S	
E1	
E2	
For Office Use Only	

Masters Project Final Report

(MCS)

2019

Project Title	Mining of people's opinions towards government's decisions/activities in the twitter posts using Sentiment Analysis
Student Name	D.L.A. Jayawickrama
Registration No. & Index No.	2017/MCS/042 17440429
Supervisor's Name	M.G.N.A.S. Fernando

For Office Use ONLY

Mining of people's opinions towards government's decisions/activities in the twitter posts using Sentiment Analysis

A dissertation submitted for the Degree of Master of Computer Science

D.L.A. Jayawickrama
University of Colombo School of Computing
2019

UCSC

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: D.L.A.Jayawickrama

Registration Number: 2017/MCS/042

Index Number: 17440429

____

Signature: Date:

This is to certify that this thesis is based on the work of

Mr./Ms. D.L.A.Jayawickrama

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: M.G.N.A.S. Fernando

____

Signature: Date:

Abstract

In modern world social media is widely used by people around the world and now it has become the most popular platform to express people's opinions on popular day today topics. This study was carried out by using one of the most popular social media data with some of the trending computer technologies like machine learning, LSTM and word embeddings.

One of the challenges that government face is, they cannot get an overall idea about how the people in the country react to the decisions and actions they take, whether the people accept those or not and these cannot be analyzed frequently. As a result, government loses votes in elections if the majority of the people were unhappy about the decisions and actions they made. In order to overcome above problem this study suggests an application to analyze how people react to different types of decisions and actions the government takes, and this can be done frequently, even in daily basis without any additional cost. To use this application developed, the government should have an official twitter page. There the government can tweet their day to day decisions and activities. When the people put comments, replies and likes to those tweets, this application can be used to get an idea about how the majority of the people react to those decisions and actions. The government can use this application in two different ways.

1. They can get an overall idea about how majority of the people reacted to an action/decision they took by analyzing the comments, replies and likes people put to that particular post they have published describing the action/decision they took.

2. They can analyze how the people reacted to the actions/decisions they took during a time period (this month, this week) and can understand whether people are happy about the government or not by analyzing the comments and likes people put to the posts published during that period. Here replies won't be taken into consideration due to limitations in the twitter API and this will be discussed further in following chapters.

To evaluate the model built during this study, metrics like accuracy, precision, recall and f1-score were calculated.

Acknowledgements

I would like to express my sincere gratitude to Dr. M.G.N.A.S Fernando, senior lecturer of the University of Colombo School of Computing, for his continuous guidance throughout the study and for the effort he has put to make this successful.

I’m also grateful to all the academic members of the University of Colombo School of Computing for their dedication and effort put to teach us throughout the master’s program.

Lastly, I would like to thank my friends who supported me by giving some ideas and my family for the encouragement given to carry out this research.

List of abbreviations

API	Application Programming Interface
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
IT	Information Technology
NLP	Natural Language Processing
GLOVE	Global Vectors
URL	Uniform Resource Locator
CNN	Convolutional Neural Network
DNN	Deep Neural Networks
TD-IDF	Term Frequency–Inverse Document Frequency

List of Figures

List of Equations

Table of Contents

Chapter 1
Introduction

1. Introduction

Getting meaningful insights from social media data has now arrived, with the evolution in the technology. In this study the data collected by twitter (tweets) - one of the most famous social media platform in the world, will be used for doing in-depth research to identify underlying sentiments (positive or negative). With the aid of machine learning techniques this study tries to analyze how people reacts to the tweets published in the government's official twitter page which describes decisions/activities the government has taken.

2. Problem Domain

Nowadays people use social media as the most powerful platform to voice their opinions regarding different day to day topics. Among those social media platforms, twitter has a large set of audience and therefore has a huge volume of sentiment rich data. Unlike other social media platforms, in twitter almost every user's tweets and the comments and replies are completely public. This is a huge advantage when trying to get a large amount of data to run analytics on. Apart from that twitter has been recognized as the most popular social media platform among professionals. Most of the politicians, governments and officials use twitter to publish their suggestions, decisions and their day to day activities. People can comment and put likes to those and express their ideas too, so this makes a direct communication channel between people and the officials. This study focusses on analyzing comments, replies and likes people put to tweets published by the government in their official page. This may help governments to understand how people react to their decisions/activities, positive or negative way.

This research is associated with several significant areas of modern computer science. Machine learning, Word embeddings and Long Short-Term Memory (LSTM) are some of them. Apart from those, this study focuses on extracting sentiments of texts, which is called a sentiment analysis.

2.1 Sentiment analysis

Sentiment analysis is an automated process which uses Natural Language Processing. This can be used to understand the underline opinion of a piece of text with respect to a particular subject. Using this method can classify the opinions in texts into categories like 'negative', 'positive' and 'neutral'. This is very useful when monitoring social media data to get an overall idea about how the people reacts to certain topics[3].

2.2 Machine Learning

Machine learning is a very famous trend in Information Technology (IT) field. It has been used for developing various kind of applications like self-driven cars, image processing, Speech Recognition, Email Classification, Spam Filtering etc. This focuses on developing intelligent applications which can access data and learn by themselves. These applications need not to be programmed with all the instructions instead they automatically learn and improve their knowledge from the experiences they get. They observe data, look for patterns and take decisions.

2.3 Word embeddings

Word embeddings are a technique which use to represent words. It represents words in a way where the similar words have a similar encoding. This technique converts strings into a numeric format so it can be fed to the model. The words are represented as vectors and they can have hundred to thousand dimensions [9][10].

2.4 LSTM

A major problem in Recurrent Neural Network is it has a short-term memory capacity hence it can't handle long-term dependencies. This is called 'Vanishing Gradient' problem. When lots of information fed into the model, at some point those past information gets lost. Therefore, Recurrent Neural Networks are useful when there are short-term dependencies. LSTM is a special type of Recurrent Neural Networks and was introduced by Hochreiter and Schmidhuber to solve this ‘Vanishing Gradient’ problem [1].

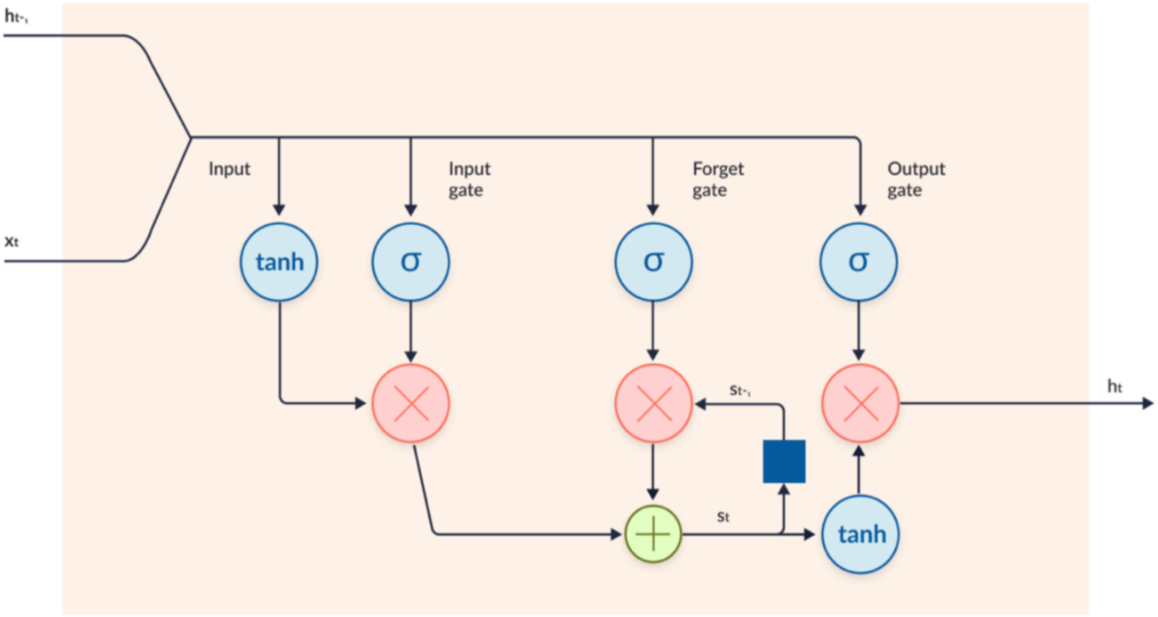


Figure 1 - LSTM

LSTM uses gated cells to store information. By using these gated cells, it decides what information need to be carried further and what need to be forgotten. These cells make decisions and by opening and closing the gates they execute those decisions. They pass two states between the cells, the ‘cell state’ and the ‘hidden state’.

LSTM has three types of gates:

- Forget gate – This gate removes unwanted information from the cell state which does not have any impact to the task it’s carrying out. This improves the network performance.
- Input gate – This gate adds important information to the cell state which has major impact to the task it’s carrying out.
- Output gate – All the information that has been added to the cell state is not important. So, this gate selects useful information from the current cell state and show it out.

3. Problem

This research focuses on analyzing the comments, replies and likes that people put to decisions and activities published in the government's official twitter page. At the end of this research an application will be built which can be used by a government authority, so the government can understand how people reacts to their decisions or the activities, whether the majority of people like or dislike them.

4. Motivation

When government takes important decisions and actions, sometimes the leaders and the government do not have any clue about people's opinions towards them. Due to this difficulty in predicting people's opinions sometimes government makes a majority of the people unhappy and this may lead them to lose votes in the next election. If there is a method to predict people's reactions to government's decisions and activities, the government can understand how good they are doing and can take necessary actions if a majority of people are unhappy with their decisions or activities.

By having an application as this study suggests, it will help government to keep its people happy and it will help them to maintain their existence without losing votes in upcoming elections.

5. Exact Computer Science Problem

The actual problem that going to be solved by this research is, finding a good approach to analyze comments, replies and likes people put to the tweets published by the government so that the government can understand the overall feedback for the decisions and activities they took. This includes building a sentiment analysis model and deriving equations to calculate the final overall percentages of each sentiment (positive, negative).

6. Research Contribution

6.1 Goal

Main goal of this study is to deliver an application which can be used by a government authority, so by using it he can take an overall idea about how the people in the country reacts to the decisions/actions the government has taken and posted in the government's twitter page. For this, some endpoints available in the twitter Application Programming Interface (API), a sentiment analysis model and some equations to calculate the overall sentiment percentages (positive, negative) will be used along with a front-end application.

Model will be developed by using latest, most efficient technologies and with a high accuracy to identify sentiments correctly. This will be used to identify the sentiment of each comment/reply.

When developing equations to calculate the overall percentage of each sentiment, there are few important facts to consider. People put likes to comments approving/accepting what's written in the comment. So, when counting likes, the sentiment of the comment is very important to identify whether the like put to that comment has a positive or a negative weight to the overall feedback for the post published by the government. If the comment is a negative comment, the likes it has should be taken as negative feedbacks and the likes for the comments which have positive sentiments should be taken as positive feedbacks. So, these equations will be derived by considering the sentiment of each comment/reply and likes counts to the post and likes counts to the each positive/negative comment/reply.

6.2 Objectives

The aim of this study will be achieved by focusing on the following objectives.

- Build a sentiment analyzer model
- Derive equations to calculate the final positive and negative sentiment percentages by considering collected comments/replies and likes.

- Build a front-end application to be used by the government authority.
- Build the backend of the application which uses the developed model, and which use derived equations to calculate the overall sentiment percentages and give a summary of people's feedback.

7. Scope

The primary focus of this research is to build an application that can be used by a government authority so the government can take an idea about how people reacts to the decisions and actions they take.

Scope of the project will include

- Building the sentiment analysis model
- Make use of endpoints available in twitter API to retrieve likes, comments and relevant data needed for the analysis
- Come up with equations to calculate the overall percentage sentiment values considering likes to the posts, likes to comments/replies and sentiment value of each comment/reply
- Front end application which can be used by the government authority to run the analysis.
- The output of the analysis – positive and negative percentages will be displayed in a graphical format

8. Structure of the Dissertation

The Chapter 1 – Introduction includes a fully detailed description of the problem domain, objectives and the scope of the study. Chapter 2 – Literature Review discuss about the researches which have been carried out relevant to the problem domain. The Chapter 3 – Methodology explains how the study was carried out. It includes details about the dataset, algorithms, techniques used and details about the end product. Chapter 4 – Results and Evaluation describes the evaluation plan carried out in order to measure the accuracy of the model built and the results of the evaluation. Chapter 5 – Conclusion includes final comments of the study and give some idea about the future work could be carried out to improve the application developed here.

The above chapters consist of relevant diagrams, equations and figures in order to give a clear picture of the work carried out in this study.

Chapter 2

Literature review

1. Introduction

It is very important to do a literature review on the problem domain and corelated fields, in order to find a better approach, most suitable technologies and to identify the challenges within the problem domain. Those information are very important on planning how the study should be executed therefore a detailed description of recent related studies will be described under this section.

2. History and recent researches

This study was planned to carryout using technologies like Machine learning, deep learning, word embeddings and LSTM. Apart from that, this study focuses on mining the underlying sentiments of texts, which is call as a sentiment analysis.

Getting meaningful insights from social media data has now arrived with the enhancement of the technology. So, here twitter social media data will be used for doing in-depth research to identify how people in the country reacts to different decisions, actions the government takes. In order to find a better approach for this study a literature survey about recent researches of corelated fields of this study was carried out and taken into consideration.

Sentiment analysis has been used widely by applying it in many areas such as e- commerce, marketing, politics etc. It is a growing research area in the field of computer science. At the beginning it was used for some applications like, for analyzing product reviews and now it has evolved to analyze the sentiments in the areas of politics, medicine, social engineering etc. Sentiment analysis is deriving subjective information from texts. By doing a sentiment analysis people can get underlying opinions, ideas hidden inside the texts [2]. Nowadays people tend to express their feelings towards current trending topics, news and products using social media and this makes huge impact to the concerned parties. Therefore, sentiment analysis is important so that the parties get impacted by them can handle and plan the outcome properly [3].

Twitter has a large amount of audience and one of the important features it has is the tweets, comments, replies that people put are public and anyone can access it hence it has a huge volume of data. This makes twitter data more suitable for analytics compared to other social media data. Even though it has this advantage when using them for analytics still there are some additional pre-processing needs to be done to make them ready for analytics. Specially this is required if someone use this data for training models and extracting sentiments. The twitter data are polluted with many unnecessary strings like Uniform Resource Locators (URLs), numbers, stop words etc, which do not have any impact on performance of the model.

Zhao Jianqiang and Gui Xiaolin from School of Electronic and Information Engineering, Xi'an Jiaotong University, China have carried out a research to explore various pre-processing methods on twitter data and have monitored how they increase or decrease the performance of twitter sentiment analysis [4]. Some of the pre-processing methods they have evaluated were removing stop words, removing URLs, reverting repeated letters, removing numbers etc. They have conducted experiments using five Twitter datasets and four classifiers. At the end of the study they have concluded that removal of stop words, numbers and URLS reduce the noise in data but it doesn't affect to the performance of the four classifiers that much. But some other pre-processing methods like replacing negation and expanding acronyms have increased the classification accuracy.

A. Krouska, C. Troussas and M. Virvou from Software Engineering Laboratory, Department of Informatics University of Piraeus Greece have also studied how the pre-processing techniques on twitter data affect the performance of models [5]. During this study they have used a series of pre-processing techniques with three different twitter datasets and have evaluated the performance of four classifiers. At the end of the study they have stated that pre-processing is very important and it affects the quality of the classification model.

LSTM has been introduced as a solution to the vanishing gradient problem in Recurrent Neural Networks (RNN). Therefore, LSTM can be used when there are longer dependencies and when the prediction depends on historical data[6]. During a sentiment analysis sometimes texts with longer sequences need to be analyzed too. In such situations RNNs won't perform well in identifying the underlying sentiment hence it may give wrong predictions. Therefore, to build sentiment analysis models LSTM is most suitable over RNNs[7].

A research has been carried out by D. Li and J. Qian from School of Science Beijing University of Posts and Telecommunications Beijing, China in order test how well RNNs and LSTMs perform when they use for sentiment analysis tasks [8]. During their study they have carried out a sentiment analysis using both RNN and LSTM and have compared the results of both models. According to the results the LSTM has performed well than the conventional RNN.

Word embeddings are very popular and currently widely use for building sentiment analysis models because they represent words as vectors and solves the problems that one hot encodings have hence it boosts performance of models. The most popular ones among different types of word embedding models are Word2Vec [9] and GloVe [10]. In those models neural networks have used to train the relations between words and documents in the training data.

A research has been done by Dang Cach, Moreno García María and De La Prieta Fernando [11]. They have evaluated Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and RNN models which use TF-IDF and word embeddings with a series of datasets. At the end of the study they conclude when doing a sentiment analysis, the word embeddings perform well when combining them with deep learning methods rather than Term Frequency–Inverse Document Frequency (TD-IDF).

Oscar Blessed, Deho & Agangiba, William & Aryeh, Felix & Ansah, Jeffery have also done a research on sentiment analysis using word embeddings. They have used Word2Vec pre trained word embeddings to train the model and they have stated the word embeddings have improved the accuracy of the model greatly.[12]

Many sentiment analysis studies have been carried out using both LSTM and word embeddings over the years. Wang, Jenq-Haur, Ting-Wei Liu, Xiong Luo and Long Wang have proposed an approach for sentiment classification using LSTM and word embeddings [13]. At the end of the study they have mentioned that deep learning methods like LSTM performs better while doing sentiment classification, when they have been trained with large amount of training data.

Ms.R.Monika, Dr.S.Deivalakshmi and Dr.B.Janet from CSIR-Central Electrochemical Research Institute Karaikudi, India have done another sentiment analysis research using US airline tweets with the aid of LSTM and Global Vectors (GloVe) pre trained word embeddings [14]. They have concluded that the model built had very good performance. F. Miedema has also done a sentiment analysis for IMDP dataset using LSTM and word embeddings and he also stated that he ended up having a pretty good model [15].

Chapter 3

Methodology

1. Introduction

This chapter provides a detailed description of the process which has been carried out while doing this study. Here it describes how the dataset was created, how the model was trained, what are the tools and techniques used, what are the equations derived and used, how the integration was done and how the frontend application was implemented.

1.1 Representation of the Problem

This study is focusing on identifying the sentiments of comments/replies and likes people put to the tweets which were published in the government's official twitter page. By analyzing these sentiments of each comment/reply and likes to the tweets, here the aim is to get an overall idea about how the majority of people reacts to those decisions/actions, whether the majority likes or dislikes them.

In order to address this, a machine learning model was built which is capable of identifying the sentiments of comments and replies with high accuracy. Apart from that twitter API was used to retrieve relevant data which are needed each time when analyzing the tweets via the application developed. After completing the implementation of sentiment analyzer model and the backend, a front-end application was developed which can be used by a government authority to analyze the overall sentimental values of comments, replies and likes they got for the tweets published in their page.

2. Proposed model/design

The proposed application has two options which can be used to analyze the people's reactions towards government's decisions and activities. One option allows the government to analyze how people reacted to one particular decision/action, so then the government can understand whether the majority of people like that particular decision/action or not. The other option allow government to analyze how people reacted to the decisions and actions they took during a particular time period (within this week, this month etc.). By using this option government can understand how well they are doing, whether people majority of people are happy with the government or not at the moment.

The high-level details of the first option in the application can be listed as follows.

1. First the government authority has to give the URL of the tweet published in their page from the front end of the application.
2. From the backend the search endpoint in the twitter API will be called using a query built with necessary search parameters with the twitter id extracted from the URL of the tweet given from the front-end. Due to some limitations in the search query provided by the twitter API, here the replies to a comment will be taken only up to two levels. This will be further explained under section 4.1 – Twitter API integration
3. Pre-process the retrieved comments and replies.
4. Call the model, get sentiment value for each comment/reply and label them accordingly (Positive or negative).
5. Calculate the total number of positive and negative comments/replies. To get the sentiment value of a like for a comment/reply, the sentiment value of the comment/reply will be taken into consideration. One like for a negative comment/reply will be taken as a negative feedback and One like for a positive comment/reply will be taken as a positive feedback. Also likes to the post will be taken as positive feedbacks.
6. Calculate the percentages of positive and negative sentiments collected and pass calculated values to the front-end
7. The calculated percentages will be displayed in a graphical format to the end user

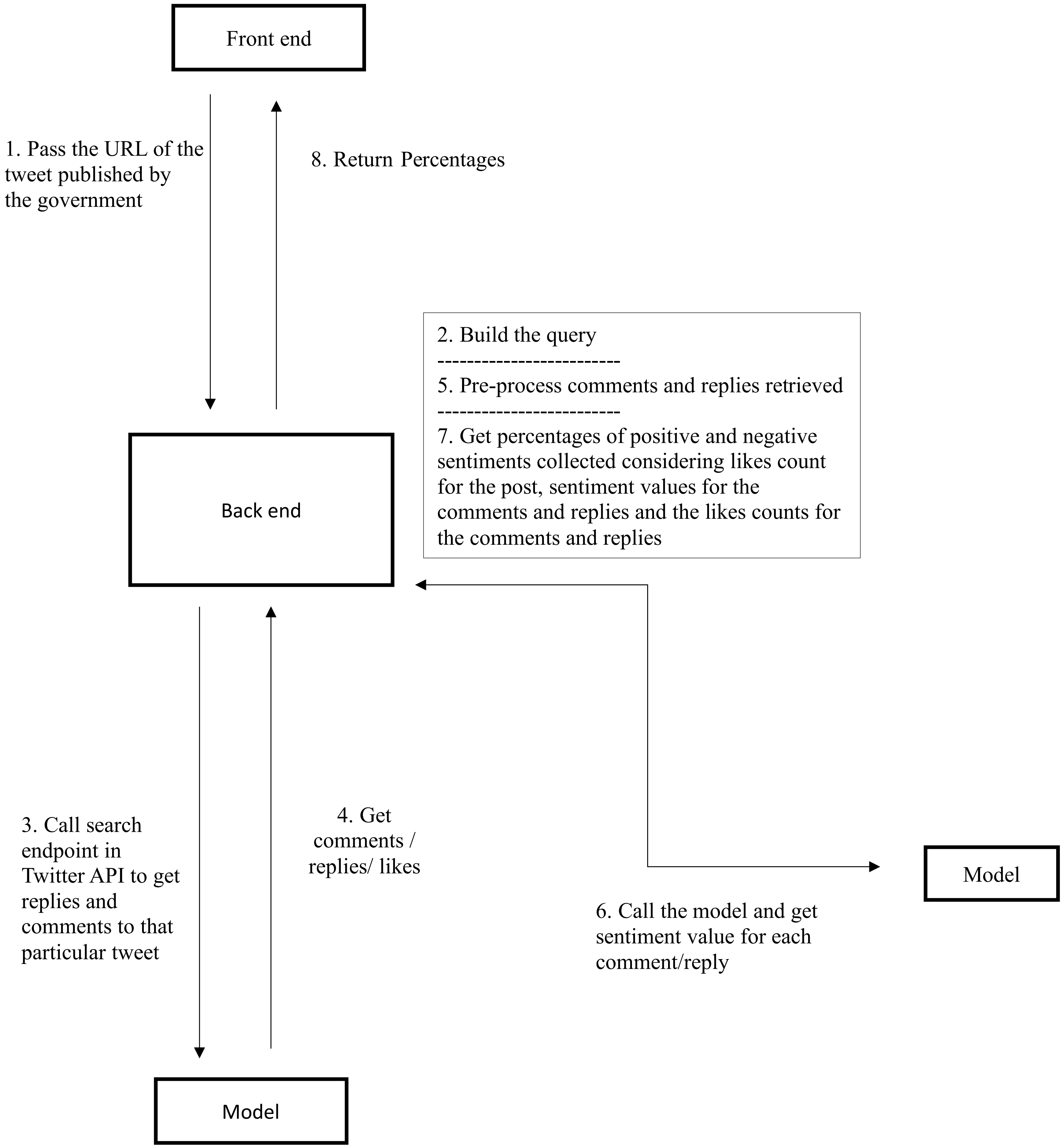

Front end
1. Pass the URL of the tweet published by the government
8. Return Percentages
2. Build the query
----
5. Pre-process comments and replies retrieved
----
7. Get percentages of positive and negative sentiments collected considering likes count for the post, sentiment values for the comments and replies and the likes counts for the comments and replies
Back end
3. Call search endpoint in Twitter API to get replies and comments to that particular tweet
4. Get comments / replies/ likes
Model
6. Call the model and get sentiment value for each comment/reply
Model

Figure 2 – Proposed design for option 1 in the application

The high-level details of the second option in the application can be listed as follows.

1. First the government authority has to select the time period that he wants to analyze people’s reactions, from the front end of the application. Due to some limitations in the twitter API this application supports only to select dates within one month period. This will be further explained under section 4.1 – Twitter API integration

2. From the backend the search endpoint in the twitter API will be called using a query built with necessary search parameters and the date range given from the front-end. Due to some limitations in the search query provided by the twitter API, here only comments will be retrieved and replies to the comments won’t be taken into consideration. This will be further explained under section 4.1 – Twitter API integration

3. Pre-process the retrieved comments for all the posts published by the government during that time period.

4. Call the model, get sentiment value for each comment and label them accordingly (Positive or negative).

8. Calculate the total number of positive and negative comments. To get the sentiment value of a like for a comment, the sentiment value of the comment will be taken into consideration. One like for a negative comment will be taken as a negative feedback and One like for a positive comment will be taken as a positive feedback. Also likes to the posts will be taken as positive feedbacks.

5. Calculate the percentages of positive and negative sentiments collected and pass calculated values to the front-end

6. The calculated percentages will be displayed in a graphical format to the end user

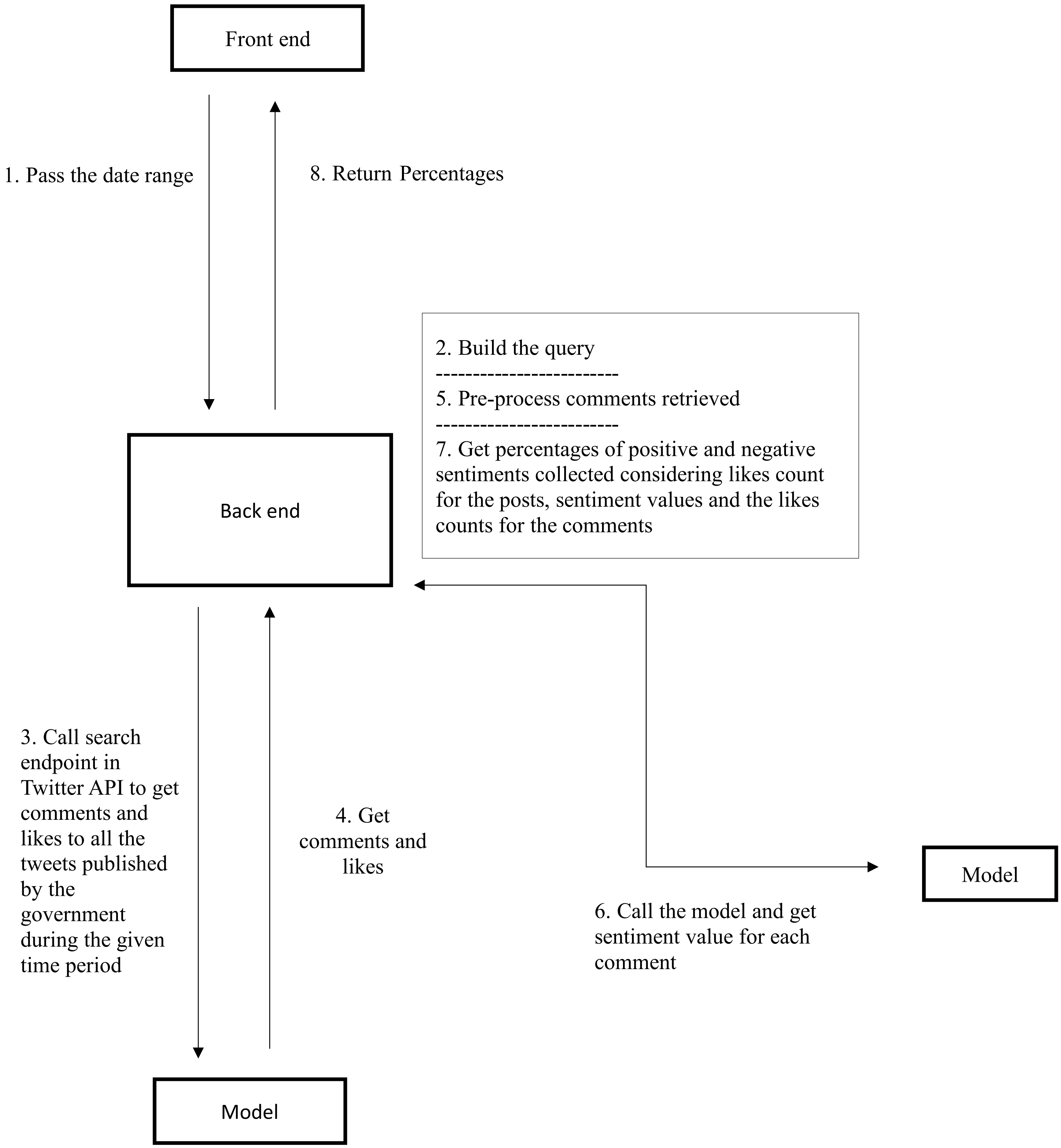

Front end
1. Pass the date range
8. Return Percentages
2. Build the query
----
5. Pre-process comments retrieved
----
7. Get percentages of positive and negative sentiments collected considering likes count for the posts, sentiment values and the likes counts for the comments
Back end
3. Call search endpoint in Twitter API to get comments and likes to all the tweets published by the government during the given time period
4. Get comments and likes
Model
6. Call the model and get sentiment value for each comment
Model

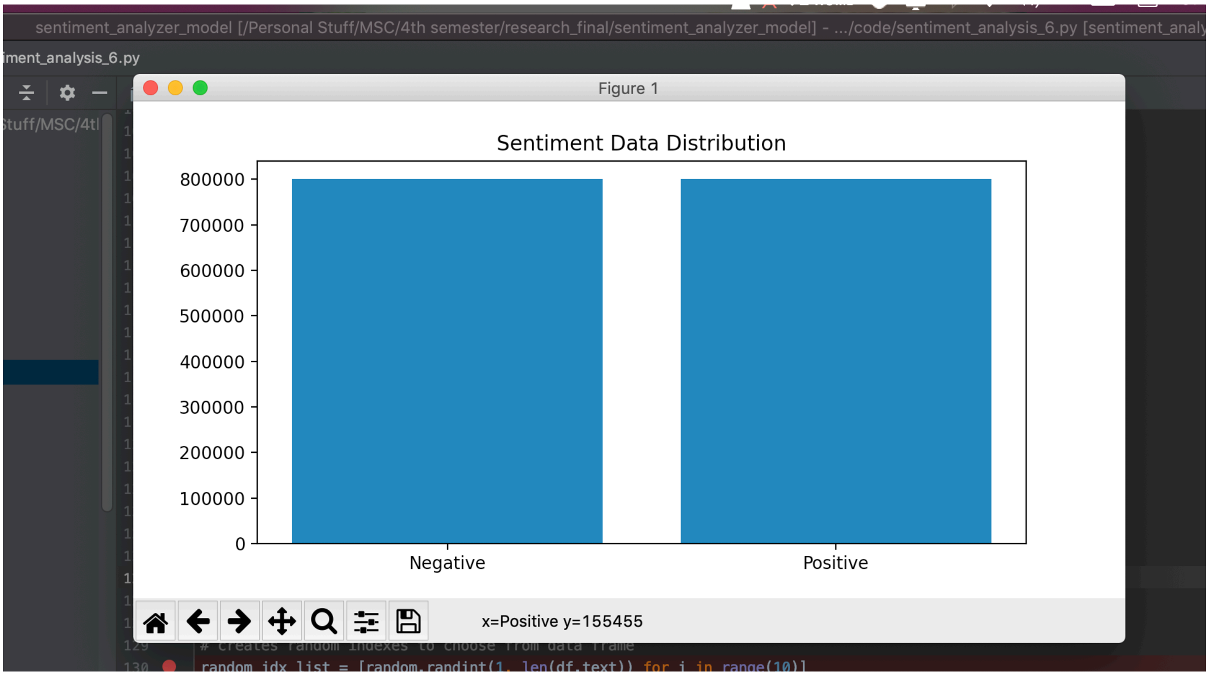
Figure 3 – Proposed design for option 2 in the application

This study assumes, there could be a very low chance of having a neutral comment or a reply for a decision/action published by the government. So, this study only focuses on negative and positive sentiments.

3. Sentiment analysis model

3.1 Dataset

Too little training data results in a poor approximation. To have a more accurate and decent performance, a dataset which has relatively good size should be used. In this study the famous Sentiment 140 dataset which contains 1.6 million tweets was used. The tweets have been labelled as negative: 0, neutral: 2 and positive: 4. Even though they mention about neutral sentiments, actually the dataset does not have any neutral data. Following figure was drawn using matplotlib library while analyzing the dataset and it shows how the sentiments of dataset has been distributed.


sentiment_analyzer_model [/Personal Stuff/MSC/4th semester/research_final/sentiment_analyzer_model] - .../code/sentiment_analysis_6.py [sentiment_analy
iment_analysis_6.py
Figure 1
Sentiment Data Distribution
800000
700000
600000
500000
400000
300000
200000
100000
0
Negative
Positive

x=Positive y=155455

```
# creates random indexes to choose from data frame
random_idx_list = [random.randint(1, len(df.text)) for i in range(10)]
```

Figure 4 – sentiment data distribution in sentiment 140 dataset

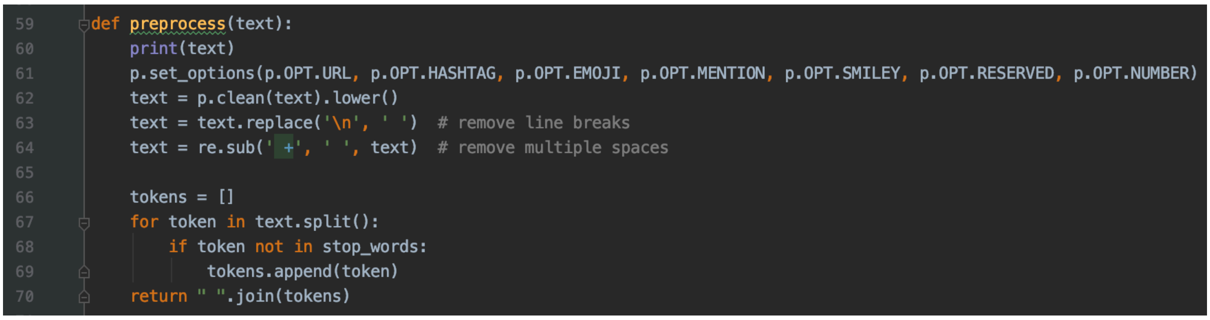
To collect data, here a different, unique approach has been used because the training data was automatically created without getting humans involved in labelling them. In their unique approach they have made an assumption. They have taken any tweet with positive emoticons as positive, and also the tweets with negative emoticons as negative. To generate this dataset, they have used Twitter Search API, keyword search.

3.2 Pre-processing

The data found in Sentiment 140 dataset is polluted with many unnecessary characters, smiles and words therefore preprocessing was needed to be done before using it for training the model.

Preprocessing data collected via twitter needs few more different mechanisms apart from the techniques which use for a standard collection of texts because the twitter data includes hashtags, user mentions, URLs, smiles etc. While doing pre-processing following things were done to clean the dataset.

- URL removal
- Hashtag removal
- Emoji removal
- User mention removal (eg: @userABC)
- Smile removal
- Twitter reserved words removal
- Digits removal
- Convert upper case to lower case
- New lines removal
- Multi space removal
- Stop word removal



```
def preprocess(text):
    print(text)
    p.set_options(p.OPT.URL, p.OPT.HASHTAG, p.OPT.EMOJI, p.OPT.MENTION, p.OPT.SMILEY, p.OPT.RESERVED, p.OPT.NUMBER)
    text = p.clean(text).lower()
    text = text.replace('\n', ' ')  # remove line breaks
    text = re.sub(' +', ' ', text)  # remove multiple spaces

    tokens = []
    for token in text.split():
        if token not in stop_words:
            tokens.append(token)
    return " ".join(tokens)
```

Figure 5 - Code snippet used for pre-processing task

3.3 Techniques

Word Embeddings

For Natural Language Processing (NLP) tasks word embeddings are very useful and therefore use commonly for better performance. They represent words in a manner where the similar words have similar encodings. This converts words into vectors which can be fed to the model. This is a great solution to handle the problems associate with one-hot encodings [14][15].

For this study famous GloVe pre-trained word embeddings have been used which contains word vectors created from twitter corpora and it has 2B tweets, 27B tokens and 200d vectors.

LSTM (Long Short-Term Memory)

These are designed to avoid the problem of vanishing gradient in RNN s. It uses gated cells to store information. Using them it decides what information should be carry further and what should be forgotten. After the whole run all the required information that has huge impact on the output will be preserved so the accuracy of this is higher compared to Recurrent Neural Networks when there are long term dependencies [6][7].

LSTM is good to build models for sentiment analysis because of this ability to remember past, historical data. So, in this study LSTM was used to build the model.

3.4 Libraries

Pandas is a python data analysis library. In this study it was used while building the model, to deal with the data read from the csv files. For pre-processing task tweet- Preprocessor and nltk libraries were used. tweet- Preprocessor library supported removal of URLs, hashtags, emoji, smiley, mentions, reserved words and digits. nltk library supported removal of stop words. Keras is a famous library for deep learning tasks which is very easy to use. In this study Keras has been used to tokenize texts and to develop, train and evaluate the model. Matplotlib library was used to plot some graphs to get deep insight about the dataset and to plot evaluation results of the model developed.

3.5 Evaluation

Followings are the metrics which were used to measure how accurate the model works.

- Accuracy: This value shows the percentage of instances predicted with the correct sentiment label
- Precision: This value shows the percentage of instances the model correctly predicted out of total number of texts that model predicted for a given sentiment label.
- Recall: This value shows the percentage of instances the model predicted for a sentiment label out of the total number of instances it should have predicted for that given sentiment label.
- F1 Score: This value shows the harmonic mean of recall and precision which can be convey in other words as the weighted average of recall and precision.

The evaluation plan and the results are further explained under Chapter 4 - Results and Evaluation section.

4. Backend

4.1 Twitter API integration

Tweepy is an easy to use python library developed to access Twitter API. For this study it was used to handle calls to the Twitter API.

As mentioned in above section 2 - Proposing model/design, the application developed in this study supports government to analyze people's reactions to their decisions and actions in two different ways.

Option 1 allows government to analyze people's reactions towards a particular post (which describes an action or decision taken by the government and published in the government official twitter page), so that the government can get an idea whether the majority of people in the country accept it or not.

Option 2 allows government to analyze people's reactions towards the posts (which describe actions or decisions taken by the government and published in the government official twitter page) which were published during a particular time period (within this week, last two weeks etc), so the government can understand how good the government is doing during that time period. Eg: The government can analyze whether the people are happy with the decisions and actions they took during this month etc.

▪ Option 1 - Analyze people's reactions towards a particular post

Here twitter API need to be accessed to retrieve comments, replies and likes relevant to the particular post published by the government. The end user of the application should give the tweet URL from the front-end. In the backend the id of the tweet will be extracted from the URL given and will call the twitter search endpoint using a query built. The twitter API still does not provide an endpoint to directly take comments and likes to a post, so have to use the available options in the search endpoint and this method is bit complex.

There are few things need to be understood before dig-in how this study uses twitter API to collect comments and replies to a post.

- According to the twitter architecture when a person comments to a tweet, it automatically tags the user who posted the original tweet. Likewise, if person replies to a comment it tags the user who has put the original comment.
- Twitter takes comments and replies also as tweets hence they have tweet ids. Comments and replies get posted after the original post, so they have ids greater than the original post id.

The query use here calls the search endpoint with following parameters.

- to: Here need to pass the username of the account who posted the tweet. In this study the value which need to be passed here is username of the government's twitter page and that value is hard coded inside the application. By using this parameter can retrieve tweets(posts/comments/replies) which have tagged the user specified here.
- since_id: The id of the tweet which was published by the government. When passing 'to' parameter along with this since_id parameter, the API returns posts/comments/replies which have been tagged the user specified in 'to' parameter and has the id greater than or equal to the tweet id pass with this 'since_id' (means posted after this particular post)
- lang: Retrieve tweets (comments and replies) written in specified language. Here the value is 'en' because our model is built to analyze only texts written in English. This value is hard coded inside the application.

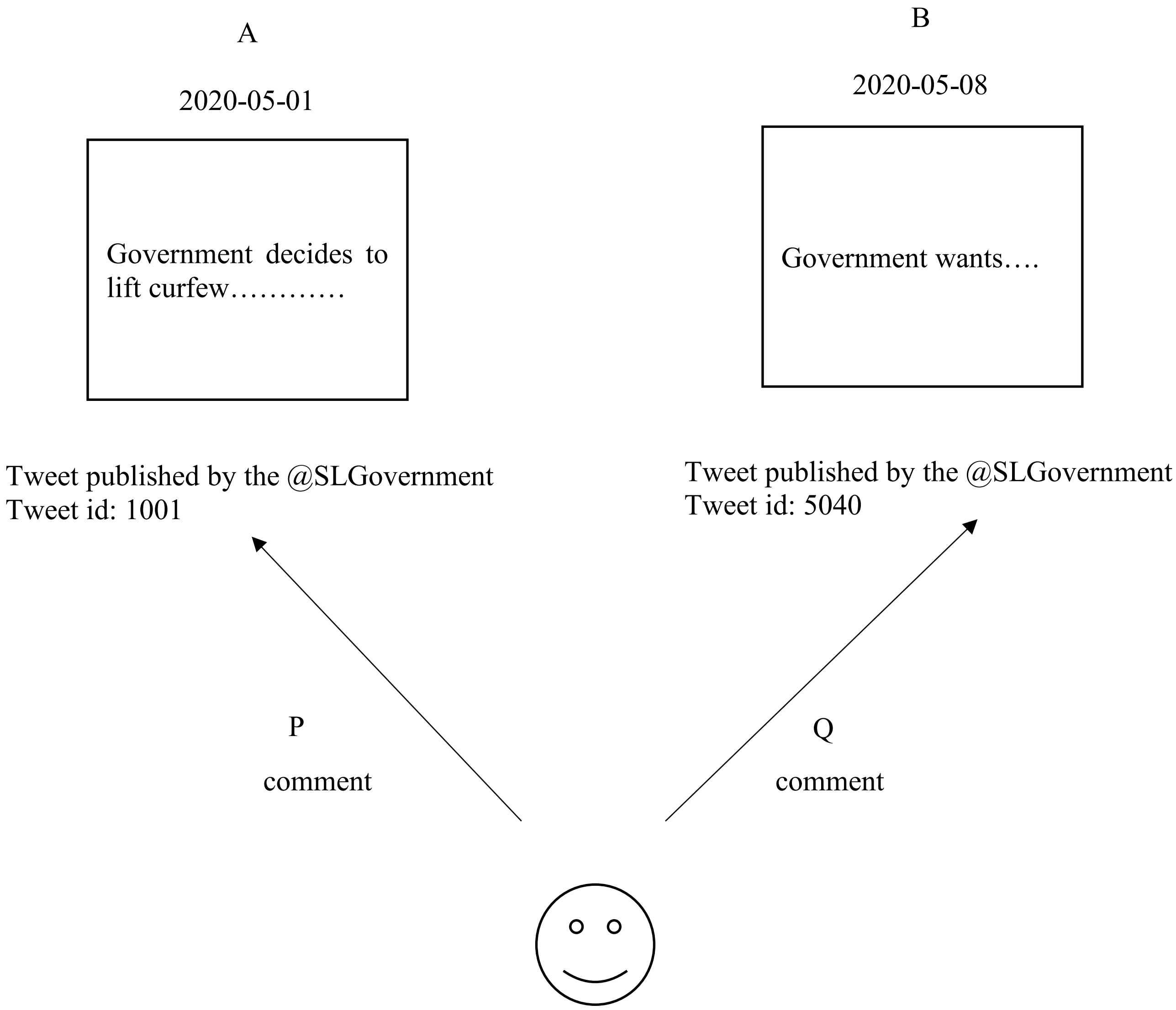

A
2020-05-01
Government decides to lift curfew…………
Tweet published by the @SLGovernment
Tweet id: 1001
B
2020-05-08
Government wants….
Tweet published by the @SLGovernment
Tweet id: 5040
P
comment
Q
comment

Figure 6 – search with to and since_id parameters

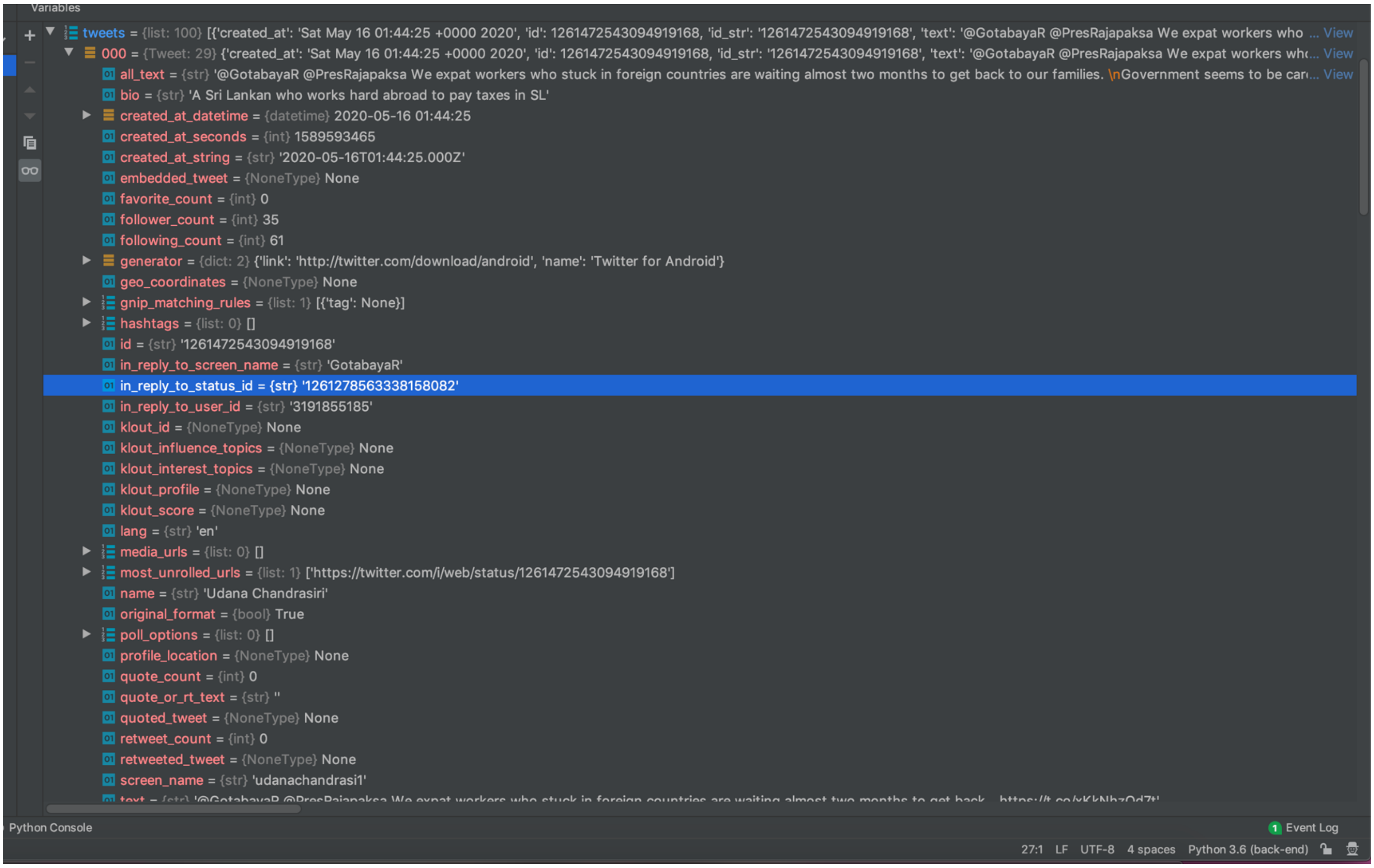
If query sends with to: SLGovernment and since_id 1001, then the result set includes comments P and Q

If query sends with to: SLGovernment and since_id 5040, then the result set includes only comment Q.

So, when calling the twitter search API with above parameters the result may include comments relevant to the tweet the government wants to analyze plus comments/replies/posts which tagged government and have published after the tweet the government wants to analyze

(tweet id greater than the tweet the government wants to analyze). So, from the result set only the comments relevant to the tweet which need to be analyzed should be filtered-out.

The each retrieved object in the result array has following properties



```
Variables
tweets = {list: 100} [{'created_at': 'Sat May 16 01:44:25 +0000 2020', 'id': 1261472543094919168, 'id_str': '1261472543094919168', 'text': '@GotabayaR @PresRajapaksa We expat workers who ... View
  000 = {Tweet: 29} {'created_at': 'Sat May 16 01:44:25 +0000 2020', 'id': 1261472543094919168, 'id_str': '1261472543094919168', 'text': '@GotabayaR @PresRajapaksa We expat workers wh... View
    all_text = {str} '@GotabayaR @PresRajapaksa We expat workers who stuck in foreign countries are waiting almost two months to get back to our families. \nGovernment seems to be car... View
    bio = {str} 'A Sri Lankan who works hard abroad to pay taxes in SL'
    created_at_datetime = {datetime} 2020-05-16 01:44:25
    created_at_seconds = {int} 1589593465
    created_at_string = {str} '2020-05-16T01:44:25.000Z'
    embedded_tweet = {NoneType} None
    favorite_count = {int} 0
    follower_count = {int} 35
    following_count = {int} 61
    generator = {dict: 2} {'link': 'http://twitter.com/download/android', 'name': 'Twitter for Android'}
    geo_coordinates = {NoneType} None
    gnip_matching_rules = {list: 1} [{'tag': None}]
    hashtags = {list: 0} []
    id = {str} '1261472543094919168'
    in_reply_to_screen_name = {str} 'GotabayaR'
    in_reply_to_status_id = {str} '1261278563338158082'
    in_reply_to_user_id = {str} '3191855185'
    klout_id = {NoneType} None
    klout_influence_topics = {NoneType} None
    klout_interest_topics = {NoneType} None
    klout_profile = {NoneType} None
    klout_score = {NoneType} None
    lang = {str} 'en'
    media_urls = {list: 0} []
    most_unrolled_urls = {list: 1} ['https://twitter.com/i/web/status/1261472543094919168']
    name = {str} 'Udana Chandrasiri'
    original_format = {bool} True
    poll_options = {list: 0} []
    profile_location = {NoneType} None
    quote_count = {int} 0
    quote_or_rt_text = {str} ''
    quoted_tweet = {NoneType} None
    retweet_count = {int} 0
    retweeted_tweet = {NoneType} None
    screen_name = {str} 'udanachandrasi1'
    text = {str} '@GotabayaR @PresRajapaksa We expat workers who stuck in foreign countries are waiting almost two months to get back… https://t.co/xKkNhzOd7t'
```

Figure 7 – Tweet response object

The property highlighted above gives the actual tweet id, to which this has been put as a comment. So, need to iterate over all the objects retrieved and filter-out only the objects which have ‘in_reply_to_status_id’ equal to the id of the tweet posted by the government which need to be analyzed.

In this option one in the application, when analyzing people’s reactions to a post, the aim is not only to consider the comments to the post but also the replies to each comment up to two levels.

To retrieve the first level replies, the same process needs to be done, same search endpoint should be called for each comment with respective values to the 'to' and 'since_id' parameters. Each time 'to' parameter value should be the username of the person who put the comment, 'since_id' should be the id of the comment. To retrieve the second level replies, 'to' parameter value should be the username of the person who put the first level reply, 'since_id' should be the id of the first level reply. In these both two cases also same filtering process need to be carried out using 'in_reply_to_status_id' property available in the resulting objects.

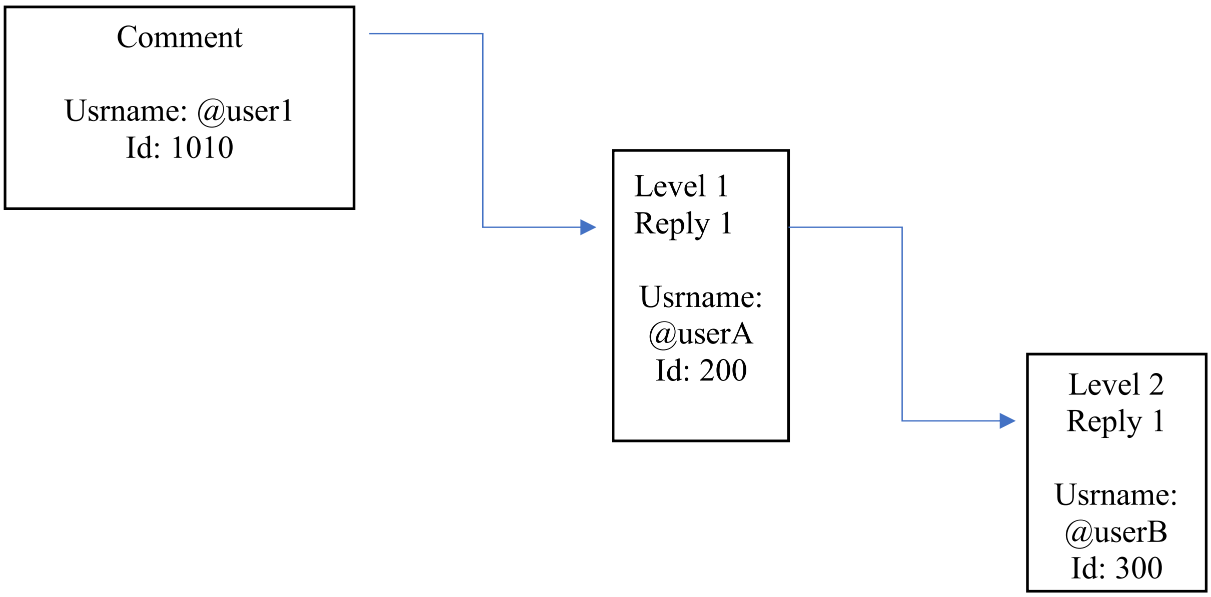
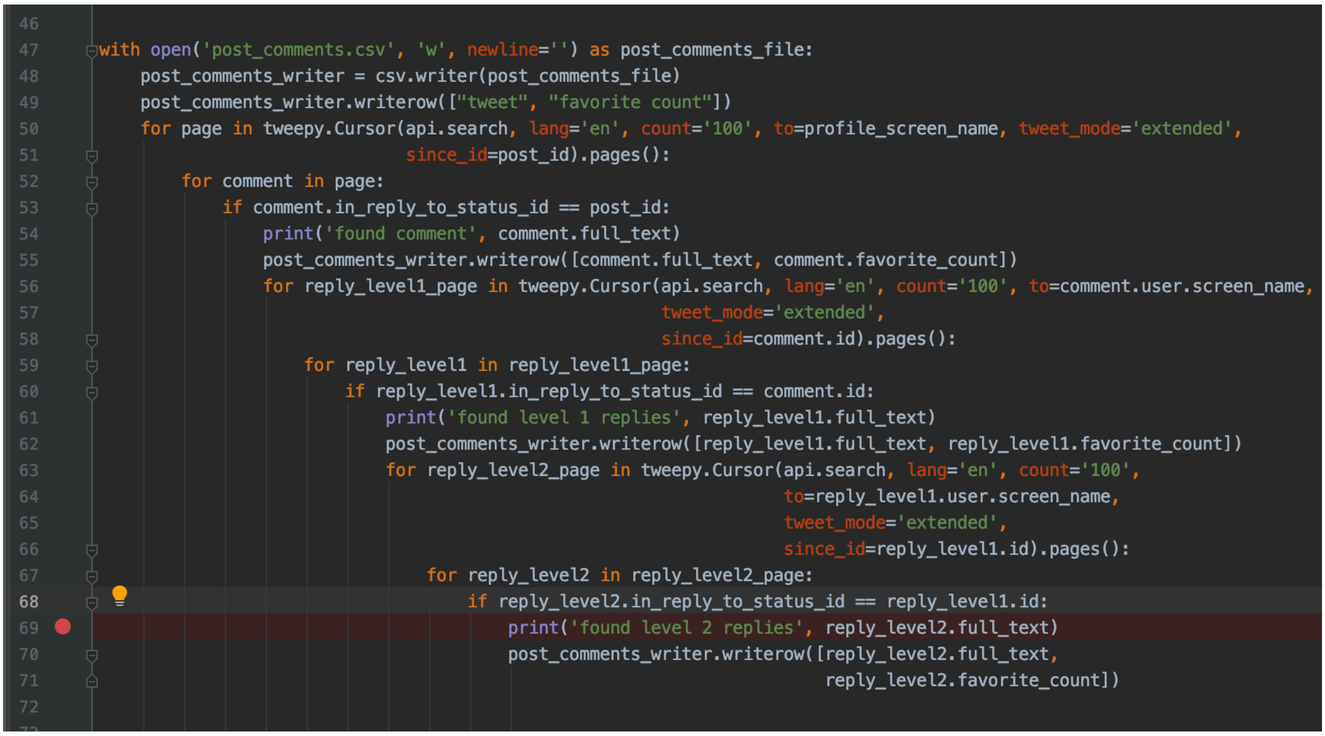


Figure 8 – Twitter comments and replies

To get reply 1 (first level replies to a comment) 'to' parameter value should be user1 and 'since_id' parameter value should be 1010. To get reply 2 (second level replies to a comment) 'to' parameter value should be userA and 'since_id' parameter value should be 200.

The code snippet for extracting comments and replies up to two levels and writing them in to a file along with favorite counts is as below.



```
with open('post_comments.csv', 'w', newline='') as post_comments_file:
    post_comments_writer = csv.writer(post_comments_file)
    post_comments_writer.writerow(["tweet", "favorite count"])
    for page in tweepy.Cursor(api.search, lang='en', count='100', to=profile_screen_name, tweet_mode='extended',
                              since_id=post_id).pages():
        for comment in page:
            if comment.in_reply_to_status_id == post_id:
                print('found comment', comment.full_text)
                post_comments_writer.writerow([comment.full_text, comment.favorite_count])
                for reply_level1_page in tweepy.Cursor(api.search, lang='en', count='100', to=comment.user.screen_name,
                                                       tweet_mode='extended',
                                                       since_id=comment.id).pages():
                    for reply_level1 in reply_level1_page:
                        if reply_level1.in_reply_to_status_id == comment.id:
                            print('found level 1 replies', reply_level1.full_text)
                            post_comments_writer.writerow([reply_level1.full_text, reply_level1.favorite_count])
                            for reply_level2_page in tweepy.Cursor(api.search, lang='en', count='100',
                                                                   to=reply_level1.user.screen_name,
                                                                   tweet_mode='extended',
                                                                   since_id=reply_level1.id).pages():
                                for reply_level2 in reply_level2_page:
                                    if reply_level2.in_reply_to_status_id == reply_level1.id:
                                        print('found level 2 replies', reply_level2.full_text)
                                        post_comments_writer.writerow([reply_level2.full_text,
                                                                       reply_level2.favorite_count])
```

Figure 9 – code snippet for extracting comments, replies and favorite counts for a post

In each resulting object there's a property called 'favourite_count' (can see in Figure 7), so need to track this as well because in this study each like (which call as a 'favorite' in twitter) to a negative comment/reply will be taking as a negative reaction and each like to a positive comment/reply will be taking as a positive reaction.

- Option 2 - Analyze people's reactions towards the posts which have been published during a particular time period

The twitter API has a separate endpoint to retrieve tweets published by a specific user but still it does not support to filter the result using dates. The result would be huge to manually iterate and filter by dates hence it is not practical. So, finding a method to do this was a challenging one. After analyzing the available options and endpoints provided by the twitter API finally a method found and there need to call the search endpoint with 'from', 'since' and 'until' parameters.

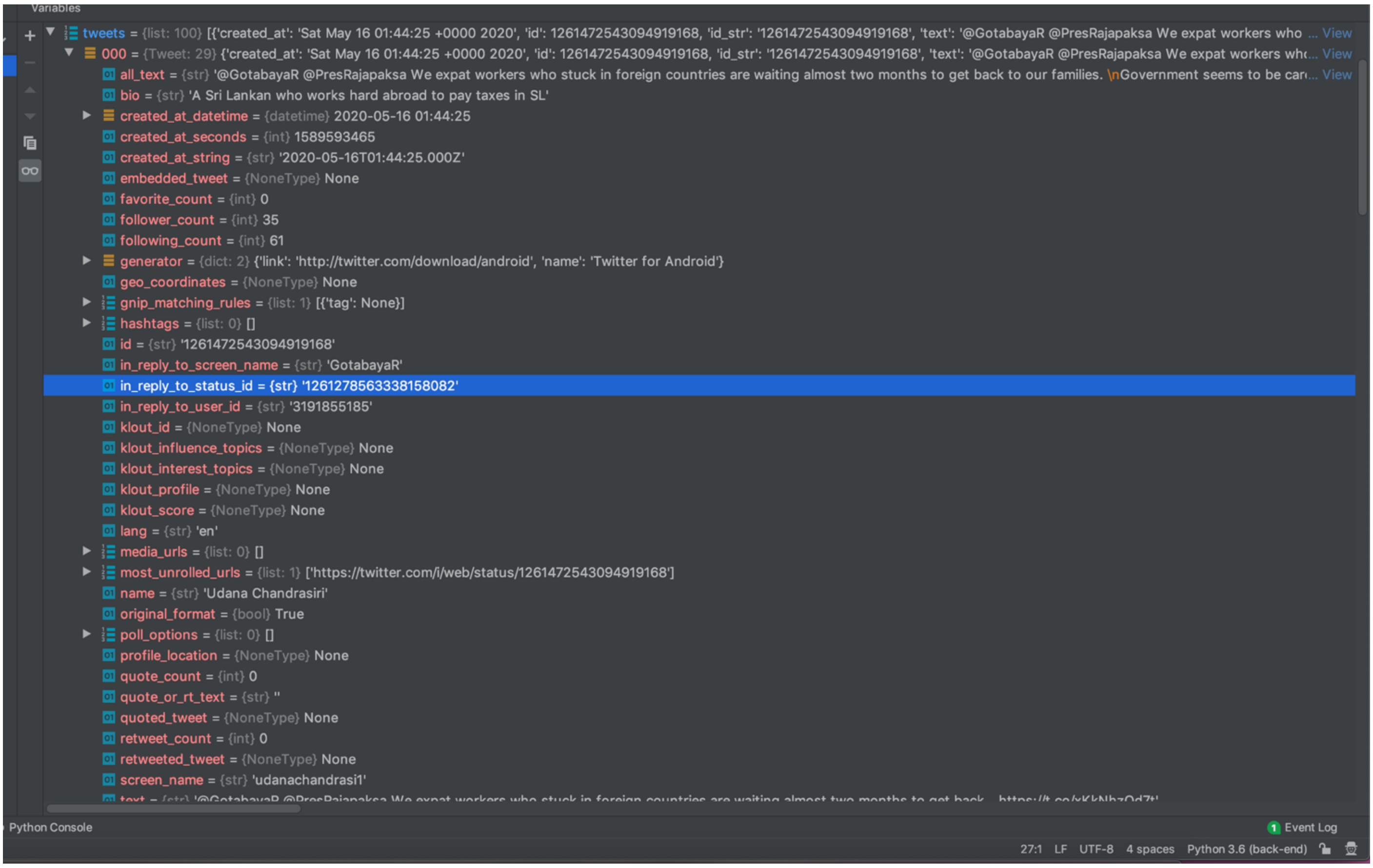
- from: Here need to pass the username of the account who posted the tweets that need retrieve and analyze. In this study the value which need to be passed here is username of the government's twitter page and that value is hard coded inside the application. By using this parameter can retrieve tweets which were published by the user specified here.

- since: The starting date of the time period should be passed here

- until: The ending date of the time period should be passed here

By using the search query with the above parameters, can retrieve posts/comments/replies/retweets posted by the owner of the account (specified by from parameter) within that given time period. After retrieving results, the next challenge was to filter out only posts and discard the comments, replies and retweets from the result set.

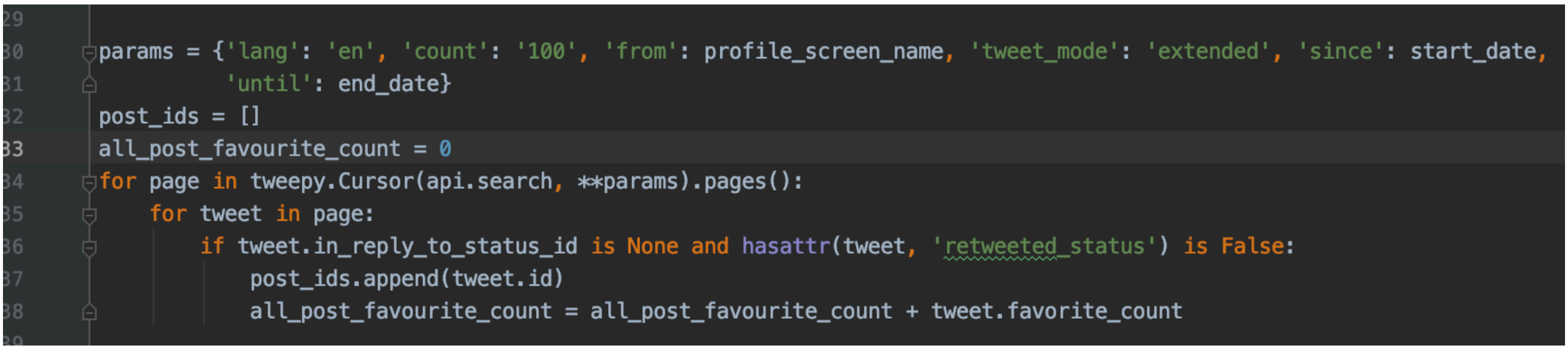
Each resulting object has following structure.



```
Variables
▼ tweets = {list: 100} [{'created_at': 'Sat May 16 01:44:25 +0000 2020', 'id': 1261472543094919168, 'id_str': '1261472543094919168', 'text': '@GotabayaR @PresRajapaksa We expat workers who ... View
  ▼ 000 = {Tweet: 29} {'created_at': 'Sat May 16 01:44:25 +0000 2020', 'id': 1261472543094919168, 'id_str': '1261472543094919168', 'text': '@GotabayaR @PresRajapaksa We expat workers wh... View
      all_text = {str} '@GotabayaR @PresRajapaksa We expat workers who stuck in foreign countries are waiting almost two months to get back to our families. \nGovernment seems to be car... View
      bio = {str} 'A Sri Lankan who works hard abroad to pay taxes in SL'
    ▶ created_at_datetime = {datetime} 2020-05-16 01:44:25
      created_at_seconds = {int} 1589593465
      created_at_string = {str} '2020-05-16T01:44:25.000Z'
      embedded_tweet = {NoneType} None
      favorite_count = {int} 0
      follower_count = {int} 35
      following_count = {int} 61
    ▶ generator = {dict: 2} {'link': 'http://twitter.com/download/android', 'name': 'Twitter for Android'}
      geo_coordinates = {NoneType} None
    ▶ gnip_matching_rules = {list: 1} [{'tag': None}]
    ▶ hashtags = {list: 0} []
      id = {str} '1261472543094919168'
      in_reply_to_screen_name = {str} 'GotabayaR'
      in_reply_to_status_id = {str} '1261278563338158082'
      in_reply_to_user_id = {str} '3191855185'
      klout_id = {NoneType} None
      klout_influence_topics = {NoneType} None
      klout_interest_topics = {NoneType} None
      klout_profile = {NoneType} None
      klout_score = {NoneType} None
      lang = {str} 'en'
    ▶ media_urls = {list: 0} []
    ▶ most_unrolled_urls = {list: 1} ['https://twitter.com/i/web/status/1261472543094919168']
      name = {str} 'Udana Chandrasiri'
      original_format = {bool} True
    ▶ poll_options = {list: 0} []
      profile_location = {NoneType} None
      quote_count = {int} 0
      quote_or_rt_text = {str} ''
      quoted_tweet = {NoneType} None
      retweet_count = {int} 0
      retweeted_tweet = {NoneType} None
      screen_name = {str} 'udanachandrasi1'
      text = {str} '@GotabayaR @PresRajapaksa We expat workers who stuck in foreign countries are waiting almost two months to get back… https://t.co/xKkNhzOd7t'
Python Console                                                                                                  1 Event Log
                                                                          27:1  LF  UTF-8  4 spaces  Python 3.6 (back-end)
```

Figure 10 - Tweet response object

The highlighted property in the figure above, ‘in_repply_to_status_id’ has a value only if the resulting object is a comment or a reply. So, by manually filtering only the result objects which have this ‘in_reply_to_status_id’ value none, can have only the posts published by the user. To remove retweets, need to check whether the result object has ‘retweeted_status’ attribute and need to remove that particular result object from the result set if it has. The code snippet for above filtering part is as below. Here as the final result we extract ids of the posts that user has published during that period and also the total number of favorite counts for those posts.

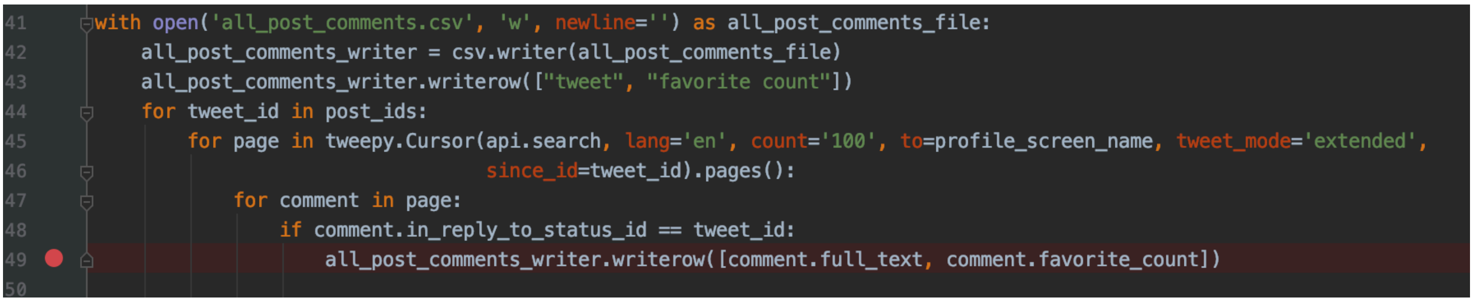


```
params = {'lang': 'en', 'count': '100', 'from': profile_screen_name, 'tweet_mode': 'extended', 'since': start_date,
          'until': end_date}
post_ids = []
all_post_favourite_count = 0
for page in tweepy.Cursor(api.search, **params).pages():
    for tweet in page:
        if tweet.in_reply_to_status_id is None and hasattr(tweet, 'retweeted_status') is False:
            post_ids.append(tweet.id)
            all_post_favourite_count = all_post_favourite_count + tweet.favorite_count
```

Figure 11 – get all posts published by the government during the given time period

The rest of the implementation of this option is almost similar to the implementation of first option but here it is difficult to consider replies because there could be a huge number of posts published by the government during that period and with the limitations in the twitter API there’s no way to directly retrieve replies as described in the option 1 above. So here only comments for the posts will be taken into consideration.

The code snippet for extracting comments and writing them in to a file along with favorite counts is as below.



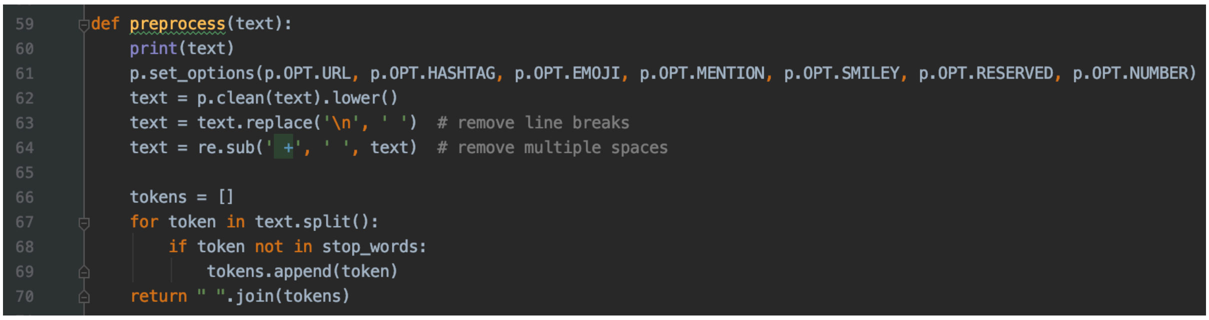
```python
with open('all_post_comments.csv', 'w', newline='') as all_post_comments_file:
    all_post_comments_writer = csv.writer(all_post_comments_file)
    all_post_comments_writer.writerow(["tweet", "favorite count"])
    for tweet_id in post_ids:
        for page in tweepy.Cursor(api.search, lang='en', count='100', to=profile_screen_name, tweet_mode='extended',
                                  since_id=tweet_id).pages():
            for comment in page:
                if comment.in_reply_to_status_id == tweet_id:
                    all_post_comments_writer.writerow([comment.full_text, comment.favorite_count])
```

Figure 12 – extract comments and favorite counts for all the posts published by the government during the given time period

4.2 Model integration

After retrieving required data from the twitter API, need to pre-process them before passing them to the model to take the sentiment values. Here the pre-processing process includes

- URL removal
- Hashtag removal
- Emoji removal
- User mention removal (eg: @userABC)
- Smile removal
- Twitter reserved words removal
- Digits removal
- Convert upper case to lower case
- New lines removal
- Multi space removal
- Stop word removal



```python
def preprocess(text):
    print(text)
    p.set_options(p.OPT.URL, p.OPT.HASHTAG, p.OPT.EMOJI, p.OPT.MENTION, p.OPT.SMILEY, p.OPT.RESERVED, p.OPT.NUMBER)
    text = p.clean(text).lower()
    text = text.replace('\n', ' ')  # remove line breaks
    text = re.sub(' +', ' ', text)  # remove multiple spaces

    tokens = []
    for token in text.split():
        if token not in stop_words:
            tokens.append(token)
    return " ".join(tokens)
```

Figure 13 - Code snippet used for pre-processing task

After cleaning the data each text will be analyzed through the model built, and a sentiment value will be calculated by the model. Here if the sentiment score is <= 0.5 it will be taken as a negative sentiment and if the score is > 0.5 it will be taken as a positive sentiment.

4.3 Equations to take the overall sentiment values

- Followings are the equations use in option 1 - Analyze people's reactions towards a particular post. Here we consider comments and replies up to two levels

$$TP = P_C + P_R + \Sigma L_{PC} + \Sigma L_{PR} + L_P$$

Equation 1: Total positive sentiment count for option 1

TP – Total of positive sentiment count

P_c – Total number of positive comments count

P_R – Total number of positive replies count

ΣL_{PC} – Summation of number of likes to each positive comment

ΣL_{PR} – Summation of number of likes to each positive reply

L_P – Number of likes to the post

$$TN = N_C + N_R + \Sigma L_{NC} + \Sigma L_{NR}$$

Equation 2: Total negative sentiment count for option 1

TN - Total of negative sentiment count

N_{c} – Total number of negative comments count

N_{R} – Total number of negative replies count

ΣL_{NC} – Summation of number of likes to each negative comment

ΣL_{NR} – Summation of number of likes to each negative reply

$$\text{Per } TP = TP/T * 100\%$$

Equation 3: Total positive sentiment percentage for option 1

TP - Total positive sentiment count

T – Total number of comments + Total number of replies + Total number of likes to comments, replies and post

$$\text{Per } TN = TN/T * 100\%$$

Equation 4: Total negative sentiment percentage for option 1

TN - Total negative sentiment count

T – Total number of comments + Total number of replies + Total number of likes to comments, replies and post

- Followings are the equations use in option 2 - Analyze people's reactions towards the posts which were published during a particular time period. Here we do not consider replies to comments

$$TP = P_C + \Sigma\ L_{PC} + L_P$$

Equation 5: Total positive sentiment count for option 2

TP – Total of positive sentiment count

P_{c} – Total number of positive comments count in all the posts published within the given time period

$\Sigma\ L_{PC}$ – Summation of number of likes to each positive comment in all the posts published within the given time period

L_P – Total number of likes to all the posts

$$TN = N_C + \Sigma\ L_{NC}$$

Equation 6: Total negative sentiment count for option 2

TN - Total of negative sentiment count

N_{c} – Total number of negative comments count in all the posts published within the given time period

$\Sigma\ L_{NC}$ – Summation of number of likes to each negative comment in all the posts published within the given time period

$$Per\ TP = TP/T * 100\%$$

Equation 7: Total positive sentiment count for option 2

TP - Total positive sentiment count

T – Total number of comments + Total number of likes to comments and posts published within the given time period

Per TN = TN/T * 100%

Equation 8: Total negative sentiment count for option 2

TN - Total negative sentiment count

T – Total number of comments + Total number of likes to comments and posts published within the given time period

5. Frontend

Front end application was implemented using react language.

- option 1 - Analyze people's reactions towards a particular post. Here we consider comments and replies up to two levels

From the front end the end user – the government authority can give the URL of the tweet they want to analyze which was published in the official government twitter page. As the output he can see a chart which show the total percentages of positive and negative sentiment values calculated by considering the comments, replies and likes relevant to that particular tweet.

- option 2 - Analyze people's reactions towards the posts which were published during a particular time period. Here we do not consider replies to posts.

From the front end the end user – the government authority can give the time period they want to analyze people's reactions. As the output he can see a chart which show the total percentages of positive and negative sentiment values calculated by considering the comments and likes relevant to all the tweets published by them during that given time period.

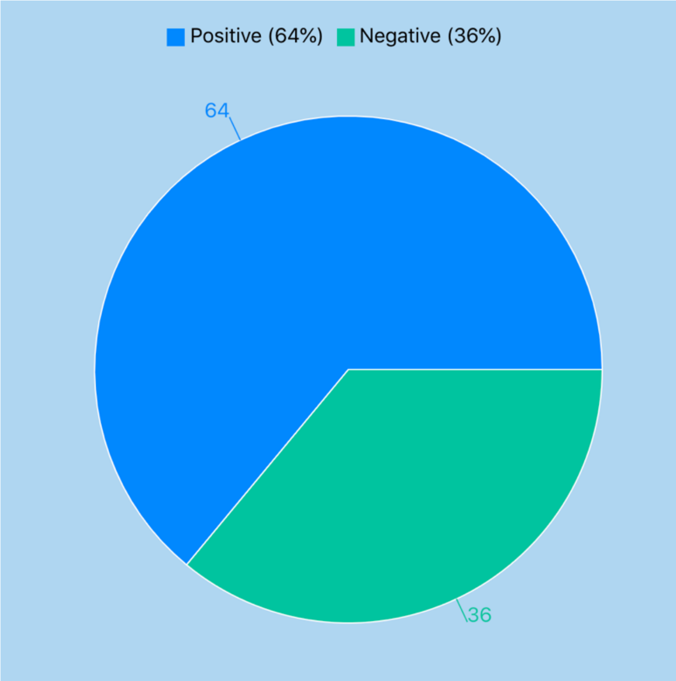


Figure 14 – Final sentiment percentage values

Chapter 4

Results and Evaluation

1. Introduction

This research focuses on building an application that can perform a sentiment analysis on comments, replies and likes that people have put to the tweets published in their government's official twitter page. In this research, a machine learning model which uses LSTM and word embeddings techniques was built and trained to classify sentiments of tweets. By using it, the sentiment of each collected comment/reply will be measured, and the sentiment of each collected comment/reply will be used when deciding the overall sentiment percentages of the tweets collected. The evaluation of this system involves in evaluating the model to decide how accurate it predicts.

2. Evaluation approach

The source of the data, the cleanliness of the language, how it is scored, the subject matter and the amount of the data tested are all important variables that can have a drastic impact on the results. To have a classifier with more accurate results, the data source should closely match the intended uses and the volume of data is very important. This application deals with twitter data so to train the model, famous sentiment140 dataset (http://help.sentiment140.com/for-students) used. It contains 1.6 million tweets. Those tweets are annotated with sentiment labels negative: 0, neutral: 2, positive: 4. Their approach was special because training data was generated automatically, rather than human manual annotated tweets. In their approach, they have taken that any tweet with positive emoticons as positive, and tweets with negative emoticons as negative. To collect these huge set of tweets they have used search endpoint available in Twitter API.

Word embeddings were used to build this model. Pre-trained word embeddings are so popular because they represent words in a very powerful way preserving the semantic and syntactic meanings. So, to increase the accuracy of prediction pre trained word embeddings set called

Stanford’s GloVe Pretrained Word Embeddings, generated with Twitter data which has 2B tweets, 27B tokens and 200d vectors used here.

Apart from above mentioned, evaluation plan for a sentiment classifier model is really important to decide how accurate it predicts. There are metrics that can be used to evaluate models such as accuracy, precision, recall etc.

- Accuracy

Accuracy of a classification model will be calculated as follows:

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative}$$

Equation 9: Accuracy

Accuracy is not a good metric to evaluate the performance of a model for imbalanced classification problems. Therefore, metrics like precision and recall should be also used to evaluate the model.

- Precision

Precision quantifies the number of positive class predictions that actually belong to the positive class. A measure of how often a sentiment rating was correct. Maximizing precision will minimize the number of false positives. Appropriate when minimizing false positives is the focus.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Equation 10: Precision

When calculating precision for the model it highlights, of all rows it predicted to be a certain class, how many did it correctly predict?

- Recall

Recall is the opposite of precision, it measures false negatives against true positives. maximizing the recall will minimize the number of false negatives. Appropriate when minimizing false negatives is the focus.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Equation 11: Recall

When calculating recall for the model it highlights, of all rows of a certain class, how many did it correctly predict?

- F1 Score

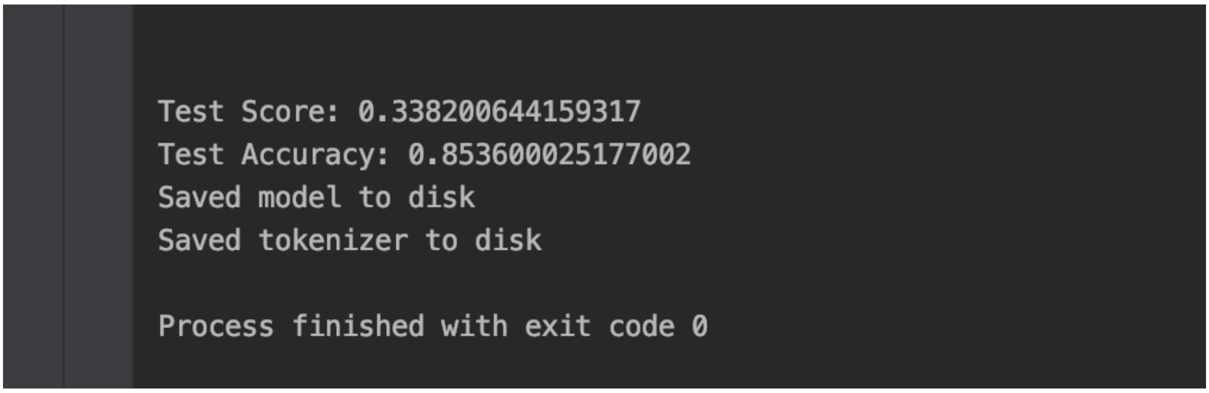
F1 Score is the weighted average of Precision and Recall. This is the harmonic mean of the two fractions. Therefore, this score takes both false positives and false negatives into account. F1 is usually more useful than accuracy, especially when there is an uneven class distribution.

$$F1 - Score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

Equation 12: F1-score

3. Results and discussion

The model was evaluated on test data using evaluate method available in keras library and the results are as follows.



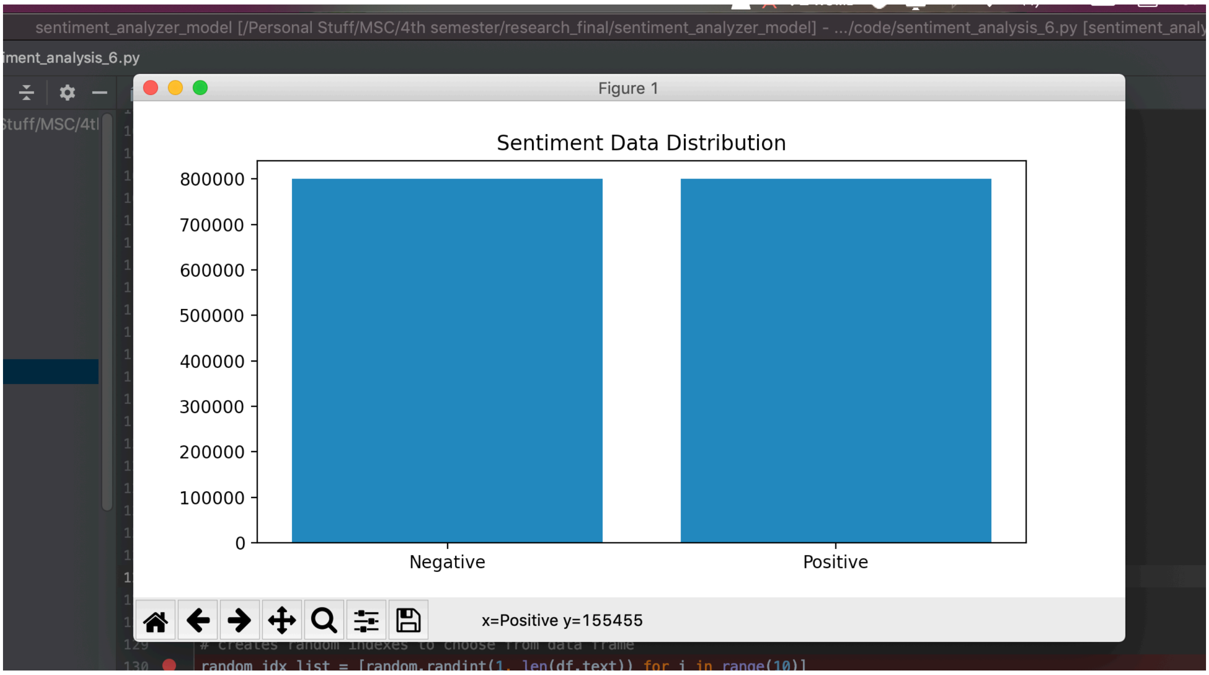
```
Test Score: 0.338200644159317
Test Accuracy: 0.853600025177002
Saved model to disk
Saved tokenizer to disk

Process finished with exit code 0
```

Figure 15 – Test score and test accuracy of the model

The test score of the model is 0.33 which is a very low value and it means the model has lower prediction error.

The accuracy of the model is above 85% which means the model behaves well in predicting sentiments. As mentioned in the above section 2 - Evaluation approach, the accuracy is not a good metric to evaluate the model for imbalanced classification problems which means if the dataset is biased the accuracy of the model does not exactly tell how accurate the model works. The distribution of the data in sentiment 140 dataset was plot while analyzing the dataset and it has following distribution.



sentiment_analyzer_model [/Personal Stuff/MSC/4th semester/research_final/sentiment_analyzer_model] - .../code/sentiment_analysis_6.py [sentiment_analy

```
# creates random indexes to choose from data frame
random_idx_list = [random.randint(1, len(df.text)) for i in range(10)]
```

Figure 16 - sentiment data distribution in sentiment 140 dataset

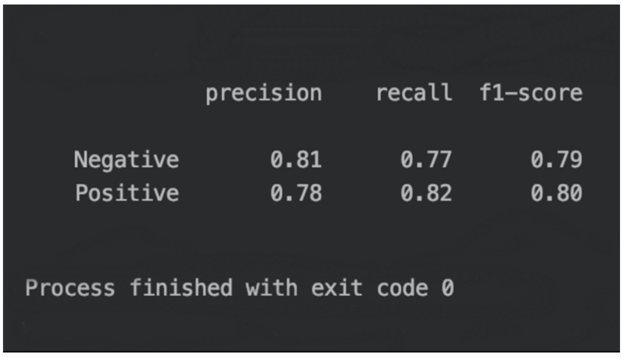
So, the dataset we used – sentiment 140 dataset is unbiased and has high volume and similar number of data for both sentiments. Therefore, accuracy can be used to decide how well this model works. The accuracy of this model is above 85% so this model can be taken as a model which performs well.

Followings are the True positive/negative and False positive/negatives percentages obtained.

True Positive	40.81%
False Positive	11.51%
True Negative	38.53%
False Negative	9.04%

Figure 17 - results

The precision, recall and f1-score for the model were generated using the methods available in the sklearn library and the scores obtained for those metrics are as follows.



```
              precision    recall  f1-score

    Negative       0.81      0.77      0.79
    Positive       0.78      0.82      0.80


Process finished with exit code 0
```

Figure 18 - Precision, recall and f1-score of the model

The model has values > 75% for precision, recall and f1-score metrics.

According to the precision values obtained for the Negative and Positive sentiments, can conclude that the model has predicted correctly most of the positive and negative identifications it has made.

According to the recall values obtained for the Negative and Positive sentiments, can conclude that the model has identified most of the actual positives and negatives in the dataset correctly.

According to the f1-score values obtained for the Negative and Positive sentiments, can conclude that the model predicted most of the instances correctly.

Chapter 5
Conclusion

1. Introduction

This chapter contains the details about limitations, final comments and future works which can be carried out to extend and improve this study. During this study an application was built which can be used by governments to take a rough idea about how people in the country reacts to the decisions and actions they take. For that this study tried to analyze the comments/replies and likes which people put to the tweets that government publish in their twitter page and tried to understand the underlying sentiments of those comments/replies and likes.

While carrying out this study the major challenges faced were due to the limitations in the twitter API. They still do not have any endpoint to retrieve comments and replies for a particular post and they do not support filtering data by dates when retrieving posts published by a particular user. Due to these limitations, the twitter API was explored thoroughly during this study to find some available options and endpoints to achieve the aims of this study and some manual filtering were done to the data retrieved. Also because of these, some limitations were set to the study. When using the first option in the application which derives sentiments for one particular post, it will take only comments and replies up to two levels and will ignore the sentiments of replies beyond that level. When using the second option in the application where it derives sentiments for posts published within a particular time period, it supports only to use dates within one month period and when deriving sentiments only comments will be taken into consideration.

During this study when calculating overall sentiment percentages, apart from the sentiment values of comments and replies, the sentiment values of like reactions also taken into consideration. To derive sentiment value for a like this study suggested a new mechanism, 1 like for a 1 positive comment/reply will be taken as a positive reaction and 1 like for a negative comment/reply will be taken as a negative reaction. Also, some equations were derived in order to calculate overall sentiment percentages.

This study assumes, there could be a very low chance of having a neutral comment or a reply for a decision/action published by the government. So, this study only focuses on negative and positive sentiments.

2. Conclusion

This study was carried out in order to find a solution to one of the major problems that governments face. The government cannot get an overall idea about how the people in the country react to the decisions and actions they take so as a result they lose votes in coming elections. Therefore, the aim of this study was to develop an application which can be use by governments to analyze people's reactions to their actions, decisions as per need, even on daily basis without any additional cost. To achieve this, government need to maintain an official twitter page and they should publish their decisions and actions there. When people put comments and replies to those posts, those will be taken into consideration to decide how people react to those decisions/actions. The application is provided with two options, either the government can analyze people's opinion towards one particular decision/action they took or the government can analyze people's opinions towards decisions/actions they took during a particular time period. During this study a sentiment analysis model was created by using LSTM and word embeddings techniques. Apart from that some equations were derived which are useful when calculating the overall sentiment percentage values.

3. Future Work

Here when training the model, in preprocessing step, smiles and emojis were removed from the dataset but actually they may carry some valuable sentiments too. So, as a future work people can consider these smile and emoji values too when training the model.

Due to the limitations of the twitter API this study focused on only replies to a comment up to two levels when deriving sentiments for a post. If API supports retrieving replies in more easy way, this research can be extended by considering all the replies to a post so that the calculated final total sentiment percentages would be more accurate.

Also, the twitter API does not support filtering results by dates when retrieving posts published by a user. In future if twitter API supports this, people can extend this option 2 in the application (analyze how people reacted to government's decisions and actions within a particular time period) and can remove the limitation set to the data range.

People can extend this study to consider sentiments of comments and replies of retweeted government's decisions actions also, so they can get more inputs from the people for the analysis which leads to more accurate overall sentiment value predictions.

References

[1]. Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[2]. Liu, B. (2009). Handbook Chapter: Sentiment Analysis and Subjectivity. In Hndbook of Natural Language Processing. New York, NY, USA: Marcel Dekker.

[3]. Zia-ul-Rehman, & Syed Anwer Hasnain. (2017). A COMPREHENSIVE STUDY ON SENTIMENT ANALYSIS. Journal of ISOSS.

[4]. Z. Jianqiang and G. Xiaolin, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis," in IEEE Access, vol. 5, pp. 2870-2879, 2017, doi: 10.1109/ACCESS.2017.2672677.

[5]. A. Krouska, C. Troussas and M. Virvou, "The effect of preprocessing techniques on Twitter sentiment analysis," 2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA), Chalkidiki, 2016, pp. 1-5, doi: 10.1109/IISA.2016.7785373.

[6] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, “Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies,” In S. C. Kremer and J. F. Kolen, editors, A Field Guide to Dynamical Recurrent Neural Networks. IEEE Press, 2001.

[7] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Journal of Neural Computation, Vol. 9 No. 8, pp. 1735-1780, 1997.

[8]. D. Li and J. Qian, "Text sentiment analysis based on long short-term memory," 2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI), Wuhan, 2016, pp. 471-475, doi: 10.1109/CCI.2016.7778967.

[9]. T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” Proceedings of the International Conference on Learning Representations 2013 Workshop.

[10] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global Vectors for Word Representation,” Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp. 1532–1543, 2014.

[11]. Dang, Cach & Moreno García, María & De La Prieta, Fernando. (2020). Sentiment Analysis Based on Deep Learning: A Comparative Study. Electronics. 9. 483. 10.3390/electronics9030483.

[12]. Oscar Blessed, Deho & Agangiba, William & Aryeh, Felix & Ansah, Jeffery. (2019). Sentiment Analysis with Word Embedding.

[13]. J.-H. Wang, T.-W. Liu, X. Luo, and L. Wang, “An LSTM Approach to Short Text Sentiment Classification with Word Embeddings,” 2018 Conf. Comput. Linguist. Speech Process. ROCLING 2018, pp. 214–223, 2018

[14]. R. Monika, S. Deivalakshmi and B. Janet, "Sentiment Analysis of US Airlines Tweets Using LSTM/RNN," 2019 IEEE 9th International Conference on Advanced Computing (IACC), Tiruchirappalli, India, 2019, pp. 92-95, doi: 10.1109/IACC48062.2019.8971592.

[15]. F. Miedema, “Sentiment Analysis with Long Short-Term Memory networks,” pp. 1–17, 2018.