DEPRESSION DETECTION USING TWITTER TWEETS

by

Aruna Arumugam 2015103005

Shri Shruthi S 2015103022

A project report Submitted to the

FACULTY OF INFORMATION

AND COMMUNICATION ENGINEERING

*in partial fulfillment of the requirements*

*for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



## ***ANNA UNIVERSITY***

CHENNAI 600 025

## ***October 2018***

**ANNA UNIVERSITY**

**CHENNAI 600 025**

**CERTIFICATE**

Certified that this project report **“DEPRESSION DETECTION USING TWITTER TWEETS ”** is the bonafide work of Aruna Arumugam (2015103005), Shri Shruthi S (2015103022) who carried out the project as part of the **CS7711 – Creative and Innovative Project Laboratory** in the Department of Computer Science and Engineering, College of Engineering, Guindy, Chennai. The work reported herein is original and does not form part of any other work.

We understand the University’s policy on plagiarism and declare that the thesis is our own work, except where specifically acknowledged and has not been copied from other sources or been previously submitted for award or assessment.

**Aruna Arumugam**

**Shri Shruthi S**

**MS. V.SUGANYA**

**COURSE INSTRUCTOR**

**ABSTRACT**

There are many people around us who suffer from depression due to varying reasons. Depression is a very sensitive mental health issue in the recent years. Depression when not detected at the right time can lead to fatigue, irritability, insomnia, and in extreme cases can lead to unpredictable actions of the affected person.

Our aim is to identify depression using twitter tweets. We use a sequence to sequence encoder-decoder as our model and train the model to predict depression by using twitter tweets.

The model is evaluated using Accuracy(Acc), Precision(P), Recall(R), F1 score and Confusion matrix(CM). Our model has performed better in comparison to existing models.

**TABLE OF CONTENTS**

ABSTRACT..............................................................................................................2

LIST OF TABLES....................................................................................................4

LIST OF FIGURES..................................................................................................5

LIST OF ABBREVIATIONS...................................................................................6

INTRODUCTION ...................................................................................................7

1.1 PROBLEM DOMAIN.............................................................................8

1.2 PROBLEM DESCRIPTION ..................................................................8

1.3 SCOPE.....................................................................................................8

1.4 STAKEHOLDERS……………………………………………………..8

1.5 CONTRIBUTION ..................................................................................9

1.6 SWOT ANALYSIS ..............................................................................10

1.7 FEASIBILITY ANALYSIS..................................................................11

1.8 ORGANIZATION OF THESIS ...........................................................12

RELATED WORKS..............................................................................................13

SYSTEM DESIGN ..............................................................................................15

3.1 SYSTEM ARCHITECTURE...............................................................15

3.2 MODULE DESIGN.............................................................................16

3.3 JUSTIFICATION OF ALGORITHMS............................................... 20

3.4 EVALUATION OF ALGORITHMS………………………………....20

TEST PLAN ........................................................................................................33

RESULTS ANALYSIS........................................................................................39

CONCLUSION AND FUTURE WORK............................................................43

REFERENCES....................................................................................................44

**LIST OF TABLES**

Table 1.1 SWOT analysis ……………………………………………………….. 12

Table 1.2 Feasibility analysis ……………………………………………………. 13

Table 3.1 Data Collection Algorithm evaluation .……………………………….. 23

Table 3.2 Data Cleaning Algorithm evaluation ……...………...……..…………. 25

Table 3.3 Corpus of Tweets Algorithm evaluation …….…….………………….. 26

Table 3.4 Encoder Algorithm evaluation…..…………………………………….. 27

Table 3.5 Decoder Algorithm evaluation ……………...……………………….... 29

Table 3.6 Prediction Algorithm evaluation ……………………...………………. 31

Table 3.7 Data Cleaning and labelling ……………………………………………32

Table 3.8 Corpus of tweets …………………………………………………….....33

Table 3.9 Sequence encoding……………………………………………………..34

Table 3.10 Sequence decoding…………………………………………………..35

Table 3.11 Classification/Prediction……………………………………………..36

Table 5.1 Results and analysis ………………….……………………………….. 37

**LIST OF FIGURES**

Figure 3.1 System architecture ………………………………………………….. 17

Figure 3.2 Word to Vector Conversion snapshot .……………………………….. 33

Figure 3.3 Encoder and Decoder snapshot 1 ...………………………………….. 33

Figure 3.4 Encoder and Decoder snapshot 2 …...……………………………….. 34

Figure 5.1 Analysis formulae ………………………………………………….... 36

**LIST OF ABBREVIATIONS**

1. PCA - Principal Component Analysis
2. LDA - Linear Discriminant Analysis
3. RNN - Recurrent Neural Networks
4. LSTM - Long Short Term Networks

**CHAPTER 1**

**INTRODUCTION**

**1.1 PROBLEM DOMAIN**

In the recent years, there has been a lot of research and methods used in the field of sentiment analysis. The sentiment analysis has helped identify various emotions among the users. They have been used to detect various emotions such as happiness, sadness, frustration, anxiety and many more.

Depression is a very common and critical mental health issue that is always tend to be ignored in the society. A lot of people around us do not have the basic knowledge about depression and its effects. The initial stages of depression may not affect the person or the the others around them in a significant way. But when the symptoms are not detected due to either no knowledge, neglectance or fear of being judged by society the effects of derepression become severe leading the victim to take any extreme decisions. Depression when I identified at the right time can be cured simply with the aid of counselling.

Social media has been the platform for the people to express their views and opinions to the people all over the world. The activities of a user on social media tell a lot about them. This includes their amount of time spent per day on the social media, the domain of topics the user is interested to look into, their character based on their status and post etc. These information help in analyzing the nature of the user.

Hence our goal is to help identify depression using twitter tweets collected by scrapping the tweets off twitter pages.

**1.2 PROBLEM DESCRIPTION**

Our objective is to identify the depression present among the users using twitter tweets to help the victims of depression in overcoming it at an earlier stage.

**1.3 SCOPE**

The life of a teenager is not as easy as it was when compared to the generations a few years back. Their lives have become more advanced and complicated. They tend to be highly competitive by achieving more in a short period of time want to have as much fun-filled activities possible. As much as this may sound amazing, this attitude also tends to add a lot of stress on these teenagers causing them to breakdown and go into depression. Hence our project can be of great help in this domain.

**1.4 STAKEHOLDERS**

One of our primary stakeholder consists of the healthcare. This tool could be used to help identify the depression of the user even without the user’s knowledge of being affected mentally. It is usually the doctor who arises various questions to the patients in order to study their mind and derive a conclusion. But we believe that with this tool, the job of the doctors would be a mere consultation only. This leads to a reduce in consultation time allowing the doctors to assess more patients per day.

Our indirect stakeholders consists of the organizations that help people with mental illness. As more and more users are identified, the organizations will receive an increased number of clients.

Our main competitors are the mental health consultations. It has been a tradition for a long period of time that these mental health consultations conduct a series of survey and study about the behaviour of the person. So it could be a little difficult for this transition from onsite to online detection of depression.

**1.5 CONTRIBUTION**

This model is supposed to take in the twitter tweets and classify them if they have depression or not. We use a sequence to sequence encoder decoder to do this. We scrap the twitter tweets using tweepy package of python from various twitter pages. This data is scrapped based on hashtags. We scrap using 4 hashtags, namely, depression, sad, frustration and anger. The data collected using depression hashtag is used to train the model for positive examples whereas the data collected using the other three hashtags are used as negative examples.

This data is then preprocessed using twitter-preprocessing python package along with some regular expressions. The data on being preprocessed is translated from word to index in batches of 24 tweets. The 24 tweets in a batch comprise of 12 tweets of depressions, 3 tweets of sad, 3 tweets of anger and 3 tweets of a frustration. These indexes are then passed to the encoder and decoder to classify and predict.

**1.6 SWOT ANALYSIS**

|  |  |
| --- | --- |
| **STRENGTH**   * Our strength is the presence of enormous amounts of data available as we are using twitter to collect them. * There are various algorithms and packages in pythons used for machine learning and deep learning which can facilitate the implementation of this project. * This project of being implemented can help in detecting depression among the victims who do not know the symptoms of depression or neglect them due to fear of being judged. | **WEAKNESS**   * The weakness lies in the validity of the tweets obtained as data. * There is no method of evaluation for checking the validity of the tweets. * The usage of relatively small dataset. |
| **OPPORTUNITY**   * We see the availability of twitter tweets as a great opportunity for us as it helps understand the emotions of the user as tweets primarily consists of a user’s thoughts and expressions. * Our opportunity is using this data and being able to apply deep learning algorithms to predict from this available data. | **THREATS**   * One of our threats is the misuse of our product. If the tool goes in the hand of a wrong person, it could lead to them identifying the mentally-affected person and threatening them. * The implementation may consume a lot of time during the training phase. |

**Table 1.1** SWOT analysis

**1.7 FEASIBILITY ANALYSIS**

|  |  |  |  |
| --- | --- | --- | --- |
| Module name | Tool/technique with source or information | Alternatives considered | Alternatives rejected  (with reason) |
| Data Collection | Using **Tweepy** python package | An alternate package to tweepy is the twython. | Since the tweepy python package is more closer to the twitter’s API and provides easier iteration, we used tweepy. |
| Data cleaning and labelling | Using **Textblob** python package | Considered using NLTK which is another good method. | Textblob seemed to be comparatively simplified to use it for our product. |
| Corpus of tweets | **GloVe** | Another similar tools is the word2vec | GloVe has a better accuracy when compared to word2vec. |
| Sequence encoder | **pytorch** | TensorFlow is another choice we considered | The framework is more tightly integrated with Python language and feels more native most of the times when compared to tensorFlow and hence we use pytorch. |
| Sequence decoder | **pytorch** |
| prediction/classification | **pytorch** |

**Table 1.2** Feasibility analysis

**1.8 ORGANIZATION OF THESIS**

Chapter 3 discusses the overall system design and the flow of the system. Chapter 4 discusses the implementation of the modules along with the explanation of algorithms. Chapter 5 discusses how the model and algorithms were tested and presents their results. Chapter 6 discusses the results and analysis of the system after the implementation. Chapter 7 gives the details on conclusion and future works and improvements that can be done.

**CHAPTER 2**

**RELATED WORKS**

This chapter gives a survey of various papers and applications that exist in this domain. This survey helped to identify various approaches, techniques and future work possible with the project.

**Woebot**

Woebot uses machine learning and natural language processing to help users manage their mood and mitigate depression. It uses facebook messenger as the platform and the bot prompts the user with questions to assess their mood. Over time the algorithm learns the method and understands the user’s emotional profile and suggests therapies.

**Wysa**

Wysa is another application that is similar to woebot. Wysa records the texts using natural language processing and machine language. It finds a pattern in the communication. It then suggests tips like meditations, counseling etc.

**TANAKA ALGORITHM**

This paper specifically gives the methods for recognition of face, classification techniques for detection of different face parts like eyes, lips etc. By using these techniques and algorithm they collect the data base for depression analysis.The images are passed through INface matlab toolbox for identification of different body parts. The Tanaka algorithm applied on these images is primarily used to detect eye-pair region. IT used Nearest Neighbor classifier to train/test the system for detecting the severity of depression.

**DEPRESSION IN ADOLESCENCE**

This paper examines the possibility of depression which is going to develop after 1 to 2 years in adolescence. In this paper , database is obtained from

the at risk adolescents facial images and not at risk facial images. Prediction of depression were followed by 2 years of data collection. For this approach ,two approaches for detection of depression by facial images were followed. First approach is eigenface. Eigenface means it uses PCA approach. And the second method is fisherface.It used PCA and LDA method. PCA and LDA are two feature extraction approaches. Nearest neighbor approach were used to classify person dependent and person independent type.

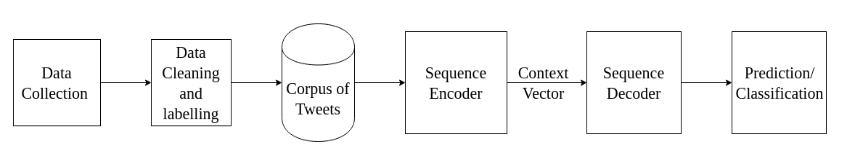
**CHAPTER 3**

**SYSTEM DESIGN**

**3.1 SYSTEM ARCHITECTURE**

Figure 3.1 represents the block diagram of the system. Our depression detection system consists of the following modules:

* Data Collection
* Data Cleaning and labelling
* Corpus of Tweets
* Encoder
* Decoder
* Prediction/Classification



**Figure 3.1** System architecture

The first module consists of the data collection. The data collected is preprocessed and converted to the corpus of tweets. This is then passed onto the sequence to sequence encoder decoder and finally predict the tweet.

**3.2 MODULE DESCRIPTION**

**3.2.1 DATA COLLECTION**

Our dataset consists of 10,800 tweets collected/scraped\cite{scraping} from Twitter using their hashtags. We primarily collected tweets that have Anger, Sad, Depression and Frustration as hashtags in the tweets. We use tweets with Anger, Sad and Frustration hashtags as negative samples for the model and ones with Depression tags as positive samples for the model.

These tweets are unlabeled. They have to be preprocessed and labeled for giving them as input to the model.

We use Tweepy python package to scrape required tweets from Twitter.

Input: twitter pages

Output: scrapped unlabeled tweets

**3.2.2 DATA CLEANING**

We process the collected dataset to clean them, that is, to remove hyperlinks, hashtags, emojis and other irrelevant text. We also change common message language words to proper words like "'n" to "In". Similarly, we sanitize each tweet to remove irrelevant text to make it suitable for the model.

We then label each tweet collected with "Depression" hashtag as "1" and those with hashtags "Anger", "Sad" and "Frustration" as "0" as targets for the model. "1" indicates that the tweet indicates symptoms of tweet and "0" indicates that the tweet does not show any symptom of depression.

We use Textblob\cite{textblob} python package to clean the tweets and to label them.

Input: unlabeled tweets

Output: labeled and cleaned tweets

**3.2.3 CORPUS OF TWEETS**

In this module, we process the corpus of tweets to convert them to a proper format to input them for the model. We tokenize each tweet and find the word to index of each tweet.We then create a mini-batch to be given as input to the model. We use mini-batches of size 12. The batch contains 4 tweets with "Anger", 4 tweets with "Frustration", 4 tweets with "Sad" and 12 tweets with "Depression". We pad the batches with 1 to get a square matrix input of size 24 \* 40 where 40 is the maximum sequence length.

We pass the corpus of tweets to get them in the form of word embeddings to input to the model using Word2Vec.

Input: labeled and cleaned tweets

Output: batches of word to indices of tweets

**3.2.4 ENCODER**

Encoder forms the first part of a sequence to sequence classifier. The encoder is a Long Short Term Memory (LSTM) model. We pass the processed corpus of tweets as input to the encoder, batch by batch. The encoder processes or encodes the batch of word of indices of tweets and gives as output a context vector. The context vector gives us a probabilistic distribution of the entire sequence/batch that was processed.

We use Pytorch framework to create a sequence model that acts as an encoder and produces the context vector.

Input: batched of word to indices of tweets

Output: context vector

**3.2.5 DECODER**

The decoder is used to get a final output vector from which a prediction can be made as to whether the tweet indicates depression or not. The decoder is also Long Short Term Memory (LSTM) model. We feed the context vector as additional input to the decoder along with the input batch to get the final decoder output tensor.

We use Pytorch framework to create a sequence model that acts as the decoder and takes the context vector as input to produce final classification.

Input: context vector and batched of word to indices of tweets

Output: final decoded output vector

**3.2.6 PREDICTION/CLASSIFICATION**

In this module, we predict whether the tweet indicates depression or not through the final decoder output tensor. We perform linear transformations on the output and apply a sigmoid function to get a value between 0 and 1. If the value is greater than or equal to 0.5, we say that the tweet has got symptoms of depression and if it is less than 0.5, we say that it does not have major symptoms of depression.

The output from the decoder is linearly transformed and used to predict if the tweet is an indication of depression or not.

Input: final decoded output vector

Output: prediction whether the tweet indicates depression or not

**3.3 JUSTIFICATION OF SELECTION ALGORITHM**

Sequence to sequence (seq2seq) prediction is a key to many tasks of machine learning.

Encoder-decoder neural models are a generic deep-learning approach to sequence-to-sequence (Seq2Seq) tasks. They encode an input sequence into a vector representation from which the decoder generates an output. These models have shown to achieve state-of-the-art or at least highly competitive results for various NLP tasks.

They have shown to outperform traditional classifiers in many machine learning and deep learning tasks.

Hence, we have used the Encoder-Decoder (Sequence-to-Sequence) classifier in the solution model of our product.

**3.4 EVALUATION OF ALGORITHM**

**3.4.1 DATA COLLECTION**

The data collection module uses the tweepy python package to collect the data from twitter pages. We collect data based on the hashtags present in the twitter tweets. We scrap using 4 hashtags, namely, depression, sad, frustration and anger.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name:** dataCollection | | |  |
| **Processing Strategy:**  The input to this stage consists of twitter pages. The algorithm uses tweepy package and searches for tweets with “#depression” and writes on to a csv file as part of positive example. Similarly, tweets with “#sad”, “#anger” and “#frustration” are collected and stored as part of negative examples. | | | |
| **Objective/ Goal of the Algorithm:** The objective is to collect data for feeding it to the models. | | | |
| **Example**  Given a twitter page, it searches for the specified # and scrapes the data and writes it onto the csv file. | **Which module of your Project this algorithm is used?**  This is used in the first module, i.e, data collection. | | |
| **Pseudocode /Flowchart of the algorithm** | | | |
| **What classes of problem is the algorithm well suited?**  It is well suited for problems of scraping data. | | | |
| **Common benchmark (or) example datasets used to demonstrate the algorithm**  This algorithm uses the twitter pages as the input. | | **References**  [scraping](https://gist.github.com/vickyqian/f70e9ab3910c7c290d9d715491cde44c) | |

**Table 3.1** Data Collection Algorithm evaluation

**3.4.2 DATA CLEANING**

This module preprocesses the data collected in the previous module in order for it to be used the further stages. We use the tweet-preprocessor package to do an initial preprocessing. Then we use regular expressions to do further processing of the data collected.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name:** DataCleaning | | |  |
| **Processing Strategy:** In data cleaning algorithm, we use regular expression to remove unwanted words/ text from the data collected in the previous step. The input to the algorithm is the csv file with previously collected data. | | | |
| **Objective/ Goal of the Algorithm**  The objective of the algorithm is to remove unnecessary words/text. | | | |
| **Example**  It collects the data from the csv file, checks for the unwanted text such as links and removes them. | **Which module of your Project this algorithm is used?**  This is part of the data cleaning module. | | |
| **Pseudocode /Flowchart of the algorithm** | | | |
| **What classes of problem is the algorithm well suited?**  The algorithm is mainly suited for problems like data cleaning or searching. | | | |
| **Common benchmark (or) example datasets used to demonstrate the algorithm**  One of the examples is the data set we have scrapped in the previous step. | | **References**  <https://docs.python.org/3/howto/regex.html> | |

**Table 3.2** Data Cleaning Algorithm evaluation

**3.4.3 DATA LABELLING**

This module aims at labeling the collected and preprocessed data. The data is labelled 1 is they were collected with the hashtag depression. Else they are labelling 0.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name:** DataLabelling | | |  |
| **Processing Strategy:**  The cleaned data is passed on as input and the data is tokenized. These tokenized words are then looked up on for removing stop words. The data is then POS tagged and then labelled as a positive or negative example. | | | |
| **Objective/ Goal of the Algorithm**  The goal of this algorithm is to label if the example is positive or negative. | | | |
| **Exampl**If there is a word depression in the tweet, it is labelled as positive, else it is labelled negative. | **Which module of your Project this algorithm is used?**  This is used in the data labelling module. | | |
| **What classes of problem is the algorithm well suited?**  The algorithm is well suited for classification problems. | | | |
| **Common benchmark (or) example datasets used to demonstrate the algorithm** | | **References**  <https://www.analyticsvidhya.com/blog/2018/02/natural-language-processing-for-beginners-using-textblob/> | |

**Table 3.3** Corpus of Tweets Algorithm evaluation

**3.4.4 ENCODER**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name:** Recurrent Neural networks | | |  |
| **Processing Strategy:**  Given word embeddings as input, it provides the context vector as output. | | | |
| **Objective/ Goal of the Algorithm**  To provide context vector to be used as input for the decoder | | | |
| **Example**  Given a tweet, it provides a context vector of the tweet that consists information about the tweet in it. | **Which module of your Project this algorithm is used?**  Sequence Encoding | | |
| **Pseudocode /Flowchart of the algorithm** | | | |
| **What classes of problem is the algorithm well suited?**  Problems involving text and images | | | |
| **Common benchmark (or) example datasets used to demonstrate the algorithm**  Shakespeare data, Paul Graham (<http://www.paulgraham.com/articles.html>) | | **References**  <https://arxiv.org/abs/1406.1078> | |

**Table 3.4** Encoder Algorithm evaluation

**3.4.5 DECODER**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name:** Recurrent Neural networks | | |  |
| **Processing Strategy:**Given the context vector as input, it should give as output a tensor which can be used to predict. | | | |
| **Objective/ Goal of the Algorithm**  Provide a output tensor containing prediction information. | | | |
| **Example** For a context vector of a tweet, it provides a tensor that contains information to predict whether the tweet indicates depression or not. | **Which module of your Project this algorithm is used?**  Sequence Decoding | | |
| **Pseudocode /Flowchart of the algorithm** | | | |
| **What classes of problem is the algorithm well suited?**  Problems involving text and images | | | |
| **Common benchmark (or) example datasets used to demonstrate the algorithm**  Shakespeare data, Paul Graham (<http://www.paulgraham.com/articles.html>) | | **References**  <https://arxiv.org/abs/1406.1078> | |

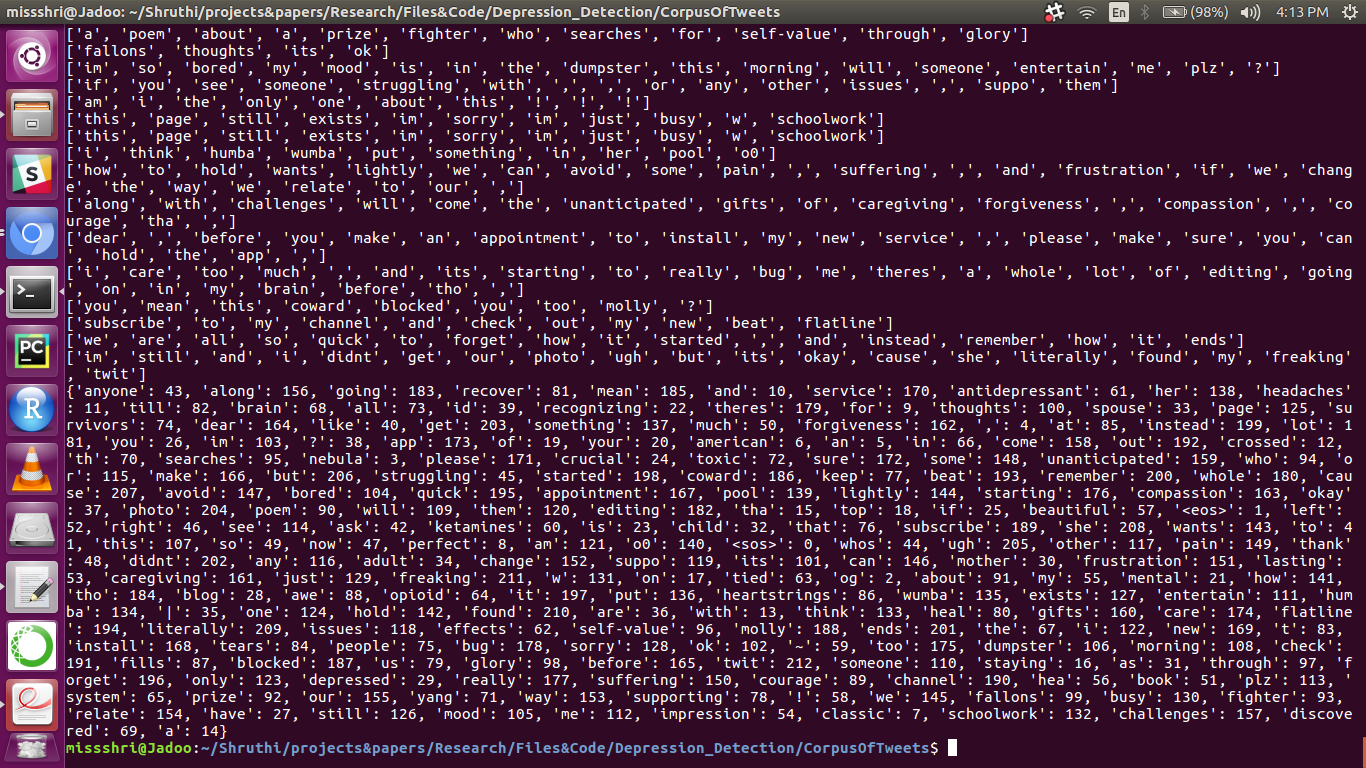
**Table 3.5** Decoder Algorithm evaluation

**3.4.6 PREDICTION/CLASSIFICATION**

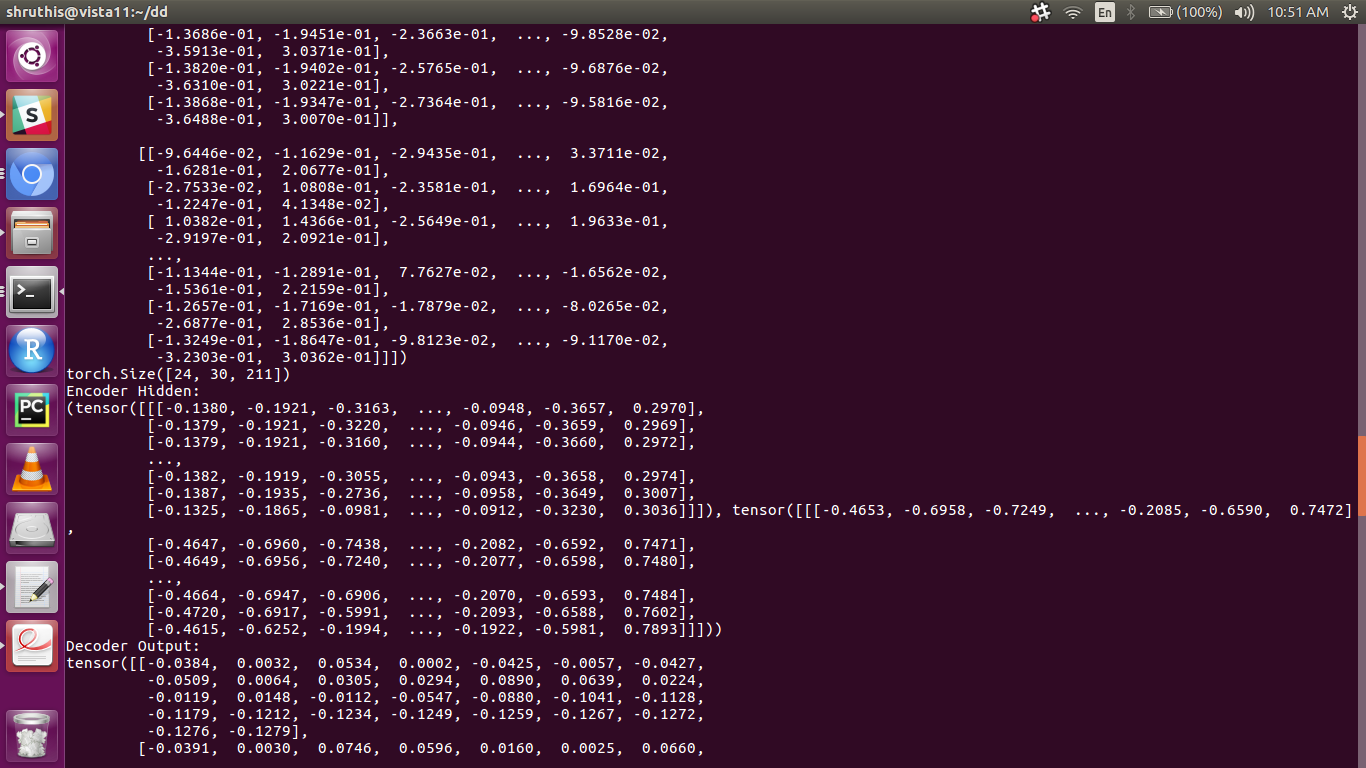
|  |  |  |  |
| --- | --- | --- | --- |
| **Name:** Sigmoid function | | |  |
| **Processing Strategy:**Given the output tensor from the decoder, it predicts 1 or 0 indicating depression or not. | | | |
| **Objective/ Goal of the Algorithm**  Used to predict. | | | |
| **Example**  For a given tweet, it predicts whether or not it indicates depression. | **Which module of your Project this algorithm is used?**  Prediction | | |
| **Pseudocode /Flowchart of the algorithm** | | | |
| **What classes of problem is the algorithm well suited?**  Problems involving binary prediction | | | |
| **Common benchmark (or) example datasets used to demonstrate the algorithm**  Pima Indian DIabetes Dataset (<https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes>) | | **References**  <https://pytorch.org/docs/stable/_modules/torch/nn/modules/activation.html#Sigmoid> | |

**Table 3.6** Prediction Algorithm evaluation

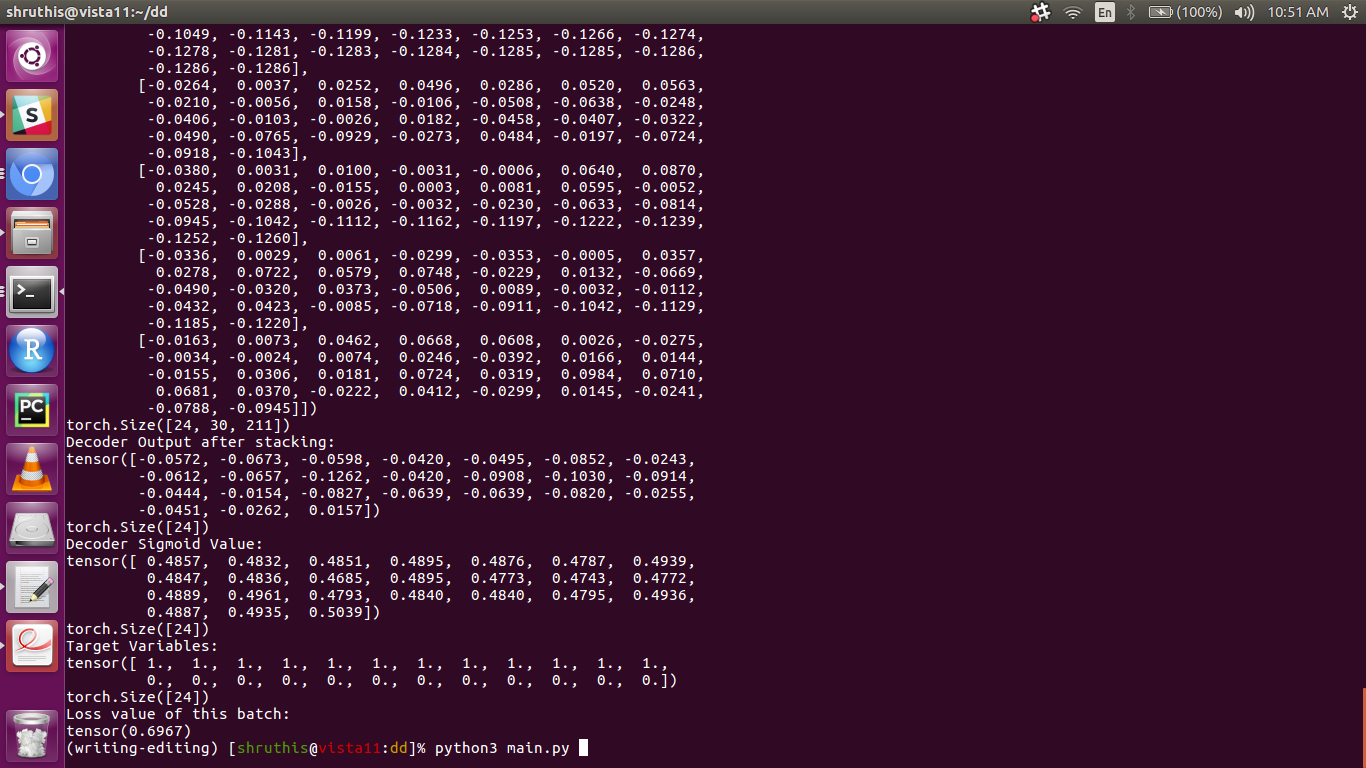
**3.5 SNAPSHOTS OF PROJECT MODULES:**



**Figure 3.2** Word to Vector conversion snapshot



**Figure 3.3** Encoder and Decoder snapshot 1



**Figure 3.4** Encoder and Decoder snapshot 2

**CHAPTER 4**

**TEST PLAN**

We have divided the data set in the ratio 60:40 for training and testing. We have performed analysis of the model using the test data. We performed unit testing and integration testing over the modules. The unit testing consisted of testing each module for their functionality individually. The integration testing tested the flow from corpus of tweets after forming batches and passing them onto the encoder and decoder to get the prediction. This was done for several epochs.

**4.1.1 Data Collection**

|  |  |  |
| --- | --- | --- |
| **Module Name** | **Sample Input** | **Expected Test Results** |
| 1.Data Collection | Twitter pages | A csv file with scrapped twitter tweets |
| **Prerequisites** | Twitter developer account credentials | |
| **Pass / Fail Criteria** | Pass: The scraping is completed successfully  Fail: scrapping is interrupted. | |
| **Test Procedure** (Only from the UI perspective! – Think you are doing Black Box Testing) | 1)give credentials  2)run code | |
| **Assumptions/Constraints** | 1. requires internet access. 2. may take upto 5 or 10 mins depending on the size of data to be scrapped. | |
| **Actual Output**  section of the collected data: | | |

**Table 3.7** Data collection

**4.1.2 Data cleaning and labelling**

|  |  |  |
| --- | --- | --- |
| **Module Name** | **Sample Input** | **Expected Test Results** |
| 2.Data cleaning and labellinng | csv file of scrapped data | preprocessed data |
| **Prerequisites** | data scraping of twitter tweets  install textblob  install re | |
| **Pass / Fail Criteria** | Pass: removal of hashtags, smiley, emoji, links  fail: presence of encodings | |
| **Assumptions/Constraints** | 1. the textblob does a very basic cleaning only 2. must include our own regular expressions to remove data that is not required after analyzing the csv file manually. | |
| **Actual Output**  cleaned/ preprocessed data | | |

**Table 3.8** Data cleaning and labelling

**4.1.3 Corpus of tweets**

|  |  |  |
| --- | --- | --- |
| **Module Name** | **Sample Input** | **Expected Test Results** |
| 3. corpus of tweets | preprocessed csv file | a word to vec converted result in batches |
| Prerequisites | Pre-processing | |
| Pass / Fail Criteria | Pass: the words are indexed on being passed in batches  Fail: the batching fails. | |
| Assumptions/Constraints | 1. requires torch preinstalled | |
| **Actual Output**  section of the output: | | |

**Table 3.8** Corpus of tweets

**4.1.4 Sequence encoding**

|  |  |  |
| --- | --- | --- |
| **Module Name** | **Sample Input** | **Expected Test Results** |
| 4. sequence encoding | corpus of tweets with indexed values | word embeddings |
| Prerequisites | Pre-processing, word2vec conversion | |
| Pass / Fail Criteria | Pass: the context vector is created | |

**Table 3.9** Sequence encoding

**4.1.5 Sequence decoding**

|  |  |  |
| --- | --- | --- |
| **Module Name** | **Sample Input** | **Expected Test Results** |
| 5. sequence decoding | word embeddings | prediction |
| Prerequisites | Pre-processing, word2vec conversion, sequence encoding | |
| Pass / Fail Criteria | Pass: gives a prediction | |

**Table 3.10** Sequence decoding

**4.1.6 Classification/ prediction**

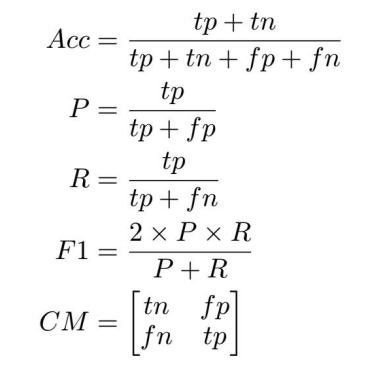
|  |  |  |
| --- | --- | --- |
| **Module Name** | **Sample Input** | **Expected Test Results** |
| 6. classification/ prediction | Prediction | Returns if prediction is true or not. |
| Prerequisites | Pre-processing, word2vec conversion, sequence encoding,sequence decoding | |
| Pass / Fail Criteria | Pass: if the returned prediction is true  fail: if the returned prediction is false | |

**Table 3.11** Classification/prediction

**CHAPTER 5**

**RESULTS AND ANALYSIS**

To evaluate our model, we mainly used Accuracy(Acc), Precision(P), Recall(R), F1 score and Confusion matrix matrix(CM) defined as follows:

****

**Figure 5.1** Analysis formulae

where tp, tn, fp, fn represents "True Positive", "True Negative", "False Positive" and "False Negative".

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LABEL** | **ACC** | **P** | **R** | **F1** |
| Non-Depressed | 0.92 | 0.88 | 0.92 | 0.91 |
| Depressed | 0.91 | 0.91 | 0.90 | 0.92 |

**Table 5.1** Results and analysis

**CHAPTER 6**

**CONCLUSION AND FUTURE WORKS**

Our model has performed really well in comparison with existing models. The relatively very high performance might be due to the small dataset. One major issue with the dataset was we scraped tweets with hashtags "Depression" from twitter which is a loose logical assumption of depressed tweets. Our model, good at classifying depressed tweets, would have been benefited by a dataset of depressed tweets confirmed by experts in the field.

Our model could use some iterations in the future. We can try Convolutional Neural Networks(CNNs) for Encoder and compare and analyze LSTMs and CNNs to find the best encoder for the system. We can also try out attention on the sequence to sequence models. Some hyperparameters that can be tuned is the batch size and the sequence length.

**REFERENCES**

[1] Woebot - Your charming robot friend who is here for you, 24/7, Woebot.io, 2018. [Online]. Available: https://woebot.io/. [Accessed: 14- Oct- 2018].

[2] Wysa - your 4 am friend and AI life coach, Wysa - your 4 am friend and AI life coach, 2018. [Online]. Available: https://www.wysa.io/.

[3] A. Pampouchidou, M. Pediaditis, F. Chiarugi, K. Marias, P. Simos, F. Yang, F. Meriaudeau, M. Tsiknakis, ”Automated characterization of mouth activity for stress and anxiety assessment”, Imaging Systems and Techniques (IST) 2016 IEEE International Conference on, pp. 356-361, 2016.

[4] R.E. Roberts, et al., ”Screening for Adolescent Depression: A Comparison of Depression Scales,” Journal of the American Academy of Child and Adolescent Psychiatry, vol. 30, pp. 58-66, 2009.

[5] Tweepy Documentation tweepy 3.5.0 documentation, Docs.tweepy.org, 2018. [Online]. Available: http://docs.tweepy.org/en/v3.5.0/.

[6] tweet-preprocessor, PyPI, 2018. [Online]. Available: https://pypi.org/project/tweet-preprocessor/.

[7] TextBlob: Simplified Text Processing TextBlob 0.15.1 documentation, Textblob.readthedocs.io, 2018. [Online]. Available: https://textblob.readthedocs.io/en/dev/.

[8] re Regular expression operations Python 3.7.1rc2 documentation, Docs.python.org, 2018. [Online]. Available: https://docs.python.org/3/library/re.html.

[9] Welcome to PyTorch Tutorials PyTorch Tutorials 1.0.0.dev20181002 documentation, Pytorch.org, 2018. [Online]. Available: <https://pytorch.org/tutorials/>.

[10] Understanding LSTM Networks – colah’s blog, Colah.github.io, 2018. [Online]. Available: http://colah.github.io/posts/2015-08-UnderstandingLSTMs/. [11] CS231n Convolutional Neural Networks for Visual Recognition, Cs231n.github.io, 2018. [Online]. Available: http://cs231n.github.io/convolutional-networks/.

[12] A Python script to download all the tweets of a hashtag into a csv, Gist, 2018. [Online]. Available: https://gist.github.com/vickyqian/f70e9ab3910c7c290d9d715491cde44c/.

[13] torch.nn.modules.activation PyTorch master documentation, Pytorch.org, 2018.[Online]. Available:https://pytorch.org/docs/stable/modules/torch/nn/modules/activation.htmlSigmoid.

[14]K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, ”Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”, Arxiv.org, 2018. [Online]. Available: https://arxiv.org/abs/1406.1078.

[15] Depression: Causes, Symptoms and Treatments, Live Science, 2018. [Online]. Available:https://www.livescience.com/34718-depressiontreatment-psychotherapy-anti-depressants.html.