North East University Bangladesh
Department of Computer Science and Engineering

Emoji Prediction from Sentence using Deep Learning

By

Jannatul Ferdous Jannah
Reg. No: 200103020003
BSc(Engg) in CSE
4th year 2nd semester

Sourov Dey
Reg. No: 200103020026
BSc(Engg) in CSE
4th year 2nd semester

Fahima Akther Moni
Reg. No: 200103020054
BSc(Engg) in CSE
4th year 2nd semester

Supervised By

Khadem Mohammad Asif Uz Zaman
Lecturer
Department of Computer Science and Engineering

21st January, 2024

Emoji Prediction from Sentence using Deep Learning

A project submitted to the Department of Computer Science and Engineering, North East University Bangladesh, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

By

Jannatul Ferdous Jannah	Sourov Dey	Fahima Akther Moni
Reg. No: 200103020003	Reg. No: 200103020026	Reg. No: 200103020054
BSc(Engg) in CSE	BSc(Engg) in CSE	BSc(Engg) in CSE
4^{th} year 2^{nd} semester	4^{th} year 2^{nd} semester	4^{th} year 2^{nd} semester

Supervised By

Khadem Mohammad Asif Uz Zaman
Lecturer
Department of Computer Science and Engineering

21^{st} January, 2024

Recommendation Letter from Project Supervisor

These Students, Jannatul Ferdous Jannah, Sourov Dey, Fahima Akther Moni whose project entitled “Emoji Prediction from Sentence using Deep Learning”, is under my supervision and agrees to submit for examination.

Signature of the Supervisor :

Khadem Mohammad Asif Uz Zaman
Lecturer
Department of Computer Science and Engineering
North East University Bangladesh

Qualification Form of BSc(Engg.) Degree

Student Name：Jannatul Ferdous Jannah, Sourov Dey & Fahima Akther Moni
Project Title：Emoji Prediction from Sentence using Deep Learning

This is to certify that the thesis is submitted by Jannatul Ferdous Jannah, Sourov Dey & Fahima Akther Moni on 09th November, 2023. It is qualified and approved by the following persons and committee.

Head of the Dept.

Rathindra Chandra Gope
Assistant Professor & Head
Department of CSE
North East University Bangladesh

Supervisor

Khadem Mohammad Asif Uz Zaman
Lecturer
Department of CSE
North East University Bangladesh

Abstract

Emojis are a small visual representation of emotions or objects that are usually used in text messages to enhance the communication experience between individuals. In this project, we propose an approach to predict multiple emoji for a given text-based tweet message. Our proposal contains three modules, where the first module is preprocesses the given text data, the second module is the model on which the data is trained, and a multi-class classifier to predict the emojis evoked by the given text. And finally testing or evaluating the models.

The objective of this model is to understand the underlying semantics of the text sentence using natural language processing techniques to predict reasonable emojis.

Keywords: Emoji, Deep Learning, Natural Language Processing, Tweets.

Table of Contents

List of Figures and Tables

Chapter 1

INTRODUCTION

Emojis serve as small visual representations of emotions or objects, commonly used in text messages to enhance communication between individuals. As social media platforms and instant messaging have gained immense popularity, users have increasingly embraced emojis to efficiently convey broad feelings that may be challenging to express through words alone. This fusion of text and emoji has become an integral aspect of modern communication, highlighting the need to explore the relationship between text messages and the emoji employed within them.

This report presents multiple methods to predict emojis for text-based tweets. The approach comprises three modules: preprocessing the data, training a model to recognize patterns between text and emoji, and using a multi-class classifier for prediction. By leveraging natural language processing techniques, the model aims to accurately predict appropriate emojis by understanding the semantics of text sentences.

By developing this predictive model, we aim to address the gap in understanding the intricate relationship between text-based messages and emojis. The model's effectiveness will be evaluated using a dataset comprising almost 50,000 English tweets, each labeled with the corresponding emoji associated with the message. Through training on this dataset, the classifier will acquire a deep understanding of natural language and the contextual nuances that trigger specific emoji. Consequently, when provided with a sentence as input, the classifier will generate the most appropriate emoji suggestion, enhancing the expressiveness and impact of the text.

Chapter 2

BACKGROUND STUDY

1. **Emoji Prediction from Twitter Data using Deep Learning Approach**
   V N Durga Pavithra Kollipara, V N Hemanth Kollipara, M Durga Prakash
   Used method: Naïve Bayes, SVM, Bi-LSTM
   Accuracy: 27%(Naïve Bayes), 29%(SVM), 89%(Bi-LSTM)

2. **Using Neural Networks to Predict Emoji Usage from Twitter Data**
   Luda Zhao, Connie Zeng
   Used method: LSTM, CNN
   Accuracy: 37%(LSTM), 40%(CNN)

3. **Context-Aware Emoji Prediction Using Deep Learning**
   By Anushka Gupta, Bhumika Bhatia, Diksha Chugh, Gadde Satya
   Used method: LSTM, BERT.
   Accuracy: 72.99%(BERT), 66.66%(LSTM)

4. **Emojify: Emoji Prediction from Sentence**
   By Chen Huang, Xueying (Shirley) Xie, Boyu (Bill) Zhang
   Used method: Multinomial Naïve Bayes, SVM, Bi-LSTM
   Accuracy: 19%(Multinomial Naïve Bayes), 16%(SVM), 15%(Bi-LSTM)

5. **Emoji Prediction with Transformer Models**
   By Wenna Qin, Jiacheng Ge
   Used method: GPT-2, BERT.
   Accuracy: 76%(BERT), 75%(GPT-2)

Chapter 3

PROPOSED METHOD

The proposed method in the project involves using regular machine learning models such as Naïve Bayes and SVM model and deep learning approach, such as LSTM, Bi-LSTM, GRU model, for emoji prediction from Twitter data.

3.1 Methodology

Figure 3.1.1: Methodology

3.2 Dataset

The twitter emoji dataset obtained from CodaLab comprises almost 50 thousand tweets along with the associated emoji label. Each tweet in the dataset has a corresponding numerical label which maps to a specific emoji. The emojis are of the 20 most frequent emojis and hence the labels range from 0 to 19 as seen in "Fig. 3.2.1". Initial dataset in raw text format is later written to a csv file using python scripts.

	Tweet	Label
0	A little throwback with my favourite person @ ...	0
1	glam on @user yesterday for #kcon makeup using...	7
2	Democracy Plaza in the wake of a stunning outc...	11
3	Then & Now. VILO @ Walt Disney Magic Kingdom	0
4	Who never... @ A Galaxy Far Far Away	2
5	Dinner in FLA tonight // Pan-seared salmon ove...	1
6	It's my fav seniors last game congrats on beat...	8

Label	Emojis	Description
0	❤️	Red heart\t
1	😍	Smiling face with hearteyes\t
2	😂	Face with tears of joy\t
3	💕	Two hearts
4	🔥	Fire
5	😊	Smiling face with smiling eyes\t
6	😎	Smiling face with sunglasses\t
7	✨	Sparkles \t

Figure 3.2.1: Sample Dataset

3.3 Pre-processing Data

3.3.1 Data Cleaning:

In general, we use a lot of punctuations and other words without any contextual meaning. Tweet texts often consist of other user mentions, hyperlink texts, emoticons and punctuations. The tweets are checked for duplicates while pre-processing and in case there is any tweet that is repeated more than once then duplicate tweets are removed.

- Firstly, the whole text is converted into lowercase for convenience.
- Removal of hashtag symbols (#), user mentions (@user), retweet tags(RT).
- Removal of hyperlinks or any HTML elements
- Removing numbers
- Removing punctuations

Tweet
A little throwback with my favourite person @ Water Wall
glam on @ user yesterday for #kcon makeup using @user in #featherette,...
Democracy Plaza in the wake of a stunnig outcome #Decision 2016 @ NBC News
Then & Now. VILO @ Walt Disney Magic Kingdom
Who never... A Galaxy Far Far Away
Dinner in FLA tonight // Pan-seared salemon over couscous veggie salad #yum #dinner
It's my fav seniors last game congrats on beating west @ West Salem...

Figure 3.3.1.1: Before Cleaning Data

Tweet
A little throwback with my favourite person water wall
glam on user yesterday for kcon makeup using in featherette...
democracy plaza in the wake of a stunnig outcome decision 2016 nbc news
then amp now vilo walt disney magic kingdom
who never a galaxy far far away
dinner in fla tonight panseared salemon over couscous veggie salad yum dinner
dinner with this priceless viewthank you s anniversarydinner columbuscircle augustrd...

Figure 3.3.1.2: After Cleaning Data

3.3.2 Data Pre-processing:

Natural language with which people communicate usually contain commonly repeated words such as ‘a’, ‘an’, ‘the’, which are commonly referred to as stop words. These words need to be filtered out before going for any further processing of the text since they do not greatly add any meaning to the sentence.

Next, lemmatization is applied on the text which determines the root word which belongs to the same language unlike stemming which only performs word reduction. Since lemmatization is typically more informative, it is opted over stemming. When lemmatization is applied to the words (‘eat’, ‘ate’, ‘eaten’), it reduces the words to a common lemma which is ‘eat’.

- Removal of Stop words
- Lemmatization of text

Tweet
A little throwback with my favourite person water wall
glam on user yesterday for kcon makeup using in featherette...
democracy plaza in the wake of a stunnig outcome decision 2016 nbc news
then amp now vilo walt disney magic kingdom
who never a galaxy far far away
dinner in fla tonight panseared salemon over couscous veggie salad yum dinner
Its my fav seniors last game congrats on beating west west salem...

Figure 3.3.2.1: Before Pre-processing Data

Tweet
little throwback favourite person water wall
glam yesterday kcon makeup featherette...
democracy plaza wake stunnig outcome decision nbc news
then amp vilo walt disney magic kingdom
galaxy far far away
dinner fla tonight panseared salemon couscous veggie salad yum dinner
dinner priceless viewthank s anniversarydinner columbuscircle augustrd

Figure 3.3.2.2: After Preprocessing Data

3.4 Background Theory of the Algorithms

3.4.1 SVM:

A Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks. It works by finding a hyperplane in a high-dimensional space that separates different classes with a maximum margin. The algorithm aims to identify the optimal hyperplane that maximally separates data points of different classes while minimizing classification errors.

SVMs use a kernel function to map input data into a higher-dimensional feature space, where a clear hyperplane separation becomes possible. The key concept is to find the hyperplane that has the maximum margin, defined as the distance between the hyperplane and the nearest data point of each class. This margin maximization ensures robust generalization to new, unseen data.

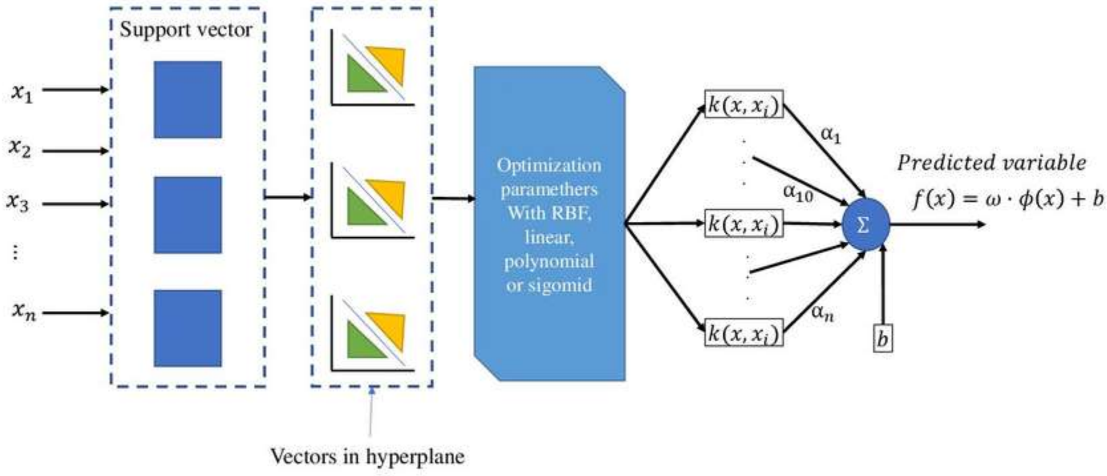

Figure 3.4.1.1: SVM Architecture

3.4.2 Naïve Bayes:

The Naive Bayes (NB) algorithm is a probabilistic classification technique based on Bayes' theorem. It assumes that the features used to describe instances are conditionally independent given the class label, which is a simplifying and often naive assumption. Despite its simplicity, Naive Bayes is efficient and works well in various text classification and document categorization tasks.

The algorithm calculates the probability of each class given a set of input features. It involves computing the prior probability of each class and the likelihood of the observed features for each class. The final classification decision is based on the class with the highest posterior probability.

3.4.3 LSTM:

LSTM (Long Short-Term Memory) is a neural network architecture designed for sequential data. It uses a memory cell, forget gate, input gate, and output gate to selectively process and store information, allowing it to capture long-term dependencies. The forget gate decides what to discard from the memory cell, the input gate determines what to add, and the output gate controls the information to output. This mechanism helps LSTM effectively learn and retain complex patterns in sequences, making it valuable for tasks like language processing and time series analysis.

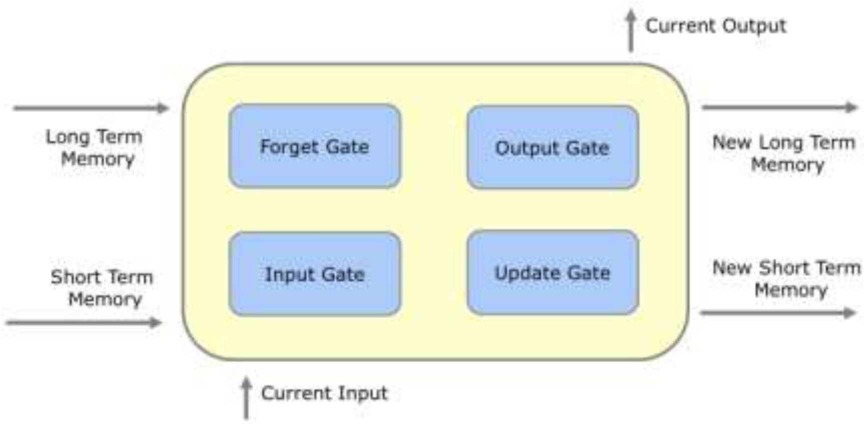

Figure 3.4.3.1: LSTM Architecture

3.4.4 Bi-LSTM:

Bi-LSTM (Bidirectional Long Short-Term Memory) is an extension of LSTM designed for sequential data. It processes input in both forward and backward directions simultaneously. The forward LSTM captures information from past to present, while the backward LSTM captures information from future to present. The outputs from both directions are typically concatenated, providing a comprehensive understanding of the sequence. This bidirectional approach enhances the model's ability to capture complex patterns and dependencies in both past and future contexts, making it effective in tasks like natural language processing and speech recognition.

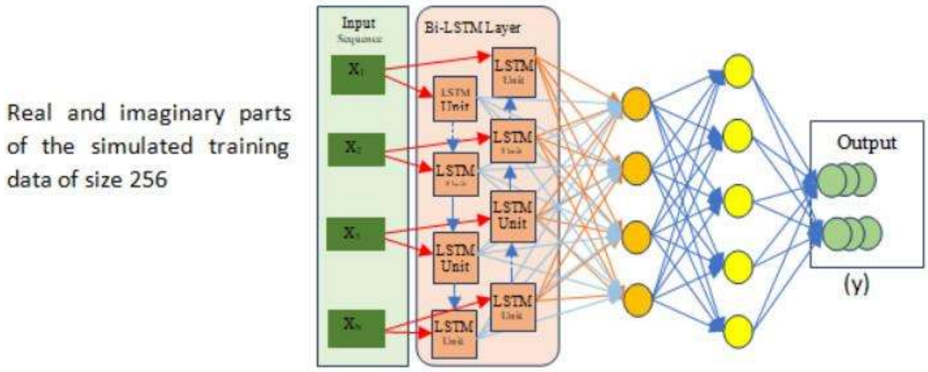

Figure 3.4.4.1: Bi-LSTM Architecture

3.4.5 GRU:

The Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture designed to address the vanishing gradient problem in traditional RNNs. It consists of a memory cell that maintains hidden state information and two gates—update gate and reset gate. The update gate determines how much of the past information to retain, while the reset gate decides how much of the past information to forget. These gates enable the GRU to selectively update and use relevant information from the input sequence.

The model processes sequential data by iterating over each element while updating its hidden state. The gates dynamically control the flow of information through the cell, facilitating the capture of long-range dependencies in sequences. The GRU's simplified structure allows for efficient training and mitigates vanishing gradient issues, making it suitable for various sequential tasks, including natural language processing.

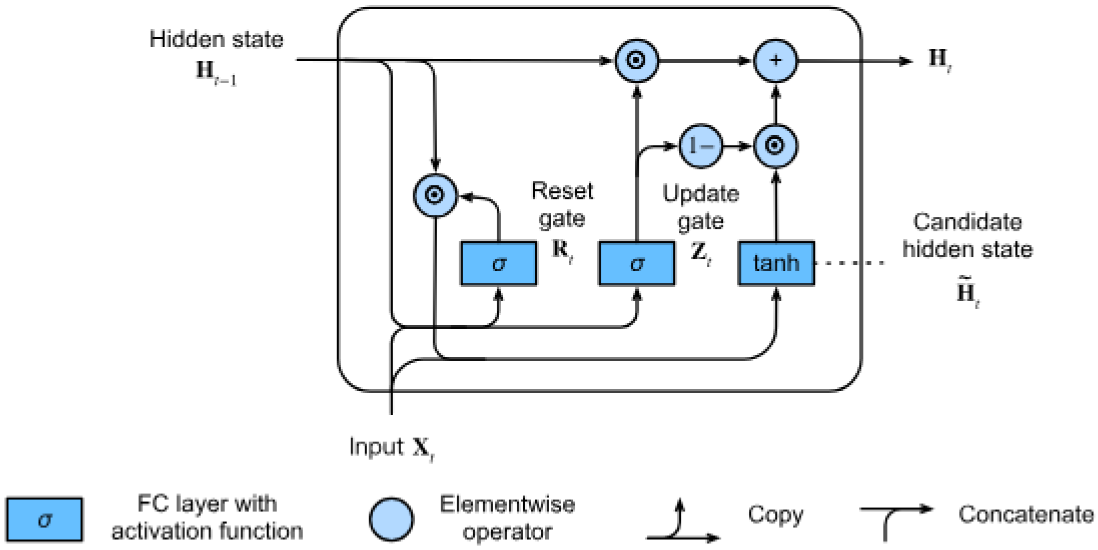

Figure 3.4.5.1: GRU Architecture

3.5 Exploratory Data Analysis:

Our dataset has no null values, i.e. we have all 50000 tweets and the emoji label.
The analysis showcases the distribution of emojis based on their respective labels. The DataFrame presents three columns: 'Emoji label', 'Number of that emoji', and 'Emojis'. It demonstrates the count of each emoji label in the dataset, along with the corresponding emojis. This information provides insights into the frequency and representation of emoji within the analyzed data.

	Emoji label	Number of emoji	Emojis
0	0	10760	❤️
1	1	5279	😍
2	2	5241	😂
3	3	2885	💕
4	4	2517	🔥
5	5	2317	😊
6	6	2049	😎
7	7	1894	✨
8	8	1796	💙
9	9	1671	😘
10	10	1544	📷
11	11	1528	us
12	12	1462	☀️
13	14	1377	😉
14	13	1346	💜
15	16	1306	😁
16	18	1286	📸
17	17	1279	🎄
18	15	1249	💯
19	19	1214	😜

Figure 3.5.1: Number of Emojis

Emoji Distribution:

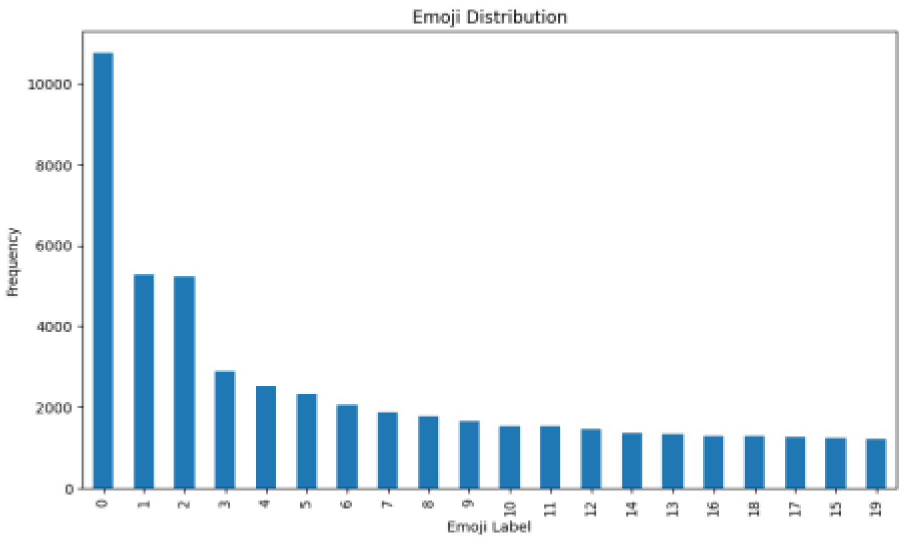

Figure 3.5.2: Emoji Distribution (Bar Plot).

The pie chart titled "Emoji Distribution" illustrates the distribution of emojis based on their occurrence in the dataset. The chart reveals that the red heart emoji, represented by label 0, has the highest occurrence, accounting for approximately 21.5% of the total emojis. On the other hand, the emoji with the label 19, which corresponds to the winking face with tongue, has the lowest occurrence, constituting only 2.4% of the total emojis. The chart provides a visual representation of the varying frequencies of emojis present in the dataset, with the red heart emoji being the most prevalent and the winking face with tongue emoji being the least frequent.

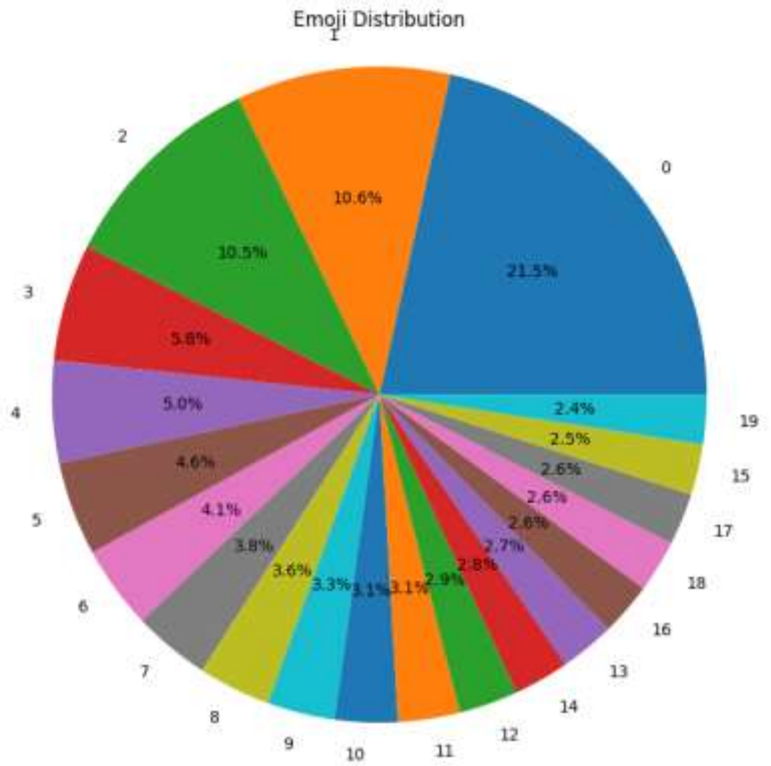

Figure 3.5.3: Emoji Distribution (Pie Chart).

Chapter 4

RESULTS AND DISCUSSION

Model	Accuracy	Precision	F1-Score	Recall
Multinomial NB	28%	29%	19%	28%
SVM	30%	32%	21%	30%
LSTM	88%	87%	87%	88%
Bi-LSTM	88%	87%	87%	88%
GRU	63%	70%	66%	63%

The chart illustrates the performance metrics of different models across four evaluation criteria: Model Accuracy, Precision, F1-Score, and Recall.

The Multinomial Naive Bayes (Multinomial-NB) and Support Vector Machine (SVM) exhibit relatively lower overall performance, with accuracies of 28% and 30%, respectively. The SVM outperforms Multinomial-NB across all metrics, demonstrating a slight improvement in Precision, F1-Score, and Recall.

In contrast, the Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM) models display remarkable results, achieving high accuracy (88%) and consistent Precision, F1-Score, and Recall scores around 87-88%. These recurrent neural network architectures, particularly LSTM and Bi-LSTM, demonstrate superior performance compared to the traditional machine learning models.

The Gated Recurrent Unit (GRU) model falls in between, with a 63% accuracy. While its performance is not as high as the LSTMs, it surpasses Multinomial-NB and SVM in all metrics, showcasing the effectiveness of GRU in capturing sequential dependencies.

4.1 Multinomial Naïve Bayes:

The "Figure 4.1.1" below shows the confusion matrix of Naive Bayes classifier. The first important insight we get from the matrix is that the model is biased towards the first class. The true positives as well as false positives are very high for the first class i.e. the 'red heart' emoji.

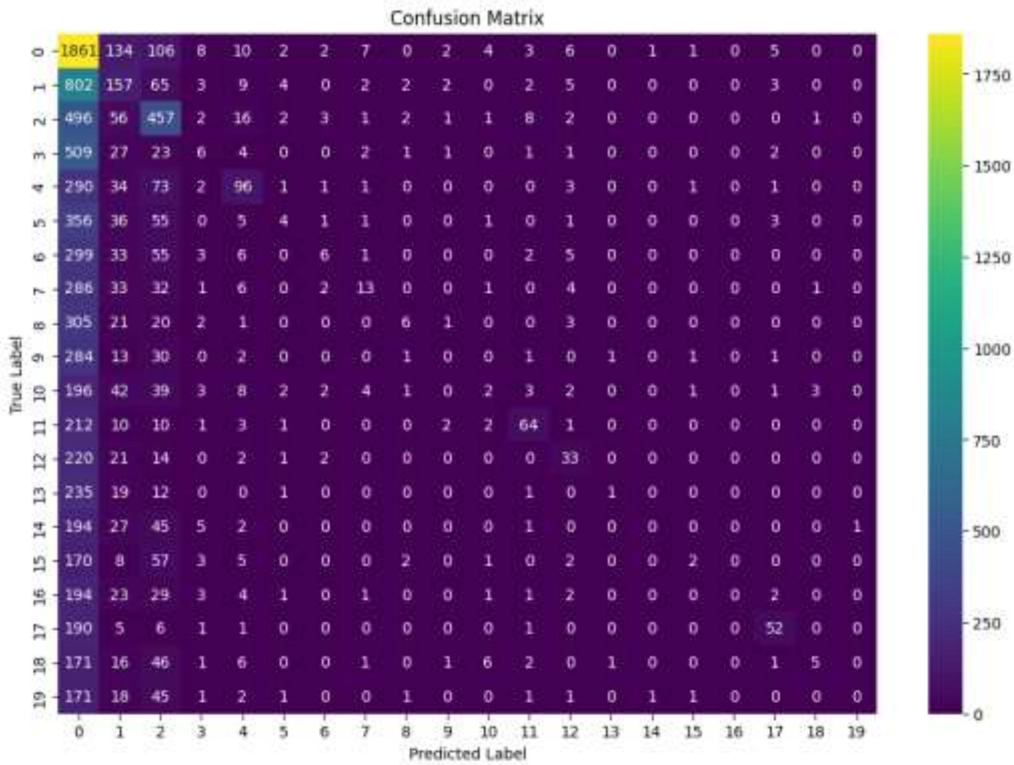

Figure 4.1.1: Naïve Bayes Confusion Matrix

- **Example:**

```
Enter a sentence: This is suppose to be funny
User Input: This is suppose to be funny
Predicted Emoji: 😂
```

```
Enter a sentence: Chase your dreams among the stars, for the sky is an open canvas of endless possibilities.
User Input: Chase your dreams among the stars, for the sky is an open canvas of endless possibilities.
Predicted Emoji: ❤️
```

4.2 Support Vector Machine:

SVM Model performed marginally better than the Naïve Bayes model by over 2% in accuracy and Recall, 10% in Precision and 3% in F1-score! Biasness of the first emoji remains the same as NB.

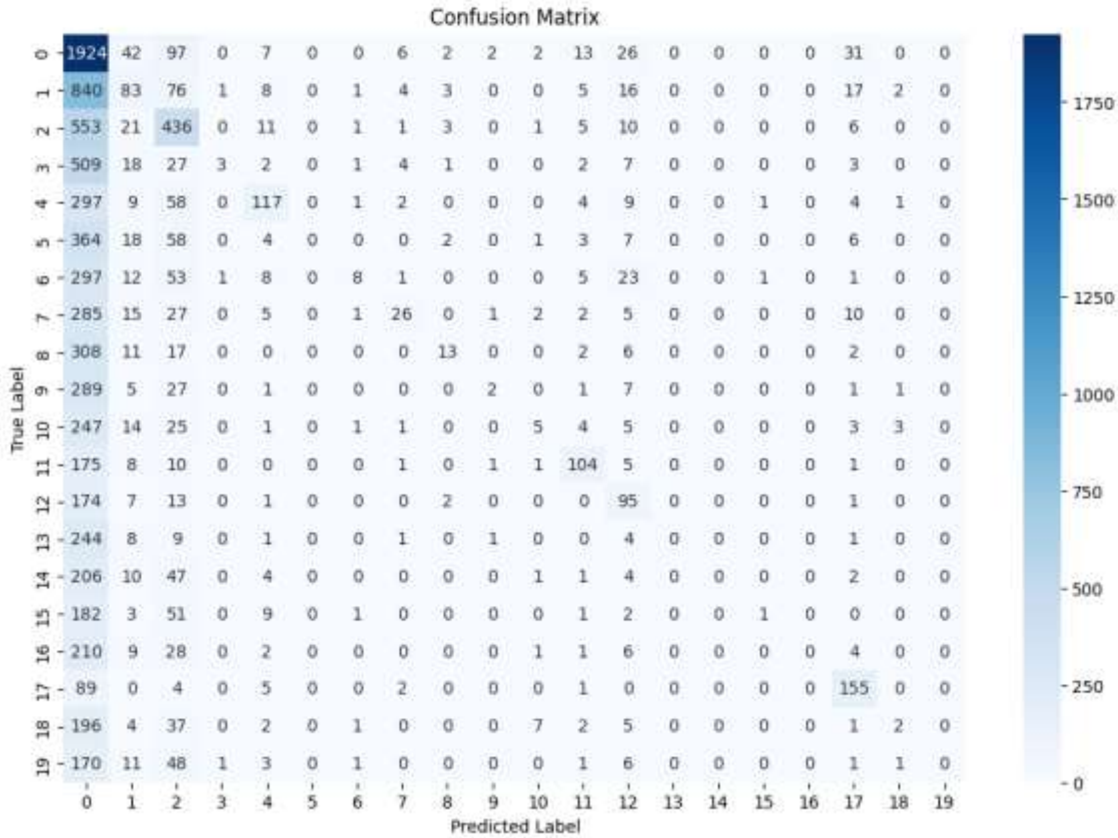

Figure 4.2.1: SVM Confusion Matrix.

- **Example:**

```
Enter a sentence: Enjoying the sunshine and good vibes!
User Input: Enjoying the sunshine and good vibes!
Predicted Emoji: ☀
```

4.3 Long Short Term Memory:

Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM) models display remarkable results, achieving high accuracy (88%) and consistent Precision, F1-Score, and Recall scores.

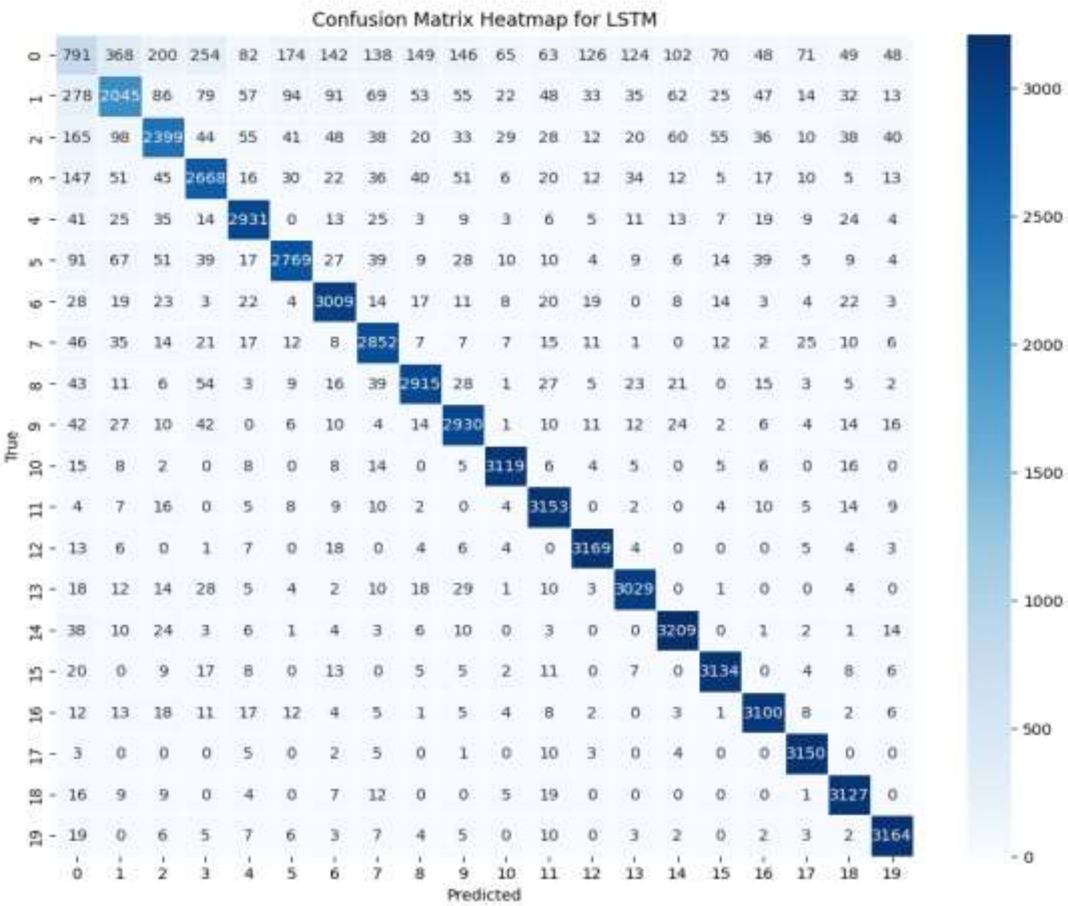

Figure 4.3.1: Long Short Term Memory Confusion Matrix

Table 4.3.1: Model Summary of LSTM

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 40, 20)            1094960

 lstm_2 (LSTM)               (None, 40, 80)            32320

 dropout_2 (Dropout)         (None, 40, 80)            0

 lstm_3 (LSTM)               (None, 40)                19360

 dropout_3 (Dropout)         (None, 40)                0

 dense_1 (Dense)             (None, 20)                820

=================================================================
Total params: 1147460 (4.38 MB)
Trainable params: 1147460 (4.38 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

- **Example:**

```
Enter tweet
Let's explore the North East University

Emojified Tweet

1/1 [==============================] - 0s 33ms/step
Let's explore the North East University 📷
```

```
Enter tweet
Asif sir is our supervisor for this project

Emojified Tweet

1/1 [==============================] - 0s 28ms/step
Asif sir is our supervisor for this project ❤️
```

```
Enter tweet
we want more holidays

Emojified Tweet

1/1 [==============================] - 0s 45ms/step
we want more holidays 😘
```

4.4 Bidirectional Long Short Term Memory:

The “Figure 4.4.1” below shows the confusion matrix of Bi-LSTM.

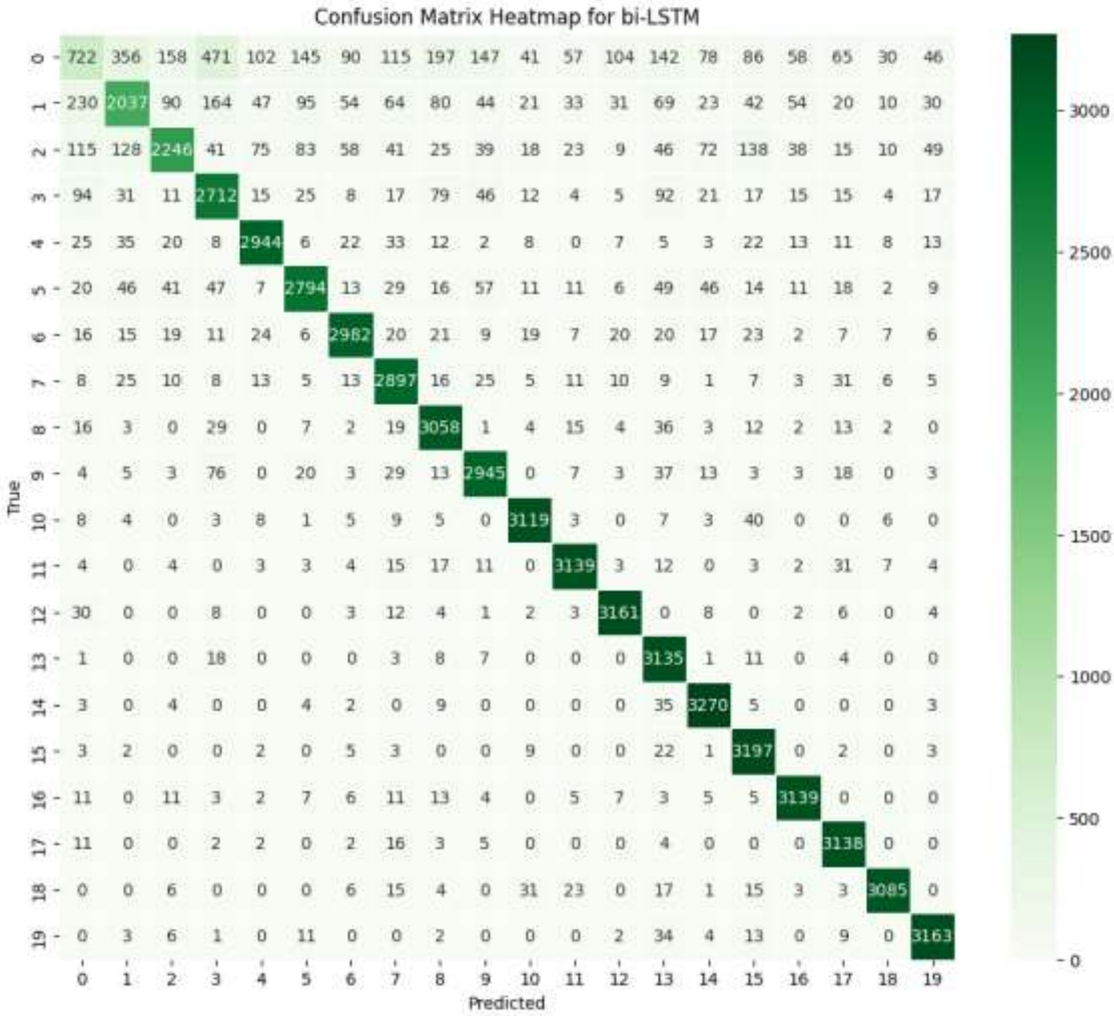

Figure 4.4.1: Bidirectional Long Short Term Memory Confusion Matrix

Table 4.4.1: Model Summary of Bi-LSTM

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 40, 128)           7007744

 bidirectional (Bidirection  (None, 40, 160)           133760
 al)

 bidirectional_1 (Bidirecti  (None, 40, 160)           154240
 onal)

 global_max_pooling1d (Glob  (None, 160)               0
 alMaxPooling1D)

 dropout (Dropout)           (None, 160)               0

 dense (Dense)               (None, 64)                10304

 dropout_1 (Dropout)         (None, 64)                0

 dense_1 (Dense)             (None, 20)                1300

=================================================================
Total params: 7307348 (27.88 MB)
Trainable params: 7307348 (27.88 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

- **Example:**

```
Enter tweet
Winter is coming

Emojified Tweet

1/1 [==============================] - 0s 71ms/step
Winter is coming ❤️
```

4.5 Gated Recurrent Unit:

The Gated Recurrent Unit (GRU) model falls in between, with a 63% accuracy. The "Figure 4.5.1" below shows the confusion matrix of GRU.

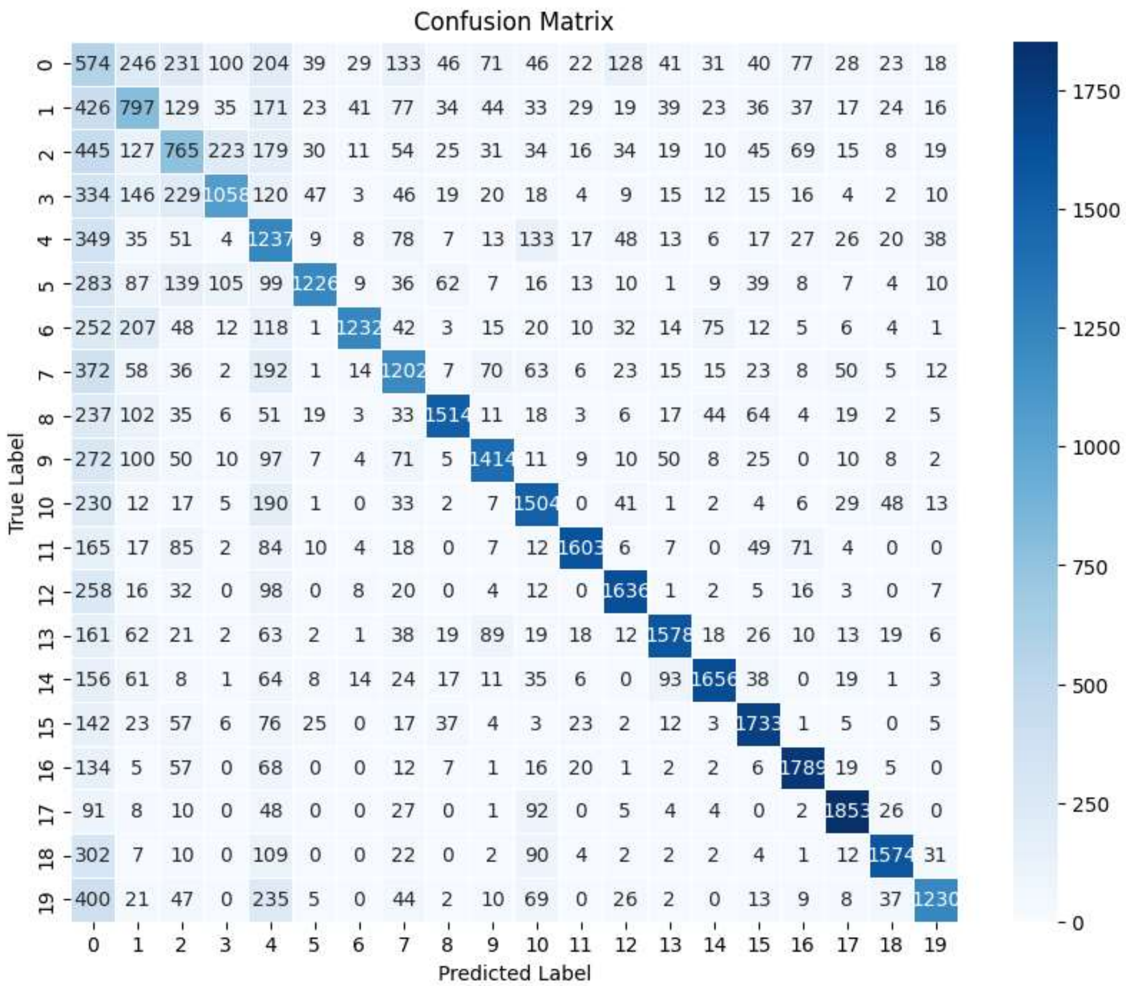

Figure 4.5.1: Gated Recurrent Unit Confusion Matrix

Table 4.5.1: Model Summary of Gated Recurrent Unit

```
model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 40, 50)            4000000

 gru (GRU)                   (None, 64)                22272

 dropout (Dropout)           (None, 64)                0

 dense (Dense)               (None, 20)                1300

=================================================================
Total params: 4023572 (15.35 MB)
Trainable params: 4023572 (15.35 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

- **Example:**

```
1/1 [==============================] - 1s 517ms/step
Input Tweet: Parvez sir seems like the toughest one to deal with
Predicted Emoji: 🔥
```

```
1/1 [==============================] - 1s 1s/step
Input Tweet: Sadat sir went to US for higher studies
Predicted Emoji: ✨
```

Chapter 5
Limitation

The project encounters several limitations primarily related to the dataset. First, the dataset exhibits imbalance, with certain emoji classes having significantly fewer instances than others.

Additionally, the dataset encompasses only a limited set of 20 emojis, potentially restricting the model's generalization to a broader spectrum of emoji expressions.

Lastly, there is an acknowledged imperfection in the mapping between tweets and assigned emojis, introducing ambiguity and noise into the training data. The lack of a perfect correspondence between text content and emoji labels impacts the model's capacity to learn robust associations between textual expressions and appropriate emoji representations.

Chapter 6
CONCLUSION

In conclusion, this project aimed to predict emojis using deep learning techniques on Twitter data. Through comprehensive exploratory data analysis, including emoji distribution, co-occurrence analysis, and sentiment analysis, we gained insights into the usage patterns and relationships between emojis in tweets. The dataset consisted of tweets and their corresponding emoji labels, which were successfully preprocessed using tokenization and vectorization techniques. We trained a Multinomial Naive Bayes classifier, an SVM classifier, LSTM, Bi-LSTM and GRU achieving an accuracy of 28%, 30%, 88%, 88% and 64% respectively. The confusion matrices provided a visual representation of the performance of these models in predicting emoji labels.

References

[1] V N Durga, V N Hemanth, M Durga(2021). "**Emoji Prediction from Twitter Data using Deep Learning Approach".** Asian Conference on Innovation in Technology (ASIANCON). Retrieved from https://ieeexplore.ieee.org/document/9544680

[2] Chen Huang, Xueying (Shirley), Boyu (Bill) (2019). **"Emojify: Emoji Prediction from Sentence. Stanford University".** Retrieved from **https://cs229.stanford.edu/proj2019aut/data/assignment_308832_raw/26647690.pdf**

[3] Pota M, Ventura M, Catelli R, Esposito M. (2021). An Effective BERT-Based Pipeline for Twitter Sentiment Analysis. A Case Study in Italian. Retrieved from https://www.mdpi.com/1424-8220/21/1/133 Pota M, Ventura M, Catelli R, Esposito M. (2021). **"An Effective BERT-Based Pipeline for Twitter Sentiment Analysis"**. A Case Study in Italian. Retrieved from https://www.mdpi.com/1424-8220/21/1/133

[4] Barbieri, Francesco, Luis Espinosa-Anke, and Horacio Saggion (2016). **"Revealing patterns of Twitter emoji usage in Barcelona and Madrid. Artificial Intelligence Research and Development)".** Retrieved from https://repositori.upf.edu/bitstream/handle/10230/30780/espinosa_FAIA288_reve .pdf

[5] Luda Zhao, Connie Zeng (2018). **"Using Neural Networks to Predict Emoji Usage from Twitter Data. Stanford University".** Retrieved from https://www.mdpi.com/1424-8220/21/1/133