Multimodal Sentimental Analysis Of Coronavirus Tweets Using Deep Learning

Abstract

With the sudden outbreak of COVID-19, there was a quick spike in microblog activity, providing a perfect opportunity to analyze public emotion about the happenings. Sentiment analysis is utilized in this context to study how coronavirus impacts public perception. The findings will help executives communicate more effectively in the virtual world while accounting for emotions. This study will explore incorporating emotional models using deep learning algorithms that surpass typical machine learning models. The investigation establishes implementing the computational model of emotion and its outcome within a multimodal framework. Multimodal-based sentiment analysis seeks to evaluate information from textual and image data. Despite recent improvements, current systems still have issues classifying social media data that combines visual and text information due to subjectivity and inter-class uniformity.

Contents

1 Introduction **1**

1.1 Background 1

1.2 Problem statement 2

1.3 Objectives of the Study 3

1.4 Significance of the Research 4

1.5 Thesis Outline 4

2 Literature Review **5**

2.1 Introduction 5

2.1.1 Why Sentiment Analysis is hard? 6

2.2 History 6

2.2.1 Lexicon-based approach 8

2.2.2 Machine learning approach 9

2.3 DEEP LEARNING 14

2.3.1 Introduction 14

2.3.2 What is Deep Learning? 16

2.3.3 Biological inspiration of Neural Networks 16

2.3.4 Working of a Perceptron: 17

2.3.5 Activation Functions 18

2.3.6 Loss Functions 19

2.3.7 Optimizer 21

2.3.8 Backpropogation 23

2.4 Convolutional neural network (CNN) 24

2.4.1 How does CNN work? . . . 24
2.4.2 Padding . . . 25
2.4.3 What's a pooling layer? . . . 26
2.4.4 Max pooling . . . 26
2.4.5 Average Pooling . . . 27
2.5 Recurrent neural network (RNN) . . . 27
2.5.1 LSTM & GRU . . . 28
2.5.2 Sequence to Sequence Models . . . 29
2.6 Transformers . . . 30
2.6.1 Self-Attention . . . 31
2.6.2 BERT . . . 33
2.6.3 Text Preprocessing . . . 34
2.6.4 Training . . . 35
2.7 Related Work . . . 37

3 Methodology 39
3.1 Data collection . . . 39
3.2 Class Labeling & Balancing . . . 40
3.3 Text Pre-Processing: . . . 40
3.4 Text Extraction . . . 41
3.5 Model Architecture . . . 41
3.5.1 Text modality . . . 42
3.5.2 Image modality . . . 44
3.5.3 Fusion Techniques . . . 45
3.5.4 Construction . . . 46
3.5.5 Hyperparameter Tuning . . . 47

4 Results 49
4.1 Model Evaluation . . . 49
4.2 Error Analysis . . . 54
4.3 Further Discussions . . . 59
4.4 Future Improvement . . . 60

4.5 Model Deployment 60

5 Conclusion **62**

6 APPENDIX **69**

Chapter 1

Introduction

1.1 Background

On 31st December 2019, the Covid-19 first reported in the Wuhan, Hubei Province, China. The virus began to spread swiftly around the planet. On 11th March 2020, the World Health Organization declared the Covid-19 outbreak a pandemic. As the virus spread swiftly across the globe, claiming the lives of millions of people every day, few countries imposed a total lockdown to reduce the epidemic's impact. During the lockdown, social media platforms played a pivotal role in spreading knowledge about the pandemic, as people used social media to voice their feelings. Users were increasingly eager to communicate their emotions and thoughts through images, text, audio, and video.

The majority of the content shared on social media is emotive. People were experiencing worry, dread, and anxiety due to the increasing number of exponential instances around the world. The global population's mental and physical health has proven to be directly proportional to this pandemic sickness.

People found it difficult to comprehend the pandemic since the tweets and information related to Covid-19 posted on social media platforms were not from legit sources. It is often difficult to discern how widespread imprecise information is, and we don't know what the public thinks about the pandemic scenario. For leaders and

politicians analyzing, understanding, and communicating public opinion has always been a top priority. This research will extract the sentiments of people regarding the COVID situation based on the tweets, images, and text extracted from the visual content, which helps to understand the overall polarity of the masses and will enable the competent authority to give the appropriate direction and take timely remedial action.

Traditionally the analysis of sentiment was performed extensively on text. These days a large amount of data is uploaded as images, emoticons, audio, and videos[40]. Since visual data contains profound sentiment indicators, there are benefits of including images rather than just texts. Facial expressions are among image characteristics that are crucial for capturing thoughts and emotions since they play a pivotal role in determining a person's current cognitive state of mind. A smile and a frown is regarded as one of the most commonly used predictive visual indicators in determining the polarity of whether a person is happy(positive) or unhappy(negative). Additionally, sentiment on the text was performed using classical machine learning algorithms such as Naïve Bayes, Logistic Regression, SVM, Random Forests, etc. With the advent of deep learning, state-of-the-art algorithms such as CNN, RNN, LSTM, and BERT can be applied to text, images, sound, and video ensuring promising results.

1.2 Problem statement

Sentiment analysis is often predominately carried out with textual data. On the other hand, Multimodal sentiment analysis aims to produce an evaluation using information from multiple sources such as text, image, sound, video. The data helps understand people's attitudes and views on some events by analyzing the sentiment using a multimodal approach. However, there are key issues that remain mostly unaddressed in this field, such as the consideration of the context in classification, the effect of speaker-inclusive and speaker-exclusive scenarios, the impact of each modality across datasets, and generalization ability of a multimodal sentiment classifier[18].

The coupling of multiple data sources allows the development of multimodal classification, in which a method leverages several types of data to perform classification [10]. It is significantly hard concerning the context of sentiment analysis, as extracting sentiments solely from textual information is easy rather than combining data from a text and other modalities like images. Even though a multimodal approach can improve performance over the sole text approach, this is challenging especially when data acquired from social media can be sparse and may carry noise. Data possess different contexts, contradictions, present irony, and different intentions.

The multimodal-based approach may reveal users' true feelings or intentions because it possesses more information from multiple modalities. Sentiment analysis requires effective feature capture since sentiment information varies from one modality to another. From a human perspective, not all the attributes of an image or the words in a sentence must relate to sentiment. The fundamental characteristics of dimensions or features must be expressed differently among discrete models. In this research, we study the behavior of the proposed method, various NLP and computer vision techniques incorporated, the generalizability of the models, model comparisons, and performance of individual modalities.

1.3 Objectives of the Study

There are three core objectives of this thesis:

1. To develop a multimodal-based deep neural network framework to predict the sentiment of social media tweets about coronavirus containing text and visual data.

2. To evaluate the performance of individual modalities and determine which performed the best.

3. To investigate whether adding more features promotes the model to yield better predictions.

1.4 Significance of the Research

Leaders are challenged with surplus explosive situations in which they must make decisions that might benefit or harm their companies, themselves, or society. Examining person feelings in the form of positive, negative, and neutral tweets could assist researchers in fathoming how people are dealing with the pandemic and its psychological repercussions. Sentiment analysis will enhance the ethical quality of government choices, ensuring that they do not have unintended consequences. It helps authorities predict the outcome of a decision and determine the influence on the affected population. This research aids us in determining the degree of susceptibility with which the population under investigation communicates, also making better decisions based on the information gathered. It will likewise be very effective in determining individuals's mental well-being and would help devise appropriate lockdown strategies and safeguard crisis management in future pandemics.

1.5 Thesis Outline

The thesis work is organised as shown below:

- Chapter 1: We walk through the background of our thesis and introduce our problem statement and objectives.
- Chapter 2: In this chapter, we discuss previous research or study on sentimental analysis and the theory behind machine/deep learning algorithms in great detail.
- Chapter 3: This chapter involves the adaptation of deep learning algorithms and other techniques to tackle our problem.
- Chapter 4: This chapter contains the results and outcomes of our study with model deployment to address real-world data.
- Chapter 5: In this chapter, we close our thesis work with a conclusion.

Chapter 2

Literature Review

2.1 Introduction

Sentiment analysis, also called opinion mining, is a computational study of reviews, sentiments, opinions, evaluations, attitudes, subjective, views, emotions, etc., expressed in the text. Sentiment is often recognized as emotions or judgments, opinions or ideas prompted or colored by emotions or susceptibility or feelings [36]. In Computational Linguistics, the focus is on opinions and sentiments rather than feelings or emotions, and the words 'sentiment' and 'opinion' is often used alternately.

There are two types of textual information: facts and opinions information. While the facts are objective expressions about objects, features, entities, events and their characteristics, and opinions are ordinarily subjective expressions that identify people's sentiments, views, or feelings toward objects, entities, events, and their characteristics [35].

Sentiment analysis can deal with the computational handling of subjective, sentiment, and opinion in the text [35]. It plans to realize the opinion of a writer concerning a topic or goal. The attitude could reflect his or her opinion and evaluation. His or her effective situation (what are the feelings of the writer at the time of recording the opinion), or the purpose of emotional communication (What is the effect, which is situated on the reader when reading the opinion of the writer). Moreover, it should

be noted that in this context 'subjective' does not mean that something is not true [41]. In sentiment analysis, studying the subjective language: the language used to rapid a private situation in the context of a text or conversation. The research identified as private situation is a general encasement term for opinions,evaluations, emotions, and assessment [50].

2.1.1 Why Sentiment Analysis is hard?

Sentiment analysis is one of the toughest problems to be addressed by Computer. Identifying some entities, features and patterns is hard for machines or even impossible while it is easy for us human beings. Below you can find some intractable situations for computers:

- Dealing with ironies or sarcasm, it is difficult to understand that the opposite meaning of a sentence is required. Sometimes irony is recognized through special punctuation marks such as (!!!) but it is not that common to be a rule or sign for these types of expressions.
- Pronoun resolution is another daunting task. Although there are some techniques and algorithms that can solve, it is still demanding task in sentiment analysis. For instance; there are opinion words in a sentence because the corresponding feature is a pronoun, it is not easy to find which feature is expressed by those sentiment words.
- Defining on the strength of an opinion also should be recognized as a demanding task in this area. Opinions have different strengths.

2.2 History

Natural language processing(NLP) is an area of computer science, artificial intelligence, and computational linguistics concerned with interactions between computers and human (natural) languages. Intrinsically, NLP is related to the field of human-computer interaction (HCI). Many challenges in NLP include natural language understanding that is, enabling computers to deduce meaning from human or natu-

ral language input, and others involve natural language generation [33].The term Natural Language Processing involves wide set of techniques for automated generation, manipulation , and analysis of natural or human languages. Despite most NLP techniques inherit from Linguistics and Artificial Intelligence, they are also affected by relatively newer domains such as Machine Learning.

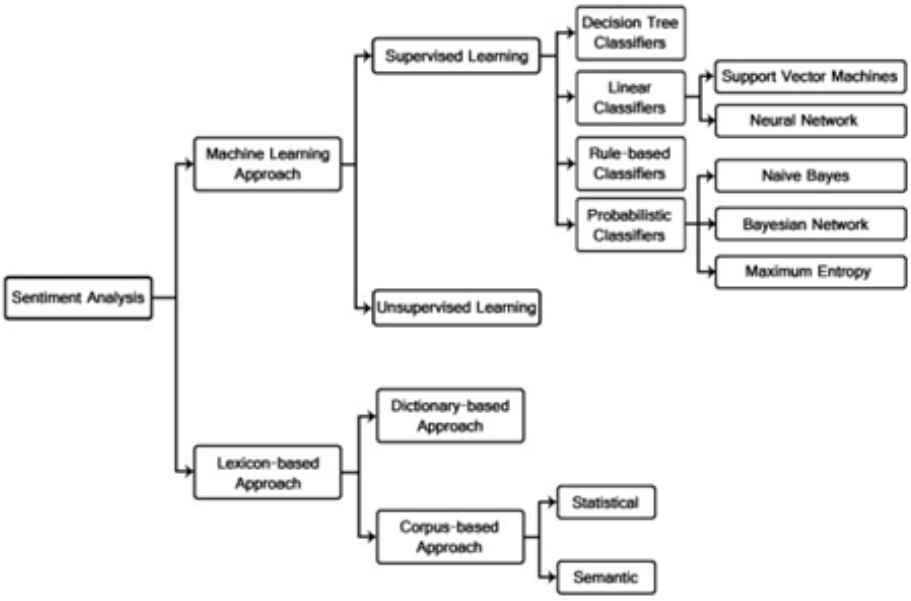

Figure 2.1. Sentiment classification techniques

From the **Figure 2.1** we can see sentiment classification techniques is roughly divided into machine learning approach, lexicon-based and hybrid approach [30]. The Machine Learning Approach (ML) applies the famous ML algorithms and uses linguistic features. The Lexicon-based Approach relies on a sentiment lexicon, a collection of known and pre-compiled sentiment terms. It's divided into the dictionary-based and corpus-based approach, that uses statistical or semantic methods to find sentiment polarity. The hybrid approach is common among sentiment lexicons playing a pivotal role in majority of methods.

2.2.1 Lexicon-based approach

Opinion words are employed in many sentiment classification tasks. Positive opinion words used to express some desired states, while negative opinion words used to express some undesired states. There are also opinion phrases and idioms together are called opinion lexicon. There are three main approaches in order to compile or collect the opinion word list. Manual approach is very time consuming and is not used alone. It is usually combined with the other two automated approaches as a final check to avoid the mistakes which results from automated methods. The two automated approaches are presented in the following subsections.

Dictionary-based approach

A small set of opinion words is collected manually with known orientations. [43] Then, this set is grown by searching in the well-known corpora WordNet [16] or thesaurus [13] for their synonyms and antonyms. The newly found words added to the seed list, later the iteration starts. The iterative process stops when no new words are found. After the process is completed, a manual inspection is carried, to remove or correct errors.

The dictionary-based approach has a major disadvantage of having inability to find opinion words with domain and context-specific orientations. Qiu and He [11] used a dictionary-based approach to identify sentiment sentences in contextual advertising. They proposed an advertising strategy to improve ad relevance and user experience. They used syntactic parsing and a sentiment dictionary and proposed a rule-based approach to tackle topic word extraction and consumers' attitude identification in advertising keyword extraction. They worked on web forums from automotvieforums.com. Their results demonstrated the effectiveness of the proposed approach in advertising keyword extraction and ad selection.

Corpus-based approach The Corpus-based approach helps to solve the problem of finding opinion words with context-specific orientations. Its methods depend on syntactic patterns or patterns that occur together along with a seed list of opinion words to find other opinion words in a large corpus. Using the corpus-based approach

alone is not as effective as the dictionary-based approach because it is hard to prepare huge corpus to cover all English words. This approach has major advantage that can help finding domain and context-specific opinion words and their orientations using a domain corpus.

Statistical approach

Finding co-occurrence patterns or seed opinion words can be done using statistical techniques. It could be done by deriving posterior polarities using the co-occurrence of adjectives in a corpus,. It is possible to use the entire set of indexed documents on the web as the corpus for the dictionary construction. It overcomes the problem of the unavailability of some words if the used corpus is not large enough [34].

The polarity of a word can be identified by studying the occurrence frequency of a word in a large annotated corpus of texts [23]. If the word occurs more frequently among positive texts, then its polarity is positive. If it occurs more frequently among negative texts, then its polarity is negative. If it has equal frequencies, then it is a neutral word.

Semantic approach

The Semantic approach gives sentiment values directly and relies on different principles for computing the similarity between words. This principle gives similar sentiment values to semantically close word. WordNet for example, provides different kind of semantic relationships between words used to calculate sentiment polarities. It can be used for obtaining a list of sentiment words by iteratively expanding the initial set with synonyms and antonyms and then determining the sentiment polarity for an unknown word by the relative count of positive and negative synonyms of this word.

2.2.2 Machine learning approach

The machine learning approach relies on the famous ML algorithms to solve the sentiment analysis as regular text classification problem that use syntactic or linguistic features.

Text Classification Problem Definition: We have a set of training records $D = \{x_1, ...,x_n\}$ where each record is labeled to a class. The classification model is related to the features in the underlying record to one of the class labels. Then for a given instance of an unknown class, the model is used to predict a class label. The hard classification problem is when only one label is assigned to an instance. The soft classification problem is when a probabilistic value of labels is assigned to an instance.

Supervised learning

The supervised learning methods depend on the existence of labeled training documents. There are many kinds of supervised classifiers in literature. In the next subsections, we present details of the most frequently used classifiers in sentimental analysis.

Probabilistic classifiers

Probabilistic classifiers use mixture models for classification. The mixture model assumes that each class is a component of the mixture. Each mixture component is a generative model that provides the probability of sampling a particular term for that component. These kinds of classifiers are also called generative classifiers. Three of the most famous probabilistic classifiers are discussed in the next subsections.

Naïve Bayes Classifier

The Naïve Bayes classifier is the simplest and most commonly used classifier. Naïve Bayes classification model computes the posterior probability of a class, based on the distribution of the words in the document. The model works with the bag of words feature extraction which ignores the position of the word in the document. It uses Bayes Theorem to predict the probability that a given feature set belongs to a particular label.

$$P(label|features) = P(label) * P(features|label)P(features)P(label)$$

is the prior probability of a label or the likelihood that a random feature set the label. P(features|label) is the prior probability that a given feature set is being

classified as a label. P(features) is the prior probability that a given feature set is occurred. Given the Naïve assumption which states that all features are independent, the equation could be rewritten as follows:

$$P(label|features) = P(label) * P(f_1|label) * ..... * P(f_n|label)P(features)$$

An improved NB classifier was proposed by Kang and Yoo [17] to solve the problem of the tendency for the positive classification accuracy to appear up to approximately 10% higher than the negative classification accuracy. This creates a problem of decreasing the average accuracy when the accuracies of the two classes are expressed as an average value. They showed that using this algorithm with restaurant reviews narrowed the gap between the positive accuracy and the negative accuracy compared to NB and SVM. The accuracy is improved in recall and precision compared to both NB and SVM.

Bayesian Network (BN)

The main assumption of the NB classifier is the independence of the features. The other extreme assumption is to assume that all the features are fully dependent. This leads to the Bayesian Network model which is a directed acyclic graph whose nodes represent random variables, and edges represent conditional dependencies. BN is considered a complete model for the variables and their relationships. Therefore, a complete joint probability distribution (JPD) over all the variables, is specified for a model. In Text mining, the computation complexity of BN is very expensive; that is why, it is not frequently used [8].

Maximum Entropy Classifier

The Maxent Classifier (known as a conditional exponential classifier) converts labeled feature sets to vectors using encoding. This encoded vector is then used to calculate weights for each feature that can then be combined to determine the most likely label for a feature set. This classifier is parameterized by a set of X{weights}, which is used to combine the joint features that are generated from a feature-set by an X{encoding}. In particular, the encoding maps each C{(feature set, label)} pair to a

vector. The probability of each label is then computed using the following equation:

$$P(fs \mid label) = \frac{dotprod(weights, encode(fs, label))}{sum(dotprod(weights, encode(fs, l))forlinlabels)}$$

ME classifier was used by Kaufmann [24] to detect parallel sentences between any language pairs with small amounts of training data. The other tools that were developed to automatically extract parallel data from non-parallel corpora use language specific techniques or require large amounts of training data. Their results showed that ME classifiers can produce useful results for almost any language pair. This can allow the creation of parallel corpora for many new languages.

Linear classifiers

Given $\bar{X}$=$\{x_1 \ldots\ldots x_n\}$ is the normalized document word frequency, vector $\bar{A}$ =$\{a_1 \ldots\ldots a_n\}$ is a vector of linear coefficients with the same dimensionality as the feature space, and b is a scalar; the output of the linear predictor is defined as p=$\bar{A}.\bar{X}$ + b, which is the output of the linear classifier. The predictor p is a separating hyperplane between different classes. There are many kinds of linear classifiers; among them is Support Vector Machines (SVM) [9] which is a form of classifiers that attempt to determine good linear separators between different classes. Two of the most famous linear classifiers are discussed in the following subsections.

Support Vector Machines Classifiers (SVM)

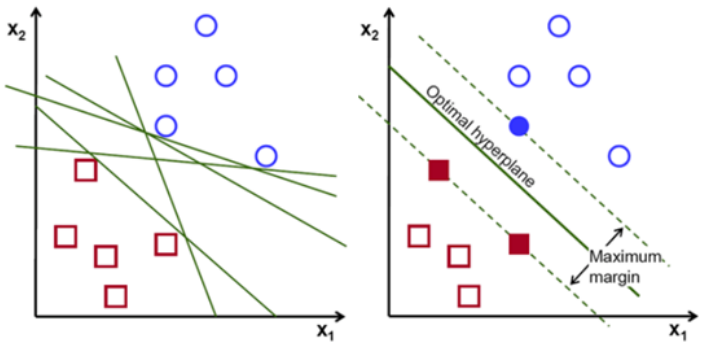

Figure 2.2. Possible hyperplanes

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.

To separate the two classes of data points, there are many possible hyperplanes as shown in Figure 2.2. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Neural Network

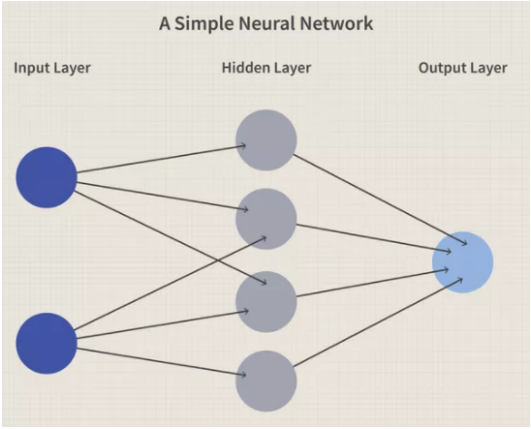

Figure 2.3. Neural Network Architecture

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics a way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial. A neural network contains layers of interconnected nodes. Each node is known as a perceptron and is similar to multiple linear regression. The perceptron feeds the signal produced by a multiple linear regression into an activation function that may be nonlinear.

Decesion tree

Decision Tree is a Supervised learning technique that can be used for classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and the Leaf Node. Decision nodes are used to make any decision and have multiple

branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

For predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of the root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node. For the next node, the algorithm again compares the attribute value with the other sub-nodes and moves further. It continues the process until it reaches the leaf node of the tree.

Rule-based classifiers

In rule-based classifiers, the data space is modeled with a set of rules. The left-hand side represents a condition on the feature set expressed in disjunctive normal form while the right-hand side is the class label. The conditions are on the term presence. Term absence is rarely used because it is not informative in sparse data.

There are several criteria to generate rules, the training phase constructs all the rules depending on these criteria. The most two common criteria are support and confidence [7]. The support is the absolute number of instances in the training data set which are relevant to the rule. Confidence refers to the conditional probability that the right-hand side of the rule is satisfied if the left-hand side is satisfied. Some combined rule algorithms were proposed in [20].

2.3 DEEP LEARNING

The driving force behind this thesis mainly incorporates a deep learning framework in the form of neural network architecture that we will explore in-depth. But before we diving deep into the actual picture, let's revisit some history behind this marvelous field.

2.3.1 Introduction

Big data has revolutionized the modern business environment in recent years. Big data is a collection of information where organizations can mine for business purposes

through machine learning, predictive modeling, and other advanced data analytics applications. At one time, the concept of big data may have seemed like a buzzword, but the reality is the impact of big data on the world around us has been tremendous. The term Big Data was used for the first time in 2005 by Roger Mougalas from O'Reilly Media [19]. It purely denotes huge amount of data collected by big companies daily.

In 1663, John Graunt dealt with "overwhelming amounts of information" [14], while he studied the bubonic plague, which was currently ravaging Europe. Graunt used statistics and is credited with being the first person to use statistical data analysis. In the early 1800s, the field of statistics expanded to include collecting and analyzing data. The evolution of Big Data includes several preliminary steps for its foundation, and while looking back to 1663 isn't necessary for the growth of data volumes today, the point remains that "Big Data" is a relative term depending on who is discussing it.

AI has been part of our imaginations and simmering in research labs since a handful of computer scientists rallied around the term at the Dartmouth Conferences in 1956 and birthed the field of AI. [47] If AI is our vehicle, then big data is the fuel to keep the engines pumping. AI helps us to understand Big Data by providing proper insight into the pattern.

On the other hand, Machine learning at its most basic is the practice of using algorithms to parse data, learn from it, and then decide or prediction about something in the world. So rather than hand-coding software routines with a specific set of instructions to accomplish a particular task, the machine is "trained" using large amounts of data and algorithms that give it the ability to learn how to perform a task.

Machine learning came directly from the minds of the early AI crowd, and the algorithmic approaches over the years included decision tree learning and inductive logic programming,clustering, reinforcement learning, and Bayesian networks among others. As we know, none achieved the ultimate goal of General AI, and even Narrow AI was mostly out of reach with early machine learning approaches.

2.3.2 What is Deep Learning?

Deep Learning can be thought as the evolution of Machine Learning which takes inspiration from the functioning of the human brain.[49] Deep Learning is used to solve complex problems where the data is huge, diverse, less structured. Deep learning models are built on top of Artificial Neural Networks, which mimic how the human brain works.

Let's look at the basic functioning of our brain to understand how Neural Networks work. Our human brain has neurons which are the basic functional units of our brain. The neurons transmit information to other nerve cells, muscles, and glands and receive input from other neurons, enabling the brain to make decisions. Our brain continuously learns from inputs from the environment and previous experiences and makes the best possible decision in every scenario. This is pretty much what Deep Learning does. It learns progressively from raw data and previous experiences and corrects itself without explicit programming.

Although Deep Learning was conceptualized in the 1980s, researchers had 2 major constraints when it comes to implementing Deep Learning models. Deep learning models require abundant data and very high computational power. As the data increases, the depth of the neural network increases and the learning becomes deep. That is the essence of Deep Learning. Another significant advantage of Deep Learning is that, as the model trains, it learns to extract features on its own and we don't have to do manual feature extraction similar to other Machine Learning algorithms.

2.3.3 Biological inspiration of Neural Networks

A neuron (nerve cell) is the basic building block of the nervous system. A human brain consists of billions of neurons that are interconnected to each other. They are responsible for receiving and sending signals from the brain. As seen in the below diagram, a typical neuron consists of the three main parts – dendrites, an axon, and cell body or soma. Dendrites are tree-like branches originating from the cell body. They receive information from the other neurons. Soma is the core of a neuron. It is responsible for processing the information received from dendrites. Axon is like a

cable through which the neurons send the information. Towards its end, the axon splits into many branches that make connections with other neurons through their dendrites. The connection between the axon and other neuron dendrites is called synapses.

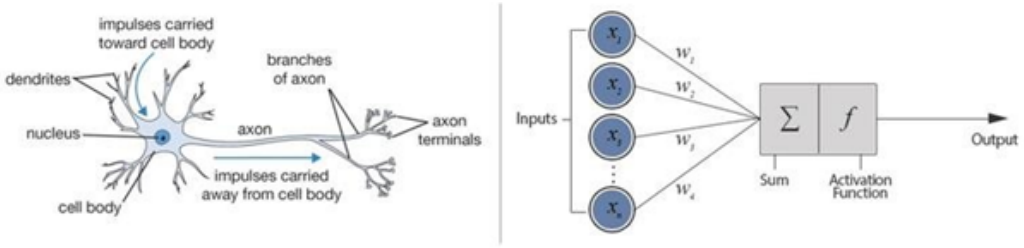

Figure 2.4. Biological Neuron Vs Artificial Neural Network

As ANN, is inspired by the functioning of the brain[15], let us see how the brain works. The brain consists of a network of billions of neurons. They communicate by means of electrical and chemical signals through a synapse, in which the information from one neuron is transmitted to other neurons. The transmission process involves an electrical impulse called action potential. For the information to be transmitted, the input signals (impulse) should be strong enough to cross a certain threshold barrier then only a neuron activates and thus transmits the signal further (output).

2.3.4 Working of a Perceptron:

In the first step, all the input values are multiplied with their respective weights and added together. The result obtained is called weighted sum :

$$\sum_{i=1}^{n} W_i.x_i$$

This sum gives an appropriate representation of the inputs based on their importance. Additionally, a bias term b is added to this sum.

$$\sum_{i=1}^{n} W_i.x_i + b$$

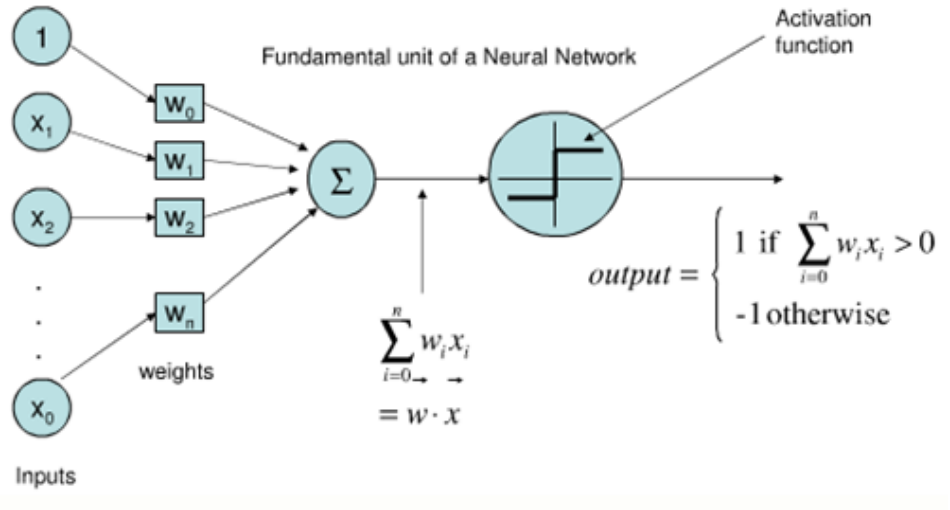

Figure 2.5. Working of Perceptron

Bias serves as another model parameter (in addition to weights) that can be tuned to improve the model's performance. In the second step, an activation function f is applied over the above sum to obtain output.

$$Y = f(\sum_{i=1}^{n} W_i.x_i + b)$$

Depending upon the scenario and the activation function used, the Output is either binary 1, 0 or a continuous value.

2.3.5 Activation Functions

A biological neuron only fires when a certain threshold exceeds. Similarly, the artificial neuron will only fire when the sum of the inputs (weighted sum) exceeds a certain threshold value, let's say 0. Intuitively, we can think of a rule-based approach like this:

if $\sum_{i=1}^{n} W_i.x_i + b > 0$ **then**
    $output \leftarrow 1$;
else
    $output \leftarrow 0$;
end

This is a unit step(threshold) activation function that was originally used by Rosenblatt [39]. But as you can see, this function is discontinuous at 0, so it causes problems in mathematical computations. A smoother version of the above function is the sigmoid function. It outputs between 0 and 1. Another one is the Hyperbolic tangent(tanh) function, which produces the output between -1 and 1. Both sigmoid and tanh functions suffer from vanishing gradients problems. Nowadays, ReLU and Leaky ReLU are the most popularly used activation functions. **Figure 2.6** describes various activation functions.

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU)[2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU)[3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1+e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Figure 2.6. Types of Activation functions

2.3.6 Loss Functions

The loss function is used to measure how good or bad the model is performing. It is used to compute to estimate the prediction given by the model in terms of generalizability. Loss function is classified into two categories that is classification and regression Loss. Classification loss is the case where the aim is to predict the output from the different categorical values. Whereas if the problem is regression if the value is continous.

CLASSIFICATION LOSSES

Cross-Entropy Loss / Log Loss: It computes the performance of classification tasks where results lie between probability values 0 and 1. As the predicted probability disunites from the true label, cross-entropy loss increases. Log loss of 0 is considered to be a perfect model. Both cross-entropy and log loss are a bit different from each other when we are computing errors between 0 and 1.

$$Loss = -\frac{1}{n}\sum_{i=1}^{n}(Y_i.log\bar{Y_i} + (1 - Y_i).log(1 - \bar{Y_i})$$

Hinge Loss: Another loss for binary classification task is the hinge loss function that was initially developed to use with the support vector machine models[3]. It is recommended to be used where the target labels are in (-1,1) in binary classification tasks. Hinge loss makes the examples have the right sign, allocating more error when there is dissimilarity in the sign of the true label and predicted label.

$$Loss = max(0, 1 - y * f(x))$$

REGRESSION LOSSES

Mean Square Loss: It is more often used regression loss that is computed by taking the average squared difference between actual and predicted observations. It mainly takes average magnitude of error into consideration, ignoring the direction. Due to squaring, the predictions that are distant from the true values are penalized laboriously capared to less diverged predictions. It is easy to compute gradients because of the mathematical properties there in L2 Loss.

$$MSE = -\frac{1}{n}\sum_{i=1}^{n}(Y_i - \bar{Y_i})^2$$

Mean Absolute Error: It is computed by taking the average of the sum of absolute differences between the true and predicted variables. Similar to MSE, it also calculates magnitude ignoring the direction. It is tough to compute the gradients in MAE as there is a need for linear programming. MAE does not use square. MSE is more sensitive to outliers than MAE.

$$MAE = -\frac{1}{n}\sum_{i=1}^{n}|x_i - x|$$

2.3.7 Optimizer

While training the deep learning model, we need to modify each epoch's weights and minimize the loss function. An optimizer is a function or an algorithm that modifies the attributes of the neural network, such as weights and learning rate. Thus, it helps in reducing the overall loss and improve the accuracy. The issue by choosing the correct weight for the model is a daunting task, as a deep learning model generally consists of millions of parameters. It raises the need to select a suitable optimization algorithm for your application.

Introduction to Gradient Descent

Using the Gradient Decent optimization algorithm, the weights are updated incrementally after each epoch.

The magnitude and direction of the weight update are computed by taking a step in the opposite direction of the cost gradient[38]

$$\Delta w_j = -\eta \frac{\delta J}{\delta w_j}$$

where η is the learning rate. The weights are then updated after each epoch via the following update rule :

$$w = w + \delta w$$

where δw is a vector that contains the weight updates of each weight coefficient w, which are computed as follows:

$$\Delta w_j = -\eta \frac{\delta J}{\delta w_j}$$

$$= -\eta \sum_{i=1}^{n} (target^i - output^i)(-x_j^i)$$

$$= \eta \sum_{i=1}^{n} (target^i - output^i)(x_j^i)$$

Essentially, we can picture Gradient Descent optimization as a hiker (the weight coefficient) who wants to climb down a mountain (cost function) into a valley (cost

minimum), and each step is determined by the steepness of the slope (gradient) and the leg length of the hiker (learning rate). Considering a cost function with only a single weight coefficient, we can illustrate this concept as shown in **Figure 2.7**.

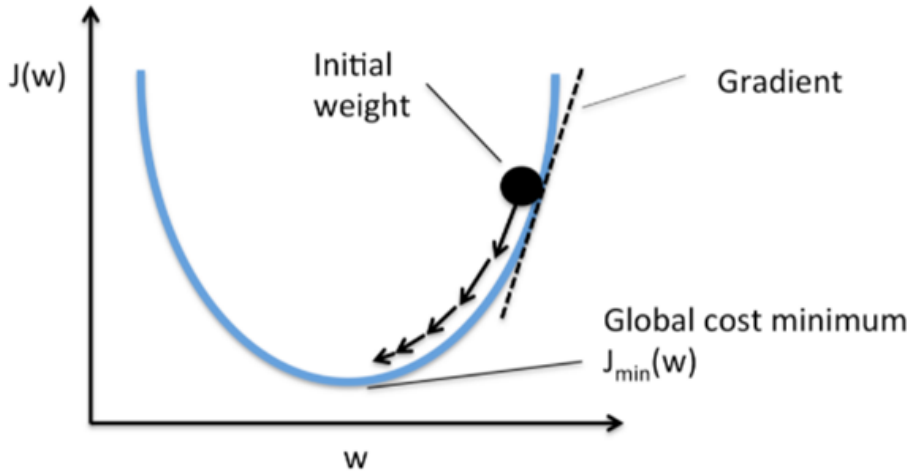

Figure 2.7

There are other variants of Gradient Descent such as Stochastic Gradient Descent (SGD), Mini-Batch Gradient Descent (MB-GD) etc

Adagrad

Adagrad adapts the learning rate specifically to individual features; that means that some of the weights in your dataset will have different learning rates than others.[53] This works well for sparse datasets where a lot of input examples are missing. Adagrad has a major issue. The adaptive learning rate tends to get small over time. Some other optimizers below seek to eliminate this problem.

RMSprop

RMSprop is a special version of Adagrad developed by Professor Geoffrey Hinton in his neural nets class. Instead of letting all of the gradients accumulate for momentum, it only accumulates gradients in a fixed window. RMSprop is similar to Adaprop, is another optimizer that seeks to solve some of the issues that Adagrad leaves open.

Adam

Adam stands for adaptive moment estimation and is another way of using past gradients to calculate current gradients[25]. Adam also utilizes the concept of momentum by adding fractions of previous gradients to the current one. This optimizer has become widespread and is used to train neural nets.

2.3.8 Backpropogation

Backpropagation is the essence of neural network training. It is the method of fine-tuning the weights of a neural network based on the error rate obtained in the previous epoch (i.e., iteration). Proper tuning of the weights allows you to reduce error rates and make the model reliable by increasing its generalization.

Backpropagation in a neural network is a short form for "backward propagation of errors." It is a standard method of training artificial neural networks. This method helps calculate the gradient of a loss function concerning all the weights in the network.[12]

How Backpropagation Algorithm Works

The Back propagation algorithm in a neural network computes the gradient of the loss function for a single weight by the chain rule. It efficiently computes one layer at a time, unlike a native direct computation. It computes the gradient and it does not define how the gradient is used. It generalizes the computation in the delta rule.

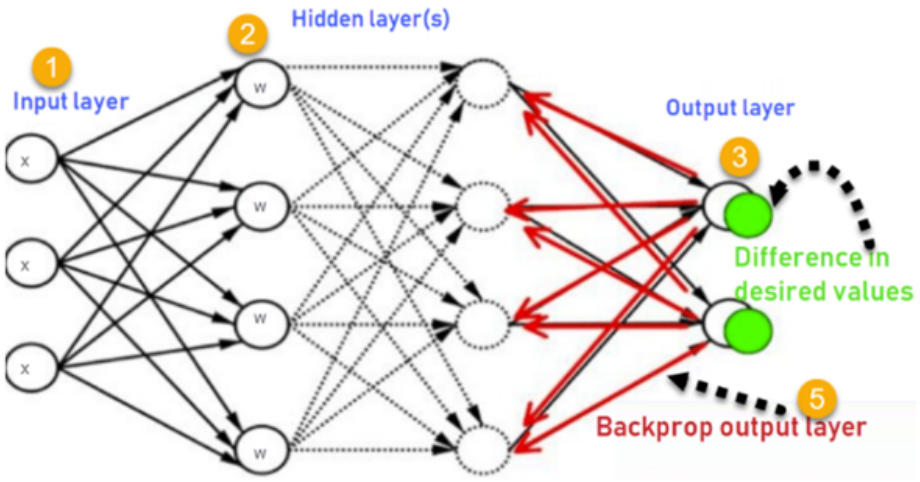

Figure 2.8. Working of Backpropogation

- Inputs X, arrive through the preconnected path
- Input is modeled using real weights W. The weights are usually randomly selected.
- Calculate the output for every neuron from the input layer, to the hidden layers, to the output layer.
- Calculate the error in the outputs:

$$ErrorB = ActualOutput - DesiredOutput$$

- Travel back from the output layer to the hidden layer to adjust the weights such that the error is decreased.

2.4 Convolutional neural network (CNN)

In deep learning, a convolutional neural network (CNN/ConvNet) is a class of deep neural networks, most commonly applied to analyze visual imagery [31]. When we think of a neural network, we regard it to be matrix multiplications, this is not the case with ConvNet. It uses a special technique called Convolution. In mathematics, convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

2.4.1 How does CNN work?

An RGB image is nothing but a matrix of pixel values having three planes whereas, a grayscale image is the same but it has a single plane. Take a look at this image to understand more

We take a filter/kernel(3×3 matrix) and apply it to the input image to get the convolved feature. This convolved feature is passed to the next layer. Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and outputs an activation value.

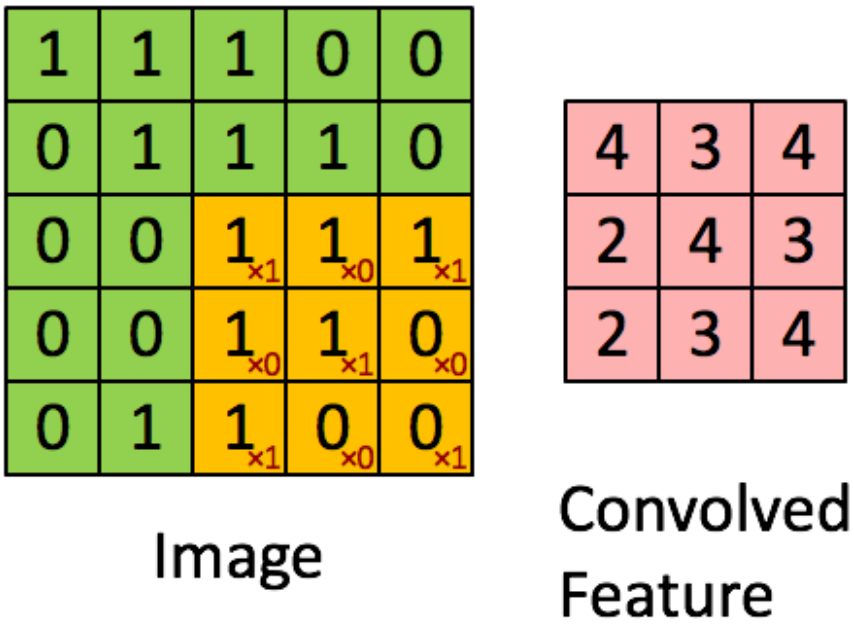

Figure 2.9. Convolution Layer

When you input an image in a ConvNet, each layer generates several activation functions that are passed to the next layer.

The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed to the next layer that detects more complex features such as corners or combinational edges. As we move deeper into the network, it can identify even more complex features such as objects, faces, etc.

2.4.2 Padding

Padding is a term relevant to convolutional neural networks as it refers to the amount of pixels added to an image when it is being processed by a kernel of a CNN. For example, if the padding in a CNN is set to zero, then every pixel value that is added will be of value zero. Padding works by extending the area of which a convolutional neural network processes an image. The kernel is a neural network filter that moves across the image, scanning each pixel and converting the data into a smaller, or sometimes larger, format. In order to assist the kernel with processing the image,

padding is added to the frame of the image to allow for more space for the kernel to cover the image. Adding padding to an image processed by a CNN allows for more accurate analysis of images.

2.4.3 What's a pooling layer?

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data by reducing the dimensions. There are two types of pooling average pooling and max pooling.

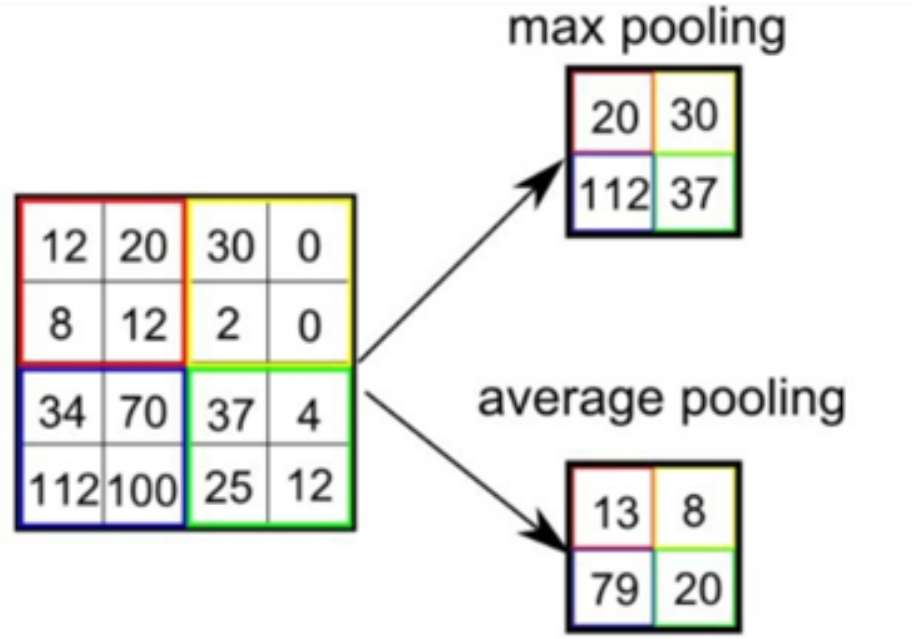

Figure 2.10. Pooling Layer

2.4.4 Max pooling

So what we do in Max Pooling is to find the maximum value of a pixel from a portion of the image covered by the kernel. Max Pooling also performs as a Noise Suppressant. It discards the noisy activations and also executes de-noising along with dimensionality reduction.

2.4.5 Average Pooling

On the other hand, average pooling returns the average of all values from the portion of the image covered by the Kernel. Average pooling performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs more promising than Average Pooling.

2.5 Recurrent neural network (RNN)

A recurrent neural network (RNN) is a special type of artificial neural network adapted to work for time series data or data that involve sequences. Ordinary feed-forward neural network is meant for data points especially independent of each other. However if we have data in a sequence such that one data point depends upon the previous data point, we need to modify the neural network to incorporate the dependencies between these data points. RNNs have the concept of 'state that helps them store the states or information of previous inputs to generate the next output of the sequence.[48]

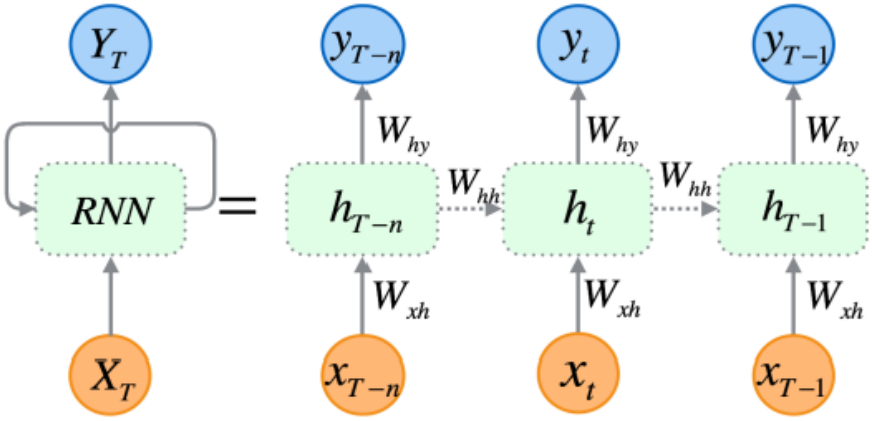

Figure 2.11. RNN architecture

The formula for the current state can be written as:

$$h_t = f(h_{t-1}, x_t)$$

Here, h_t is the new state, h_{t-1} is the previous state while x_t is the current input. We

now have a state of the previous input instead of the input because the input neuron would have applied the transformations on our previous input. So each successive input is called a time step.

Taking the simplest form of a recurrent neural network, let's say that the activation function is tanh, the weight at the recurrent neuron is W_{hh} and the weight at the input neuron is W_{xh}, we can write the equation for the state at time t as :

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

The Recurrent neuron in this case is just taking the immediate previous state into consideration. For longer sequences the equation can involve multiple such states. Once the final state is calculated we can go on to produce the output Now, once the current state is calculated we can calculate the output state as:

$$y_t = W_{h_y}h_t$$

RNNs suffer from vanishing gradient problems when we ask them to handle long term dependencies. They also become severely difficult to train as the number of parameters become extremely large. If we unroll the network, it becomes so huge that its convergence is a challenge.

2.5.1 LSTM & GRU

Long Short Term Memory networks – usually called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber[21]. They work tremendously well on a large variety of problems, and are now widely used. LSTMs also have this chain like structure, but the repeating module has a slightly different structure. Instead of having a single neural network layer, there are multiple layers, interacting in a very special way. They have an input gate, a forget gate and an output gate.

Another efficient RNN architecture is the Gated Recurrent Units i.e. the GRUs[28]. They are a variant of LSTMs but are simpler in their structure and are easier to train. Their success is primarily due to the gating network signals that control how

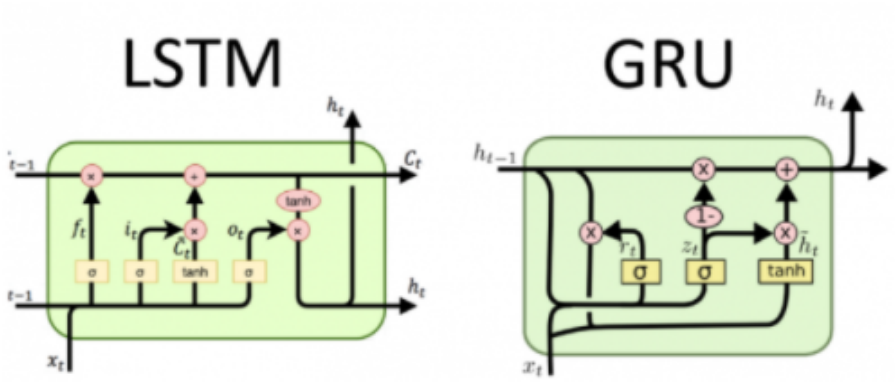

Figure 2.12. LSTM vs GRU cell

the present input and previous memory are used, to update the current activation and produce the current state. These gates have their own sets of weights that are adaptively updated in the learning phase. We have just two gates, the reset and the update gate.

2.5.2 Sequence to Sequence Models

Sequence-to-sequence (seq2seq) models in NLP are used to convert sequences of Type A to sequences of Type B. For example, translation of English sentences to Italian sentences is a sequence-to-sequence task.

Recurrent Neural Network (RNN) based sequence-to-sequence models have gained a lot of attraction since it was introduced in 2014. Most of the data in the current world are in the form of sequences . It can be a number sequence, text sequence, a video frame sequence or an audio sequence.

The performance of these seq2seq models was further enhanced with the addition of the Attention Mechanism in 2015 [32]. These sequence-to-sequence models are pretty versatile and is used in a variety of NLP tasks, such as Machine Translation, Text Summarization, Speech Recognition ,Question-Answering System, and so on[51].

From Figure 2.13 seq2seq model is converting a German phrase to its English counterpart. Both Encoder and Decoder are RNNs. At every time step in the Encoder, the RNN takes a word vector x_i from the input sequence and a hidden

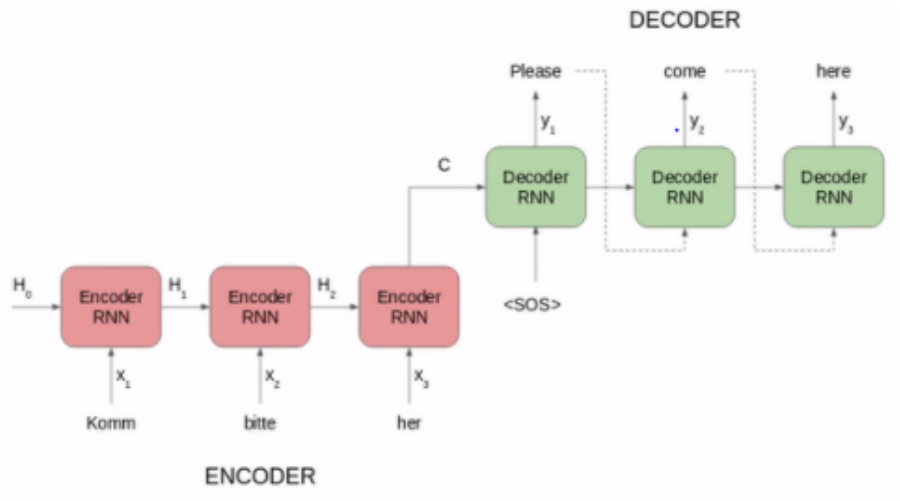

Figure 2.13. Sequence to Sequence architecture

state H_i from the previous time step. The hidden state is updated at each time step. The hidden state from the last unit is known as the context vector. This contains information about the input sequence. This context vector is then passed to the decoder and it is used to generate the target sequence (English phrase). If we use the Attention mechanism, then the weighted sum of the hidden states is passed as the context vector to the decoder. Despite being so good at what it does, there are certain limitations of seq2seq models with attention:

- Dealing with long-range dependencies is still challenging
- The sequential nature of the model architecture prevents parallelization. These challenges are addressed by Google Brain's Transformer.

2.6 Transformers

The Transformer in NLP is a novel architecture that aims to solve sequence-to-sequence tasks while handling long-range dependencies with ease. It relies entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution.

The Encoder block has one layer of a Multi-Head Attention followed by another layer of Feed Forward Neural Network. The decoder on the other hand, has an extra

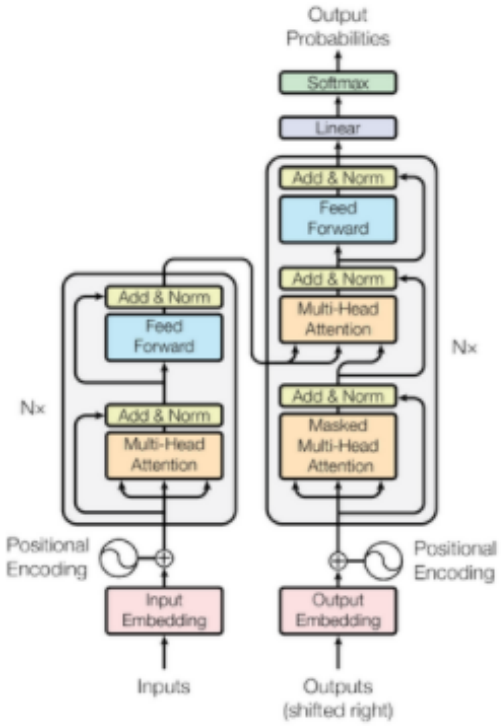

Figure 2.14. The Encoder Block of the Transformer Architecture

Masked Multi-Head Attention. The encoder and decoder blocks are actually multiple identical encoder and decoder stacked on top of each other. The encoder stack and the decoder stack have the same number of units. In addition to the self-attention and feed-forward layers, the decoders also have one more layer of Encoder-Decoder Attention layer. This helps the decoder focus on the relevant parts of the input sequence.

2.6.1 Self-Attention

"Self-attention, sometimes called intra-attention, is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence."

- The first step in calculating self-attention is to create three vectors from each of the encoder's input vectors (in this case, the embedding of each word). So for each word, we create a Query vector, a Key vector, and a Value vector.These

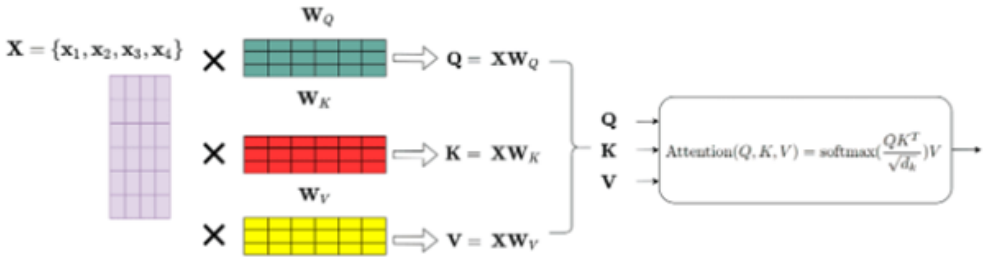

Figure 2.15. Self attention layer

vectors are created by multiplying the embedding by three matrices that was trained during the training process.

- The second step in calculating self-attention is to calculate a score The score determines how much focus to place on other parts of the input sentence as we encode a word at a certain position. The score is calculated by taking the dot product of the query vector with the key vector of the respective word we're scoring.
- The third and fourth steps are to divide the scores by the square root of the dimension of the key vectors. The result is passed through softmax. Softmax normalizes the scores so they're all positive and add up to 1. This softmax score determines how much each word will be expressed at this position. Clearly the word at this position will have the highest softmax score, but sometimes it's useful to attend to another word that is relevant to the current word.
- The fifth step is to multiply each value vector by the softmax The intuition here is to keep intact the values of the word(s) we want to focus on, and drown-out irrelevant words
- The sixth step is to sum up the weighted value vectors. This produces the output of the self-attention layer at this position

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Self-attention is computed not once but multiple times in the Transformer's architecture parallelly and independently. It is therefore referred to as Multi-head Attention.

The outputs are concatenated and linearly transformed as shown in the **Figure 2.16**:

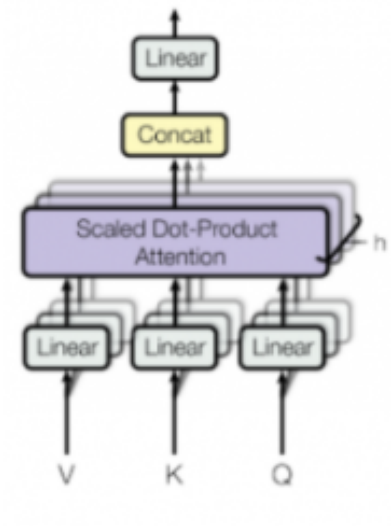

Figure 2.16. Multihead Attention layer

2.6.2 BERT

BERT is an acronym for Bidirectional Encoder Representations from Transformers. The BERT architecture is composed of several Transformer encoders stacked together. Further, each Transformer encoder is composed of two sub-layers: a feed-forward layer and a self-attention layer.

BERT uses a Transformer that learns contextual relations between words in a sentence/text[26]. The transformer includes two separate mechanisms: an encoder that reads the text input and a decoder that generates a prediction for any given task. BERT makes use of only the encoder as its goal is to generate a language model. In contrast to state-of-the-art models, the Transformer encoder reads the entire sentence at once as it is bidirectional and thus more accurate. The bidirectional characteristic allows the model to learn all surroundings (right and left of the word) of words to better understand the context.

BERT's Architecture

The BERT architecture builds on top of Transformer. We currently have two variants

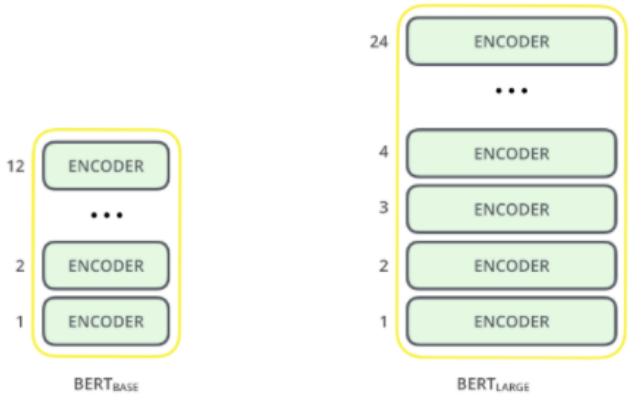

Figure 2.17. BERT Architecture

available:

- BERT Base: 12 layers (transformer blocks), 12 attention heads, and 110 million parameters
- BERT Large: 24 layers (transformer blocks), 16 attention heads and, 340 million parameters

2.6.3 Text Preprocessing

Every input embedding is a combination of 3 embeddings:

1. **Position Embeddings**: BERT learns and uses positional embeddings to express the position of words in a sentence. These are added to overcome the limitation of Transformer. Unlike an RNN, it is not able to capture "sequence" or "order" information.
2. **Segment Embeddings**: BERT can also take sentence pairs as input for tasks (Question-Answering). That's why it learns a unique embedding for the first and the second sentences to help the model distinguish between them. In the above example, all the tokens marked as EA belong to sentence A (and similarly for EB).
3. **Token Embeddings**: These are the embeddings learned for the specific token from the WordPiece token vocabulary.

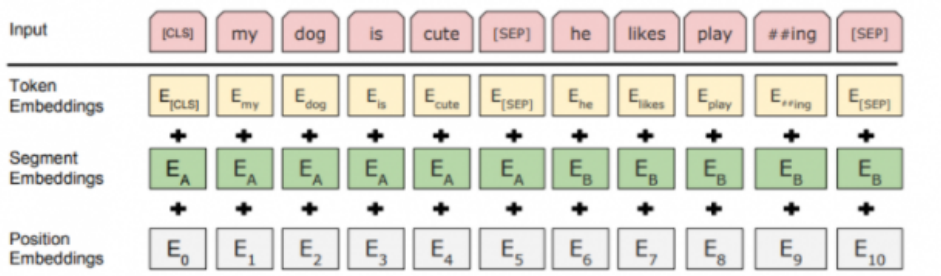

Figure 2.18. Text Preprocessing

For a given token, its input representation is constructed by summing the corresponding token, segment, and position embeddings.

2.6.4 Training

When training language models, there exists a challenge of defining a prediction goal. Many models predict the next word in a sequence a directional approach which inherently limits context learning. To overcome this challenge, BERT uses two training strategies:

Masked LM(MLM)

Before feeding word sequences into BERT, 15% of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence. In technical terms, the prediction of the output words requires:

- Adding a classification layer on top of the encoder output.
- Multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension.
- Calculate the probability of each word in the vocabulary with softmax.

The BERT loss function takes only prediction of the masked values into consideration and ignores the prediction of the non-masked words. As a consequence, the model converges slower than directional models, a characteristic which is offset by its

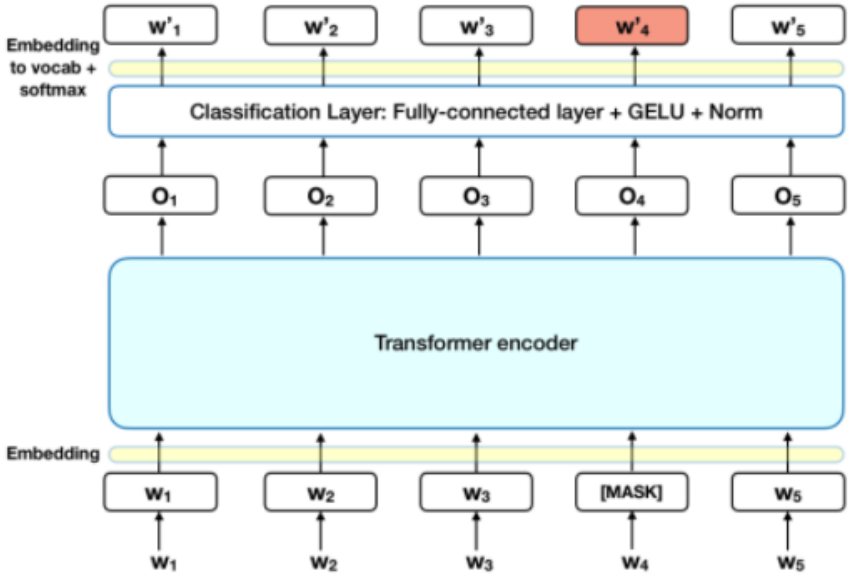

Figure 2.19. Bidirectional Encoder Representation

increased context-awareness. In the BERT training process, the model receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the original document. During training, 50% of the inputs are a pair in which the second sentence is the subsequent sentence in the original document, while the other 50% a random sentence from the corpus is chosen as the second sentence. The assumption is that the random sentence will be disconnected from the first sentence.To help the model distinguish between the two sentences in training, the input is processed in the following way before entering the model:

- A [CLS] token is inserted at the beginning of the first sentence and a [SEP] token is inserted at the end of each sentence.
- A sentence embedding indicating Sentence A or Sentence B is added to each token. Sentence embeddings are similar in concept to token embeddings with a vocabulary of 2.
- A positional embedding is added to each token to indicate its position in the sequence.

To predict if the second sentence is indeed connected to the first, the following steps

are performed:

- The entire input sequence goes through the Transformer model.
- The output of the [CLS] token is transformed into a 2×1 shaped vector, using a simple classification layer (learned matrices of weights and biases).
- Calculating the probability of IsNextSequence with softmax.

When training the BERT model, Masked LM and Next Sentence Prediction are trained together, with a goal of minimizing the combined loss function.

2.7 Related Work

Multimodal sentiment analysis is a new dimension of the traditional text-based sentiment analysis, which goes beyond the analysis of texts and includes other modalities such as audio and visual data [29]. It can be bimodal, which includes various combinations of two modalities, or trimodal, which incorporates three modalities [5]. The traditional text-based sentiment analysis has evolved into more complex models of multimodal sentiment analysis to better understand human behaviors due to the vast amount of social media data that is now readily accessible online in the form of videos, audio, and pictures. Adopting fusion techniques is necessary due to the intricacy of assessing text, audio, and visual elements to conduct such a task. The types of textual, auditory, and visual information included in the sentiment analysis depend on how well these features are fused and the classification algorithms applied.

This thesis work draws influences from research work based on this multimodal subject. The authors Akshi Kumar & Geetanjali Garg proposed a multimodal sentiment analysis model to determine the sentiment polarity and score for any incoming tweet that is textual, image, info-graphic, and typographic [27]. Image sentiment scoring was performed using SentiBank and SentiStrength.Textual sentiment scoring was done using a context-aware hybrid (lexicon and machine learning) technique. It also involved separating text from image using an optical character recognizer and then aggregating the independently processed image and text hence determining the

overall polarity. The primary limitation of the model is that the text recognition was restricted by the capability of the Computer Vision API.

Anthony Hu & Seth Flaxman[22] proposed an approach to multimodal sentiment analysis using a deep neural network combining visual analysis and natural language processing to predict the emotional word tags attached by users to their Tumblr posts. The authors demonstrated an architecture where an input image was fed into the inception network and the text was projected through an LSTM. The two modalities are then concatenated and fed into a dense layer that gives the probability distribution of the emotional state of the user.

Using these seminal papers as references for my thesis work involved combining ideas and aggregating them into a new approach. From Akshi Kumar & Geetanjali Garg's work, the notion of extracting text from an image, and Anthony Hu & Seth Flaxman's research involved leveraging LSTM and the inception network for their task. I incorporated state-of-the-art BERT & ResNet/InceptionResNet architecture for classification. The mentioned papers were limited to using data from 2 sources. I extended by utilizing the idea of extracting text from the images as my third feature along with text from the tweets and images. Upon adding more features, the classifier will be able to perform better. More the data better will be the results, which will serve as the ground truth of the investigation.

Chapter 3

Methodology

In this chapter, we will briefly describe the implementation details of our deep learning model. All theoretical concepts, techniques, and algorithms that were discussed in the previous chapter will be implemented practically to address the challenge.

3.1 Data collection

Since the dataset relating to Coronavirus tweets (if available, it's related to textual data but none with images) is not publicly available, we need to scrape the tweets along with images using an open-source python library called **TWINT**. It is an open-source python library used for twitter scraping. We can use Twint to extract data from Twitter without using the Twitter API. Certain features of Twint makes it more useable and unique from other Twitter scraping API. Twitter API has a limit of fetching only 3200 tweets, while Twint has no limit on downloading tweets. It can download almost all the tweets, easy to use, and is very fast. I wrote a small script that extracted about 400,000 tweets along with the images from a period ranging from Jan 10th, 2020 to June 1st, 2021. The most challenging part of this task was to find relevant images that possessed text inside the image. As a result, it involved a lot of manual work in carefully selecting the image that contained text. The final dataset consists of 1086 relevant tweets with it their respective image.

3.2 Class Labeling & Balancing

Sentiment analysis is purely a subjective process at its root. It was not an easy effort to label the classes. It was simple to categorize an image as animals or cars since objective opinion is generally applied to describe things such as observations and decisions based on unbiased analysis. Objectivity is not influenced by an individual's perspective. On the other hand, assigning labels to images with expressions, contradictory tweets, and other elements was challenging. The human decision-maker determines the goal or purpose. For different tasks, different responses will be appropriate. The objective in AI is always subjective. It is up to the project owner to make those decisions. As a result, my decision to designate the classes was entirely on subjectivity.

We aim to train a model that generalizes well for all possible circumstances. So before incorporating any machine or deep learning algorithm, we needed to balance the dataset. It was essential to analyze the class distribution, to rectify the class balancing issues. I balanced the dataset, each consisting of 543 positive labels & 543 negative labels.

3.3 Text Pre-Processing:

Text processing is a technique used in NLP to clean the text and prepare it for the model building. Raw text is versatile and contains unsolicited noise in various forms like emoticons, punctuation, and text written in numerical or special characters. We have to handle these nuisances because machines will never understand characters, it understands only numbers. Some of the text cleaning performed on this dataset involved the removal of URLs, non-ASCII characters, special characters, whitespaces, and HTML tags.

In today's online communication, emojis and emoticons are becoming the universal language to express their inner thoughts in a shortcut manner. Emojis and emoticons play a pivotal role in text analysis by capturing people's emotions and feelings. They might give us insightful information. We need to retain as emojis hold some intrinsic

information, especially in Sentiment Analysis and discarding them might not be an ideal solution. Hence I replaced frequently occurring emoticons with a meaningful word assigned to them.

3.4 Text Extraction

The best open-source tool to extract text from images is **Easy-OCR**. It is a python package that holds PyTorch as a backend handler. EasyOCR does pre-processing steps like gray scaling within its library and later extracts the text. It applies the CRAFT(Character Region Awareness for Text Detection) algorithm for text detection. CRAFT is a scene text detection technique to effectively detect text areas by analyzing each character and the relationship between the characters. The recognition model uses a Convolutional Recurrent Neural Network (CRNN). Easy-OCR comprises three main components: feature extraction currently using Resnet and VGG, sequence labeling performed by LSTM, and decoding done by Connectionist Temporal Classification(CTC) meant for labeling the unsegmented sequence data with RNN.[1] Easy-OCR provides an end-to-end training pipeline to build new OCR models. Easy-OCR is image specific OCR tool. If text is inside the image and the fonts and colors are unorganized, Easy-ocr provides better results. Also, EasyOCR works better with noisy images.

3.5 Model Architecture

Adaptation of deep learning models forms the crux of constructing the model architecture. As discussed, our dataset consists of data from 3 sources. The first is our text from the tweet, second is the image the third is the text extracted from the image. We have to integrate embeddings consisting of text and image data into the final classification block. We can get our text embedding using state-of-the-art BERT concerning images, we can use any CNN variants. Figure 3.1 illustrates our proposed model's schema.

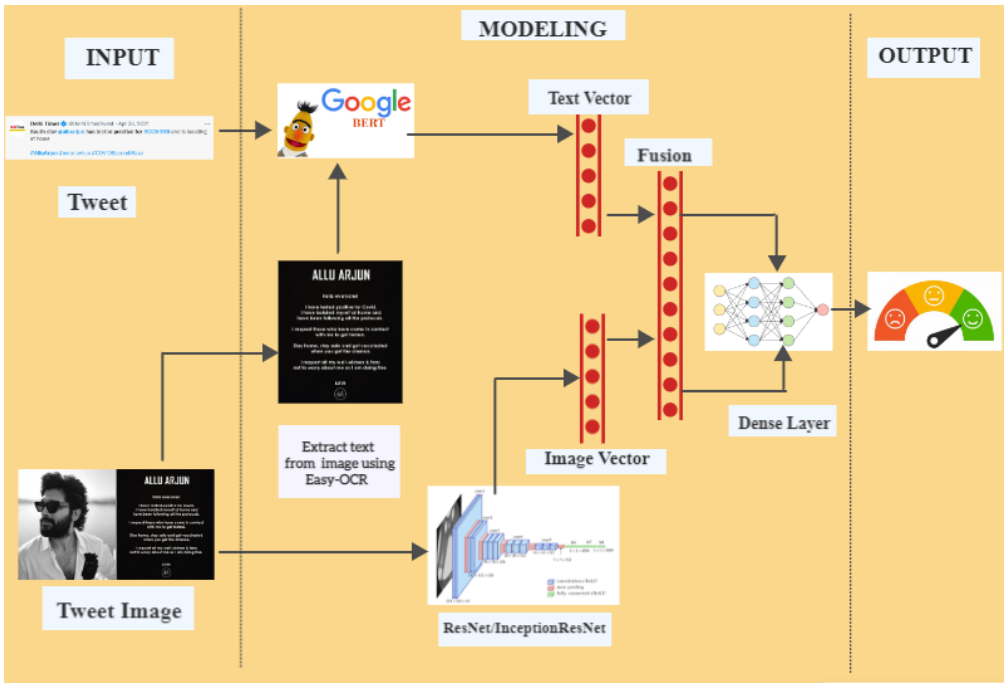

Figure 3.1. Model Schema

3.5.1 Text modality

Firstly text data of the tweets and text extracted from images are concatenated and fed into our BERT model. BERT is a complex neural network architecture containing a large number of parameters that can range from 100 million to over 300 million. So, training a BERT model from scratch would be computationally expensive. So, it is better to use a pre-trained BERT model as a starting point. We can then further train the model on our relatively small dataset. This process is known as model fine-tuning. We can freeze all the layers and attach a neural network layer and train this new model. The weights of only the newly attached layers will get updated during model training.

BERT is empirically powerful and also it is very easy to use which is included in TensorFlow implementation, credit to the hugging face transformer library. There are a lot of different implementations of BERT fine-tuned. I tested two different backbones:

- **ALBERT**

- **ROBERT**

ALBERT(base) is a transformer model pre-trained on a large corpus of English data in a self-supervised fashion. This means it was pre-trained on the raw texts only, with no humans labeling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts. The model learns an inner representation of the English language that can then be used to extract features useful for downstream tasks. ALBERT is particular in that it shares its layers across its Transformer. Therefore, all layers have the same weights. Using repeating layers results in a small memory footprint, however, the computational cost remains similar to a BERT-like architecture with the same number of hidden layers as it has to iterate through the same number of (repeating) layers. ALBERT(base) model comes with 12 repeating layers, 768 hidden, 128 Embedding, 12 attention heads, and 11 million parameters.[45]

ROBERTA(base) is pretrained with the Masked language modeling (MLM) objective. Taking a sentence, the model randomly masks 15% of the words in the input then run the entire masked sentence through the model and has to predict the masked words. This is different from traditional recurrent neural networks (RNNs) that usually see the words one after the other, or from autoregressive models like GPT which internally mask the future tokens. It allows the model to learn a bidirectional representation of the sentence.This way, the model learns an inner representation of the English language that can then be used to extract features useful for downstream tasks. ROBERTA(base) model has 12 encoding layers, 768 hidden state, 12 attention heads and 125 million parameters.[44]

Text inputs need to be transformed to numeric token ids and arranged in several tensors before being fed for classification. Tokenizer transforms text into tokens. The tokens are converted into numbers which is used to build tensors as input to a model. BERT pre-trained tokenizer returns a dictionary consisting of inputs ids and attention mask. Input ids are the indices corresponding to each token in the sentence attention mask indicates whether a token should be attended or not.

In order to extract key information from the context, we provide a word sequence as

input that traverses through all the stacked encoding layers. The input flows through the stacks. The max sequence length of processed text data was 250 but only 16% texts were greater than 200 with 249 being the maximum length of the tweet, so we decided to set max length limit to 250 for the tweets and for text extracted from image, we set the limit to 245. If the sequence of tokens are greater than limit, then the sequence of text will be truncated.

The tokenized and padded input sequence consisting of input Ids and attention masks from tweet and text from images are concatenated. After concatentenation it is passed through the BERT fine-tuned model whose layers are frozen. The obtained tensor is multi-dimensional(3D). We must unstack into a 2D tensor if we wish to employ a dense layer. A flattening layer is an option but when we used it, the resultant tensor size was unusually large, increasing computing complexity and contributing to the overfitting problem. We used the GlobalAveragePooling1D layer to tackle this problem since this downsamples the input representation from 3D to 2D. For each feature dimension, it computes a single average value for each of the input channels, making it more effective. The resulting tensor shrunk to 2 dimensions with shape (batch_size,768).

3.5.2 Image modality

We will incorporate the concept of transfer learning for our image module. Transfer learning is leveraging feature representations from a pre-trained model, so we don't have to train a new model from scratch. Transfer learning is utilized for tasks where dataset has too little data to train a full-scale model from scratch[6]. The general steps involved are:

- Instantiate the convolutional base of any CNN model and load its weights. We loaded "Imagenet" weights since it is a defacto standard for image classification.
- Freeze all layers in the base model.
- Train the model. This process is called feature extraction.
- Use that output as input data for a new classification layer.

For CNN I have used:

- **ResNet50V2**
- **InceptionResNetV2**

ResNet50V2 is a convolutional neural network consists of 25.6M parameters trained on more than a million images from the ImageNet database and is 103 layers deep.ResNet50V2 is a modified version of ResNet50 that performs better than ResNet50 and ResNet101 on the ImageNet dataset. In ResNet50V2, a modification was made in the propagation formulation of the connections between blocks.[42]

Inception-ResNet-v2 is a convolutional neural network consists of 55.9M parameters are trained on more than a million images from the ImageNet database. The network is 449 layers deep and can classify images into 1000 object categories. Inception-ResNet-v2 incorporates residual connections(replacing the filter concatenation stage of the Inception architecture).[2]

The image network block has an input of 4 dimensions with shape (batch_size,200,200,3). Similar to features extraction related to text, we have passed the images to fine-tuned CNN base model to obtain image feature tensor. Since the obtained tensor has to pass through a dense network, it should be of 2 dimensions. Using Flatten layer, the resultant tensor was massive promoting computational complexity and posing an overfitting problem. This underlying issue was resolved with the help of the GlobalAveragePooling2D layer that downsamples the input representation from 4D to 2D. For ResNet50V2 we obtained a feature tensor of 2 dimensions with shape (batch_size,2048) and concerning InceptionResNetV2, we achieved a feature tensor of 2 dimensions with shape (batch_size,1536).

3.5.3 Fusion Techniques

Multimodal sentiment analysis undergo a fusion process in which data from different modalities (text, audio, or visual) are fused and analyzed together[4]

Early Fusion

Features from multiple modalities are concatenated to form a single long feature

vector. This feature vector will then be used for classifying the final class. Early fusion captures the true essence of multimodal collaboration as all the features are combined in a unified representation.

Late Fusion

Instead of concatenating the feature vectors as in feature-level fusion, there will be a separate classifier for each modality. The output of each classifier is assigned a probability score. The final label of the classification is obtained based on the majority votes. There are a few drawbacks associated with this approach. The late fusion approach increases the computational time due to the high number of classifiers needed for training.The second obstacle is that the classifier model is not synchronized with information from different models, so correlations between those models are not considered in the classifier outputs.

Hybrid Fusion

Hybrid fusion is a combination of early and late fusion techniques, which exploits complementary information from both methods during the classification process [52]. It usually involves a two-step procedure wherein early fusion is initially performed between two modality, and late fusion is then applied as a second step, to fuse the initial results from the early fusion, with the remaining modality[37].

3.5.4 Construction

The resulting tensors from both modalities of text and image are concatenated using early fusion technique to form a single long feature tensor and is fed into a batch normalization layer. This layer standardizes the inputs to a layer for each mini-batch that stabilizes the learning process and significantly reduces the number of epochs needed to train deep networks enabling regularization hence minimizing generalization error. This block is connected to a fully connected dense layer consists of 115 Neurons with Relu as an activation function. Ultimately, we connected this to the final classification layer consists of 1 neuron with a sigmoid activation function since it's a binary classification. All the dropout layers in the architecture have a probability of 0.2 to avoid overfitting. Figure 3.3 represents our proposed model's

neural network architecture.

The model has been trained with a batch size of five and ten epochs. We made use of Adam optimizer for optimization. The reason behind selecting the batch size as five was while training, we encountered an out-of-memory memory(OOM) error. The batches having a size greater than ten exceeded available GPU memory(since BERT had too many parameters to be trained and we didn't have enough computational resources(our model ran on an 8GB RAM, i3 processor without GPU). As a result, we had to reduce our batch size and restrict our epochs to 10. After several attempts, we found that the batch size of 5 was optimal for our requirements.

3.5.5 Hyperparameter Tuning

All the experiments were performed with the following system configurations. Intel ® Core™ i3-4005U CPU @ 1.70GHz processor, with a memory of RAM 8 GB and 64-bit operating system. Hyperparameters are crucial as they regulate a machine learning or deep learning model's overall behavior. The ultimate goal is to find the right combination of hyperparameters that minimizes a predefined loss function, improves accuracy, and minimize overfitting issues. We have listed all the hyperparameters used for our model as shown in **Figure 3.2**. We conducted numerous iterations with all possible distinct sets of hyperparameters. After analyzing the model performance on various hyperparameters, we selected the one giving the best optimal result on our dataset.

PARAMETER	VALUE
No. of neurons in dense layer	115
Activation Function	RELU
Layer weight regularizers with penalty value	L2(0.01)
Dropout	0.02
Optimizer	Adam
Learning Rate	0.01
Batch Size	5
Sequence Length	250 Words

Figure 3.2. Overview of Hyperparameters

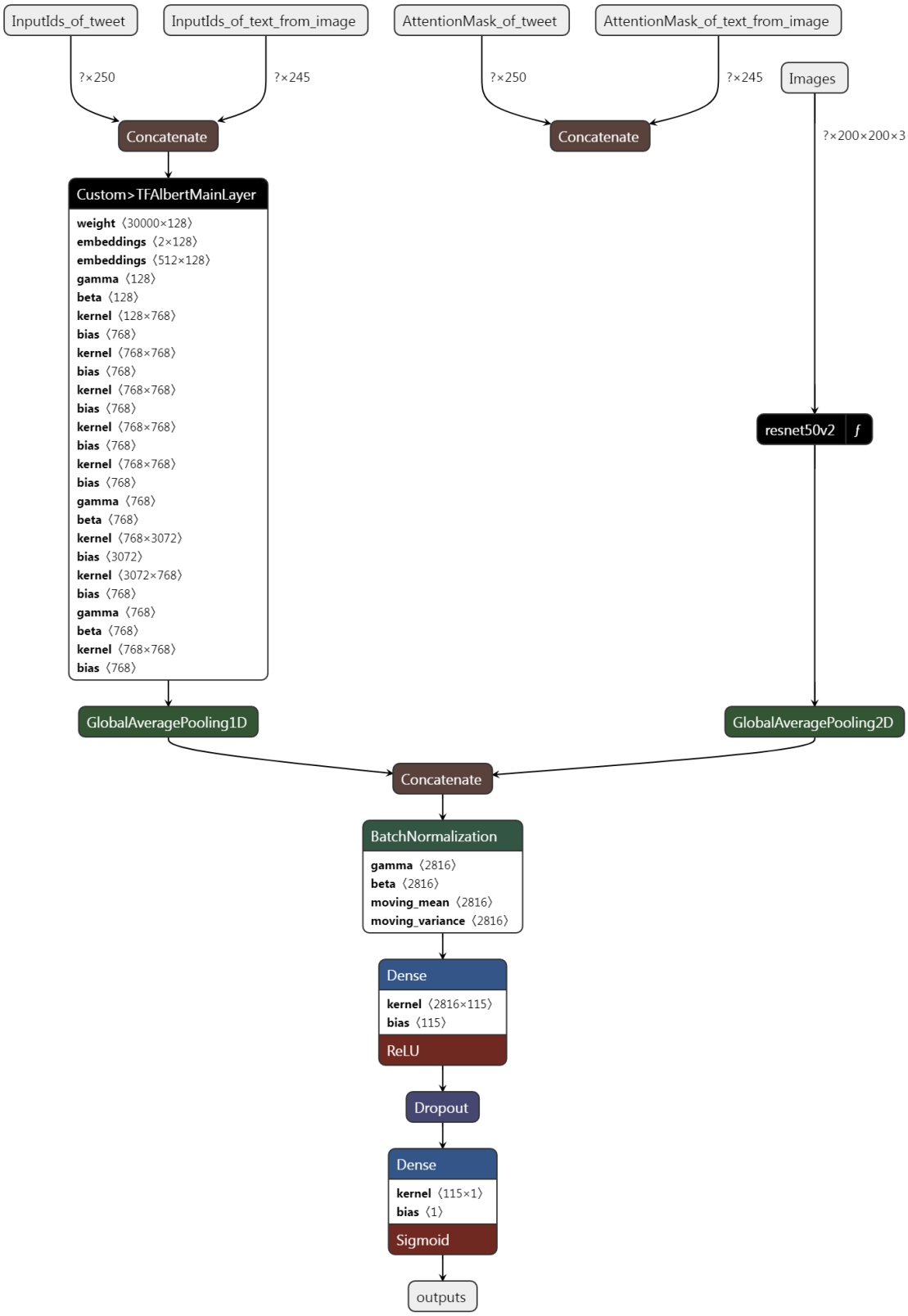

Figure 3.3. Neural Network Architecture

Chapter 4

Results

This section will discuss the outcome after performing the experiments in great detail. The results were produced after executing for ten epochs. Firstly we have conducted our experiments by incorporating artificial neural networks such as LSTM and Bi-LSTM for text modality and VGG19 for image modality used for benchmarking model performance against the state-of-the-art BERT and ResNet.

4.1 Model Evaluation

As a baseline, we have used an LSTM consisting of 200 LSTM units with a bag of words(BOW) as feature extraction. The embedding layer has an input dimension of 7250, the size of vocabulary from the text corpus, the output dimension of 64 is the size of the vectors for each word, input length sequence is 250 since it is the maximum length of the tweet. For image embedding, we used fine-tuned VGG19. VGG19 is a convolutional neural network that is 19 layers deep[46]. Feature vectors across modalities are concatenated and passed through the classification layer. You can find the baseline model neural network architecture under APPENDIX section. This classifier obtains a test accuracy of 62.13%, a precision of 0.6618, a recall of 0.617, and an F1 score of 0.639. Upon using a Bi-LSTM, we noticed a decrease in accuracy by 5.14%, which was quite amusing since Bi-LSTM retains background data from both sides of an expression within a phrase. Bi-LSTM reported a test

accuracy of 60.81% precision of 0.6025, recall of 0.631 and F1 score 0.616 as shown in **Figure 4.1**.

MODEL	TRAINING ACC.	TESTING ACC.	PRECISION	RECALL	F1
LSTM + VGG19	65.60%	62.13%	0.661	0.617	0.639
Bi-LSTM + VGG19	60.81%	56.99%	0.602	0.631	0.616

Figure 4.1. Baseline Model Performance

MODALITY	MODELS	TRAINING ACC.	TESTING ACC.	PRECISION	RECALL	F1
Unimodal	*Only Tweet*					
	Albert	**70%**	**64.71%**	**0.783**	**0.510**	**0.618**
	Robert	68.79%	54.78%	0.547	1	0.708
	Only Image					
	InceptionResNet	**55.41%**	**55.15%**	**0.670**	**0.396**	**0.498**
	ResNet	54.6%	58.09	0.596	0.745	0.633
Bimodal	*Tweet + Image Text*					
	Albert	**72.60%**	**73.90%**	**0.879**	**0.638**	**0.739**
	Robert	69.60%	67.65%	0.779	0.617	0.689
	Tweet + Image					
	Albert + ResNet	**70.88%**	**66.91%**	**0.762**	**0.604**	**0.674**
	Robert + InceptioResNet	67.81%	55.58%	0.557	0.980	0.710
Multimodal	*Tweet+Image Text+Image*					
	Albert + ResNet	**75.43%**	**74.63%**	**0.810**	**0.718**	**0.762**
	Albert + InceptionResNet	74.32%	71.69%	0.858	0.611	0.714
	Robert + ResNet	66.19%	60.66%	0.677	0.550	0.607
	Robert + InceptionResNet	70.88%	62.50%	0.656	0.691	0.673

Figure 4.2. Summary of Model Performance

The **Figure 4.2** summarises the performance of our unimodal,bimodal and multimodal.

Under the unimodal section consisting of only text, the Albert model performs the best with a test accuracy of 64.71%. Robert's model has a high recall of one with very low precision of 0.547 conveys to us that most of the predicted labels are incorrect. Concerning images, the InceptionResNet model performs the best with a test accuracy of 55.15% when compared to ResNet. InceptionResnNet model, despite having a high test accuracy, there is a trade-off between training and testing

accuracy. Also, the results for image modality are poorer when compared to text. Test accuracy for tweets outperforms image features by almost 8%. Precision, recall, and F1 score are significantly higher.

In the bimodal section consisting of the tweet with image text data(both textual data), the Albert model outperforms the Robert model with a test accuracy of 77.94%. The Precision, recall, and F1 score are significantly higher for the Albert model. Models consisting of image and textual data, Albert + Resnet reported a test accuracy of 66.91% which is 11% more compared to Robert+InceptionResnet. It displayed some overfitting issues. Training accuracy is more compared to testing accuracy. Results of the tweet along with in-image text(both textual data) exceed that of text along with image data. Another inference drawn from the comparisons is that test accuracy of bimodal outperforms unimodal by 8.84% on an average.

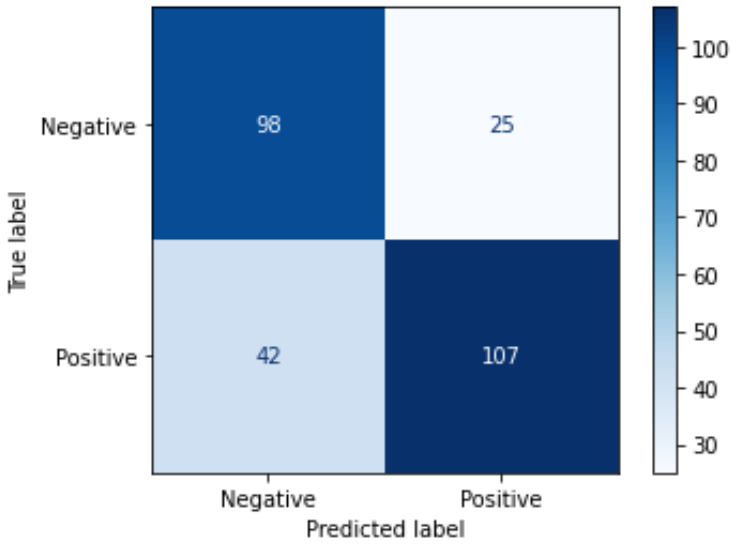

Figure 4.3. Confusion Matrix

Concerning the multimodal approach, Albert with ResNet model performs the best among the four combinations. Our proposed model reported the highest test accuracy of 74.3%. Precision and recall scores are higher. The F1 score of 0.762 reported the highest among all the other modalities. Our proposed multimodal outperforms other modalities. Upon comparing with our baseline model LSTM,

our proposed multimodal surpass the baseline model with an increase of 12.5% in accuracy. We can conclude that upon adding more and more features, model performance is enhanced, which is quite evident from our results table as shown in Figure 4.2.

Figure 4.3 represents the confusion matrix of the validation dataset of our best-performing proposed multimodal that is Albert+ResNet. The model was able to predict 79.67% negative and 71.81% positive labels correctly.

	precision	recall	f1-score	support
Negative	0.70	0.80	0.75	123
Positive	0.81	0.72	0.76	149
accuracy			0.75	272
macro avg	0.76	0.76	0.75	272
weighted avg	0.76	0.75	0.75	272

Figure 4.4. Classification Report

Overall classification report of positive and negative classes of the validation dataset is shown in **Figure 4.4**.

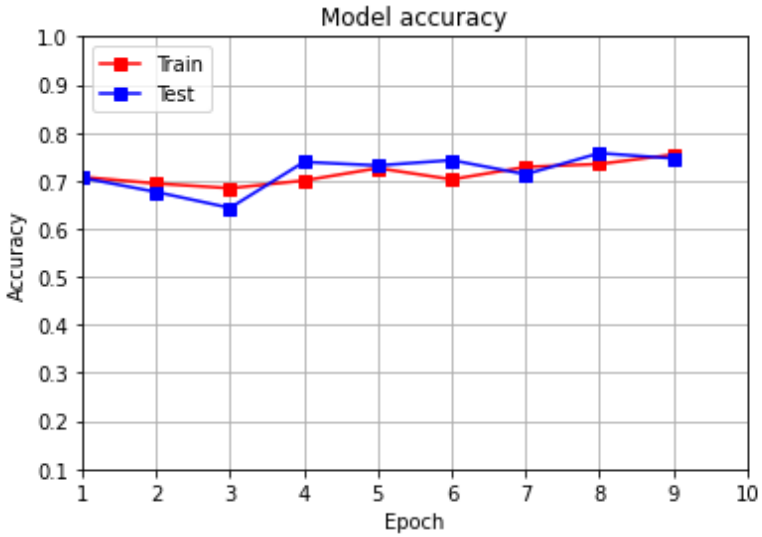

Figure 4.5. Accuracy Graph

By inspecting our accuracy graph in **Figure 4.5** of best performing multimodal,

the trade-off between training and test accuracy is more or less the same, which signifies that the model is not overtrained or undertrained. The model seems to be generalized.

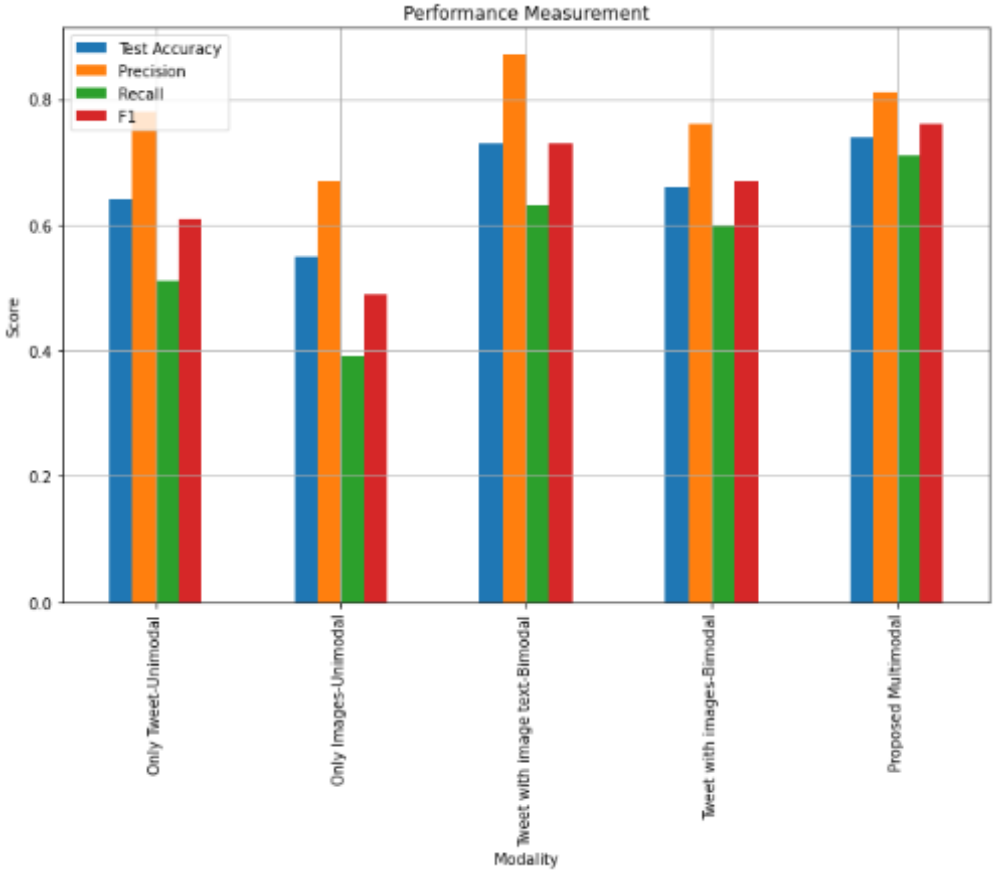

Figure 4.6. Overall Performance

Figure 4.6 portrays the overall performance measurement scores of the different modalities. From our bar graph, unimodal consisting only of image features performs the worst among all contrary, our proposed multimodal performs the best. All our models reported test accuracy of over 60% apart from our only image modality with our proposed multimodal 74.63% with the highest test accuracy, followed by our bimodal consisting of the tweet and text image with 73.9%. The F1 score of our proposed model is the clear winner. Our unimodal consisting of images reported a poor F1 score. The Precision and Recall scores dominated by our proposed model rank multimodal as our winner.

4.2 Error Analysis

After scrutinizing some errors, I discovered irregular patterns that could help us determine the limitations of our proposed model.

1) Images as abstract representation

The images under these sections are just static illustrations that are just a reflection of the text. Text illustration with pictures generally enhances learners' performance serving just as a visualization tool promoting imagination capabilities but does not capture the in-depth concept of the context.

i) From **Figure 4.7** it is clear that the tweet text and text inside the image seem to be on a positive scale whereas the photo is just a mere representation of the text that I believe is from a neutral standpoint. But looking at the overall vision and text polarity seems to be positive as the message conveys that it's a remarkable feat. IIT is a famous research institute in India, and its building is represented in the image. Our model is unable to capture this text-vision interaction feature.

TWEET: A surface that makes #coronavirus evaporate faster @iitbombay @iitmadras @ROBCHENNAI_MIB @BOCMIB @airnews_Chennai @AIRDharmapuri @TNDPHPM @MH_TN @MIB_India @COVID19 @COVID19india @Unite2FightCorona

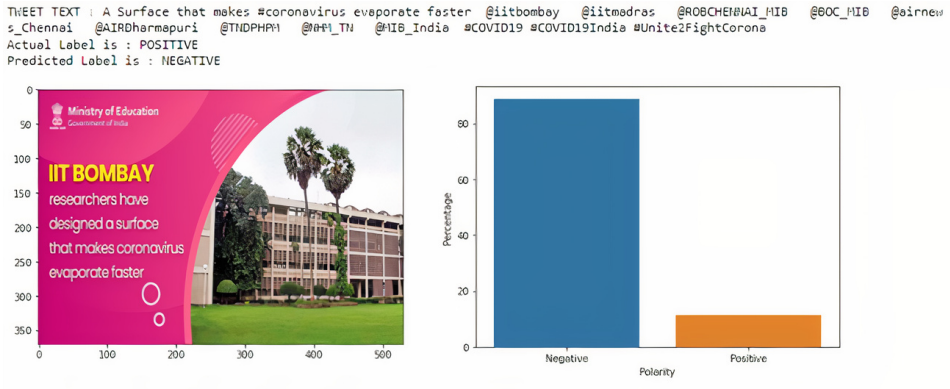

Figure 4.7

ii) From Figure 4.8, this example is similar to the first one. The model is unable to capture the relationship between text and visual information.

TWEET: What a gesture: Switzerland projected GB US IN flag (different days) onto Switzerland's iconic #Matterhorn as a sign of solidarity in the fight against #COVID19. Thank you #Switzerland for being a steadfast partner of India and recognising India effort in fight against #Covid_19

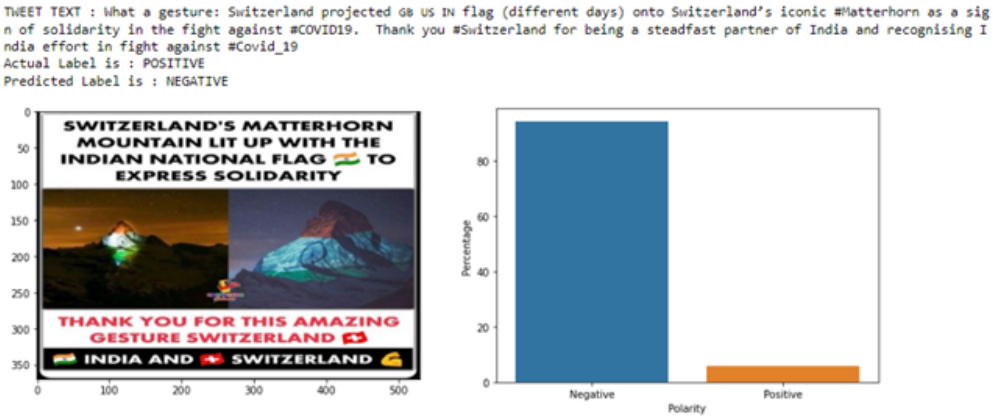

Figure 4.8

2) Contradictions in multimodal approach occurs in situations where two modality that is text and the image is unlikely to be true when considered together.

i) By inspecting the text and the image from the **Figure 4.9** the context is very contradicting concerning each other. The overall polarity seems to be negative but our model predicted it to be positive.

TWEET: They have an amazing capacity for denial, don't they? #COVID19

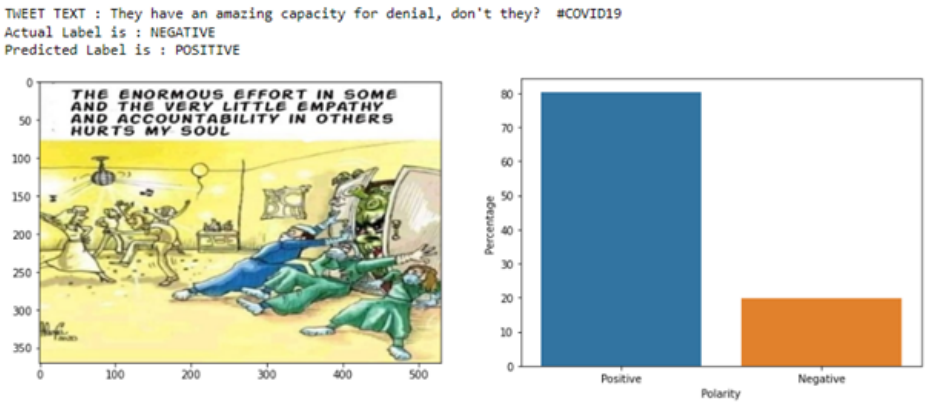

Figure 4.9

ii) From **Figure 4.10**, this case seems to be quite tricky for the model to determine the polarity correctly. The tweet conveys that except for the wife, the entire family was tested positive for covid. The image, on the other hand, displays a joyful family. It is a contradiction since our model is failing to capture the dependency information between the two modality. It's hard to represent the polarity from a human perspective because it's contradicting. However, the tweet text and the text inside the image appear to be negative, so I labeled the target negative, but our model predicted it to be positive.

TWEET:#ShilpaShetty's family members including husband Raj Kundra and kids test COVID-19 positive; actress tests negative

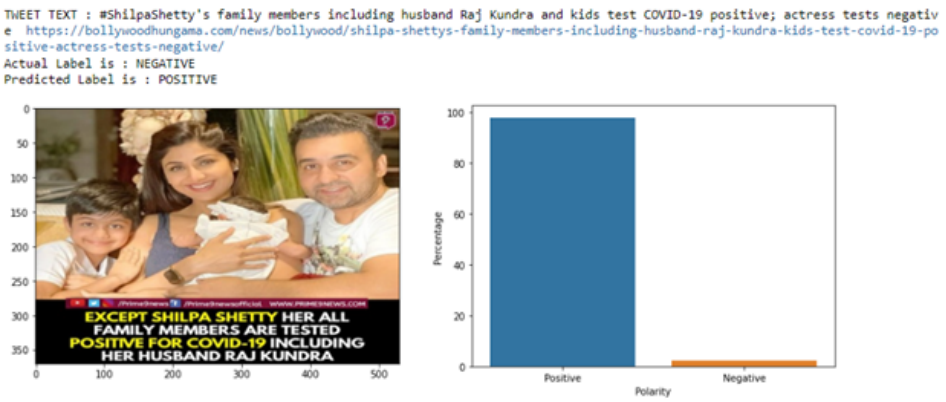

Figure 4.10

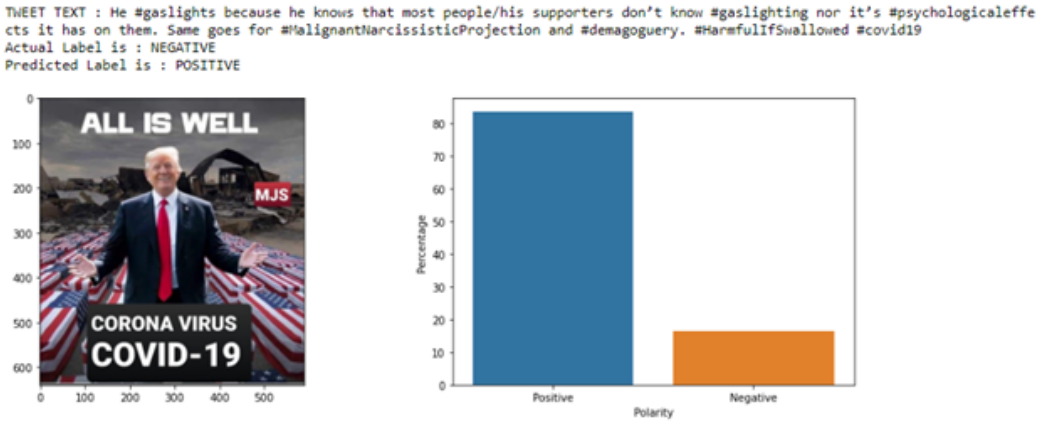

Figure 4.11

iii) From **Figure 4.11** this case too is quite similar to the above one. The text "All is

well" contradicts the image . Concerning the image, we can think of a person whose facial expression is delightful on the contrary, the backdrop of an image surrounded by many funeral boxes.

TWEET: He #gaslights because he knows that most people/his supporters don't know #gaslighting nor it's #psychologicaleffects it has on them. Same goes for #MalignantNarcissisticProjection and #demagoguery. #HarmfulIfSwallowed #covid19

3) The Model fails to capture multiple objects in an image of similar type

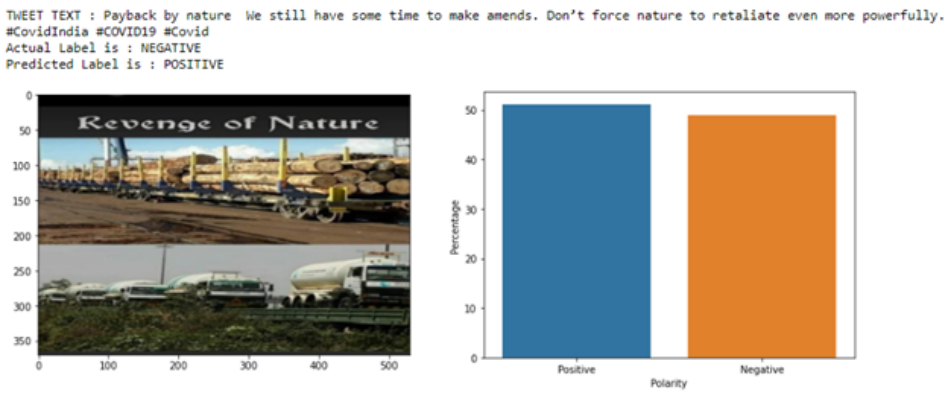

Figure 4.12

i) From **Figure 4.12** the text is clearly on the negative end. A human can perceive this image as going against nature, human have an adverse effect. The irony here is that human depends on tree for oxygen which is available for free and natural, but men are exploiting nature by cutting down trees for their profit or business venture, and in turn mother earth conveys to us that we will have to survive on artificial oxygen by purchasing it if the trees get exhausted. There are multiple objects in the image that is similar to each other. The subject in our image seems to slowly fade on nearing to end making it blurred and out of context. The model treats it as **noise** and unable to capture some intrinsic details.

TWEET: Payback by nature We still have some time to make amends. Don't force nature to retaliate even more powerfully. #CovidIndia #COVID19 #Covid

ii) From the Figure 4.13, the model is unable to decipher the meaning behind the image. The model fails to recognize the corpses burnt in the backdrop since only

remnants are left making it harder to capture the subtle details

TWEET: Join hands with Hemkunt Foundation and provide support to critical COVID patients with FREE Oxygen Cylinders. Follow the link to contribute and save their lives. https://bit.ly/HemkuntOxygenRelief #donateoxygen #helpthembreathe #OxygenCrisis #EffortsForGood #hemkuntfoundation #COVID19

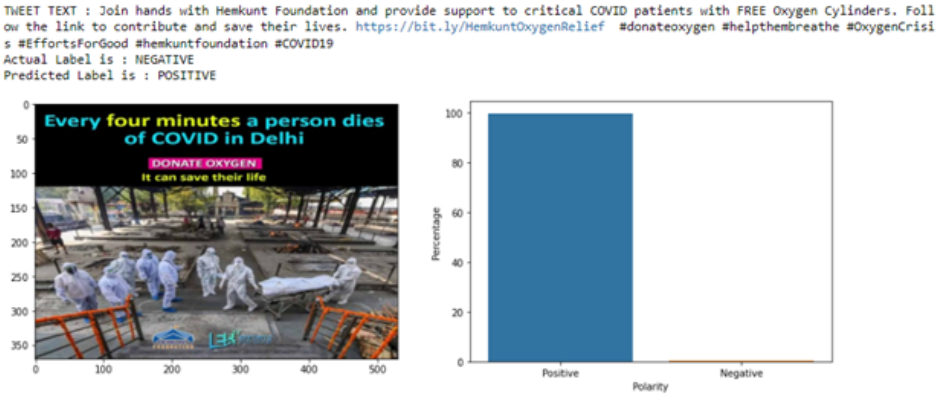

Figure 4.13

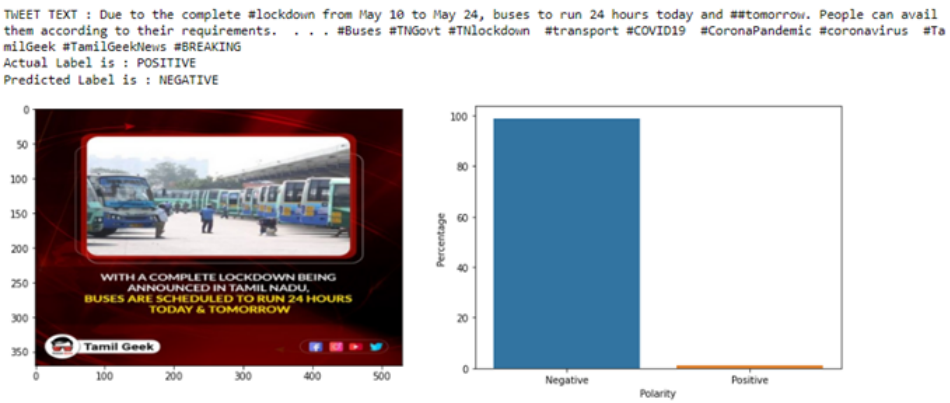

Figure 4.14

iii) From the **Figure 4.14** this tweet seems to be on a positive note as the government is the availing facility for its citizens to travel to their respective homes before a total lockdown is about to be imposed. The image seems to stand on a neutral viewpoint, depicting not much information serving only as representation intent. But taking a close look at the photo multiple objects are buses where the model fails to capture the details.

TWEET: Due to the complete #lockdown from May 10 to May 24, buses to run 24 hours today and ##tomorrow. People can avail them according to their requirements. #Buses #TNGovt #TNlockdown #transport #COVID19 #Corona-Pandemic #coronavirus #TamilGeek #TamilGeekNews #BREAKING

4.3 Further Discussions

According to our findings, the proposed multimodal model performed best among all other modalities. BERT has revolutionized the NLP field since its inception. It has achieved state-of-the-art results in contrast with existing NLP algorithms. BERT has been pre-trained on a large text corpus of about 2500 million words, suggesting that the model can perform well on newer words or characters. Previously, word embeddings produced the same vector regardless of a word's position in a sentence. BERT generates alternative vectors for the identical word based on the surrounding words in a sentence. The features evoked from an image are characterized by several factors such as textures, patterns, pixels, resolution, number of channels, and dimensions. Image features are combined with the text to create a common vector space in which correlations of features among different modalities displayed a modest improvement in accuracy. The model's performance gradually enhanced with the addition of more features. It's obvious to employ a multimodal strategy since we were aware of the potential prospects of the deep learning algorithm.

The feature embeddings from the CNN variant by just using the images(Unimodal) did not perform effectively. But this was expected since the sentiment analysis is generally performed predominantly on text data. Multimodal approach for some tweets, the image conveys no information serving the picture as an abstract representation with contradictions as shown under the error analysis section. Real-world data contains statistical noise. It's more challenging for the learning algorithm to map the input variables to target variables if more statistical noise persists. With fewer images, it would be difficult for our sentiment model to extrapolate the relationship between text and image features since CNN requires different patterns to learn decisive variations from images.

4.4 Future Improvement

Perfection is a myth. There is always scope for improvements. Though our proposed model performs well, it isn't flawless. Firstly to improvise embeddings of images, the best approach is to train the model with more additional images to learn different patterns. More data boost the model's accuracy, but high-quality data aids in building better models. If we have a huge data repository with features that are too noisy or do not possess enough variation to capture subtle details, any models will effectively be worthless regardless of the data volume. Hence quality data is always better than quantity. The next step was to allocate the task of assigning labels to multiple people since the collective views and opinions can be analyzed and later drawn on common grounds leading to better annotation. With more sophisticated computational resources and vigorous hyper-parameter tuning, models can be trained for better accuracy.

Our current multimodal restricts embeddings from images, tweets, and text extracted from an image may not reach the potential of determining the polarity as expected. Integrating other modalities such as audio or video may contain more delicate or hidden patterns paving the direction for our future efforts. For our research, we have used the early fusion technique. In the future, we can extend by incorporating late, hybrid, and other fusion techniques. Multiple modalities merged into a single unit. This approach aims to extract and mix information from individual modalities. It seems to be a black box as we are unsure how well the features are getting integrated. If we can work around it by focusing on some information rather than considering complete information, we might capture dependency information among the features hence improving performance.

4.5 Model Deployment

Model Deployment is a technique where we integrate a machine/deep learning model into a production environment to make practical decisions on real-world data. This feature helps deliver interactive analysis or insights to end-users. A model has to be deployed into production to provide a decision-making action plan. The impact

of the model is severely constrained and does not add much value if you cannot derive constructive insights from your model's output. Since our combination of Albert coupled with the ResNet model gave us the best accuracy, we have selected this model for deployment for real-world predictions. We have used an open-source python framework called **Streamlit**. We have built a simple web application using Streamlit and placed it on a Streamlit cloud where it can receive multiple requests for processing. After the model deployment, it is capable of taking inputs from the users and providing output to as many different requests provided for prediction purposes.

To try the application please click on the link: https://share.streamlit.io/sapzamt/presentation/main/app.py

Before using the application, click on the setting button located on the top right-hand corner of the screen next select the clear cache option.

You can find samples of our application predictions results under the APPENDIX section.

Chapter 5

Conclusion

Public sentiment demonstrated deep concerns about COVID-19, leading to mixed feelings such as fear, anxiety, confusion, chaos, worry, anger, sadness, etc. To address the issue relating to the sentimental analysis, we proposed a multimodal-based approach using coronavirus tweets data with the aid of state-of-the-art deep learning algorithms. BERT architecture has achieved a breakthrough in the NLP field. Employing BERT fine-tuned models, we can solve problems for multiple tasks. By harnessing the power of BERT coupled with feature extraction using transfer learning technique from CNN variants, followed by an idea to extract text from images using an OCR tool, we integrated textual and visual features embedding using an early fusion technique. This layer connects to our densely connected neural network for predictions. In our study, we have experimented with several combinations of BERT and CNN variants and exhibited performance comparisons among different modalities.

Finally, we concluded that our proposed multi-model backed with Albert and ResNet model provided us with the best results in terms of testing accuracy of 74.63%, Precision of 0.8106, Recall of 0.718, and F1 score of 0.762. We demonstrated that upon adding more features, model performance magnified. Though the results were good, it's still far from being perfect. Extracting appropriate features from an image has been a conundrum. Albeit understanding public perspectives, and sentiment will

enable policymakers to cater to the needs of people to implement specific strategies to deal with any crisis management in the future pandemic. In the future, we plan to extend our work to study sentiment for different time frames, for instance, to analyze events on covid lockdown, post lockdown, after vaccination, the resurrection of covid waves, and life after covid.

Bibliography

[1] https://github.com/JaidedAI/EasyOCR.

[2] Christian S. Sergey I. Vincent V. Alex A. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *arXiv:1602.07261* (2016).

[3] Rosasco L. De Vito E. Caponnetto A. Piana M. Verri A. "Are Loss Functions All the Same?" In: *Neural Computation. 16 (5): 1063–1076. CiteSeerX 10.1.1.109.6786. doi:10.1162/089976604773135104. PMID 15070510* (2004).

[4] Poria Soujanya Cambria Erik Bajpai Rajiv Hussain Amir. "A review of affective computing: From unimodal analysis to multimodal fusion". In: *Information Fusion. 37: 98–125* (2017).

[5] Karray Fakhreddine Milad Alemzadeh Saleh Jamil Abou Mo Nours Arab. "Human-Computer Interaction: Overview on State of the Art"". In: *International Journal on Smart Sensing and Intelligent Systems. 1: 137–159* (2008).

[6] Nisha Arya. "What is Transfer Learning?" In: *KDnuggets* (2022).

[7] Liu B. *Sentiment Analysis and Subjectivity". In Nitin I. and Fred J. (eds). Handbook of Natural Language Processing. 2nd Ed Machine Learning and pattern recognition series.* 2010.

[8] Aggarwal Charu C and Zhai Cheng Xiang. "Mining Text Data". In: *Springer New York Dordrecht Heidelberg London: © Springer Science+Business Media, LLC'12* (2012).

[9] Cortes C and Vapnik V. "Support-vector networks". In: *presented at the Machine Learning* (1995).

[10] Vasco L Antonio G Luis A Joao C. "An AutoML-based Approach to Multimodal Image Sentiment Analysis". In: *arXiv:2102.08092* (2021).

[11] Guang Q. Xiaofei H. Feng Z. Yuan S. Jiajun B. Chun C. "DASA: dissatisfaction-oriented advertising based on sentiment analysis". In: *Expert Syst Appl 37 pp. 6182-6191* (2010).

[12] Ian G. Yoshua B. Aaron C. "Back-Propagation and Other Differentiation Algorithms". In: *MIT Press. pp. 200–220* (2016).

[13] Mohammad S. Dunne C. and Dorr B. "Generating high-coverage semantic orientation lexicons from overly marked words and a thesaurus". In: *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP'09)* (2009).

[14] Keith D. "A Brief History of Big Data". In: *Dataversity* (2017).

[15] Crina Grosan and Ajith Abraham. "Artificial Neural Networks". In: *Intelligent Systems. Intelligent Systems Reference Library, vol 17. Springer, Berlin, Heidelberg* (2011).

[16] G. Miller R. Beckwith C. Fellbaum D. Gross and K. Miller. "WordNet: an on-line lexical database". In: *WordNet: an on-line lexical database Oxford Univ. Press* (1990).

[17] Hanhoon K. Seong J. Dongil H. "Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews". In: *Expert Syst Appl 39 pp. 6000-6010* (2012).

[18] Soujanya P. Navonil M. Devamanyu H. Erik C. Alexander G. Amir H. "Multimodal Sentiment Analysis: Addressing Key Issues and Setting up the Baselines". In: *arXiv:1803.07427* (2019).

[19] G. Halevi and H. Moed. "The evolution of big data as a research and scientific topic: Overview of the literature". In: *Res. Trends 3–6* (2012).

[20] W. Medhat A. Hassan and H. Korashy. "Combined algorithm for data mining using association rules". In: *Ain Shams J Electric Eng, 1 (1)* (2008).

[21] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural Computation. 9 (8): 1735–1780* (1997).

[22] Anthony Hu and Seth Flaxman. “Multimodal Sentiment Analysis To Explore the Structure of Emotions”. In: *arxiv: 1805.10205v1* (2018).

[23] Read J. and Carroll J. “Weakly supervised techniques for domain-independent sentiment classification”. In: *Proceeding of the 1st international CIKM workshop on topic-sentiment analysis for mass opinion . p. 45–52.* (2009).

[24] Kaufmann JM. and J MaxAlign. “A Maximum Entropy Parallel Sentence Alignment Tool”. In: *Proceedings of COLING’12: Demonstration Papers, Mumbai p. 277–88* (2012).

[25] Diederik P. Kingma and Jimmy Lei Ba. “Adam: a Method for Stochastic Optimization”. In: *International Conference on Learning Representations, pages 1–13* (2015).

[26] Devlin Jacob Chang Ming-Wei Lee Kenton Toutanova Kristina. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv:1810.04805v2* (2018).

[27] Akshi Kumar and Geetanjali Garg. “Sentiment analysis of multimodal twitter data”. In: *Multimedia Tools and Applications 78:24103–24119* (2019).

[28] Cho K. van M. Bart Bahdanau DZmitry Bengio Yoshua. “On the Properties of Neural Machine Translation: Encoder-Decoder Approaches”. In: *arXiv:1409.1259* (2014).

[29] Soleymani Mohammad Garcia David Jou Brendan Schuller Björn Chang Shih Pantic Maja. “A survey of multimodal sentiment analysis”. In: *Image and Vision Computing. 65: 3–14* (2017).

[30] Diana Maynard and Adam Funk. “Automatic detection of political opinions in tweets”. In: *Proceedings of the 8th international conference on the semantic web ESWC p. 88–99* (2011).

[31] Valueva M.V. Nagornov N.N. Lyakhov P.A. Valuev G.V. Chervyakov N.I. “Application of the residue number system to reduce hardware costs of the convolutional neural network implementation”. In: *Mathematics and Computers in Simulation. Elsevier BV. 177: 232–243* (2020).

[32] Ashish V. Noam S. Niki P. Jakob U. Llion J. Aidan N. Łukasz K. Illia P. “Attention Is All You Need”. In: *NeurIPS* (2017).

[33] Bo P. and Lillian L. “Opinion mining and sentiment analysis”. In: *Foundations and Trends in Information Retrieval, Vol. 2* (2008).

[34] Turney P. “semantic orientation applied to unsupervised classification of reviews”. In: *Proceedings of annual meeting of the Association for Computational Linguistics* (2002).

[35] Apoorv A. Boyi X. Ilia V. Owen R. and P." Rebecca. “Sentiment Analysis of Twitter Data”. In: *LSM Proceedings of the Workshop on Languages in Social Media, Pages 30-38 USA* (2011).

[36] Banea C. Mihalcea R. and Wiebe J. “Multilingual Sentiment and Subjectivity Analysis”. In: *Multilingual Natural Language Processing”, editors Imed Zitouni and Dan Bikel, Prentice Hall* (2011).

[37] Shahla Naghsh N. Ahmad R. “Exploiting evidential theory in the fusion of textual, audio, and visual modalities for affective music video retrieval”. In: *IEEE Conference Publication* (2017).

[38] Sebastian Raschka. “What are gradient descent and stochastic gradient descent?” In: *https://sebastianraschka.com/faq/docs/gradient-optimization.html* ().

[39] F. Rosenblatt. “The perceptron: a probabilistic model for information storage and organization in the brain”. In: *Psychological review 65(6):386* (1958).

[40] Mohammad U. Syeda M Shamim A Nabadita S. “Exploiting evidential theory in the fusion of textual, audio, and visual modalities for affective music video retrieval”. In: *ICT Express Volume 6 Issue 4 Pages 357-360* (2020).

[41] Anuj S. and Shubhamoy D. “Performance Investigation of Feature Selection Methods and Sentiment Lexicons for Sentiment Analysis”. In: *International Journal of Computer Applications (0975 – 8887) on Advanced Computing and Communication Technologies for HPC Applications - ACCTHPCA* (2012).

[42] Kaiming H. Xiangyu Z. Shaoqing R. Jian S. “Identity Mappings in Deep Residual Networks”. In: *arXiv:1603.05027* (2016).

[43] Kim S. and Hovy E. “Determining the sentiment of opinions”. In: *Proceedings of interntional conference on Computational Linguistics (COLING’04)* (2004).

[44] Yinhan L. Myle O. Naman G. Jingfei D. Mandar J.Danqi C. Omer L. Mike L. Luke Z. Veselin S. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *arXiv:1907.11692* (2019).

[45] Zhenzhong L. Mingda C. Sebastian G. Kevin G. Piyush S. Radu S. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations". In: *arXiv:1909.11942* (2019).

[46] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *arXiv:1409.1556* (2014).

[47] Santhosh Srirambhatla. "Machine learning and AI neural networks". In: *The Evolving Enterprise* (2021).

[48] Ahmed Tealab. "Time series forecasting using artificial neural networks methodologies: A systematic review". In: *Future Computing and Informatics Journal. 3 (2): 334–340* (2018).

[49] Lakshana G V. "Artificial Intelligence Vs Machine Learning Vs Deep Learning: What exactly is the difference". In: *Analytics Vidhya* (2021).

[50] Liu B. Hsu W. and Ma Y. "Integrating classification and association rule mining." In: *Presented at the ACM KDD conference* (1998).

[51] Mani Wadhwa. "seq2seq model in Machine Learning". In: *GeeksforGeeks* (2018).

[52] Kenji Morency Louis P. Wollmer Martin Weninger Felix Knaup Tobias Schuller Bjorn Sun Congkai Sagae. "YouTube Movie Reviews: Sentiment Analysis in an Audio-Visual Context". In: *IEEE Intelligent Systems. 28 (3): 46–53* (2013).

[53] Duchi J. Hazan E. Singer Y. "Adaptive subgradient methods for online learning and stochastic optimization". In: *JMLR. 12: 2121–2159* (2011).

Chapter 6

APPENDIX

Below you can find screenshots of our application prediction results that we built using **Streamlit**. Our model was tested, deployed, and successfully predicted real-world data with a distinct variation of tweets.

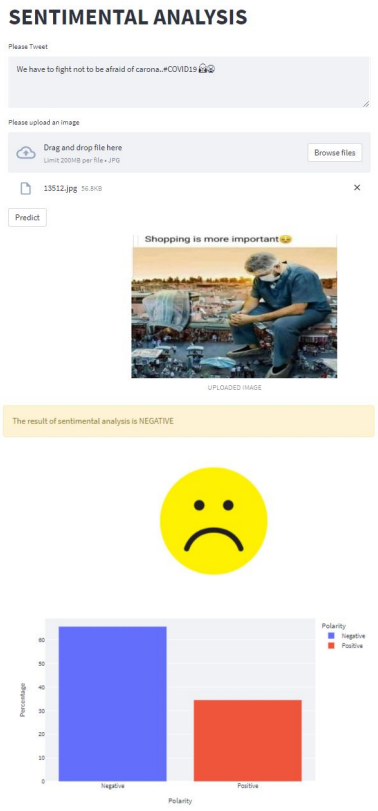

Figure 6.1

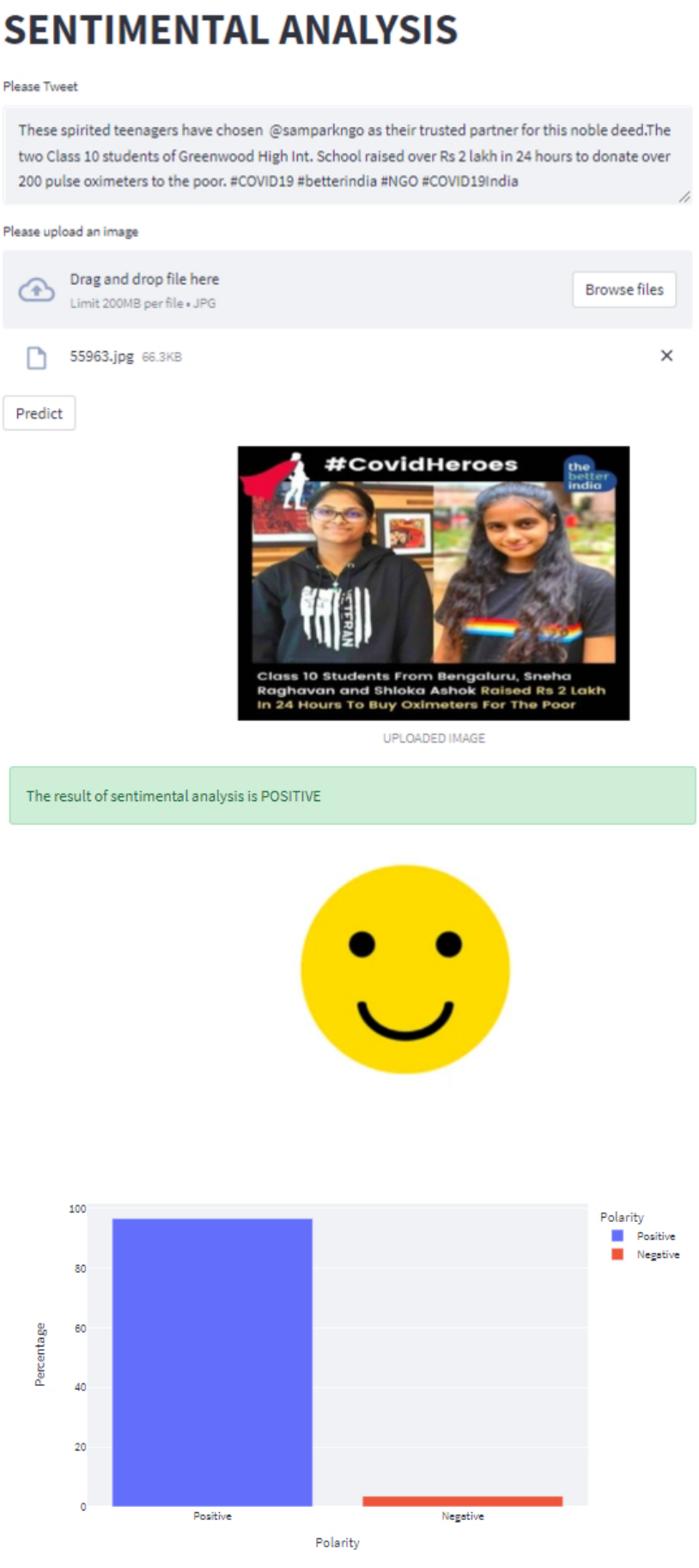
SENTIMENTAL ANALYSIS
Please Tweet
These spirited teenagers have chosen @samparkngo as their trusted partner for this noble deed.The two Class 10 students of Greenwood High Int. School raised over Rs 2 lakh in 24 hours to donate over 200 pulse oximeters to the poor. #COVID19 #betterindia #NGO #COVID19India
Please upload an image
Drag and drop file here
Limit 200MB per file • JPG
Browse files
55963.jpg 66.3KB
Predict
#CovidHeroes
the better india
Class 10 Students From Bengaluru, Sneha Raghavan and Shloka Ashok Raised Rs 2 Lakh In 24 Hours To Buy Oximeters For The Poor
UPLOADED IMAGE
The result of sentimental analysis is POSITIVE
Polarity
Positive
Negative
Percentage
100
80
60
40
20
0
Positive
Negative
Polarity

Figure 6.2

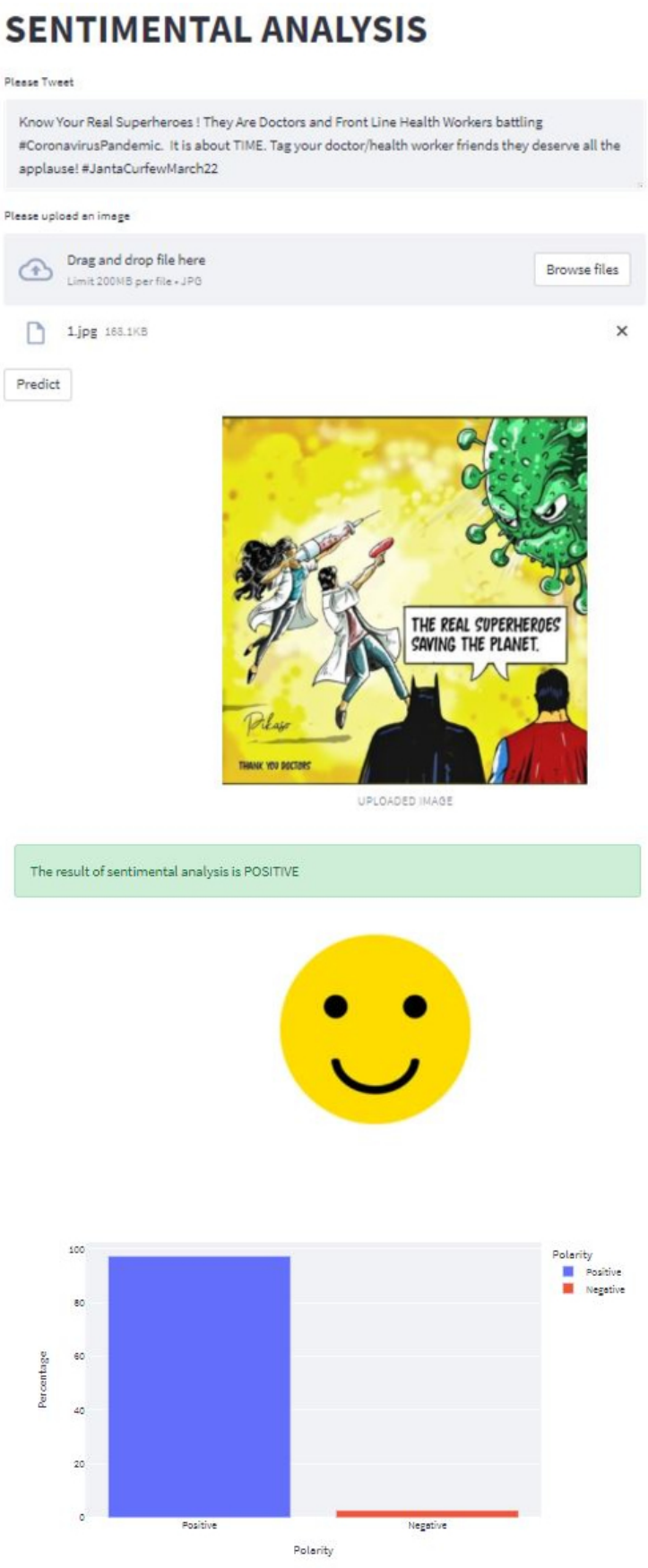
SENTIMENTAL ANALYSIS
Please Tweet
Know Your Real Superheroes ! They Are Doctors and Front Line Health Workers battling #CoronavirusPandemic. It is about TIME. Tag your doctor/health worker friends they deserve all the applause! #JantaCurfewMarch22
Please upload an image
Drag and drop file here
Limit 200MB per file • JPG
Browse files
1.jpg 163.1KB
Predict
THE REAL SUPERHEROES SAVING THE PLANET.
THANK YOU DOCTORS
UPLOADED IMAGE
The result of sentimental analysis is POSITIVE
Polarity
Positive
Negative
Percentage
100
80
60
40
20
0
Positive
Negative
Polarity

Figure 6.3


SENTIMENTAL ANALYSIS
Please Tweet
In Opinion "Getting ready now might give us a fighting chance to avoid a repeat of India's nightmare," Abhijit Banerjee and Esther Duflo, who won the Nobel in economic science in 2019, write in this guest essay.
Please upload an image
Drag and drop file here
Limit 200MB per file • JPG
Browse files
25722.jpg 64.9KB
Predict
India's Problem Is Now the World's Problem
UPLOADED IMAGE
The result of sentimental analysis is NEGATIVE
Polarity
Negative
Positive
Percentage
100
80
60
40
20
0
Negative
Positive
Polarity

Figure 6.4

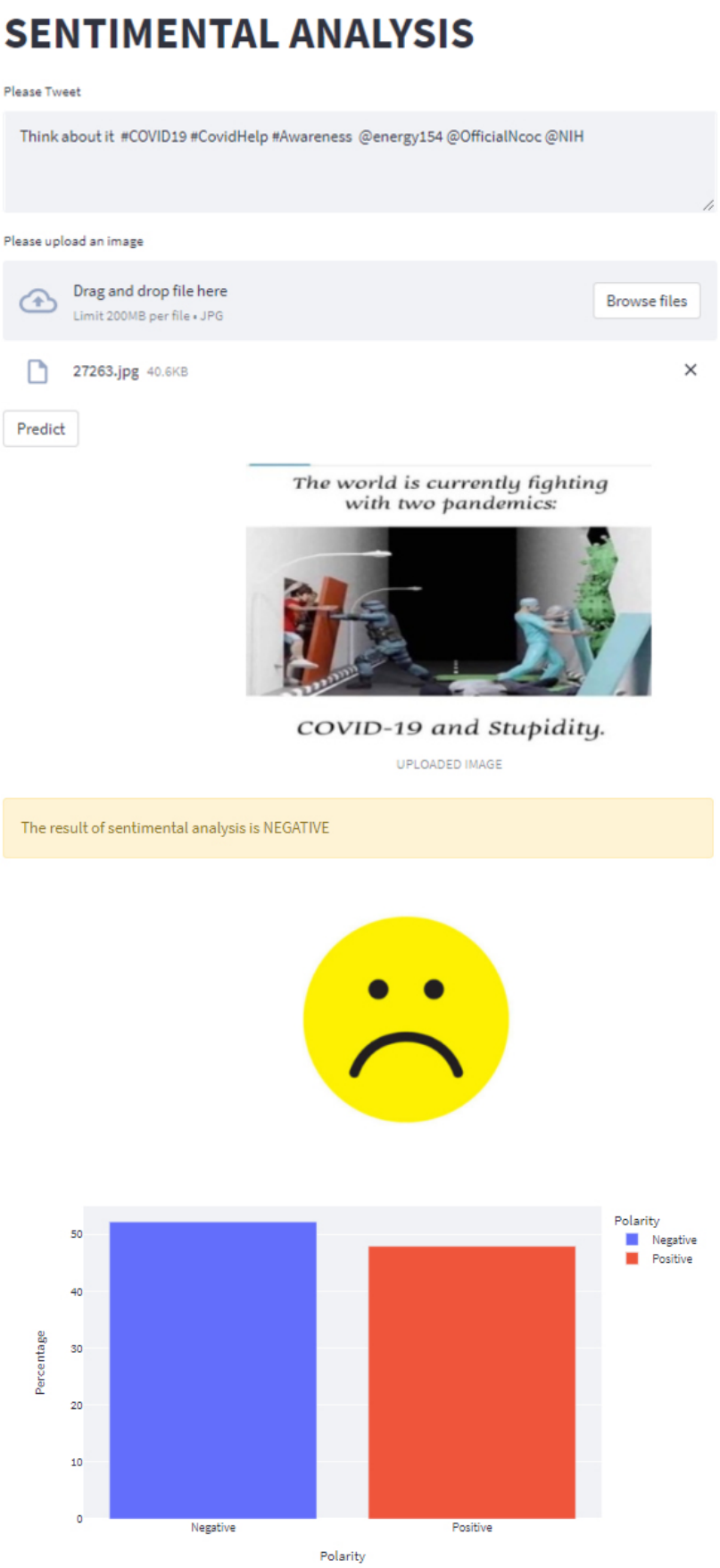
SENTIMENTAL ANALYSIS
Please Tweet
Think about it #COVID19 #CovidHelp #Awareness @energy154 @OfficialNcoc @NIH
Please upload an image
Drag and drop file here
Limit 200MB per file • JPG
Browse files
27263.jpg 40.6KB
Predict
The world is currently fighting with two pandemics:
COVID-19 and Stupidity.
UPLOADED IMAGE
The result of sentimental analysis is NEGATIVE
Polarity
Negative
Positive
Percentage
50
40
30
20
10
0
Negative
Positive
Polarity

Figure 6.5

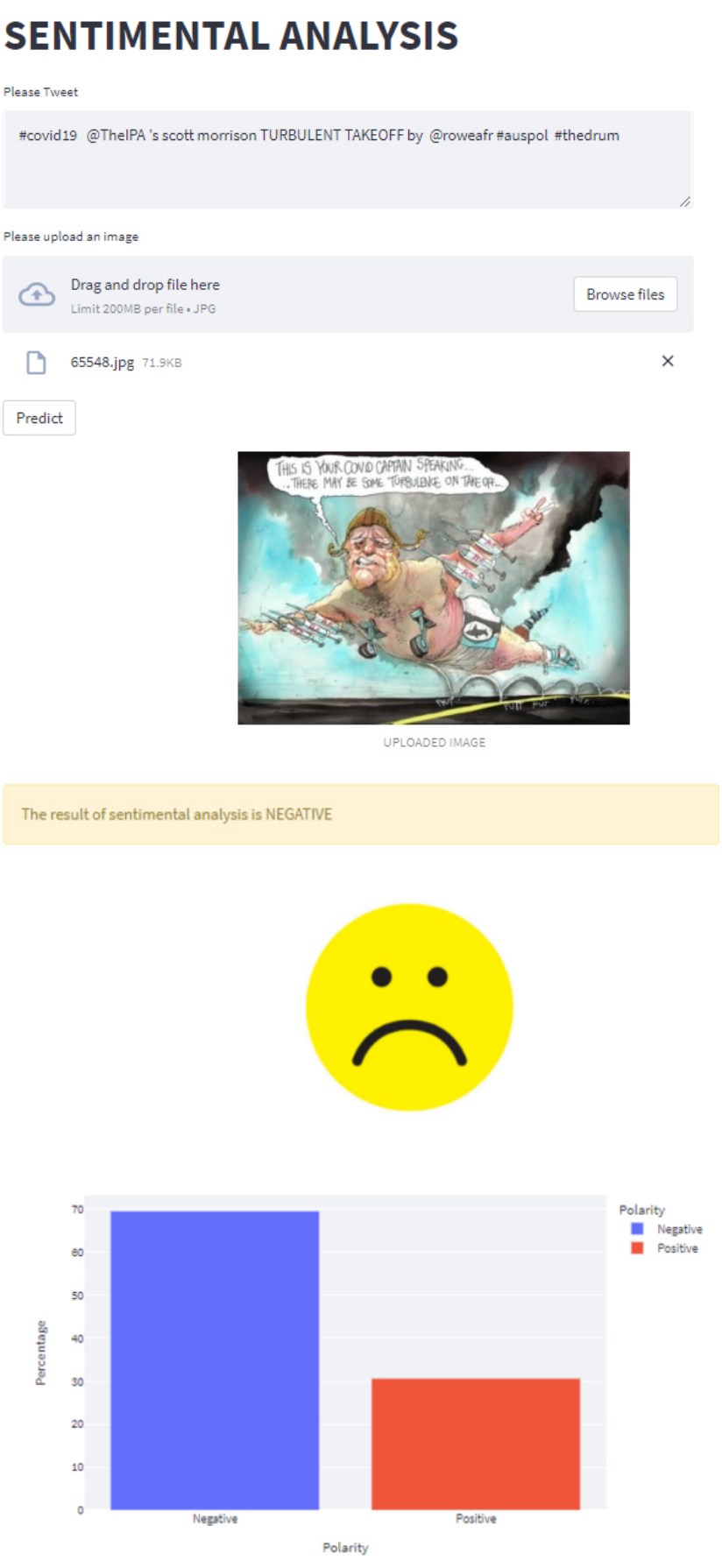
SENTIMENTAL ANALYSIS
Please Tweet
#covid19 @TheIPA 's scott morrison TURBULENT TAKEOFF by @roweafr #auspol #thedrum
Please upload an image
Drag and drop file here
Limit 200MB per file • JPG
Browse files
65548.jpg 71.9KB
Predict
THIS IS YOUR COVID CAPTAIN SPEAKING...
..THERE MAY BE SOME TURBULENCE ON TAKE OFF...
UPLOADED IMAGE
The result of sentimental analysis is NEGATIVE
Polarity
Negative
Positive
Percentage
70
60
50
40
30
20
10
0
Negative
Positive
Polarity

Figure 6.6


SENTIMENTAL ANALYSIS
Please Tweet
@Sakariya55 hope you will face this tough times 🙏🙏#chetansakariya #covid19 #COVID19 #CovidDeaths
Please upload an image
Drag and drop file here
Limit 200MB per file • JPG
Browse files
5696.jpg 58.2KB
Predict
CHETAN SAKARIYA'S FATHER LOSES BATTLE WITH COVID-19
HARD TIME FOR CHETAN SAKARIYA
Sakariya's father had tested positive while his son was playing in IPL 2021, and is the second family member Sakariya has lost this year. Weeks before the IPL auction his younger brother Rahul died by suicide.
UPLOADED IMAGE
The result of sentimental analysis is NEGATIVE
Polarity
Negative
Positive
Percentage
100
80
60
40
20
0
Negative
Positive
Polarity

Figure 6.7

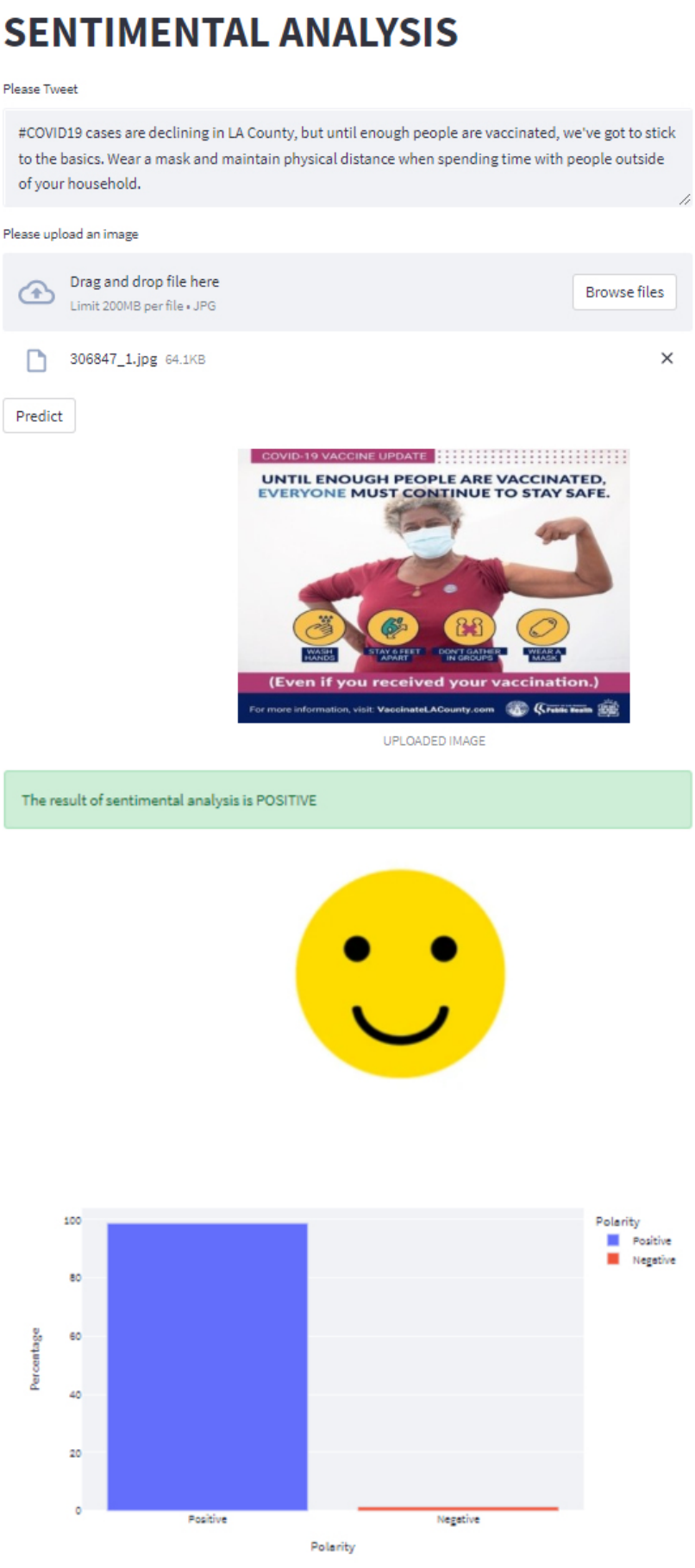
SENTIMENTAL ANALYSIS
Please Tweet
#COVID19 cases are declining in LA County, but until enough people are vaccinated, we've got to stick to the basics. Wear a mask and maintain physical distance when spending time with people outside of your household.
Please upload an image
Drag and drop file here
Limit 200MB per file • JPG
Browse files
306847_1.jpg 64.1KB
Predict
COVID-19 VACCINE UPDATE
UNTIL ENOUGH PEOPLE ARE VACCINATED, EVERYONE MUST CONTINUE TO STAY SAFE.
WASH HANDS
STAY 6 FEET APART
DON'T GATHER IN GROUPS
WEAR A MASK
(Even if you received your vaccination.)
For more information, visit: VaccinateLACounty.com
UPLOADED IMAGE
The result of sentimental analysis is POSITIVE
Polarity
Positive
Negative
Percentage
100
80
60
40
20
0
Positive
Negative
Polarity

Figure 6.8

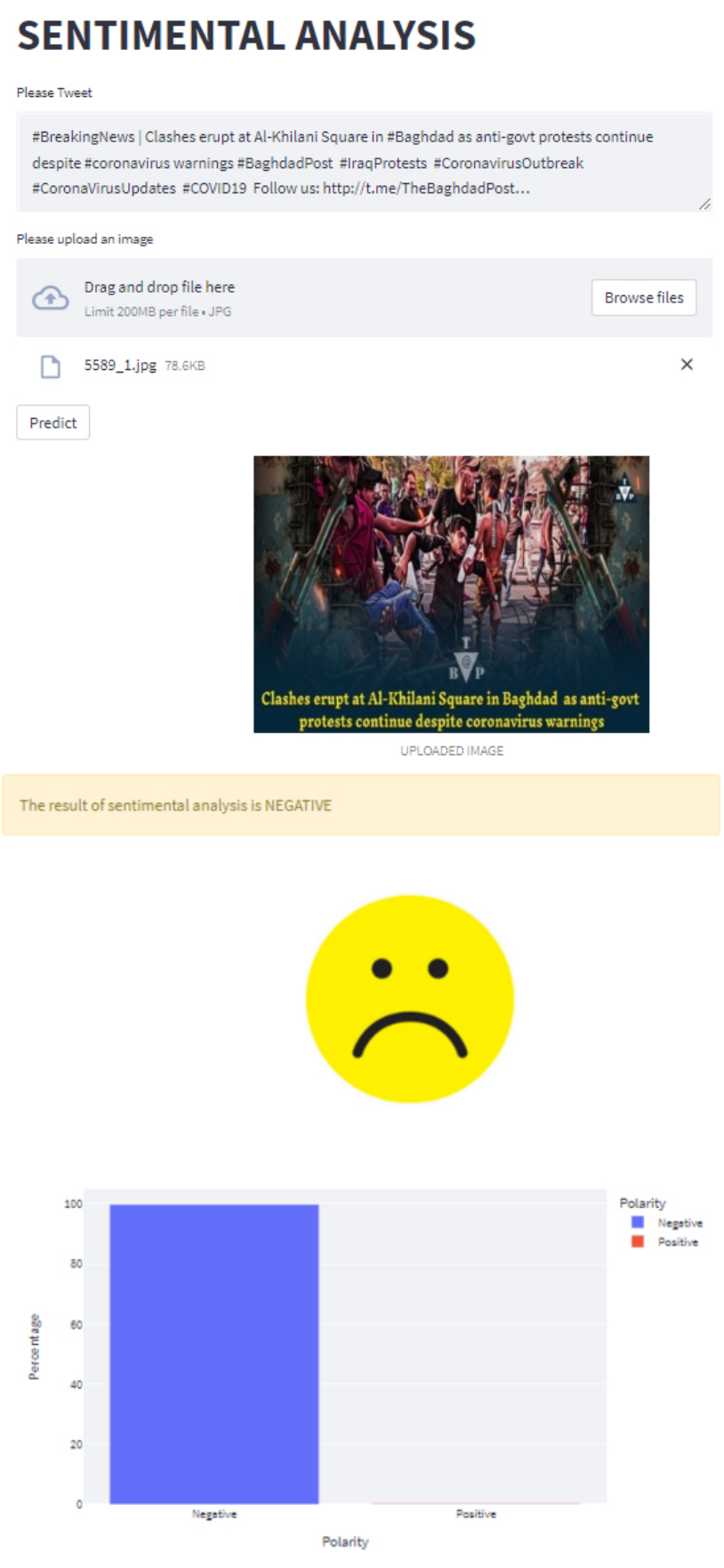
SENTIMENTAL ANALYSIS
Please Tweet
#BreakingNews | Clashes erupt at Al-Khilani Square in #Baghdad as anti-govt protests continue despite #coronavirus warnings #BaghdadPost #IraqProtests #CoronavirusOutbreak #CoronaVirusUpdates #COVID19 Follow us: http://t.me/TheBaghdadPost...
Please upload an image
Drag and drop file here
Limit 200MB per file • JPG
Browse files
5589_1.jpg 78.6KB
Predict
Clashes erupt at Al-Khilani Square in Baghdad as anti-govt protests continue despite coronavirus warnings
UPLOADED IMAGE
The result of sentimental analysis is NEGATIVE
Polarity
Negative
Positive
Percentage
100
80
60
40
20
0
Negative
Positive
Polarity

Figure 6.9

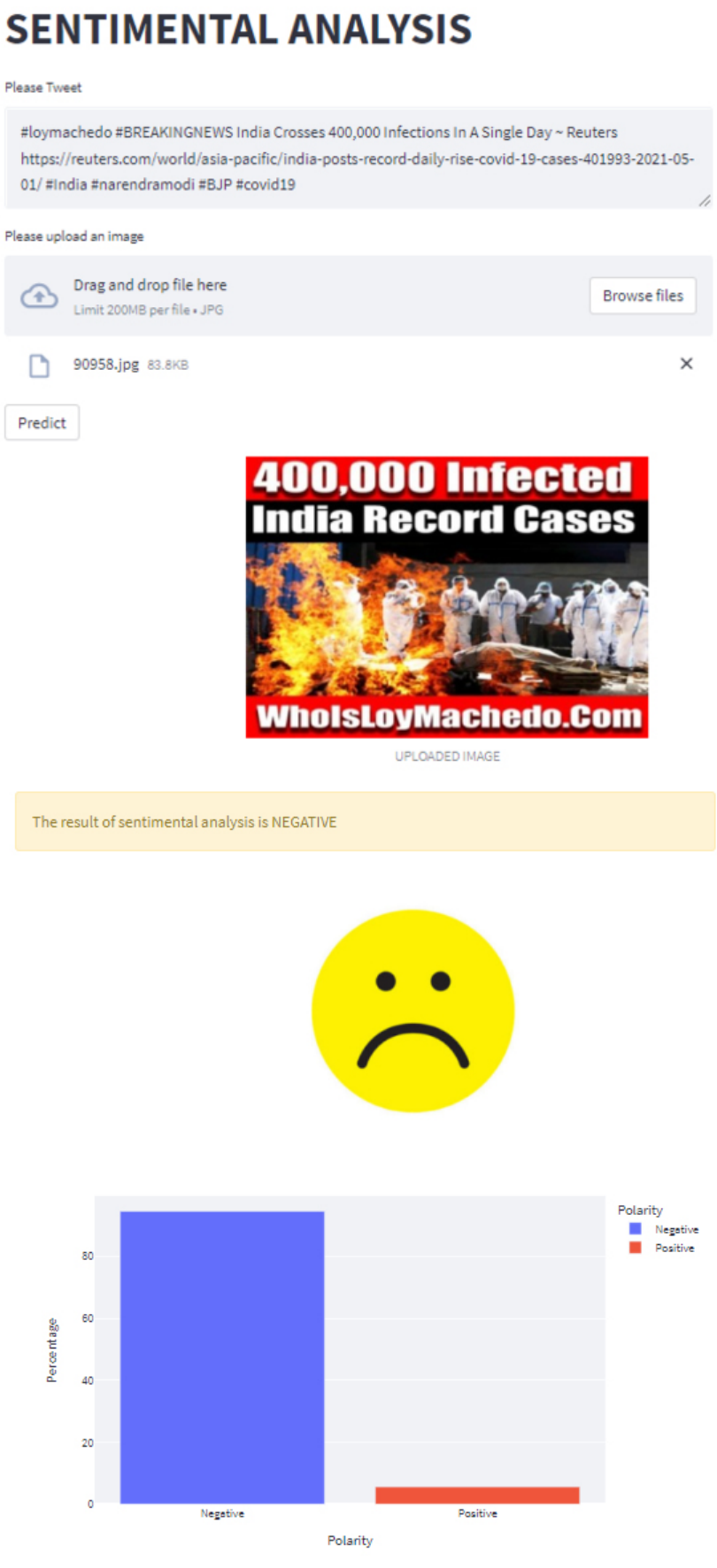
SENTIMENTAL ANALYSIS
Please Tweet
#loymachedo #BREAKINGNEWS India Crosses 400,000 Infections In A Single Day ~ Reuters https://reuters.com/world/asia-pacific/india-posts-record-daily-rise-covid-19-cases-401993-2021-05-01/ #India #narendramodi #BJP #covid19
Please upload an image
Drag and drop file here
Limit 200MB per file • JPG
Browse files
90958.jpg 83.8KB
Predict
400,000 Infected
India Record Cases
WhoIsLoyMachedo.Com
UPLOADED IMAGE
The result of sentimental analysis is NEGATIVE
Polarity
Negative
Positive
Percentage
80
60
40
20
0
Negative
Positive
Polarity

Figure 6.10

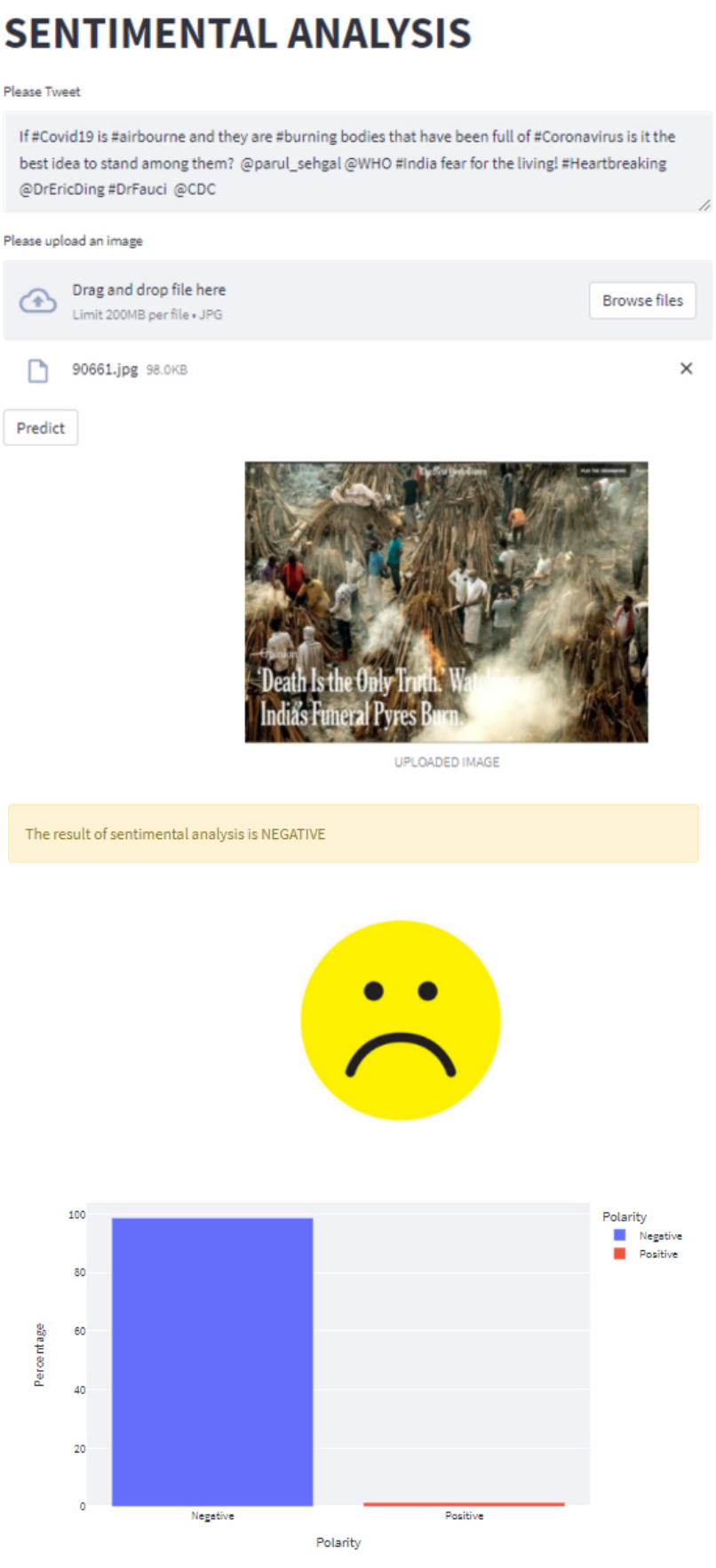
SENTIMENTAL ANALYSIS
Please Tweet
If #Covid19 is #airbourne and they are #burning bodies that have been full of #Coronavirus is it the best idea to stand among them? @parul_sehgal @WHO #India fear for the living! #Heartbreaking @DrEricDing #DrFauci @CDC
Please upload an image
Drag and drop file here
Limit 200MB per file • JPG
Browse files
90661.jpg 98.0KB
Predict
UPLOADED IMAGE
The result of sentimental analysis is NEGATIVE
Polarity
Negative
Positive
Percentage
100
80
60
40
20
0
Negative
Positive
Polarity

Figure 6.11

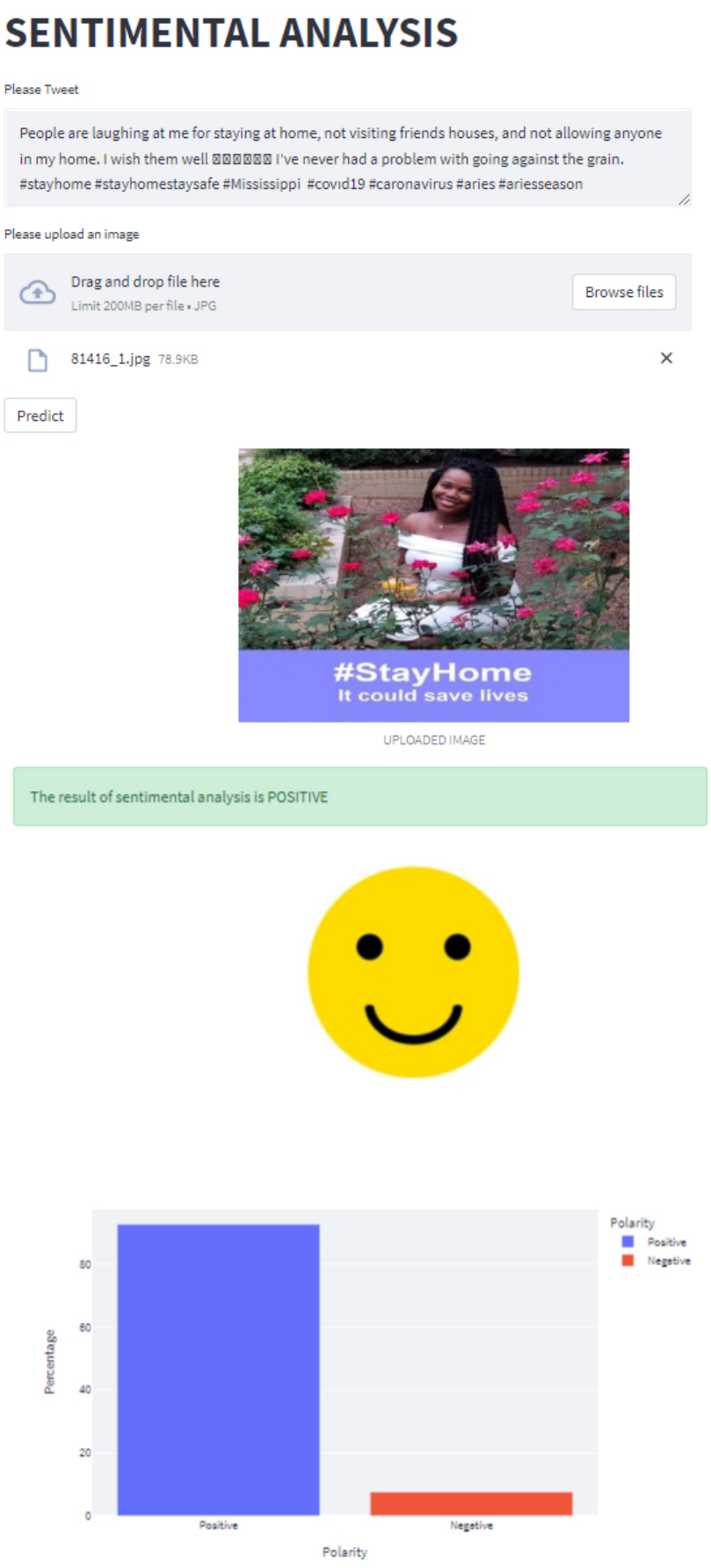
SENTIMENTAL ANALYSIS
Please Tweet
People are laughing at me for staying at home, not visiting friends houses, and not allowing anyone in my home. I wish them well ▯▯▯▯▯▯▯ I've never had a problem with going against the grain. #stayhome #stayhomestaysafe #Mississippi #covid19 #caronavirus #aries #ariesseason
Please upload an image
Drag and drop file here
Limit 200MB per file • JPG
Browse files
81416_1.jpg 78.9KB
Predict
#StayHome
It could save lives
UPLOADED IMAGE
The result of sentimental analysis is POSITIVE
Polarity
Positive
Negative
Percentage
0
20
40
60
80
Positive
Negative
Polarity

Figure 6.12

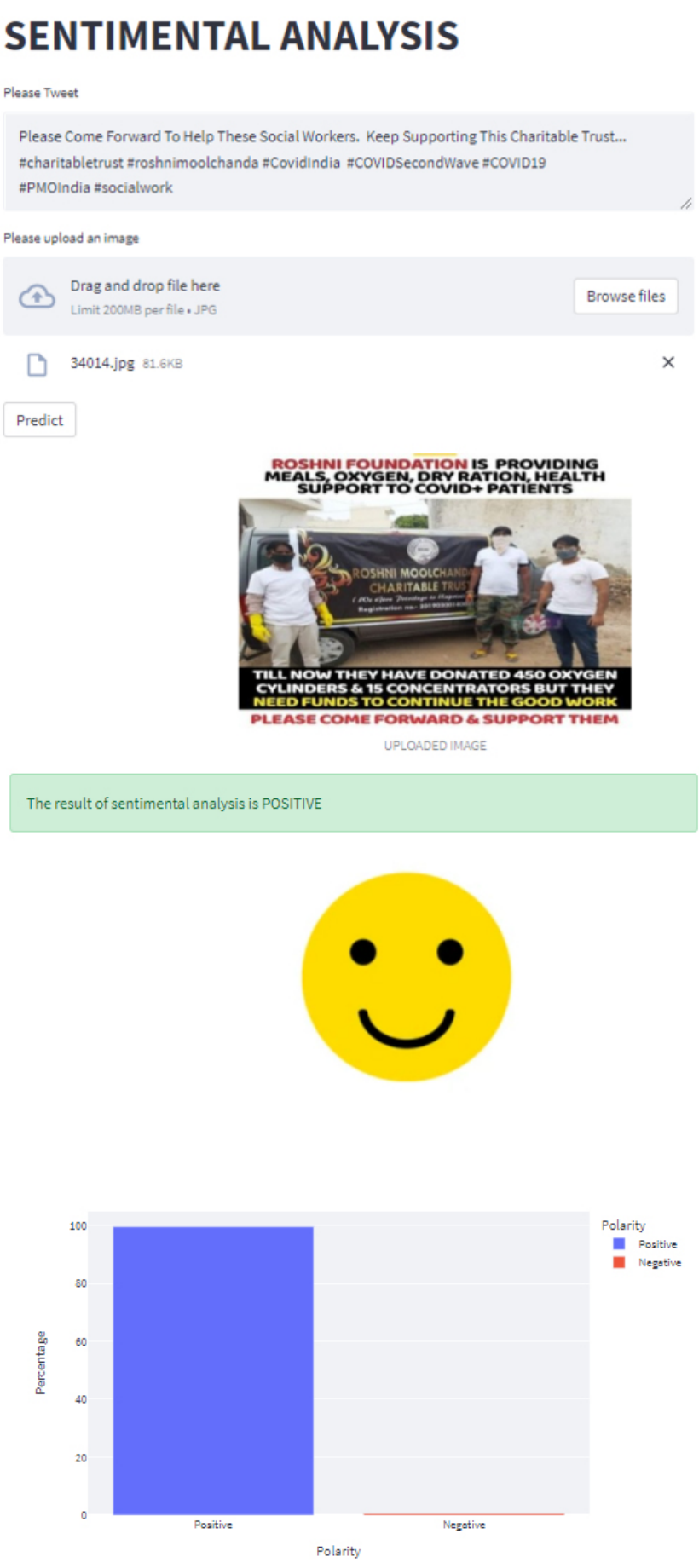
SENTIMENTAL ANALYSIS
Please Tweet
Please Come Forward To Help These Social Workers. Keep Supporting This Charitable Trust...
#charitabletrust #roshnimoolchanda #CovidIndia #COVIDSecondWave #COVID19
#PMOIndia #socialwork
Please upload an image
Drag and drop file here
Limit 200MB per file • JPG
Browse files
34014.jpg 81.6KB
Predict
ROSHNI FOUNDATION IS PROVIDING MEALS, OXYGEN, DRY RATION, HEALTH SUPPORT TO COVID+ PATIENTS
TILL NOW THEY HAVE DONATED 450 OXYGEN CYLINDERS & 15 CONCENTRATORS BUT THEY NEED FUNDS TO CONTINUE THE GOOD WORK
PLEASE COME FORWARD & SUPPORT THEM
UPLOADED IMAGE
The result of sentimental analysis is POSITIVE
Polarity
Positive
Negative
Percentage
100
80
60
40
20
0
Positive
Negative
Polarity

Figure 6.13


SENTIMENTAL ANALYSIS
Please Tweet
Don't Let The Opportunity To Be Part Of Medical History Pass You By. Join Coronavirus Vaccine Studies. Earn Up To $1,325, Which Varies By Study, If You Qualify. For more information visit: https://bit.ly/3nPz7zg #COVID19 #ClinicalTrials #ClinicalResearch #medicalresearch
Please upload an image
Drag and drop file here
Limit 200MB per file • JPG
Browse files
26667.jpg 73.6KB
Predict
HEALTHY VOLUNTEERS 18+
JOIN TRIALS TO MAKE
MORE COVID VACCINES
JOIN TODAY
NO INSURANCE NEEDED
GET QUALIFIED IN MINUTES
UPLOADED IMAGE
The result of sentimental analysis is POSITIVE
Polarity
Positive
Negative
Percentage
100
80
60
40
20
0
Positive
Negative
Polarity

Figure 6.14

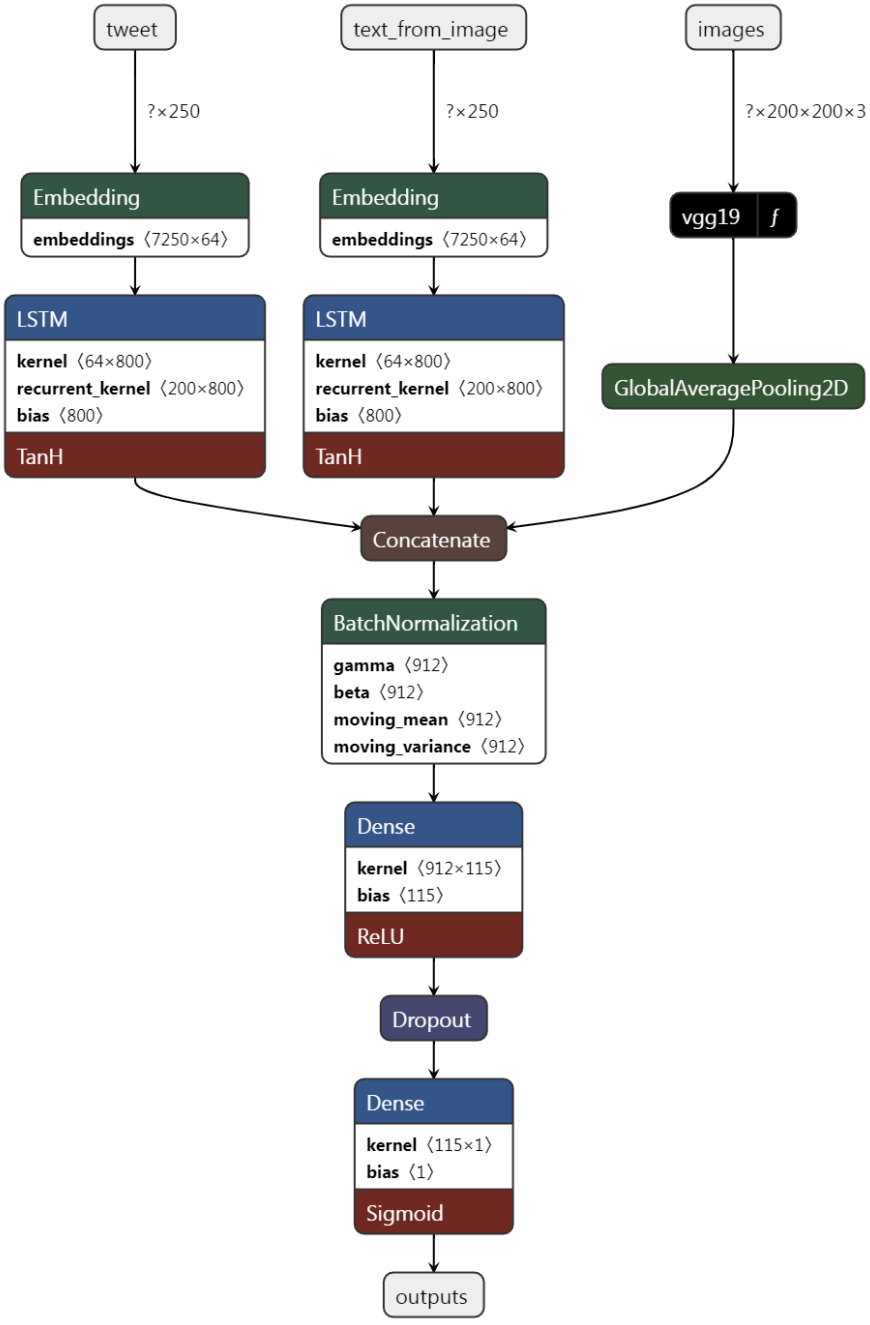

Figure 6.15. Baseline(LSTM) Neural Network Architecture