer to the model
17: Predict the emotion
18: End
```

3.4. Emotion Detection Using Bi-LSTM

Bidirectional Long Short-Term Memory (Bi-LSTM) is an advanced recurrent neural network architecture that differs from LSTM in its ability to process information from both past and future data points. While LSTM only considers past data to make predictions, Bi-LSTM employs two separate LSTM layers, one processing data forwards and the other in reverse. This bidirectional approach enables Bi-LSTM to capture dependencies from both earlier and later points in a sequence, providing a more comprehensive context for understanding the data. It excels in tasks like speech recognition, sentiment analysis, and machine translation, where considering both past and future information is crucial for accurate predictions and modeling complex dependencies. FastText is a word embedding model developed by Facebook AI Research (FAIR) that excels in natural language processing. It differs from traditional word embeddings by leveraging subword information. FastText represents words as a combination of character n-grams (subword units). For example, the word "apple" can be deconstructed into n-grams like "ap," "pp," "pl," "le," and even the whole word itself ("apple"). This approach captures both word meaning and morphological structure. FastText employs a skip-gram model, a technique used in word2vec, to learn word representations. In this model, the objective is to predict context words given a target word. FastText applies this to full words and their constituent n-grams, allowing it to generate embeddings for both. Additionally, it employs a hierarchical softmax technique for efficient training by organizing the vocabulary into a binary tree. The advantages of FastText include its ability to handle out-of-vocabulary words effectively, capture morphological information, achieve efficient training, and serve as a text classification framework. It's especially useful for languages with complex vocabularies and NLP tasks like sentiment analysis and text classification.

Proposed algorithm for BI-LSTM

```
1: Start
2: Import Bidirectional, Tokenizer, Dense, Embedding,
Dropout
```

```
3: Read the csv file ”train.txt” into df_train
4: Apply Label Encoding on y_train
5: Initialise Tokenizer() as tokenizer
6: Apply padding to X_train
7: Initialise Sequential() as model
8: Add Embedding Layer to the model
9: Add Bidirectional LSTM Layer(128)
10: Add Dropuot Layer to the model
11: Add Bidirectional LSTM Layer(256)
12: Add Dropout Layer to the model
13: Add Bidirectional LSTM Layer(128)
14: Add more hidden layers to the model
15: Compile the model
16: Predict the emotion
17: End
```

4. Model Building of LSTM and Bi-LSTM

Figure 8 shows the layers that are included in the sequential model : The first layer is the Embedded layer that uses a vector of length 200 for representing each word and it maps word indices to vectors. The second is the dropout layer which was chosen because there is a risk of over-fitting with a larger number of factors. It stops all neurons in a layer from maximizing their weights at the same time and aids in improving the accuracy score. Here, the maximum dropout is 25%. The third layer is the Convolution layer, which is mostly utilized for filter feature extraction. If the kernel size is 3, it will extract a 1x3 size for 1D and a 3x3 size for 2D. A kernel is a tiny matrix with dimensions less than the training dataset to be twisted. It's also referred to as a convolution matrix or a convolution mask.The final component of the convolutional layer is an activation function, which increases nonlinearity in the output. In general, the ReLu function and Tanh function are the two most commonly utilized activation functions in a convolution layer. The Max-pooling layer comes next, it is used to compress the input image and accelerate computation. The LSTM layer follows, with 128 memory units (smart neurons) representing the dimensionality of outer space. It is mostly used for time series data classification, processing, and prediction. The Dense layer is the final layer, it is used with a single neuron and a sigmoid activation function to create 0 or 1 predictions for each emotion because this is a classification task. Softmax generates a probability distribution from a vector of values. The output vector's members are in the range (0, 1) and add up to 1.

5. Evaluation results

The overall accuracy of model is found to be 95.15% after iteration of 30 with a loss of 17.51%

Class label	Precision	Recall	F1-score	Support
Anger	0.90	0.92	0.91	275
Fear	0.96	0.88	0.92	224
Joy	0.96	0.99	0.97	695
Sadness	0.95	0.94	0.95	581

Table 1: Evaluation Metrics of LSTM

Class label	Precision	Recall	F1-score	Support
Anger	0.91	0.94	0.92	275
Fear	0.95	0.93	0.94	224
Joy	0.99	0.98	0.99	695
Sadness	0.95	0.95	0.95	581

Table 2: Evaluation Metrics of BI-LSTM

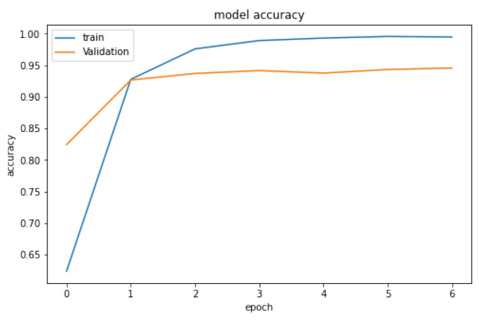

Figure 9: Model accuracy in each epoch

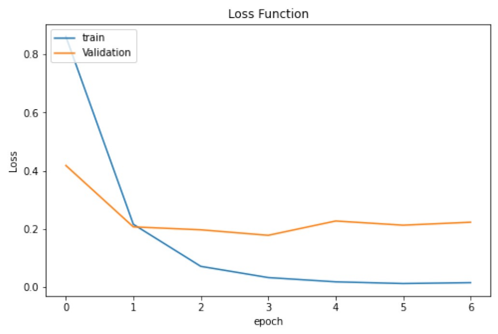

Figure 10: Model Loss in each epochs

Figure 9 illustrates the relationship between the number of iterations and model accuracy during training, whereas Figure 10 illustrates the relationship between iteration and model loss. The model is trained up to 16000 iterations with an average accuracy of 95.15% and a loss of 17.51%. Figure 9 shows that towards the conclusion, training accuracy is slightly higher than validation accuracy and Figure 10 shows that training loss is slightly lower than validation loss. This suggests more epochs will be required to train to widen the gap, in order to prevent overfitting.

6. Conclusion

The significance of social network analysis is explored throughout this study. This research asserts that exploratory data analysis provides a gateway to the undiscovered world of WhatsApp Chats and successfully was able to use the necessary tools to develop an analysis of a WhatsApp group chat and visually depict the top 15 people in the chat group, analyze the emojis and most active day/month/year, and so on. Ultimately, the analysis produced the desired results and was able to demonstrate the level of engagement of all participants in the exported WhatsApp chat. Emotion Detection using LSTM model needed several intermediate layers for passing memory to get the desired output. In this project, seven intermediate layers are used to make the sequential model and to predict the emotion in four different categories ['sadness', 'anger', 'joy', 'fear']. A dataset of 16,000 records has been used, but most of the records have either 'joy' or 'sadness'. In this sequential model, a word embedding layer needs to be employed to understand how the model works and responds to the reviews or textual data. The unsupervised word embedding learning algorithm is used.

7. Future Work

Although we observed a lot of insights in exploratory data analysis, such as day-by-day messages, most active time in the group, most used terms during the conversation, and so on, in addition, we noticed a spike in hate speech, cyberbullying, heckling, and greater impudence on social media. This study lacks in features like time-wise topic modeling (which will indicate the conversations about topics between two periods), code recognition (in case programming language code is provided in chat), and product feature extraction (to collect specifics if the user discusses purchasing any product).
Majority records in Emotion detection relate to either joy or sadness; consequently, additional records need to be included for more accurate and diverse output. Furthermore, it now predicts emotions in four separate categories; it must be improved to categorize at least 15 emotions. In addition, the use of some Google API for emotion detection in any language can be implemented. Deep learning models' performance can be enhanced by varying word embedding algorithms and hyperparameters. It can produce the most accurate results by applying pre-trained NLP called Bi-directional Encoder Representation from Transformer (BERT) and training on English Wikipedia (2,500 million words) and Book Corpus (800 million words).

References

[5] D. Kanojia and A. Joshi, "Chapter 3 - Applications and challenges of SA in real-life scenarios," ScienceDirect, Jan. 01, 2023. https://www.sciencedirect.com (accessed Nov. 05, 2023).

[13] G. Rao, W. Huang, Z. Feng, and Q. Cong, "LSTM with sentence representations for document-level sentiment classification," Neurocomputing, vol. 308, pp. 49–57, Sep. 2018, doi: https://doi.org/10.1016/j.neucom.2018.04.045.

[1]IS Srajan, K Ravishankara, and Vaisakh Dhanush, "Whatsapp chat analyzer", International Journal of Engineering Research & Technology, 9(5):897–900, 2020.

[2] Huang Zou, Xin hua Tang, Bin Xie and Bing Liu, "Sentiment classification using machine learning techniques with syntax features", pages 175–179, 2015.

[3] J.Samuel, G.G. Ali, M. Rahman, E. Esawi, Y. Samuel, "Covid-19 public sentiment insights and machine learning for tweets classification", Information, 11 (6) (2020) 314.

[4]Sunil Joshi, "Sentiment analysis on whatsapp group chat using r", Data, Engineering & Applications, pages47–55,2019.

[6] U. B. Mahadevaswamy and P. Swathi, "Sentiment Analysis using Bidirectional LSTM Network," Procedia Computer Science, vol. 218, pp. 45–56, 2023, doi: https://doi.org/10.1016/j.procs.2022.12.400.

[7] R. K. Behera, M. Jena, S. K. Rath, and S. Misra, "Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data," Information Processing & Management, vol. 58, no. 1, p. 102435, Jan. 2021, doi: https://doi.org/10.1016/j.ipm.2020.102435.

[8] M. Arbane, R. Benlamri, Y. Brik, and A. D. Alahmar, "Social media-based COVID-19 sentiment classification model using Bi-LSTM," Expert Systems with Applications, vol. 212, p. 118710, Feb. 2023, doi: https://doi.org/10.1016/j.eswa.2022.118710.

[9] W. Zhang, L. Li, Y. Zhu, P. Yu, and J. Wen, "CNN-LSTM neural network model for fine-grained negative emotion computing in emergencies," Alexandria Engineering Journal, vol. 61, no. 9, pp. 6755–6767, Sep. 2022, doi: https://doi.org/10.1016/j.aej.2021.12.022.

[10] I. N. Khasanah, "Sentiment Classification Using fastText Embedding and Deep Learning Model," Procedia Computer Science, vol. 189, pp. 343–350, 2021, doi: https://doi.org/10.1016/j.procs.2021.05.103.

[11] S. Aslan, "A deep learning-based sentiment analysis approach (MF-CNN-BILSTM) and topic modeling of tweets related to the Ukraine-Russia conflict," Applied Soft Computing, vol. 143, pp. 110404–110404, Aug. 2023, doi: https://doi.org/10.1016/j.asoc.2023.110404.

[12] Z. Ahmed and J. Wang, "A fine-grained deep learning model using embedded-CNN with BiLSTM for exploiting product sentiments," Alexandria Engineering Journal, vol. 65, pp. 731–747, Feb. 2023, doi: https://doi.org/10.1016/j.aej.2022.10.037.

[14] S. Sadiq, A. Mehmood, S. Ullah, M. Ahmad, G. S. Choi, and B.-W. On, “Aggression detection through deep neural model on Twitter,” Future Generation Computer Systems, vol. 114, pp. 120–129, Jan. 2021, doi: https://doi.org/10.1016/j.future.2020.07.050.

[15] S. Zeng, C. Ma, J. Liu, M. Li, and H. Gui, “Sequence-to-sequence based LSTM network modeling and its application in thermal error control framework,” Applied Soft Computing, vol. 138, p. 110221, May 2023, doi: https://doi.org/10.1016/j.asoc.2023.110221.