International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7186
***Article Received**: 20 April 2023 **Revised:** 14 June 2023 **Accepted:** 26 June 2023*

Leveraging Sentiment Analysis for Twitter Data to Uncover User Opinions and Emotions

Tushar Sharma[1], Dr. Priyanka Kaushik[2]
[1]Computer Science and Engineering Department
Chandigarh University Mohali, tushar1510sharma@gmail.com
[2]Professor Computer Science Engineering Dept (AIT CSE (AIML)
Chandigarh University
Kaushik.priyanka17@gmail.com

Abstract— Huge amounts of emotion are expressed on social media in the form of tweets, blogs, and updates to posts, statuses, etc. Twitter, one of the most well-known microblogging platforms, is used in this essay. Twitter is a social networking site that enables users to post status updates and other brief messages with a maximum character count of 280. Twitter sentiment analysis is the application of sentiment analysis to Twitter data (tweets) in order to derive user sentiments and opinions. Due to the extensive usage, we intend to reflect the mood of the general people by examining the thoughts conveyed in the tweets. Numerous applications require the analysis of public opinion, including businesses attempting to gauge the market response to their products, the prediction of political outcomes, and the analysis of socioeconomic phenomena like stock exchange. Sentiment classification attempts to estimate the sentiment polarity of user updates automatically. So, in order to categorize a tweet as good or negative, we need a model that can accurately discern sarcasm from the lexical meaning of the text. The main objective is to create a practical classifier that can accurately classify the sentiment of twitter streams relating to GST and Tax. Python is used to carry out the suggested algorithm.

Keywords— Sentiment Analysis, Natural Language Toolkit, Python, Deep Learning, Bidirectional LSTM, Twitter, Social Media.

I. INTRODUCTION

As they enable users to share and express their thoughts about specific issues and disseminate their views throughout the globe in an organised way, social media platforms like Twitter and Facebook have grown significantly in relevance for many readers today. The analysis of tweet sentiment falls within the categories of "Pattern Classification" and "Data Mining". Both phrases refer to the process of identifying important patterns in a big body of data, whether through labelled or unlabelled data.

This project is completed using the techniques like “Natural Language Processing”, “text analysis” which are helpful in extracting the significant patterns and features from the large cleaned data extracted from the twitter and then using the pre-processed data to feed in the “Deep Neural Network” for accurate classification of the tweet sentiments.

Both computational learning techniques and semantic approaches can be used to solve this challenge. The usage of word dictionaries (lexicons) with the semantic orientation of polarity or opinion characterises semantic methods. Systems typically pre-process the text, break it up into words, remove any unnecessary stop words, normalise the language by stemming or lemmatizing, and then check to see if each lexical term is present or not. The global polarity value of the text is then determined by adding the polarity values of all the terms. The ideal threshold can then be established in order to correctly classify the sentiment as positive or negative.

The learning-based approaches also involve training a classifier with any supervised learning algorithm from a set of annotated texts, where each text is typically represented by a vector of words (bag of words), n-grams, or skip-grams,in addition to other types of semantic features that aim to simulate the syntactic structure of sentences, intensification, negation, subjectivity, or irony. The most prevalent classifiers used by systems are those based on SVM (Support Vector Machines), Naive Bayes, and KNN (K-Nearest Neighbour). Recent studies have used more sophisticated techniques, including LSA (Latent Semantic Analysis), wordembeddings (such as Word2Vec, Fast Text, GloVe, etc.), and Deep Learning.

The benefit of learning-based approaches is that a sentiment/opinion analysis engine can be quickly and easily built and taught using a collection of tagged texts. As a result, creating classifiers customised for a particular domainis not too difficult. Contrarily, because building a vocabularyfrom scratch for a certain domain requires a lot of laborious manual work, these systems are typically less versatile.

Learning-based systems appear to be superior since they are better able to adapt to the suggested domain, but generally speaking, full retraining is required to migrate them

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7186
***Article Received**: 20 April 2023 **Revised:** 14 June 2023 **Accepted:** 26 June 2023*

to another domain. They perform well in "simple" cases because the complexity of natural language is not elaborately treated, but they are unable to handle coordination and subordination, comparisons, sentences with a mix of polarities and multiple inversion terms, and other complex sentences.

Following the classification of the tweets, we must now contrast the model's predictions with expert judgement. For calculating and comparing the outcomes of our studies, a variety of metrics have been presented. Precision, Recall, Accuracy, F1-measure, True rate, and False alarm rate are some of the most well-liked measures (each of these metrics is calculated separately for each class and then averaged for the entire classifier).

Below is a sample confusion table for our issue along with an example of how to calculate the necessary metric.

Table 1: 2x2 confusion matrix

	Model says Negative	Model says Positive
Human says Negative	True Negative	False Positive
Human says Positive	False Negative	True Positive

Despite the incredible work being done in the area of sentiment words and phrases, it is still far from enough to do an accurate sentiment analysis. The issue is considerably more intricate. We draw attention to the following issues:

1. In certain application areas or phrase situations, a term with a positive or negative feeling may have opposite orientations or polarities. If a sentiment or opinion is positive, negative, or neutral, it is what we mean by its orientation or polarity. For instance, the word awful can be used to convey both positive and negative feelings (for instance, "This hoover is terribly good" and "This camera is terrible"). As a result, we assert that the orientations of sentiment words may vary by domain or even by sentence context.

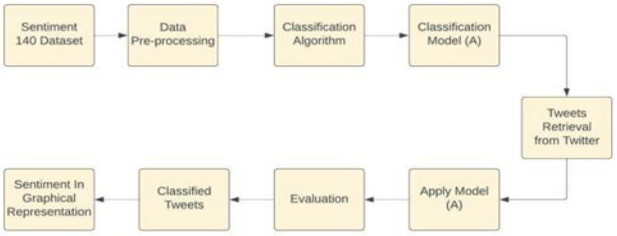
Content

It contains the following 6 fields:

i. target: the polarity of the tweet (0 = negative, 4 = positive)
ii. ids: The id of the tweet (ex:-2087)
iii. date: the date of the tweet (ex:-Sat May 1623:58:44 UTC 2009)
iv. flag: The query (ex:-lyx). If there is no query, then this value is NO_QUERY.
v. user: the user that tweeted (ex:-robotickilldozr)
vi. text: the text of the tweet (ex:-Lyx is cool)

II. METHODOLOGY

This project elaborates the entire process of designing, implementing, training, and evaluating the main performance of our model. The project is divided into 2 phases. First, we trained our model using the well-known dataset Sentiment140. In phase 2, new tweets are fetched from the Twitter and then applying the pre-trained model from phase1, to the newly fetched tweets.



III. IMPLEMENTATION

The whole process of the Twitter sentiment analysis project can be divided into some basic categories.

1.1. Phase 1:

A. Data Acquisition

The process of locating an appropriate dataset that can truly increase the performance and accuracy of our model. Sentiment140, a dataset with 1.6 million tweets from Kaggle, is what we used to train our model. 1,600,000 tweets that were retrieved using the Twitter API are included. The tweets can be used to gauge sentiment because they have been annotated (0 = negatively inclined, 4 = positively inclined).

B. Text Pre-Processing

After looking at our dataset, it is evident from the above section that there are many special characters and extraneous pieces of data in the text area of the data, which is what we need to handle. Therefore, pre-processing this data is crucial so that we can continue without analysis.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7186
***Article Received**: 20 April 2023 **Revised:** 14 June 2023 **Accepted:** 26 June 2023*

- **Regular Expression**

For text manipulation during the text cleaning step of natural language processing, regular expression is highly helpful. Emojis, misspelt words, short words, special symbols, etc. are all present in the actual human-written text data. A tweet must be cleaned up of uselessinformation.

i. For removing URL's, @mentions and hashtags.

ii. Regular expression for removing emojis.

- **Stop-words removal**

It can be very handy to delete these words because they are a list of terms that are incredibly prevalent in a language but don't actually contain any information and are therefore useless to the model.

All stop words other than "not" have been eliminated from this project because doing so will allow our model to better understand the pattern and forecast mood.

- **Lowercase Conversion**

It is possible to normalise a tweet by changing its case to lowercase, which facilitates comparison with an English dictionary.

- **Stemming**

Reducing a derivative term to its root or stem is a text normalisation technique. For instance, a stemmer would break down the words "playing," "plays," and "played" to their basic "play" form.

- **Tokenization**

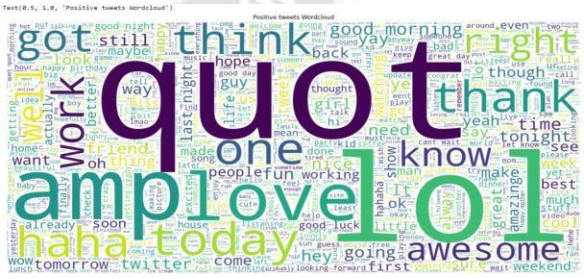
It is the process of dividing a continuous stream of text into words, symbols, and other "tokens" that have meaning. Whitespace and/or punctuation characters can be used to separate tokens. This allows us to view tokens as the discrete parts that make up a tweet. This allows us todetermine the number of words in the corpus that are unique (vocabulary=290576) as well as their specific index(word-index=290575), which we can then use to transform the words into a matrix of integers.

- **Pad sequencing**

We now have each row with cleaned and tokenized data following tokenization. The issue now is that they are all of different length, therefore we must pad each row with a fixed length (in this case, we chose 30). If a sentence is longer than that, certain words may be omitted.

We'll move on to the list of features we looked at after discussing some of the text formatting strategies we use. A feature is any variable that can aid our classifier in discriminating between the various classes, as we will see below. In our approach, there are two differentclassifications: the positivity/negativity classification and the objectivity/subjectivity classification (both of which will be covered in more detail in the following section). The former is used to distinguish between objective and subjective classes, while the latter is used to distinguish between positive and negative classes, as the name implies. After, doing all the steps to clean and frame our data into numbers we can finally visualize our text data . For this I have used the library Word Cloud for presenting the cloud of thepositive and negative tweets on the basis of the frequently occurring words.

i. Positive word cloud



ii. Negative word cloud

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7186
***Article Received**: 20 April 2023 **Revised:** 14 June 2023 **Accepted:** 26 June 2023*

C. Feature Extraction

The idea of feature extraction is to transform unprocessed data into the inputs that a specific machine learning algorithm needs. These extrapolated characteristicsfrom the raw data are truly pertinent to solving the underlying issue. Word embeddings, on the other hand, are essentially dispersed representations of text in an n- dimensional space. The sequence of each word in the matrix is used to convert our pad sequenced data, which aremerely numbers. Giving our future deep learning model merely these numbers makes no sense at this time and will produce awful results because the matrix does not contain any information about the semantic meaning of each word in the corpus.

Therefore, the word embedding layer will be our first layer and our deep learning network's input layer.

I. Word Embedding

In essence, word embeddings are a type of word representation that connects a computer's comprehension of language to that of a human. They have mastered text representations in an n-dimensional space, where words with the same meaning are represented similarly. translates to two comparable words being represented by very closely spaced, practically identical vectors.

As a result, each individual word is represented by a real-valued vector in a predetermined vector space when utilising word embeddings. The values of the vectors are learned in a manner resembling a neural network, and each word is mapped to a single vector. We will do this by using GloVe, a pre-built unsupervised learning technique, to learn word embeddings for each individual word in our corpus.

- **GloVe**

Another approach to producing word embeddings is called GloVe (Global Vectors for Word Representation). It is based on word-context matrix factorization algorithms. Asa result, I decided to use the non-trainable GloVe algorithm in this project as our embedding layer. We have an embedding index, which is a dictionary with 4 lakh distinct words as keys and a 300-digit value for each one. From the dictionary's embedding index, I produced a matrix with the dimensions (290576x300). Only 300-dimension vectors thatare part of the vocabulary are included in this matrix. The embedding layer will employ this embedding matrix as well as other parameters like embedding dimension, vocab_size, etc. as the weight matrix.

Cost function of the GloVe model: - This equation is the result of the Global Vectors for word representation Paper.

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2 \quad [1]$$

D. Model Building

To build our neural network model, I used theKeras module from TensorFlow. One of the most well- liked Python libraries for deep learning is Keras. You can use a variety of tools from this library to interact with neural network models. These tools let you see and comprehend your model.[10]

Keras is a straightforward and approachable library that emphasises the concept of models. Learn more about Keras Models by reading on. The neural network model is represented by these. These models classify items into layers. The Sequential model and the Functional model are the two types of models that are offered by Keras.[11]

Since text data is the focus of our research, using a sequential approach as opposed to a functional one is more effective.

i. Sequential Model

Sequence models are machine learning models thatinput or output data sequences. Sequential dataincludes text streams, audio and video snippets, time- series data, and other categories. The invention of Sequence Models was driven by the study of discrete sequential data, including time-series, text phrases, andother sequential data.

- **RNN**

It is a Deep Learning and Artificial Neural Networkdesign that is appropriate for processing sequential input. Its name, RNN, stands for Recurrent Neural Network. RNNs are frequently employed in NaturalLanguage Processing (NLP). RNNs are particularly helpful for machine learning applications that need sequential input since they contain internal memory. RNNs can also be used to forecast time series data.[12]

However, the issue with these conventional RNN isthat they are unable to capture long-distancedependencies. This issue is often referred to asvanishing gradients. While training very deep networks,gradients or derivatives are exponentially worse as theygo deeper into the network.

- **LSTM**

The LSTM was created to address the vanishing gradient issue in RNN; in fact, the name of the algorithm comes from this issue. With LSTM, the RNN hidden layer is altered. Thanks to LSTM, RNNs can retain their inputs for a long period. In LSTM, in addition to the concealed state, a cell state is also carriedover to the subsequent time step. At its core, LSTM uses the hidden state to preserve data from inputs that

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7186
***Article Received**: 20 April 2023 **Revised:** 14 June 2023 **Accepted:** 26 June 2023*

have already been processed.[13]

Due to the fact that it has only ever received inputs from the past, a unidirectional LSTM can only preserveinformation from the past. We must utilise BidirectionalLSTM because this will produce fewer results that are actually useful.[14]

- **Bidirectional LSTM**

When using bidirectional, your inputs will be processed in two different directions: one from the present to the future and the other from the future to thepresent. This method differs from unidirectional in that information from the future is preserved in the LSTM that runs backwards, and by combining the two hidden states, you can preserve data from both the present and the future at any given time.

The input sequence is provided as-is to the first layer as input and a reversed copy of the input sequenceto the second layer after duplicating the first recurrent layer in the network to create two layers side by side.

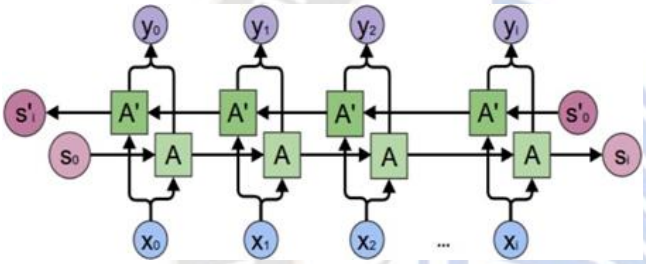
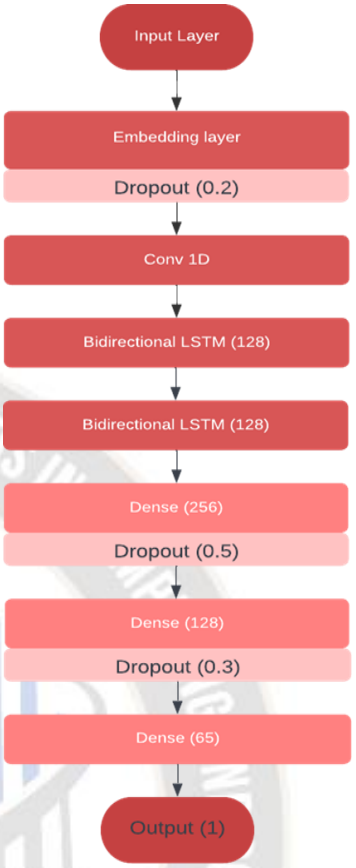


Fig: A Bidirectional long short-term memory with one hidden layer. The forward LSTM cells unfolded in time are denoted by A, while the backward LSTM cells unfolded in time are denoted by A'. [2]

ii. Model Architecture

Our project's architecture is divided into four primary components. We begin with the previously mentioned embedding layer, which takes the input sequences and outputs word embeddings. The convolution layer receives these embeddings and transforms them into tiny feature vectors. The bidirectional LSTM layer comes next. We have two such bidirectional layers and a few Dense (completely linked layers) after the LSTM layers for classification.[15]

- **Our Best Model Architecture:- [3]**



iii. Model Training and Results

The model's architecture has been finished. Let's proceed to using the dataset to train the model. Adam will be our chosen optimizer. We can use the Binary Cross-Entropy loss function since the classification problem is binary (positive or negative sentiment). I have experimented with training using different epochs in this project. However, 30 is the optimal number of epochs, which allowed me achieve a training accuracy of 80.40%.

- **Hyperparameter Tuning**

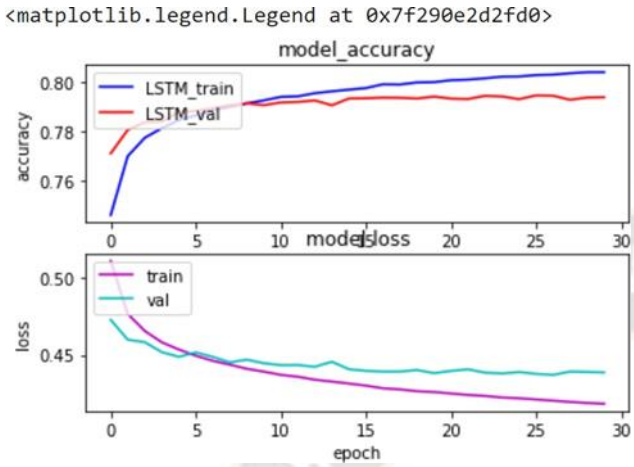
Goal: To train the best model with higher accuracy.	Number of tasks completed: 8 MODEL trained successfully	Optimizer:- Adam	Loss :- Binary cross entropy loss

Model	Epochs	Bidirectional LSTM	Dense layer	Trian accuracy
1	10	1 layer (64)	2 layers (512, 512)	78.29%
2	15	1 layer (64)	3 layers(512, 256, 256)	78.67%
3	15	1 layer (128)	3 layers(512, 256, 256)	78.5%
4	`15	2 fully connected (64)	3 layers(256, 128, 64)	78.96%
5	20	2 fully connected (128)	3 layers(512, 256, 256)	79.31%
6	20	2 fully connected (128)	3 layers(256, 128, 64)	79.32%
7	20	2 fully connected (64)	3 layers(256, 128, 64)	79.01%
8	30	2 fully connected (128)	3 layers(256, 128, 64)	80.40%

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7186
***Article Received**: 20 April 2023 **Revised:** 14 June 2023 **Accepted:** 26 June 2023*

iv. Evaluation Of Model

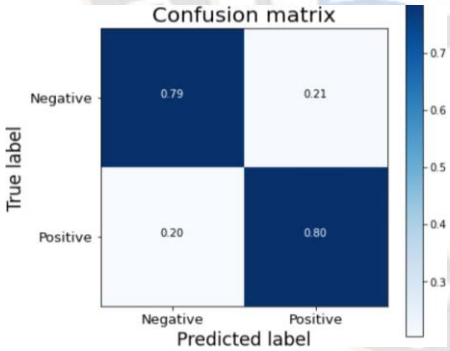
I made a graph showing the 30 epochs against the training and validation accuracy. As shown in the graphic below, the validation accuracy is approximately 0.80.



Source:- Console of the Google Colab.

We can now use the trained model to generate predictions. This is ultimately a case of binary classification. Therefore, selecting a threshold value to categorise the data samples would be helpful. I've set a cut-off of 0.5, so a tweet is considered positive if the prediction is higher than 0.5. It is categorised as negative if not.[16]

The confusion matrix for our model prediction is as follows:-



Source:- Console of the Google Colab.

1.2. Phase 2:

E. Fetching Data From Twitter

Due to social media's widespread use, many social media platforms are becoming more and more well-liked asa source of data. Data collecting utilizing APIs is evolving into a highly sought-after expertise in many data science positions as a result of the emergence of social media as a data source. To gather social media posts from themicroblogging and social networking site Twitter, we used the Twitter API v2 in our project.[17]

i. Making a Basic Request with the Twitter API

There is just one thing left to do after organizing the API access keys: test the API! Your credentials need to be loaded first in this process. I kept them in a CSV file and used them.

ii. Steps to fetch data from the Twitter:-

1. Created a CSV file with the API credentials oftwitter developer app.
2. Loaded the CSV file in the google colab (using files.upload()).
3. Created variables for each column name.
   - Creating the authentication object.
   - Then set the access token and access tokensecret to the authentication object.
   - Create the API object and pass theauthentication info created above.
4. Now we will extract the tweets with the help of the user credentials and store them in a data frame.
5. We will save these raw tweets in an excel file so that we can do human labelling.[18]

All the tweets that we fetched, were related to the GST and TAX. These two words were used as the search filter. We finally got 705 tweets from the twitter so that we can test our pre-trained model that we created using the labelled dataset named sentiment140.

F. Human Labelling

For the purpose of human labelling, I have stored all the raw tweets in an excel file. I have labelled all the tweets in two classes according to the sentiments expressed in the tweets: positive, negative.

- **Positive:** If the entire tweet has a positive/happy/excited/joyful attitude or if somethingis mentioned with positive connotations. Also, if more than one sentiment is expressed in the tweet but the positive sentiment is more dominant.
- **Negative:** If the entire tweet has a negative/sad/displeased attitude or if something is mentioned with negative connotations. Also if more than one sentiment is expressed in the tweet but the negative sentiment is more dominant.

We arrived at the following statistics for each class after going through majority voting.

- Positive: 222 tweets
- Negative: 483 tweets

G. Cleaning Of the Fetched Tweets

All of the tweets that were downloaded from Twitter using

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7186
***Article Received**: 20 April 2023 **Revised:** 14 June 2023 **Accepted:** 26 June 2023*

the Twitter API are extremely soiled because they also include emojis, URLs, #tags, and @mentions, which are useless for solving the classification issue. Due to the fact that we are just interested in analyzing the tweets' text.

The user name, user location, user verified, date, content, hashtags, and source are just a few of the columns found in the raw tweets. But not all of them are necessary for us to analyze the tweets. I have therefore dropped them. We will simply retain the text column.

H. Applying Model On New Data

The model will then be given the pre-processed data in the shape (705x30) to forecast the results. The data frame (final dataset) was then updated with the model predicted column. The new shape of the data frame is (705x2).

Our model predicted the sentiment into two classes :-

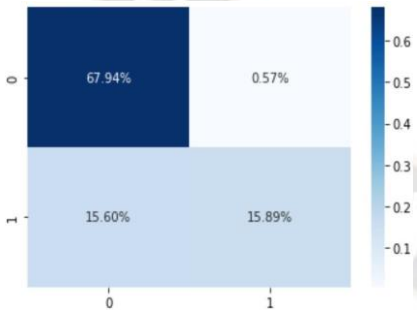
- Positive :- 116
- Negative :- 589

I. Human Evaluation

To assess the performance of our model, we will use the confusion matrix. Below is the confusion matrix we obtained by comparing our computer-predicted outcomes with the human-labelled findings:-

```
[[479   4]
 [110 112]]
```

To show it in a fancy way, we can make use of the heatmap from the seaborn library. We get the matrix as :-



Now, lets make some helpful conclusions from the visualization.

- Accuracy = 83.83%
- Precision = 96.55%
- Recall = 50.45%

Finally, we conclude that our classification approachprovides improvement in accuracy by using even the simplest features . However, there are still a number of things we would like to consider as future work which we mention in the next section.

IV. CONCLUSION

The task of sentiment analysis, particularly in the context of microblogging, was characterised as being in an advanced stage and far from being finished. In order to further improve the performance of GST, we have suggested a few approaches that we believe are worth pursuing in the sentiment prediction for tweets relating just to GST.

I concentrated on the issue of sentiment analysis in tweets in my essay. We used machine learning and deep learning models after data processing and embedding in an effort to identify the most effective model for our issue. First, we selected multi-layer perceptron models as our foundational models. Both of these models had respectable accuracy. We also experimented with RNN and LSTM, which are well-liked approaches to text-related issues. Gradient vanishing became a very serious issue as a result of RNNs back-propagation across time.

In order to resolve the gradient vanishing issue, we employed LSTM. First, we tested a straightforward one- direction LSTM, which we trained in about an hour and hadan accuracy of 78.5%. Once we converted our model to a bidirectional LSTM, we were able to obtain 83.83% accuracy while training it for more than 3 hours.

We want to simulate how people feel confident in our system in our upcoming work. A tweet can be plotted on the 2-dimensional objectivity /subjectivity and positivity/negativity planes using five human labellers, for instance, to distinguish between tweets where all five labels agree, only four agree, only three agree, or when no majority vote is attained.

For creating optimised class boundaries, we might create a custom cost function that gives the maximum weight to tweets with agreement from all five labels and decreases in weight as the number of agreements increases. In this approach, sentiment analysis can illustrate the effects of human confidence.Additionally, we can introduce GRU, which can increasetraining pace and provide us with greater accuracy at a fasterrate.

REFERENCES

[1] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. 2014.
[2] http://colah.github.io/posts/2015-09-NN-Types-FP/
[3] https://lucid.app/lucidchart/593a533b-fb37-45ff-86ff-6c7a6ef7e10d/edit?beaconFlowId=762A83F8A3E0FD29&invitationId=in v_37d72f1f-6f56-43ae-897a-e5b269eec2e0&page=0_0#?folder_id=recent

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7186
***Article Received**: 20 April 2023 **Revised:** 14 June 2023 **Accepted:** 26 June 2023*

[4] Theresa Wilson, Janyce Wiebe and Paul Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In the Annual Meeting of Association of Computational Linguistics: Human Language Technologies (ACL-HLT), 2005.

[5] Alec Go, Richa Bhayani and Lei Huang. Twitter Sentiment Classification using Distant Supervision. Project Technical Report, Stanford University, 2009.

[6] Efthymios Kouloumpis, Theresa Wilson and Johanna Moore. Twitter Sentiment Analysis: The Good the Bad and the OMG! In Proceedings of AAAI Conference on Weblogs and Social Media (ICWSM), 2011.

[7] https://www.mygreatlearning.com/blog/word-embedding/

[8] Shivadekar, S. ., Kataria, B. ., Hundekari, S. ., Kirti Wanjale, Balpande, V. P., & Suryawanshi, R. . (2023). Deep Learning Based Image Classification of Lungs Radiography for Detecting COVID-19 using a Deep CNN and ResNet 50. International Journal of Intelligent Systems and Applications in Engineering, 11(1s), 241–250. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2499

[9] Mäntylä, Mika V., Daniel Graziotin, and Miikka Kuutila. "The evolution of sentiment analysis—A review of research topics, venues, and top cited papers." Computer Science Review 27 (2018): 16-32.

[10] Zhang, Lei, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, and Bing Liu. "Combining lexicon-based and learning-based methods for Twitter sentiment analysis." HP Laboratories, Technical Report HPL- 2011 89 (2011): 1-8.

[11] Rathore, R. (2022). A Study on Application of Stochastic Queuing Models for Control of Congestion and Crowding. International Journal for Global Academic & Scientific Research, 1(1), 1–6. https://doi.org/10.55938/ijgasr.v1i1.6

[12] Ana Oliveira, Yosef Ben-David, Susan Smit, Elena Popova, Milica Milić. Improving Decision Quality through Machine Learning Techniques. Kuwait Journal of Machine Learning, 2(3). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/202

[13] Sharma, V. (2022). A Study on Data Scaling Methods for Machine Learning. International Journal for Global Academic & Scientific Research, 1(1), 23–33. https://doi.org/10.55938/ijgasr.v1i1.4

[14] Rathore, R. (2022). A Review on Study of application of queueing models in Hospital sector. International Journal for Global Academic & Scientific Research, 1(2), 1–6. https://doi.org/10.55938/ijgasr.v1i2.11

[15] Kaushik, P. (2022). Role and Application of Artificial Intelligence in Business Analytics: A Critical Evaluation. International Journal for Global Academic & Scientific Research, 1(3), 01–11. https://doi.org/10.55938/ijgasr.v1i3.15

[16] Kaushik P., Deep Learning and Machine Learning to Diagnose Melanoma; International Journal of Research in Science and Technology, Jan-Mar 2023, Vol 13, Issue 1, 58-72, DOI: http://doi.org/10.37648/ijrst.v13i01.008

[17] Kaushik P., Enhanced Cloud Car Parking System Using ML and Advanced Neural Network; International Journal of Research in Science and Technology, Jan-Mar 2023, Vol 13, Issue 1, 73-86, DOI: http://doi.org/10.37648/ijrst.v13i01.009

[18] Kaushik, P. (2023). Artificial Intelligence Accelerated Transformation in The Healthcare Industry. Amity Journal of Professional Practices, 3(01). https://doi.org/10.55054/ajpp.v3i01.630

[19] Kaushik, P. (2023). Congestion Articulation Control Using Machine Learning Technique. Amity Journal of Professional Practices, 3(01). https://doi.org/10.55054/ajpp.v3i01.631

[20] Rathore, R. (2023). A Study Of Bed Occupancy Management In The Healthcare System Using The M/M/C Queue And Probability. International Journal for Global Academic & Scientific Research, 2(1), 01–09. https://doi.org/10.55938/ijgasr.v2i1.36