A Project on
Analysis and prediction of user sentiment on COVID-19 pandemic through deep learning approach using tweets

ISLAMIC UNIVERSITY

This project paper is submitted to the Department of Information and Communication Technology in Islamic University, Kushtia, Bangladesh for the partial fulfilment and requirements for degree of B.Sc. (Hon's) final examination, 2019

Supervisor
Dr. Paresh Chandra Barman
Professor, Department of Information & Communication Technology
Islamic University, Kushtia

Co-supervisor
Dr. Bikash Chandra Singh
Professor, Department of Information & Communication Technology
Islamic University, Kushtia

Submitted by
Most Nilufa Yeasmin
Roll No. 1518036
Reg. No. 1327
Sesssion: 2015-2016
Department of Information & Communication Technology
Islamic University, Kushtia

Certificate

I am pleased to certify that examination roll no. 1518036, has performed a project work entitled **"Analysis and prediction of user sentiment on COVID-19 pandemic through deep learning approach using tweets"** under my supervision in the academic year 2015-2016 for all the fulfillment of the partial requirement of B.Sc. Degree. So far as I concern this is an original research work that she carried out in the Department of Information and Communication Technology, Islamic University, Kushtia, Bangladesh

I strongly declare that this project has not been copied from, any project or submitted elsewhere prior to this department

Signature of the Supervisor
(Bikash C. Singh, PhD)

Acknowledgement

First of all, I remember the Almighty Allah for giving me the opportunity and patience to carry on and complete this project work successfully.

I would like to thank my honorable supervisor (........), Professor, Department of Information and Communication Technology, Islamic University, Kushtia, Bangladesh for his continuous guidance, kind help, valuable suggestions and other fruitful help throughout the duration of this work.

I am so much grateful to my honorable co-supervisor (......), Associate Professor, Department of Information and Communication Technology, Islamic University, Kushtia. I am also grateful to all the teachers of the Information and Communication Technology department to support me to do better in my life and for teaching me to boost my skills and make me hungry to learn.

I also wish to acknowledge my parents, my other family members, and some of my close friends for their encouragement and continuous support throughout the project.

August 2020
Islamic University, Kushtia

Abstract

Corona-virus disease 2019 (COVID-19) is a global pandemic affecting 188 countries and territories and more than 18.7 million patients worldwide according to August 6, 2020. Corona-viruses are zoonotic, meaning they are transmitted between animals and people, but the ways in which it is transmitted, animal reservoirs, prophylaxis, and precise clinical manifestations. There is currently no vaccine or appropriate treatment for COVID-19 that‘s why a number of countries have resorted to a complete lockdown. During this lockdown, people have taken social networks to express their feelings and find a way to calm themselves down. There are many types of social networks for example: 1. Microblogging platforms: such as Twitter; 2. Blogging platforms: such as WordPress and Blogger; 3. Instant messaging Apps: such as WhatsApp and Telegram; 4. Networking platforms: such as Facebook and LinkedIn. Twitter social networking is a micro-blogging platform considered by researchers as a result of useful applications. There are over 320 million active subscribers on the social network, which daily generates approximately 6 million tweets containing instant news and comments; due to the wealth of information and their easy access. In this research work, sentiment analysis from tweeter data set has been done. This project work has taken a data set and tweets have been collected from the data set, pre-processed, and then used for sentiment analysis. Moreover, we use deep learning approaches so we have a huge amount of tweets to predict the user concern label on COVID-19. The experiment results show that the LTSM model with autoencoder provides better accuracy than other deep learning models for detecting users' concerns about COVID-19.

Keywords: COVID-19, Pandemic, Corona Virus, Twitter, Sentiment Analysis, NLP, LTSM, CNN

Dedicated To-

My Parents
And my honorable teacher Bikash Chandra Singh, Ph.D.

Contents

Chapter 1

Introduction

1.1 Introduction

The origin of COVID-19 is said to be in the starting of December 2019, when several patients from Wuhan, Hubei Province reported severe respiratory infections. These patients had a background of working in the wholesale fish and seafood market, also known as wet markets [6]. In January 2020, the markets were completely closed down and disinfectants were used to sanitize them. On 7th January 2020, the researchers isolated a novel corona-virus which was referred as SARS-CoV-2 or 2019-nCov. Initially the World Health Organization denied the possibilities of human-to-human transmission 2019-nCoV on 11th January 2020 [3]. However, the confirmed cases continued to soar and on 30th January 2020, World Health Organization declared this COVID-19 a Public Health Emergency of International Concern (PHEIC) and an epidemic [5].

By the end of January, the novel corona-virus had already started spreading out to other countries steadily.The disease is highly infectious, and, on average, each patient can spread the infection from 2 to 4 other individuals [19].Worldwide, a total of more than 18.7 million cases of COVID-19 and more than 706,000 deaths were confirmed in 188 countries by Aug 6, 2020.

With the worldwide spread of the COVID-19 infection,individual activity on social media platforms such as Facebook,Twitter, and YouTube began to increase. One of the most famous micro blogging site, Twitter has been one of the major ways for information sharing and self-documentation [8]. As the world is fighting with COVID-19 since last two month and majority of the people are under lock down, the importance of Twitter has increased more than ever. Even in the past, people have been using twitter to communicate, express and disseminate information related to the crisis, be it cyclones [16], ebola [17], floods [14] or Zika [4]. Twitter has been one of the platforms for millions to express their emotions regarding different issues.

This project study has been to analysis the sentiments of different countries regarding COVID-19 and identify what emotions people have been sharing from different parts of the world and then use some deep learning methods includes Simple Neural Networks,Computational Neural Networks(CNN) and Recurrent Neural Networks(RNN)/LTSM.

1.2 Problem Statement

Social media sentiment analysis is a great task running into the field of Natural Language Processing.Now a days it becomes more popular for several reasons. The plenty of thousands methods had been performed for this task.

In this project we will analysis the twitter sentiment of COVID-19 kaggle dataset.And finally we will build some deep leaning model to predict the sentiment of the citizens of several countries regarding COVID-19 based on their tweets.

1.3 Motivation

Sentiment analysis is extremely useful in social media monitoring as it allows us to gain an overview of the wider public opinion behind certain topics. In essence, it is the process of determining the emotional tone behind a series of words, used to gain an understanding of the the attitudes, opinions and emotions expressed within an online mention.

The motivation behind this project is that analyzing sentiment can help policy makers and health care organizations assess the needs of their stakeholders and address them in an appropriate and relevant manner.

1.4 Contribution

In late December 2019, the outbreak of a novel corona-virus causing COVID-19 was reported [10]. Due to the rapid spread of the virus, the World Health Organization declared a state of emergency. In this paper, we focused on analysing COVID-19 related comments to detect sentiment ideas relating to COVID-19 based on the public opinions of people on twitter.Specifically, we used automated extraction of COVID-19 related discussions from social media and a natural language process(NLP) method based on topic modeling to uncover various issues related to COVID-19 from public opinions. The main contributions of this paper are as follows:

1. We present a systematic framework based on NLP that is capable of extracting meaningful topics from COVID-19 related comments on Twitter.

2. We propose Three deep learning models includes Convolutional Neural Nets(CNN), Long Short-Term Memory (LSTM) and Simple Neural Nets,where LTSM is provides better result then others.

3. We Use two different methods first is Word Embedding method and second is GloVe Embedding method for get better accuracy.

4. We detect and uncover meaningful topics that are being discussed on COVID-19 related issues on Twitter Data, as primary research.

5. We calculate the polarity and subjectivity of the COVID-19 comments related to sentiment.

1.5 Project Organization

In my project paper there are total five chapters here. The chapters are belongs to:

- Chapter one is the reflection of Introduction. moreover this chapter describes the brief introduction,contribution and project organization.
- Discussion of COVID-19 related issues and some similar works is provided in this section.
- Discuss development tools and all necessary libraries.
- We describe the data pre-processing methods adopted in my project, and the NLP and deep-learning methods applied to the COVID-19 comments .
- We present the results and discussion.
- We conclude and discuss future works based on deep learning approaches for analysing the online community in relation to the topic of COVID-19.

Chapter 2

Related Works

There are several works that are related to my projects. They are given below:

Machine and deep-learning approaches based on sentiment and semantic analysis are popular methods of analyzing text-content out in online health forums. Many researchers have used these methods on social media such as Twitter, Reddit, and health information websites [13]. For example; Halder and colleagues focused on exploring linguistic changes to analyze the emotional status of a user over time. They utilized a recurrent neural network (RNN) to investigate user content in a huge data set from the mental-health online forums of health boards.com. McRoy and colleagues [12] investigated ways to automate the identification of the information needs of breast cancer survivors based on user posts of online health forums. Chakravorti and colleagues [2] extracted topics based on various health issues discussed in online forums by evaluating user posts of several subreddits (e.g., r/Depression, r/Anxiety)from 2012 to 2018. VanDam and colleagues [18] presented a classification approach for identifying clinic-related posts in online health communities. For that dataset, the authors collected 957 thread-initiating posts from WebMD, which is a health information website.

Although there are similar works regarding various health issues in online forums, to the best of our knowledge, this is the first study to utilize NLP methods to evaluate COVID-19–related comments from Twitter forums. I propose utilizing the NLP technique to automatically extract meaningful topics and design a deep-learning model that includes: CNN, LSTM RNN, and simple neural nets for sentiment classification on COVID-19 comments and to understand the positive,negative, and neutral opinions of people as they relate to COVID-19 issues to inform relevant decision-making.

Chapter 3

Development tools and libraries

We have used Python programming language along with necessary development tools and different useful machine learning libraries. Python is a great general-purpose programming language on its own, but with the help of a few popular libraries, it becomes a powerful environment for scientific computing. We choose Python for sentiment analysis and to build my model because Python has many highly developed machine-learning libraries which help me build this model easily and more accurately. A list of used Python machine learning libraries and development tools is given below:

Python libraries for sentiment analysis and model building:

1. Numpy
2. Pandas
3. Matplotlib
4. TextBlob
5. WordCloud
6. Scikit-Learn

Development tools:

1. Jupyter Notebook

3.1 Python libraries

Let's see some details about some Python libraries that are useful for building our model:

3.1.1 Numpy

Numpy is a library for the Python programming language, that adds support for large,multi-dimensional arrays and matrices, along with a large collection of high-level basic and advanced mathematical functions to operate on these arrays. Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases [11].

3.1.2 Pandas

Pandas is a popular Python-based data analysis toolkit. It is a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool, built on top of the Python programming language [11].

3.1.3 Matplotlib

Matplotlib is a Python package for data visualization. It allows the easy creation of various plots, including line, scattered, bar, box, and radial plots, with high flexibility for refined styling and customized annotation. The versatile artist module allows developers to define basically any kind of visualization. For regular usage, Matplotlib offers a simplistic object-oriented interface, the pyplot module, for easy plotting. Besides generating static graphics, Matplotlib also supports an interactive interface that not only aids in creating a wide variety of plots but is also very useful in creating web-based applications. Matplotlib is readily integrated into popular development environments, such as Jupyter Notebook, and it supports many more advanced data visualization packages.

3.1.4 Scikit-Learn

Scikit-learn exposes a wide variety of machine learning algorithms, both supervised and unsupervised, using a consistent, task-oriented interface, thus enabling easy comparison of methods for a given application. Since it relies on the scientific Python ecosystem, it can easily be integrated into applications outside the traditional range of statistical data analysis. Importantly, the algorithms, implemented in a high-level language, can be used as building blocks for approaches specific to a use case [15].

3.1.5 TextBlob

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more [9].

3.2 Development tools

We also discuss in detail the development tools that use to build our model:

3.2.1 Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows one to create and share documents that contain live code, equations, visualizations, and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. It is a great tool for exploratory data analysis and is widely used by data scientists.

Chapter 4

Framework Methodology

4.1 Framework Methodology

This section clarifies the methods used to investigate the main contributions to this study, which proposes the NLP model, with collaborative deep-learning methods including LSTN(RNN), CNN, and Simple Neural Nets to analyze COVID-19-related comments from the Twitter data set. The developed framework, shown in Figure- 1, uses sentiment analysis for mining and opinion analysis of COVID-19-related comments.

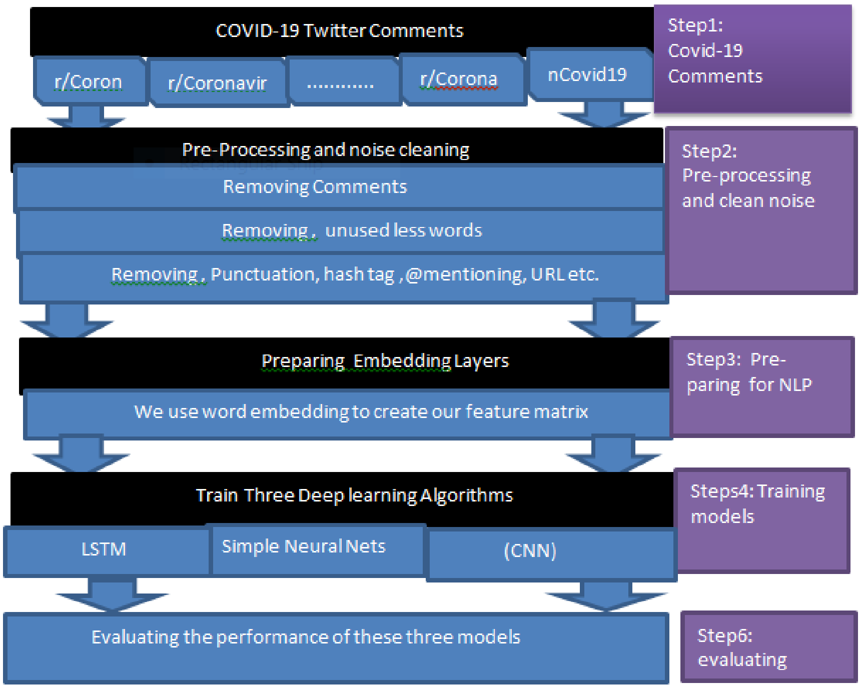

COVID-19 Twitter Comments
r/Coron
r/Coronavir
....
r/Corona
nCovid19
Step1: Covid-19 Comments
Pre-Processing and noise cleaning
Removing Comments
Removing , unused less words
Removing , Punctuation, hash tag ,@mentioning, URL etc.
Step2: Pre-processing and clean noise
Preparing Embedding Layers
We use word embedding to create our feature matrix
Step3: Pre-paring for NLP
Train Three Deep learning Algorithms
LSTM
Simple Neural Nets
(CNN)
Steps4: Training models
Evaluating the performance of these three models
Step6: evaluating

Figure 4.1: The project framework for obtaining results of COVID-19 Twitter data

4.2 Preparing the input data:

Twitter social networking is a micro-blogging platform, a discussion website for various topics that include web content ratings. In this social media, users are able to post questions and comments and respond to each other regarding different subjects, such as COVID-19. The posts are organized by subjects created by online users which cover a variety of topics like news, science, healthcare, video, books, fitness, food, and image-sharing. This website is an ideal source for collecting health-related information about COVID-19-related issues. This project focuses on COVID-19-related posts based on an existing data set as the first step in producing this model.

4.3 Pre-processing:

One of the most important steps in pre-processing COVID-19-related comments is removing useless words/data. For the sentiment analysis, I am gonna only work on the text data and remove other data from dataset. An index is not in sequential order so I reset the index. The Twitter data set has around 375k rows but limiting my computer‘s computational power I used 10k rows for my easy. All of the pre-processing steps are as:

4.3.1 Text cleaning:

Text cleaning is one of the text mining processing to clean the words or other component that is hard to analyze or figure the meaning of the text. Text data or sentence data contains white spaces, punctuation, stop words, etc. These characters do not inform much information and are hard to process for sentiment analysis. such as:

1. Stemming or lemmatization (ways of combining words that have the same linguistic root or stem)

2. Convert the text to lowercase, so that words like “write” and “Write” are considered the same word for analysis

3. Remove numbers

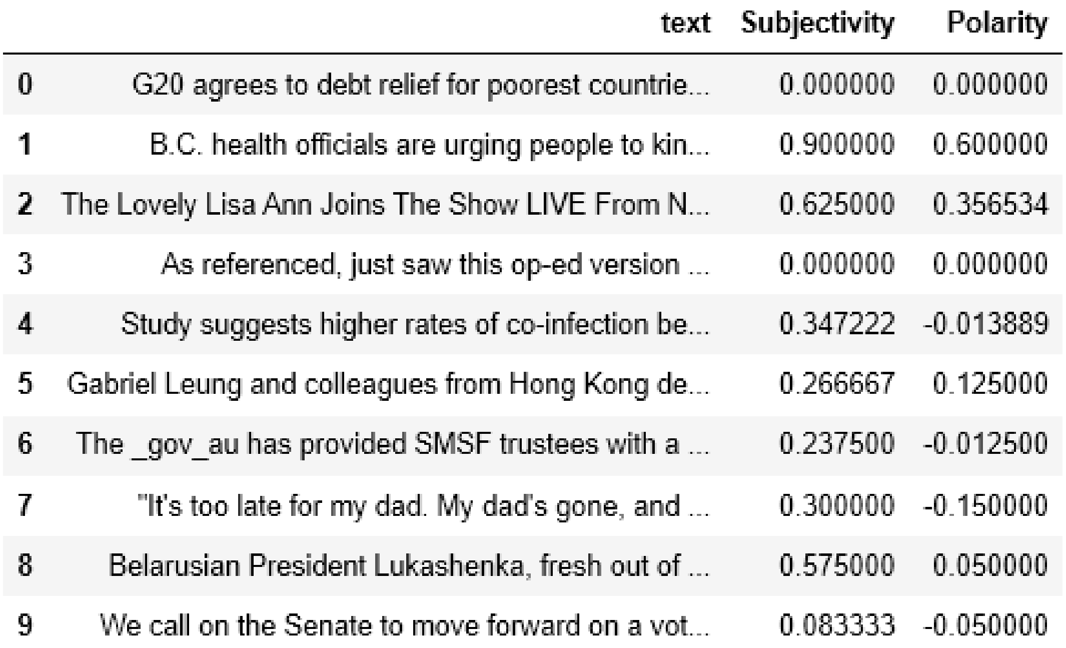
4. Remove English stop words e.g. “the”, “is”, “of”, etc.

5. Remove punctuation e.g. “?”, etc.

6. Eliminate extra white spaces

Finding subjectivity and polarity:

User responses can be positive, negative, or neutral. For each tweet, subjectivity value (range 0 to 1) and polarity value (range -1 to +1) was calculated and the total value of sentiment was calculated as the summation of the product of subjectivity and polarity values of the individual tweet.



	text	Subjectivity	Polarity
0	G20 agrees to debt relief for poorest countrie...	0.000000	0.000000
1	B.C. health officials are urging people to kin...	0.900000	0.600000
2	The Lovely Lisa Ann Joins The Show LIVE From N...	0.625000	0.356534
3	As referenced, just saw this op-ed version ...	0.000000	0.000000
4	Study suggests higher rates of co-infection be...	0.347222	-0.013889
5	Gabriel Leung and colleagues from Hong Kong de...	0.266667	0.125000
6	The _gov_au has provided SMSF trustees with a ...	0.237500	-0.012500
7	"It's too late for my dad. My dad's gone, and ...	0.300000	-0.150000
8	Belarusian President Lukashenka, fresh out of ...	0.575000	0.050000
9	We call on the Senate to move forward on a vot...	0.083333	-0.050000

Figure 4.2: Polarity and Subjectivity

Then plotting the word cloud. This figure is given below:

Figure 4.3: Word Cloud of Tweets from the data set

Finding Negative, Neutral and Positive Sentiment:

After calculating subjectivity and polarity we can analyze the sentiment of the Twitter data set. And we get Positive 45739, Neutral36460, Negative17801. we can express that using a Bar chart as

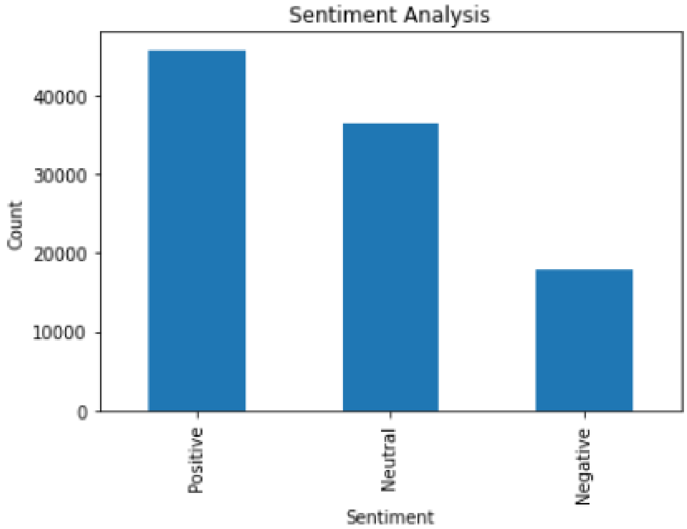


Figure 4.4: Bar chart of the Sentiment Analysis

Adding Label of the Twitter data set:

Creating a function we can add a Label such as if the sentiment analysis is positive then the label is "1" and if it is negative then it is "-1" and if it is neutral then it is "0". It is very important for training any data set.

Splitting the data set into training and testing set:

The train test split function is for splitting a single data set for two different purposes: training and testing. The testing subset is for building our model. The testing subset is for using the model on unknown data to evaluate the performance of the model. Sklearn train test split has several parameters. I use a basic syntax that would look like this:

$$X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.20)$$

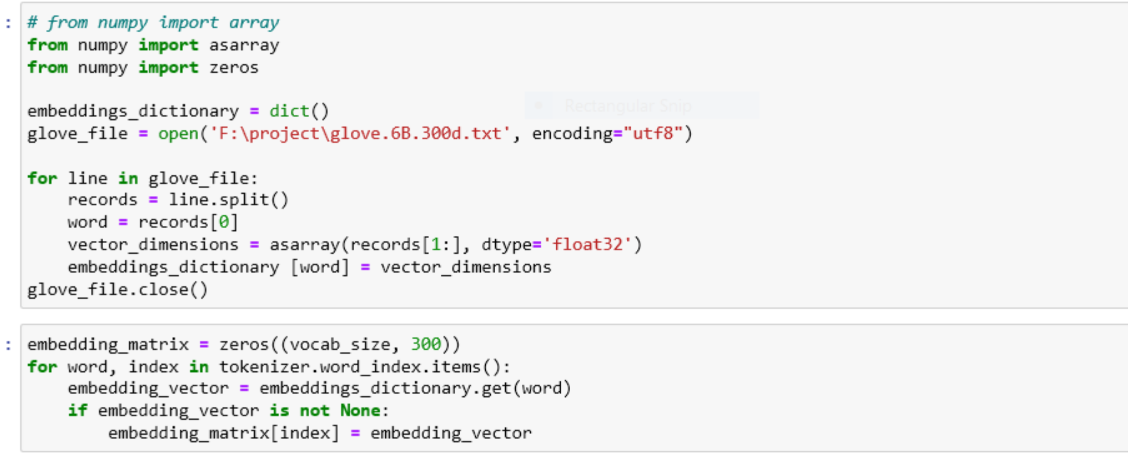
4.4 Preparing the Embedding layer:

We use two methods to prepare the embedding layer :

1. GloVe Embedding
2. Word Embedding

4.4.1 GloVe Embedding:

As a first step, we will use the "Tokenizer" class from the keras. preprocessing.text module to create a word-to-index dictionary. In the word-toindex dictionary, each word in the corpus is used as a key, while a corresponding unique index is used as the value for the key. Then, We will use GloVe embeddings to create our feature matrix. We load the GloVe word embeddings and create a dictionary that will contain words as keys and their corresponding embedding list as values. We will create an embedding matrix where each row number will correspond to the index of the word in the corpus. The matrix will have 300 columns and each column will contain the GloVe word embeddings for the words in our corpus as shown in the figure. Now we are ready to create our deep learning models.



```
# from numpy import array
from numpy import asarray
from numpy import zeros

embeddings_dictionary = dict()
glove_file = open('F:\project\glove.6B.300d.txt', encoding="utf8")

for line in glove_file:
    records = line.split()
    word = records[0]
    vector_dimensions = asarray(records[1:], dtype='float32')
    embeddings_dictionary [word] = vector_dimensions
glove_file.close()
```

```
embedding_matrix = zeros((vocab_size, 300))
for word, index in tokenizer.word_index.items():
    embedding_vector = embeddings_dictionary.get(word)
    if embedding_vector is not None:
        embedding_matrix[index] = embedding_vector
```

Figure 4.5: GloVe Embedding to create out feature matrix

4.4.2 Word Emdedding:

Inputs to Deep learning models need to be in numeric formats. This can be achieved by the following:

1. Assign a number to each word in the sentences and replace each word with their respective assigned numbers.

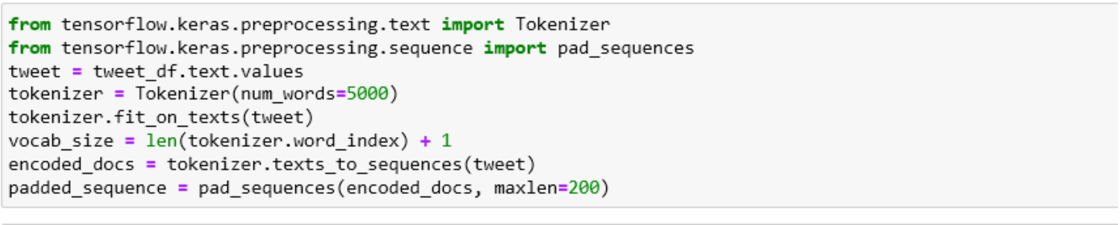
2. Use word embedding. This is capable of capturing the context of a word in a sentence or document.

3. we get the actual texts from the data frame. Initialize the tokenizer with a 5000-word limit.

4. Initialize the tokenizer with a 5000-word limit. This is the number of words we would like to encode.

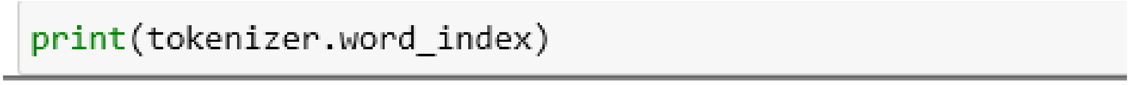
5. we call fit_on_texts to create associations of words and numbers as shown in figure 4.7.

6. calling $text\_to\_sequence$ replaces the words in a sentence with their respective associated numbers. This transforms each sentence into sequences of numbers as shown in figure 4.8.



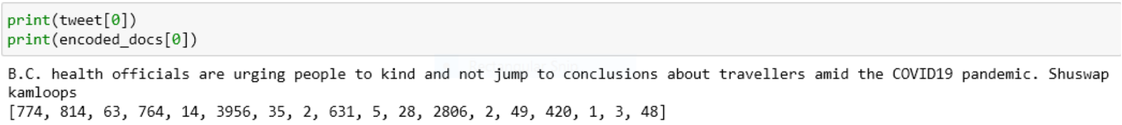
```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
tweet = tweet_df.text.values
tokenizer = Tokenizer(num_words=5000)
tokenizer.fit_on_texts(tweet)
vocab_size = len(tokenizer.word_index) + 1
encoded_docs = tokenizer.texts_to_sequences(tweet)
padded_sequence = pad_sequences(encoded_docs, maxlen=200)
```

Figure 4.6: word embedding to get feature matrix



```
print(tokenizer.word_index)
```

Figure 4.7: Create associate words and numbers



```
print(tweet[0])
print(encoded_docs[0])
```

```
B.C. health officials are urging people to kind and not jump to conclusions about travellers amid the COVID19 pandemic. Shuswap
kamloops
[774, 814, 63, 764, 14, 3956, 35, 2, 631, 5, 28, 2806, 2, 49, 420, 1, 3, 48]
```

Figure 4.8: Sentence into sequences of numbers

The sentences or tweets have a different number of words, therefore, the length of the sequence of numbers will be different. Our model requires inputs to have equal lengths, so we will have to pad the sequence to have the chosen length of inputs. This is done by calling the *pad_sequence* method with a length of 200. All input sequences will have a length of 200.

4.5 Deep-Learning and Sentiment Classification:

4.5.1 Deep Learning:

Deep learning is a specific part of machine learning in artificial intelligence (AI) and consists of algorithms that allow the software to train itself to perform tasks by exposing multilayer neural networks to massive amounts of data [15]. Recently, deep learning algorithms have given effective performance in natural language processing uses, comprising sentiment analysis over multiple datasets [16]. The greatest value of deep learning is that we do not need to manually extract features, instead of that, they take word embedding, as input which contains context information, and the middle layers of the neural network learn the features during the training phase by themselves. Words are expressed in the high dimensional vector and feature extraction is performed by the neural network [17]. The main reason deep learning starts very rapidly due to provides superior performance on various issues and also makes problem-solving much easier because it is fully automatic [18]. The deep neural network is about assigning inputs to targets through a deep chain of simple data transformations (layers), and such layers are learned by observing many samples of input and targets.

Transformation is performed through a layer that is parameterized by its own weights, also termed parameters. Learning layers means discovering a series of values for the weights of all layers in the network in such a way the network will precisely set the input samples for the targets associated with them. The deep neural network can include many million parameters, and getting the right value for each parameter looks like a dispiriting duty because changing the value of the individual parameter will influence the behavior of all other parameters, for this purpose loss function also called the objective function will be used to computes a distance score by comparing the prediction of the network and the real object to estimate how far the predicted output is from the real object. After computing the distance score between the predictions of the network and the real target by using a loss function, this score is utilized as a feedback sign to slightly improve the value of weights, in a way that will reduce the loss score, this improvement is a function of the optimizer that performs what's called the Backpropagation algorithm. In the Back propagation algorithm, at first weights of the network are appointed with random values, hence the network only performs a sequence of arbitrary shifts. Generally, the results are ideally far from what they should be, so the loss score is too high. But, in each case, the network handles, the weights are slightly modified in the right trend, and the loss score decreases, that's the training loop that iterated enough times, giving weight values that reduce the loss function. The lowest loss network is the network where the outputs are closest to the targets.

Deep Learning processes are given below:

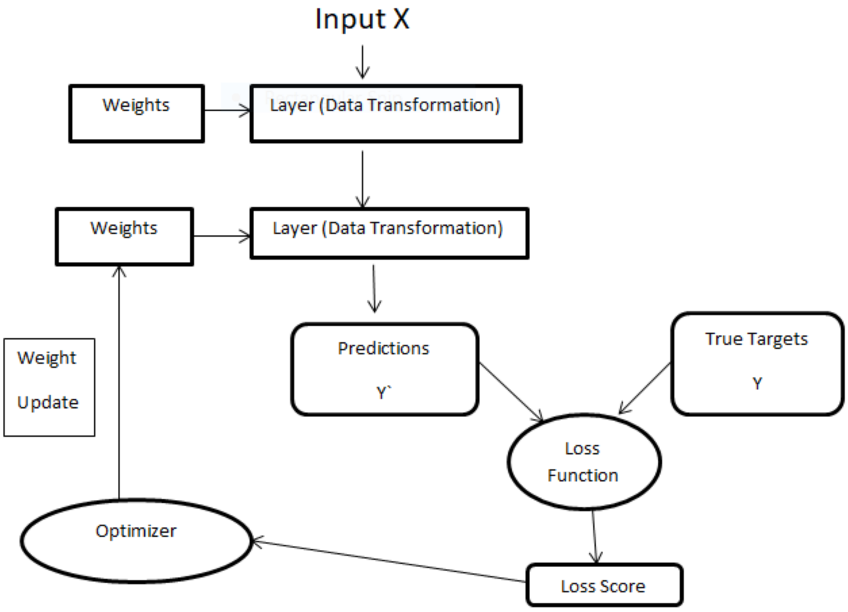


Figure 4.9: Deep Learning Process

4.5.2 Sentiment Analysis

Sentiment Analysis is the process of determining the emotional tone behind a series of words, used to gain an understanding of the attitudes, opinions, and emotions expressed within an online mention [1]. In this project, we want to do sentiment analysis for the Twitter dataset using different deep learning methods, specially LTSM and CNN methods. Before describing the different deep learning methods, we want to show a basic workflow of a text-based general sentiment analysis pipeline. The basic workflow of a text-based general sentiment analysis pipeline is given below:

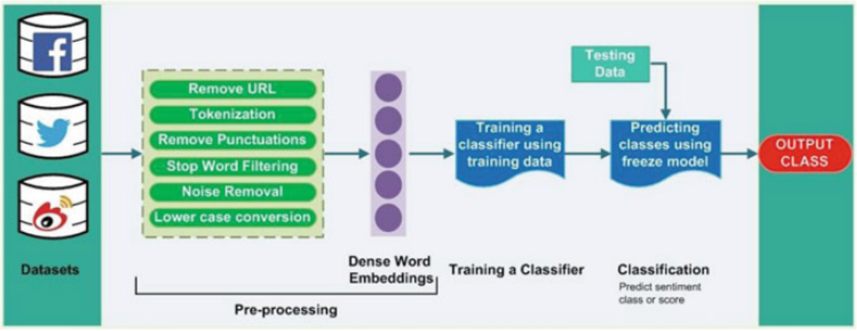


Figure 4.10: General sentiment analysis pipeline

4.5.3 CONVOLUTIONAL NEURAL NETWORK:

Let us now describe the architecture of the CNN we worked with. Its architecture is almost identical to the CNN of [7]. A smaller version of our model is illustrated in Figure. The input of the network is the tweets, which are tokenized into words. Each word is mapped to a word vector representation, i.e. a GloVe embedding, such that an entire tweet can be mapped to a matrix of size $s \times d$, where s is the number of words in the tweet and d is the dimension of the embedding space(we chose $d = 200$). We follow [7] padding strategy such that all tweets have the same matrix dimension $X \epsilon \mathbb{R}^(s \times d)$, where we chose $s = 80$. We then apply several convolution operations of various sizes to this matrix. A single aconvolution involves a filtering matrix $w \epsilon \mathbb{R}^(h \times d)$ where h is the size of the convolution, meaning the number of words it spans. The convolution operation is defined as

$$c_i = f(\sigma_(j,k)(w_(j,k)(X_[i : i + h - 1])_{j,k} + b))$$

where $b \epsilon \mathbb{R}$ is a bias term and $f(x)$ is a nonlinear function, which we chose to be the ReLu function. The output $c \epsilon \mathbb{R}^s + h + 1$ is, therefore, a concatenation of the convolution operator over all possible windows of words in the tweet. Note that because of the zero-padding strategy we use, we are effectively applying wide convolutions. We can use multiple filtering matrices to learn different features, and additionally, we can use multiple convolution sizes to focus on smaller or larger regions of the tweets. In practice, we used three filter sizes and we used a total of 300 filtering matrices for each filter size. We then apply a max-pooling operation to each convolution $c_{max} = max(c)$. The max-pooling operation extracts the most important feature for each convolution, independently of where in the tweet this feature is located. In other words, the CNN's structure effectively extracts the most important n-grams in the embedding space, which is why we believe these systems are good at sentence classification. The max-pooling operation also allows us to combine all the c_{max} of each filter into one vector $c_{max} \epsilon \mathbb{R}^m$ where m is the total number of filters (in our case $m = 3 \times 200 = 600$). This vector then goes through a small fully connected hidden layer of size 30, which is then in turn passed through a softmax layer to give the final classification probabilities. To reduce overfitting, we add a dropout layer after the max-pooling layer and after the fully connected hidden layer, with a dropout probability of 50% during training. The figure is given below:

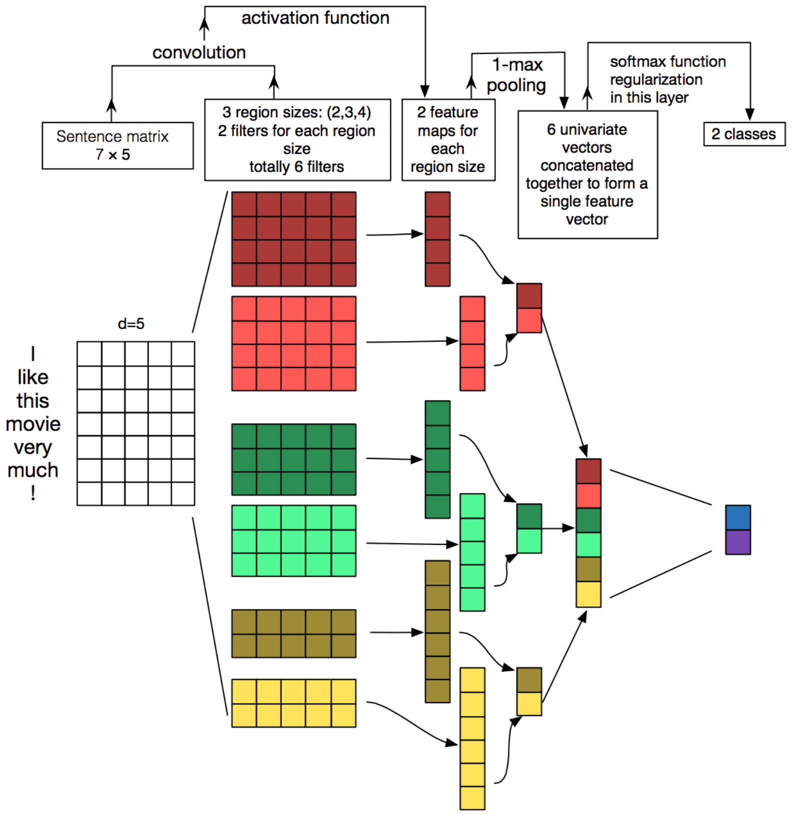

activation function
convolution
1-max pooling
softmax function regularization in this layer
Sentence matrix 7 × 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
6 univariate vectors concatenated together to form a single feature vector
2 classes
d=5
I
like
this
movie
very
much
!

Figure 4.11: Architecture of CNN used for text analysis. Picture is taken from [21]

4.5.4 LONG SHORT-TERM MEMORY:

Long short-term memory (LSTM) networks are a particular kind of RNN capable of analyzing and learning long-term dependencies. LSTMs are explicitly considered to deal with the long-term dependency problem. They can easily remember any piece of information as long as it is required, best while working on a sequence of sentences. Long short-term memory units are modules that can be used inside recurrent neural networks. At an advanced level, it makes sure that it can encapsulate information about long-term dependencies in the text. LSTM is adopted whenever dealing with long-term dependencies. Let us look at the simple LTSM cell

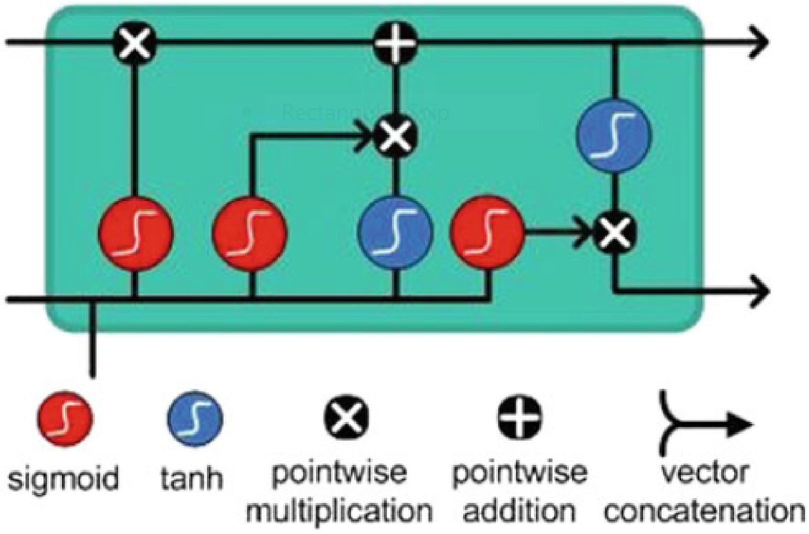


Figure 4.12: A LTSM cell

Let us now describe the architecture of the LSTM system we worked with. A smaller version of our model is illustrated in Fig. Its main building blocks are two LSTM units. LSTMs are part of the recurrent neural networks (RNN) family, which are neural networks that are constructed to deal with sequential data by sharing their internal weights across the sequence. For each element in the sequence, that is for each word in the tweet, the RNN uses the current word embedding and its previous hidden state to compute the next hidden state. In its simplest version, the hidden state $h_t \epsilon \mathbb{R}^m$ (where m is the dimension of the RNN, which we pick to be $m = 200$) at time t is computed by

$$h_t = f(W_h \times x_t + U_h \times h_(t - 1) + b_h)$$

where x_t is the current word embedding,$W_h \epsilon \mathbb{R}^m \times d$ and $U_h \epsilon \mathbb{R}^m \times m$ are weight matrices, $b_h \epsilon \mathbb{R}^m$ is a bias term and f(x) is a non-linear function, usually chosen to be tanh. The initial hidden state is chosen to be a vector of zeros. Unfortunately, this simple RNN suffers from the exploding and vanishing gradient problem during the backpropagation training stage. LSTMs solve this problem by having a more complex internal structure which allows LSTMs to remember information for either long or short terms. The hidden state of an LSTM unit is computed by [20]

$$
\begin{aligned}
f_t &= \sigma(W_f \times x_t + U_f \times h_(t-1) + b_f) \\
i_t &= \sigma(W_i \times x_t + U_i \times h_(t-1) + b_i) \\
o_t &= \sigma(W_o \times x_t + U_o \times h_(t-1) + b_o)
\end{aligned}
$$

$$
\begin{aligned}
c_t = f_t \times c_(t-1) + i_t \times \tanh(W_c \times x_t + U_c \times h_(t-1) + b_c) \\
h_t = o_t \times \tanh(c_t)
\end{aligned}
$$

where it is called the input gate, f_t is the forget gate, c_t is the cell state, h_t is the regular hidden state, σ is the sigmoid function, and o is the Hadamard product. of size 30, and then passed through a softmax layer to give the final classification probabilities. Here again, we use dropout to reduce over-fitting; we add a dropout layer before and after the LSTMs, and after the fully connected hidden layer, with a dropout probability of 50% during training. The figure is given below:

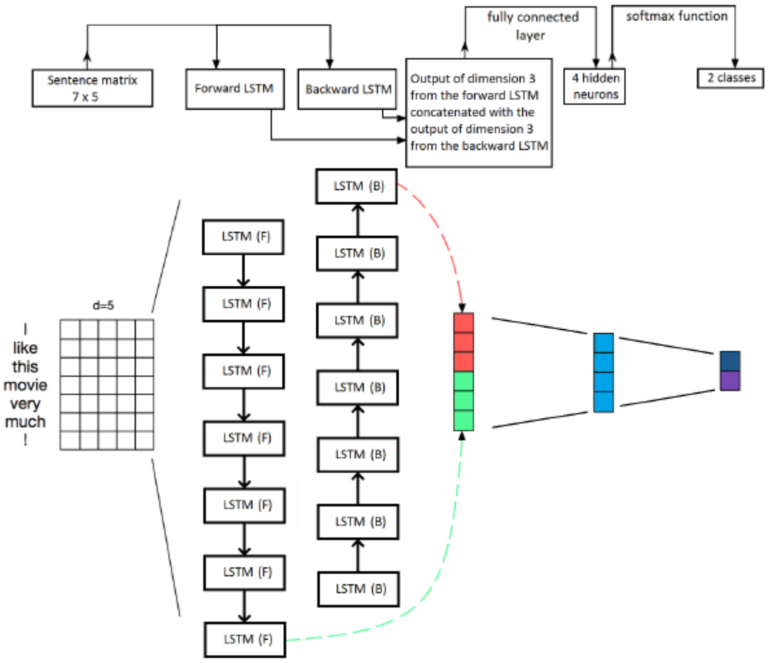


Figure 4.13: Architecture of LTSM used for text analysis. The picture is inspired by CNN architecture of [21]

Chapter 5

Result and Discussion

5.1 Sentiment Result

Analyzing social media comments on platforms such as Twitter provide meaningful information for understanding people's opinions, which might be difficult to achieve through traditional techniques, such as manual methods. The text content on Twitter has been analyzed in various studies; to the best of our knowledge, this is the first study to analyze comments by considering sentiment aspects of COVIDrelated comments from Twitter for online health communities. Overall, we extended the analysis to check whether we could find a dependency of s aspects of user comments for different issues on COVID-19-related topics. In this case, we considered an existing dataset that included 375k data but we use 10k data from this dataset. We found and detected meaningful latent topics of terms about COVID-19 comments related to various issues. Thus, user comments proved to be a valuable source of information. A variety of different visualizations was used to interpret the generated results.

After doing the sentiment analysis we see that 54% sentiments are positive,30% sentiments are neutral and only 16% sentiments are negative.

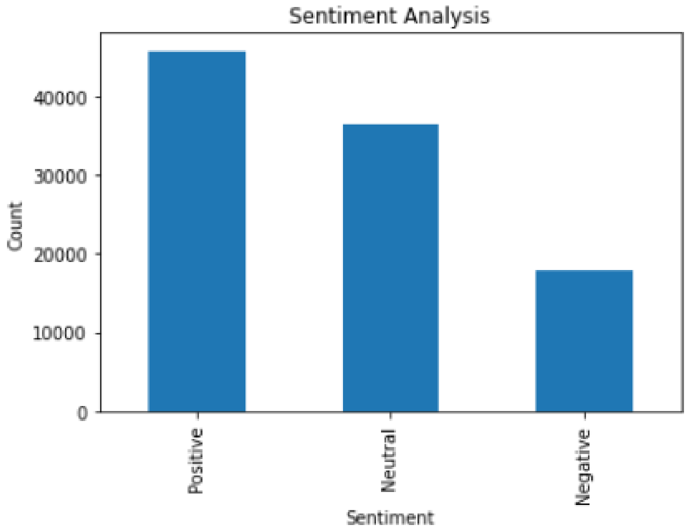


Figure 5.1: Sentiment Result

The above figure clearly shows that positive sentiments are higher than neutral and negative.

5.2 MODEL RESULT AND EVALUATION:

In this part, we discuss the results obtained through CNN, LSTM, and Simple Neural Nets, all trained on a 100k Twitter dataset, and the results are compared based on metrics like accuracy, precision, recall, and f1 score. In building any deep learning model, one of the primary tasks is to evaluate its performance; the performance of each technique used in this work is measured, by computing different metrics and the ultimate purpose behind working with different metrics is to understand how well a deep learning model is going to perform on unseen data. In this work, the following metrics are used: Accuracy is the proportion of the accurately analyzed samples to the total number of samples.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

In the above equations, TP is the true positive and predicted correctly, FP is the false positive and predicted incorrectly, TN is the true negative and predicted correctly, and FN is the false negative and predicted incorrectly.

Table-1 illustrates the results and performance comparisons of the models, in this work, the experimented dataset is split into training, and testing sets. The training set was given 80% of the dataset, while the testing set was each given 20% of the dataset.

Table 5.1: Diffrents model and their Accuracy

Using Methods	Models	Accuracy
GloVe Embedding	**CNN**	61%
GloVe Embedding	**LTSM**	38%
GloVe Embedding	**Simple Neural Nets**	54%
Word Embedding	**LTSM**	94%
Word Emdeding	**CNN**	93%

In this work, the result of each technique is achieved in the following configurations: each of the models is configured with a dropout layer to restrict the neural network from memorizing the training set, which is useful to prevent overfitting. The models compiled with the Adam optimizer with a batch size of 128 for 6 epochs, the output layer in all models is a fully-connected dense layer with sigmoid activation that makes a binary prediction. In the CNN model using Word Embedding; the network has a three-layer of 1d-CNN and all layers are implemented with 128 filters and a kernel size of 1, 2,3,4,5 respectively, after each layer a max-pooling layer with 2 pooling filter sizes is applied that selects the value with the highest weight only and ignores the rest values which significantly enhance the results of the convolutional layer and reduces the input to the next layer. Also, it has a flattened layer that transforms a two-dimensional matrix of features into a vector that can be fed into the output layer. In LTSM using Word Embedding; the network has a three-layer of LSTM and each layer has 128 neurons. In LTSM using GloVe Embedding; the network has several layers. It has also a dropout layer to prevent overfitting. As shown in the table-1 using Word Embedding LTSM model outperforms all other models. We can learn that the CNN model performs 7% better than Simple Neural nets. And using Word Embedding; we can learn that the LTSM model performs 1% better than the CNN model. LSTM showed the best performance in all metrics and achieved the highest accuracy of 93.70%.

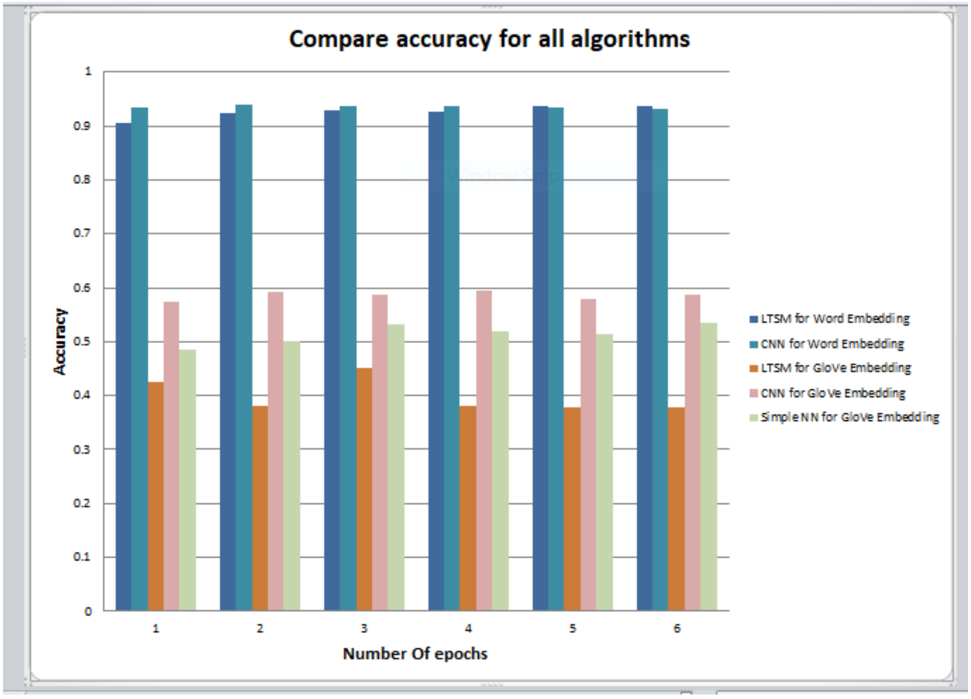


Figure 5.2: Compare accuracy for all algorithms

From the above figure, we see that the CNN and LTSM models using Word Embedding give more than 93% accuracy.They perform very well. These two models give more than 93% accuracy which is pretty good for this project. Also, we can see that the two models (CNN and Simple Neural Nets) using GloVe Embedding also performs well.

5.3 Limitation:

This project was limited to English-language text, which was considered a selection criterion. Therefore, the results do not reflect comments made in other languages. In this study, no geographical restrictions were applied to the tweets analyzed considering the worldwide spread of the disease. In addition, we don‘t use API to collect data, we use direct Twitter data from Kaggle. Moreover, this study could not collect tweets from accounts marked as private. Therefore, findings may not represent all the topics discussed by users on Twitter related to COVID-19. Only posts on Twitter were analyzed in this study, thereby, our findings may not be generalizable to other social media platforms. Furthermore, the findings reported in this study are limited to only those that have access to and use Twitter. Therefore, caution is advised before assuming the generalizability of the results, as Twitter is not used by everyone in the population.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

To our knowledge, this is the first study to analyze COVID-19 sentiment on Twitter data. This project aimed at analyzing the sentiments and emotions of the people during the pandemic COVID-19. The COVID-19 pandemic has been affecting many healthcare systems and nations, claiming the lives of many people. As a vibrant social media platform, Twitter projected this heavy toll through the interactions and posts people made related to COVID-19. It is clear that coordinating public health crisis response activities in the real world and online is paramount, and should be a top priority for all healthcare systems. The main goal of this project, however, was to show a novel application for NLP based on LSTM, CNN, and simple neural network models to detect meaningful latent topics and sentiment-comment-classification on COVID-19-related issues from healthcare forums, such as Twitter. Moreover, our findings may aid in improving practical strategies for public health services and interventions related to COVID-19.

6.2 Future Work

It is expected that as the spread of this pandemic will increase, the sentiments and emotions in the tweets may change along the lines of what was seen in the case of China, the US, Italy, etc. There much further work can be done, such as considering the word2vec tool, multilayer convolutional neural network, combination of CNN and LTSM model, larger training dataset, and other situation or status analyses. In the future, we can also try to connect our model to other natural language processing technology like parts-of-speech tagging trying to get better results in issues of NLP.

References

[1] https://www.brandwatch.com/blog/understanding-sentiment-analysis/.

[2] Dante Chakravorti, Kathleen Law, Jonathan Gemmell, and Daniela Raicu. Detecting and characterizing trends in online mental health discussions. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 697–706. IEEE, 2018.

[3] Akash Dutt Dubey. Twitter sentiment analysis during covid19 outbreak. *Available at SSRN 3572023*, 2020.

[4] King-Wa Fu, Hai Liang, Nitin Saroha, Zion Tsz Ho Tse, Patrick Ip, and Isaac Chun-Hai Fung. How people react to zika virus outbreaks on twitter? a computational content analysis. *American journal of infection control*, 44(12):1700–1702, 2016.

[5] Leiwen Fu, Bingyi Wang, Tanwei Yuan, Xiaoting Chen, Yunlong Ao, Tom Fitzpatrick, Peiyang Li, Yiguo Zhou, Yifan Lin, Qibin Duan, et al. Clinical characteristics of coronavirus disease 2019 (covid-19) in china: a systematic review and meta-analysis. *Journal of Infection*, 2020.

[6] Chaolin Huang, Yeming Wang, Xingwang Li, Lili Ren, Jianping Zhao, Yi Hu, Li Zhang, Guohui Fan, Jiuyang Xu, Xiaoying Gu, et al. Clinical features of patients infected with 2019 novel coronavirus in wuhan, china. *The lancet*, 395(10223):497–506, 2020.

[7] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[8] Ivy LB Liu, Christy MK Cheung, and Matthew KO Lee. Understanding twitter usage: What drive people continue to tweet. *Pacis*, 92:928–939, 2010.

[9] Steven Loria. textblob documentation. *Release 0.15*, 2, 2018.

[10] Monica Malta, Anne W Rimoin, and Steffanie A Strathdee. The coronavirus 2019-ncov epidemic: Is hindsight 20/20? *EClinicalMedicine*, 20, 2020.

[11] Wes McKinney. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython.* " O'Reilly Media, Inc.", 2012.

[12] Susan McRoy, Majid Rastegar-Mojarad, Yanshan Wang, Kathryn J Ruddy, Tufia C Haddad, and Hongfang Liu. Assessing unmet information needs of breast cancer survivors: Exploratory study of online health forums using text classification and retrieval. *JMIR cancer*, 4(1):e10, 2018.

[13] Hamid Naderi, Sina Madani, Behzad Kiani, and Kobra Etminani. Similarity of medical concepts in question and answering of health communities. *Health informatics journal*, 26(2):1443–1454, 2020.

[14] Meera R Nair, GR Ramya, and P Bagavathi Sivakumar. Usage and analysis of twitter during 2015 chennai flood towards disaster management. *Procedia computer science*, 115:350–358, 2017.

[15] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[16] Cheryll Ruth Soriano, Ma Divina Gracia Roldan, Charibeth Cheng, and Nathaniel Oco. Social media and civic engagement during calamities: the case of twitter use during typhoon yolanda. *Philippine Political Science Journal*, 37(1):6–25, 2016.

[17] Liza GG Van Lent, Hande Sungur, Florian A Kunneman, Bob Van De Velde, and Enny Das. Too far to care? measuring public attention and fear for ebola using twitter. *Journal of medical Internet research*, 19(6):e193, 2017.

[18] Courtland VanDam, Shaheen Kanthawala, Wanda Pratt, Joyce Chai, and Jina Huh. Detecting clinically related content in online patient posts. *Journal of biomedical informatics*, 75:96–106, 2017.

[19] TP Velavan and CG Meyer. La epidemia de covid-19. *Trop Med Int Health*, 2020.

[20] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[21] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.