

# Tweet Sentiment Analysis

*Natural Language Processing*

2021

MAHESH BHARADWAJ K, 185001089

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Dataset</b>	<b>3</b>
2.1	Dataset Preprocessing . . . . .	5
<b>3</b>	<b>Model Architecture</b>	<b>5</b>
3.1	Transformer . . . . .	5
3.1.1	Model Results . . . . .	6
3.1.2	Graphs . . . . .	6
3.1.3	Model Predictions . . . . .	7
3.2	BERT . . . . .	8
3.2.1	Model Results . . . . .	8
3.2.2	Graphs . . . . .	8
3.2.3	Model Predictions . . . . .	9
<b>4</b>	<b>Conclusion</b>	<b>9</b>

## 1 Introduction

The aim of this project is to classify given tweets into 5 classes based on their sentiment. The classes are:

1. Extremely Negative
2. Negative
3. Neutral
4. Positive
5. Extremely Positive

The features present in the dataset are given below in table 1:

UserName	ScreenName	Location
TweetAt	OriginalTweet	Sentiment

Table 1: Features in the dataset

Of the features listed above, UserName, ScreenName, Location and TweetAt have been ignored because:

1. They are not available for all data points
2. UserName and ScreenName are integers with no meaningful impact on the sentiment and may lead to bias towards certain numbers

## 2 Dataset

The dataset consists of 41,157 training data points with the features mentioned in table 1. The distribution of classes was plotted to check if there is a case for imbalance in the samples for classes and as can be seen in figures 1 and 2, the classes are distributed fairly equally and require no sampling. This dataset is split into train and validation set using 80:20 split.

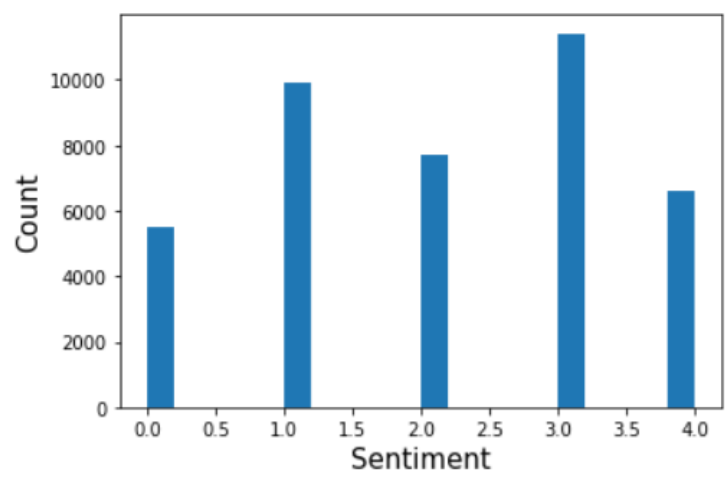


Figure 1: Bar Chart showing class distribution

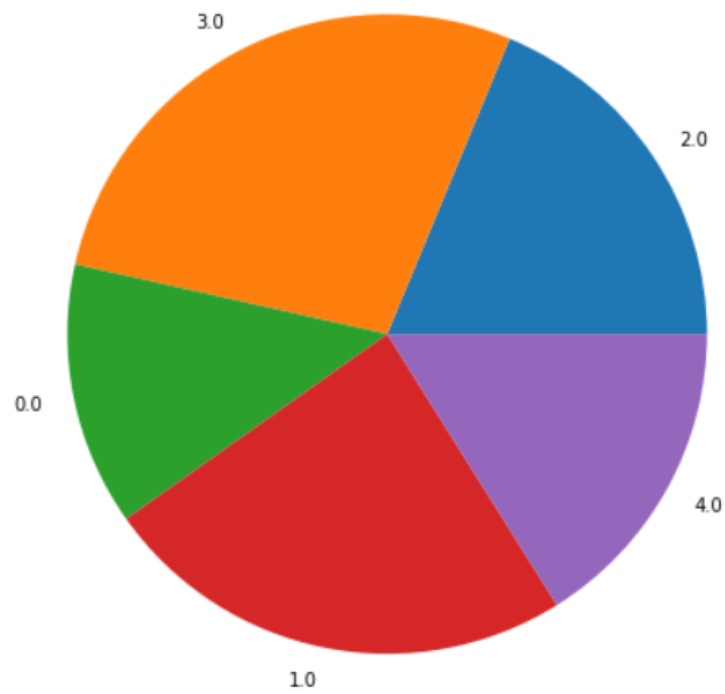


Figure 2: Pie Chart showing class distribution

## 2.1 Dataset Preprocessing

The dataset was pre-processed by performing the following:

- Removed old style re-tweet tags ‘RT’
- Removed hyperlinks to other sites on the internet.
- Stripping the ‘’ alone from hashtags since the hashtags may contain important indicator to the sentiment of the tweet
- All numbers in the tweets are removed
- Stopwords are removed(ie If, so, then).
- Punctuations are removed.
- Handles of other users are replaced by @HANDLE

## 3 Model Architecture

Two models were tested on this dataset to compare their performance:

### 3.1 Transformer

The Transformer in NLP is a novel architecture that aims to solve sequence-to-sequence tasks while handling long-range dependencies with ease. It relies entirely on self-attention to compute representations of its input. In this problem, the encoder half of the transformer alone is used in order to obtain representations for tweets to feed into a fully connected network with softmax activation function to predict the class. Table 2 below shows the hyper-parameters used:

HyperParameter	Value
Epochs	13(Stopped by EarlyStopping)
Learning Rate	$5 \times 10^{-3}$
Optimizer	Adam
Loss	Sparse Categorical CrossEntropy
Attention Heads	2
Embedding Dimension	128
Feed Forward Dimension	1024
Total Parameters	3,043,589

Table 2: Transformer Hyper Parameters

### 3.1.1 Model Results

The model performance on the validation set is given below in figure 3

Classification Report:					
	precision	recall	f1-score	support	
0	0.72	0.75	0.74	1056	
1	0.69	0.70	0.70	2006	
2	0.90	0.71	0.79	1553	
3	0.69	0.72	0.71	2287	
4	0.72	0.82	0.77	1330	
accuracy			0.73	8232	
macro avg	0.75	0.74	0.74	8232	
weighted avg	0.74	0.73	0.73	8232	

Figure 3: Transformer Performance on validation set

### 3.1.2 Graphs

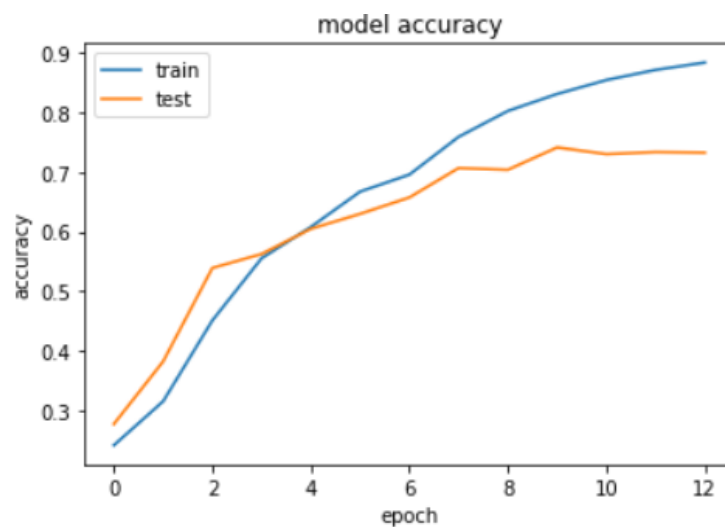


Figure 4: Accuracy vs Epochs

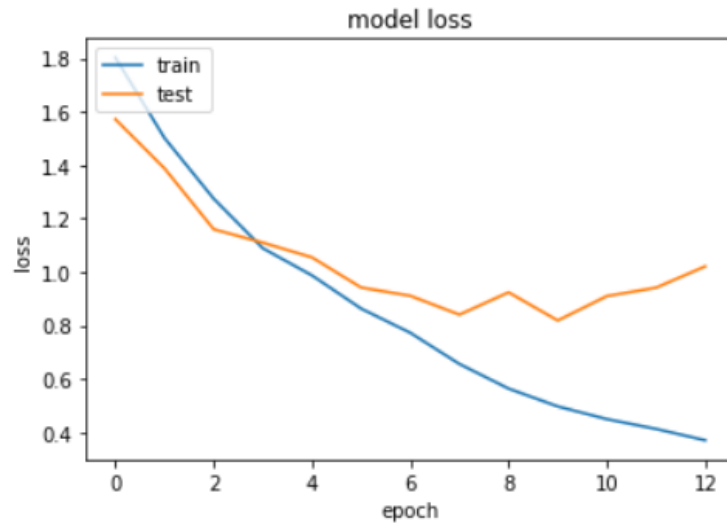


Figure 5: Loss vs Epochs

### 3.1.3 Model Predictions

This model was used to predict the sentiment of 3799 tweets and submitted on kaggle. The scores obtained and leaderboard position at the time of writing this report are given in figure 6.

Your most recent submission				
Name	Submitted	Wait time	Execution time	Score
submission2.csv	just now	1 seconds	0 seconds	0.71202
Complete				
<a href="#">Jump to your position on the leaderboard.</a>				

Figure 6: Performance of Transformer on test data

### 3.2 BERT

BERT is pretrained Bi-Directional Encoder Representations trained by google on unlabeled data extracted from the BooksCorpus with 800M words and English Wikipedia with 2,500M words. The encoded representations are fed to a Fully Connected layer with 5 nodes having softmax activation function to get probabilities. Huggingface Transformer package is used to tokenize, convert data to required format and also for the BERT model. Model hyperparameters are given below in table 3.

HyperParameter	Value
Epochs	5
Learning Rate	$3 \times 10^{-5}$
Optimizer	Adam
Loss	Sparse Categorical CrossEntropy
Encoders	12(BERT-BASE)
Attention Heads	12(BERT-BASE)
Parameters	<b>109,486,085</b>

Table 3: BERT Hyper Parameters

#### 3.2.1 Model Results

The model performance on the validation set is given below in figure 7

classification_report:					
	precision	recall	f1-score	support	
0	0.86	0.87	0.87	1056	
1	0.83	0.81	0.82	2006	
2	0.80	0.89	0.84	1553	
3	0.84	0.83	0.83	2287	
4	0.91	0.85	0.88	1330	
accuracy			0.84	8232	
macro avg	0.85	0.85	0.85	8232	
weighted avg	0.84	0.84	0.84	8232	

Figure 7: BERT Performance on validation set

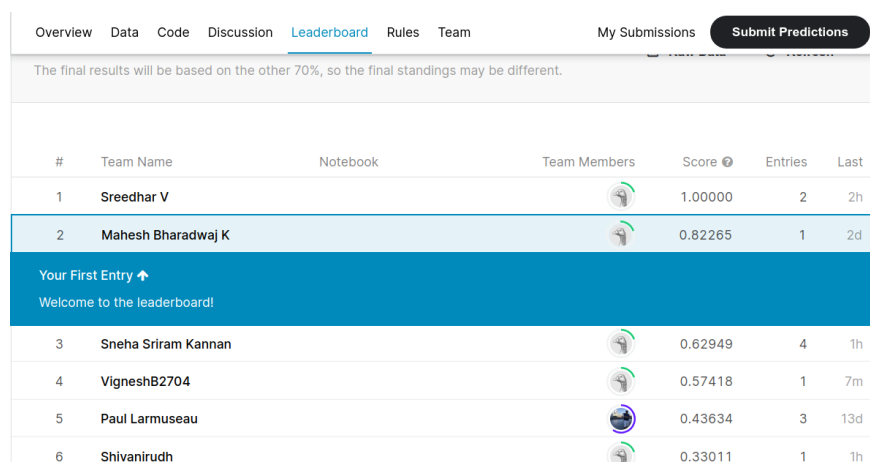
#### 3.2.2 Graphs

Unfortunately, only model weights were saved in drive and the history data of the training across 5 epochs was lost at the time of writing the report and training takes around 4 hours for BERT hence images are not attached.



### 3.2.3 Model Predictions

This model was used to predict the sentiment of 3799 tweets and submitted on kaggle. The scores obtained and leaderboard position at the time of writing this report are given in figure 8.



#	Team Name	Notebook	Team Members	Score	Entries	Last
1	Sreedhar V			1.00000	2	2h
2	Mahesh Bharadwaj K			0.82265	1	2d
Your First Entry Welcome to the leaderboard!						
3	Sneha Sriram Kannan			0.62949	4	1h
4	VigneshB2704			0.57418	1	7m
5	Paul Larmuseau			0.43634	3	13d
6	Shivanirudh			0.33011	1	1h

Figure 8: Performance of BERT on test data

## 4 Conclusion

The best results were obtained using BERT model however it used 33x the parameters( 109M) used by vanilla transformer(3M) which, with further tweaking of hyper paramters can be made to perform at the level of the BERT given time.