

ANALYZING PUBLIC SENTIMENT ON COVID-19 PANDEMIC

ANALYZING PUBLIC SENTIMENT ON COVID-19 PANDEMIC

A PROJECT

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

by

Pradeepika Gedupudi

June 2021

© 2021

Pradeepika Gedupudi

ALL RIGHTS RESERVED

The Designated Project Committee Approves on the Project Titled
ANALYZING PUBLIC SENTIMENT ON COVID-19 PANDEMIC

by

Pradeepika Gedupudi

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

June 2021

Dr. Robert Chun

Department of Computer Science

Dr. Fabio Di Troia

Department of Computer Science

Mr. Avinash Chander Gottumukkala

Software Engineer, SAP Ariba

ABSTRACT

Sentiment analysis is a method of understanding the user sentiment expressed in the form of text. Social media is the best place to capture the public's opinion regarding how they feel about current events. The Corona Virus Disease-2019 (COVID-19) is one of the worst pandemics we have experienced so far. An important observation is that this pandemic has not only affected the public's physical health but also took a toll on their mental health. Reddit is a social news discussion site where people discuss topics around current affairs in smaller groups called subreddits. The project's primary focus is to build a deep learning model that can classify and help analyze user sentiments about Covid-19 on Reddit. The model has been built by evaluating the performance of different classifiers on the Twitter dataset, called Sentiment140. Experiments with varying feature combinations have been evaluated on deep learning models, including Convolutional Neural Networks (CNN) and Long-Short Term Memory (LSTM). The idea is to build a model by combining the best of both architectures. LSTM excels at storing the forward information, whereas CNN can capture the local features. After reviewing these experiments, the best-performing model has been used to classify and analyze the sentiment of the Reddit users over different changes due to the Covid-19 pandemic. Overall, there have been some interesting changes in user reaction trends for posts related to Covid-19 under each subreddit over thirteen months, starting from Mar '20 to Mar '21.

Keywords: COVID-19, Sentiment Analysis, Sentiment140, Deep Learning, Convolutional Neural Networks, Long-Short Term Memory

ACKNOWLEDGMENTS

I want to utilize this opportunity to sincerely thank my project advisor, Dr. Robert Chun, for his continuous assistance and valuable supervision at every stage of this project. His work ethic and constant encouragement to strive for perfection have been a great source of inspiration and pushed me to accomplish this project.

I want to express sincere gratitude to my defense committee members Dr. Fabio Di Troia and Mr. Avinash Chander Gottumukkala, for their valuable recommendations to my project.

Lastly, I am incredibly grateful to my family and friends for their constant support and motivation throughout my graduation.

TABLE OF CONTENTS

I	Introduction.....	1
	1.1. Motivation.....	2
II	Background.....	4
	2.1 Text Pre-processing techniques.....	4
	2.2 N-grams.....	5
	2.3 Word Embeddings.....	6
III	Deep Learning Models.....	10
	3.1 Convolutional Neural Networks.....	10
	3.2 Recurrent Neural Networks.....	14
	3.3 Long-Short Term Memory.....	15
	3.4 CNN-LSTM.....	17
IV	Dataset Preparation.....	20
	4.1 Dataset 1 – Sentiment140.....	20
	4.2 Dataset 2 – Reddit data.....	24
V	Proposed Models.....	29
	5.1 Proposed Model-A.....	29
	5.2 Proposed Model-B.....	31
VI	Experiments and Model Results.....	32
	6.1 Experimental Environment	32
	6.2 Experiments with stop words.....	32
	6.3 Experiments with N-gram models.....	33

6.4 Experiments with Word Embeddings.....	35
6.5 Experiments with Proposed Model-A.....	37
6.6 Experiments with Proposed Model-B.....	38
VII Results on Test Data.....	41
7.1 Test Results on Topic 1: Economy.....	43
7.2 Test Results on Topic 2: Layoffs.....	45
7.3 Test Results on Topic 3: Social-Distancing.....	46
7.4 Test Results on Topic 4: Academics.....	48
VIII Conclusion.....	50
IX Future Scope.....	51
LIST OF REFERENCES.....	52

LIST OF TABLES

Table 1	Subreddits grouped under each topic.....	27
Table 2	Accuracies obtained with different combinations of n-grams.....	33
Table 3	Evaluation of different word embedding techniques with CNN.....	35
Table 4	Evaluation of different CNN-LSTM hybrid models.....	37
Table 5	Metrics used for scaling the reactions.....	43

LIST OF FIGURES

1	Impact of Covid-19.....	2
2	Skip-gram model.....	7
3	Selecting a row corresponding to 1 in row matrix.....	7
4	Model architecture of Continuous Bag of Words.....	8
5	Sample implementation Glove model.....	9
6	5*5 image & 3*3 filter.....	10
7	Convolved Feature.....	11
8	Average pooling.....	12
9	1-dimensional Convolution operation.....	12
10	Working of 1Conv - CNN.....	13
11	Feed-forward network & Recurrent neural network.....	14
12	Single-cell LSTM.....	16
13	Combining CNN-LSTM.....	18
14	Raw data from the Sentiment140 dataset.....	20
15	Dictionary built to handle negative words.....	22
16	Data cleaning process.....	23
17	Frequency of terms.....	24
18	Custom built list of stop words.....	24
19	A sample comment returned by the API.....	26
20	Data collection process.....	27
21	CNN-LSTM model with CBOW + Skip-gram features.....	30

22	CNN-LSTM model with three parallelly arranged CNNs.....	31
23	Accuracies plotted against the number of features.....	33
24	Graph showing increase in accuracy for combination of 2 n-gram models....	34
25	Performance of different word embeddings with CNN.....	36
26	Accuracies and F1-scores of the experimented models.....	39
27	Performance of Proposed Models vs Textblob classifier.....	39
28	Graph showing the demographics of Reddit users..	41
29	Increase in number of users through 2020.....	42
30	U.S. Economy GDP variation over 2020.....	44
31	Scaled Reactions over 13-months for Economy	44
32	Scaled Reactions over 13-months for Layoffs	45
33	Variation in the unemployment rate from Mar'20 to Mar'21.....	46
34	Scaled Reactions over 13-months for Social Distancing.....	47
35	Scaled Reactions over 13-months for Academics.....	48
36	A graph showing change in sentiment in all the four areas impacted.....	49

I. INTRODUCTION

In the present day, social media is one of the primary platforms used by people to discuss current news and express opinions. Understanding the sentiment of active social media users to the various changes provides insight into the effect of the pandemic.

For example, many enterprises, big or small, are under losses due to a slowdown in manufacturing essential goods. The supply chain has been disrupted, and revenue growth has dropped steeply. Lockdown restrictions and the fear of the spread of infections have reduced labor supply.

In addition, layoffs have impacted laborers. According to Lenzen et al. in [1], waged and self-employed workers have been working without proper safety and protection mechanisms, leading to greater exposure to the virus. Migrant workers have been plagued by displacement and travel restrictions.

One of the root causes of the massive increase in the number of cases is the lack of proper quarantine commitment [1]. Furthermore, lack of proper quarantine commitment has led to a massive increase in the number of Covid-19 positive cases, which has overwhelmed hospitals and the healthcare system. Doctors and healthcare professionals are at high risk of infection [2]. Therefore, social media can be harnessed

to analyze public reactions and user sentiment around the pandemic to assess its impact on mental health better.

1.1 Motivation

The objective of this research is to answer the following questions – i) *Can a robust classifier that analyses sentiment on other social media sites be built?* ii) *How has Covid-19 affected the mental health of the public for over 2020 and early 2021?* iii) *Is there a correlation between the change in Reddit users' sentiments and the severely impacted areas?*



Figure 1: Impact of Covid-19 [2]

Figure 1 shows the impact of the pandemic on everyday life [2]. Since Covid-19, websites like Reddit and Twitter have become the primary source of sharing opinions. Reddit, also known as "the front page of the Internet", is one such social sharing website where people discuss different topics in smaller communities called subreddits. Redditors make posts under these subreddits and also post their opinions as comments

under these posts. Valuable insights can be drawn by extracting the comments and analyzing users' sentiments about Covid-19. Moreover, Reddit's user base has doubled over the year 2020 [3].

II. BACKGROUND

Natural Language Processing (NLP) is an area of intersection of Computer Science and Artificial Intelligence. The aim is to understand the natural language and perform tasks like machine language translation, information extraction, etc. Sentiment Analysis is a subsection of NLP. It is a method of understanding whether the emotion behind the text is positive or negative. It combines NLP and machine learning or deep learning techniques to assign weighted sentiment scores for a sentence. It helps researchers understand if the public opinion towards a product or brand is positive or negative. Many enterprises use sentiment analysis to gather feedback and provide a better experience to the customer. There is a set of general pre-processing steps that are followed for any machine learning classifier to understand the sentiment of the text. Text pre-processing is the first step to make input data ready before feeding it to the classifier. N-grams are a set of co-occurring words within a given window, which can be used as the building blocks for the classifier. Word embeddings can be generated from the input data to capture the context of the sentences. These steps are explained below in more detail.

2.1 Text pre-processing techniques

2.1.1 Tokenization

Tokenization is the process of breaking sentences into tokens and ignoring all the punctuation. Each token is an instance of a character sequence that holds meaning.

2.1.2 Stop words

Stop words are commonly occurring words like 'the' 'I', 'in' etc., that hold very little meaning and are not useful in understanding the sentence's sentiment. Such words can be avoided entirely from the vocabulary by using built-in functions made for the English language. Another approach is to build a custom list. This can be done by sorting the terms in decreasing order of their frequency. By removing stop words that don't add value to the analysis, the vocabulary size can be reduced.

2.1.3 Building a vocabulary

Count vectorizer helps convert a collection of text documents to a vector of term counts. For every sentence, the value of each cell would be the number of words in that sentence.

2.2 N-grams

This is a set of words occurring within a given window. N-gram can be thought of as a sequence of N words; for example, the trigram is a sequence of 3-words like 'How are you?'. N-gram follows the Markovian assumption, which means a word's probability depends on the preceding term [4]. The intuitive formula is $P(w/h)$ for the word 'w,' given some history 'h', which means the probability of the number of times you see the history of sentences followed by the word w.

Uni-gram is built by assuming that adjacent words are independent of each other, and there is no mutual information.

$$P(w_1 w_2 \dots w_n) \approx \prod_i P(w_i) [4]$$

Bi-gram is built with an assumption that for two contiguous words, the probability of a particular word depends on the previous word.

$$P(w_i | w_{i-1}) \approx P(w_{i-1}, w_i) / P(w_{i-1}) [4]$$

The choice of the n-gram model to be used heavily depends on the data sparsity.

2.3 Word Embeddings

Word2Vec is a set of architectures and is optimized for enormously large datasets, and these are useful to learn word embeddings. Word2Vec focuses on training a neural network with one hidden layer, and it uses the neighboring words to predict a target word. This approach is excellent for classifying words without losing the context of their occurrence.

In word embeddings, the main task is to map words to vectors. The embedding from multi-dimensional space per word is mapped to a continuous vector space by reducing the dimensions. Mikolov et al. in [5] discuss two important architectures in Word2Vec models, namely Continuous Bag of Words (CBOW) and Skip-Gram.

2.3.1 Skip-Gram

This model focuses on classifying an entire sentence using a single word. Once the vocabulary has been built, each word would be present as a one-hot vector. The output from the network is the probability of a nearby randomly selected word [6].

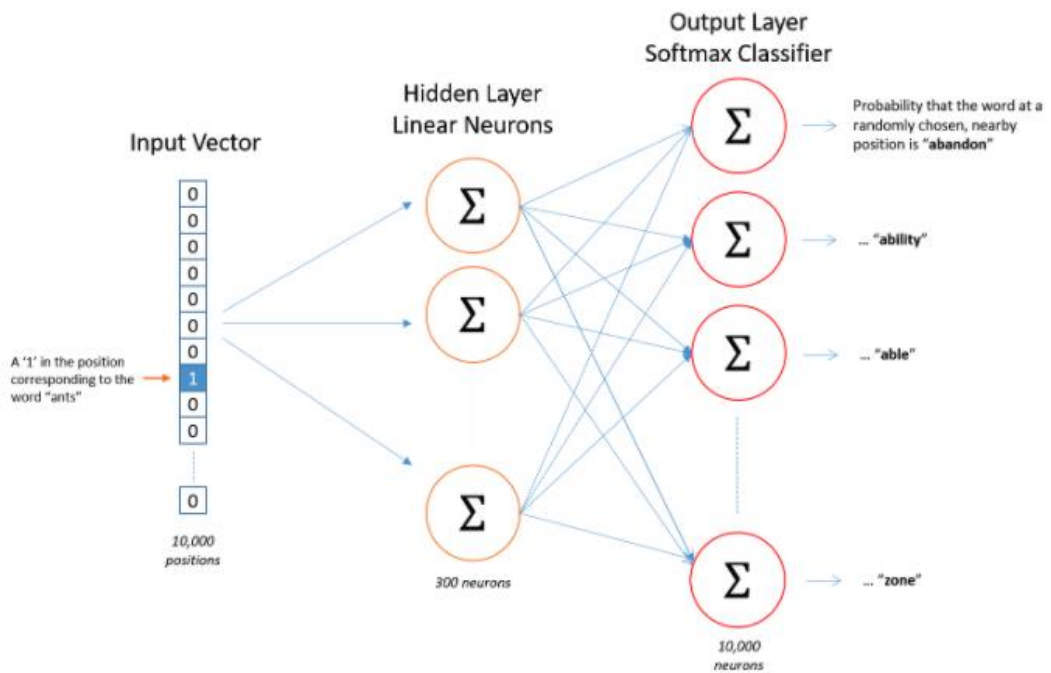


Figure 2: Skip-Gram model [6]

As seen in figure 2, input is the one-hot vector. In the following layers, there is no activation function used in the hidden layers. Figure 3 shows that only the weights corresponding to the 1's present in the row matrix are learned.

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 17 & 18 & 20 \\ 21 & 23 & 65 \\ 23 & 45 & 12 \\ 34 & 21 & 15 \\ 22 & 34 & 14 \\ 13 & 24 & 25 \end{bmatrix} = \begin{bmatrix} 23 & 45 & 12 \end{bmatrix}$$

Figure 3: Selecting a row corresponding to 1 in row matrix

The output layer is a classifier which is a SoftMax regression. Every neuron will produce output between 0 and 1. Every output neuron will contain a weight vector that is multiplied by the hidden layer's weight vector. Although the Skip-Gram model doesn't work very well for words with high frequency, it is a good choice for words that occur less frequently.

2.3.2 CBOW

This model tries to focus on predicting the current word based on the neighboring terms. As shown in figure 4, the input layer consists of one-hot encoded context words. A hidden layer follows the one-hot encoded vectors layer.

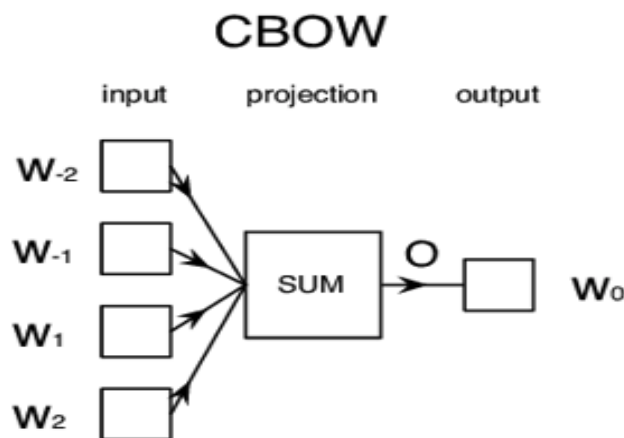


Figure 4: Model architecture of Continuous Bag of Words [7]

For example, a sentence such as 'It is raining today' would be converted to ([input words], target word). For window size 2, the word pairs would be ([It, is], raining) and ([is, raining], today). With the help of these word pairs, the target word is predicted. For predicting a single context word with the help of 'n' context words, an input layer

of size $n * W * N$ is used where 'W' indicates the word, and 'N' indicates the total number of words.

2.3.3 Glove2Vec

Glove2Vec, developed by Pennington et al. in [8], is another popular way to train word embeddings. It works by fitting vectors to a huge word co-occurrence matrix that has been built from a corpus. Glove2Vec is a "count-based" model as opposed to word2vec being a predictive model. A co-occurrence matrix is constructed by calculating the frequency of a particular word in the context. A lower dimension matrix is built by reducing the dimensions. Each row in this matrix is a vector representation of a word.

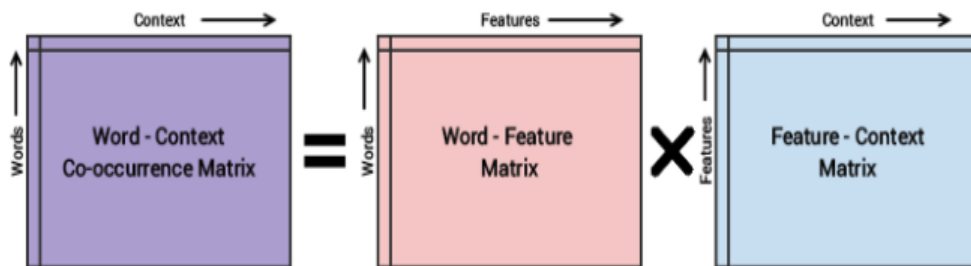


Figure 5: Sample implementation of Glove model [8]

As shown in figure 5, the count matrix is built by normalizing the counts and performing log smoothing, ensuring good quality in the learned representation.

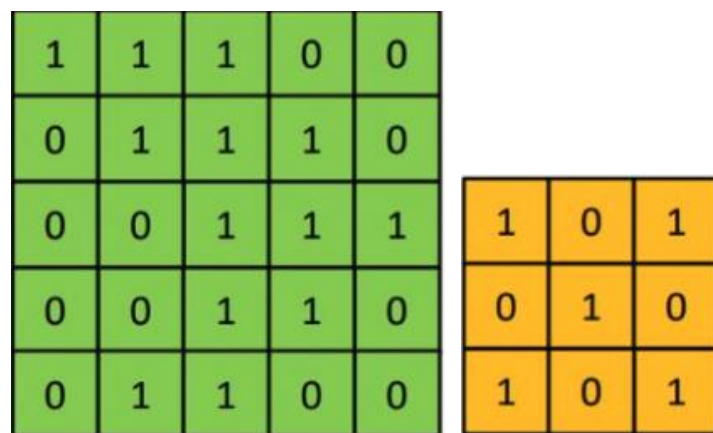
III. Deep Learning Models

3.1 Convolutional Neural Networks

Convolutional Neural Network was first introduced by LeCun et al. It is a common class of neural network that is very commonly applied to visual data [9]. It is widely used to classify images, object detection, etc. Over the last few years, CNNs have become increasingly popular for dealing with NLP tasks like Sentiment Analysis, Machine Translation, Text Summarization, Sentence Classification, etc. The CNN architecture is mainly composed of convolution layers, down-sampling layers, and fully connected layers, which are explained below in more detail.

3.1.1 Convolution Layer

This layer comprises a series of "feature maps," where each feature map has a linear map obtained by applying a convolution operation over the layers. An activation function is applied to the output.



1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

Figure 6: 5*5 image & 3*3 filter [9]

4	3	4
2	4	3
2	3	4

Figure 7: Convoluted Feature [9]

Consider the following example shown in figure 6, where 5*5 sized binary pixel values are taken and a 3*3 filter is applied over it [9]. A convolution operation is used over the image with a kernel filter. Figure 7 shows the resulting feature map obtained [9].

3.1.2 Down-Sampling Layer

Down-sampling is a method used to reduce the number of parameters. There are various down-sampling approaches, and the most common ones are:

- (i) **Average Pooling:** The approach of taking the average of the elements from the map
- (ii) **Max Pooling:** The approach of taking the largest component of each window
- (iii) **Sum Pooling:** The approach of taking the sum of all parts in the feature map

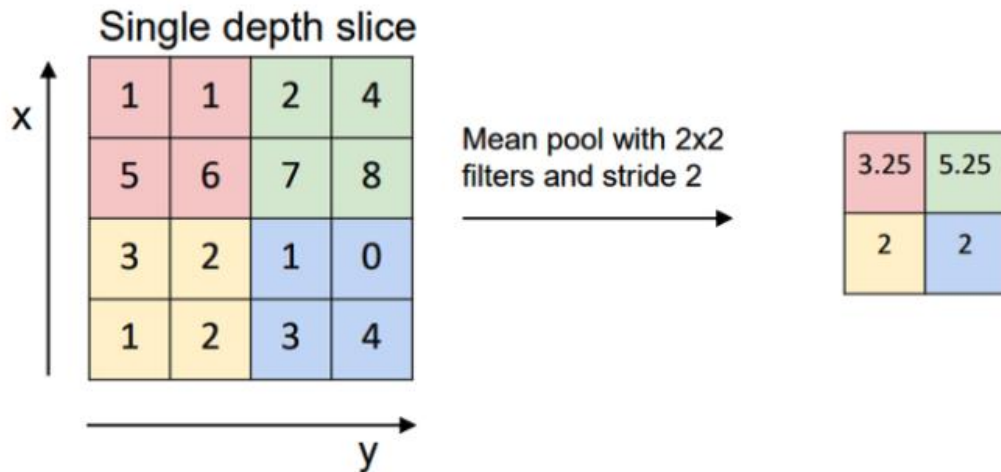


Figure 8: Average Pooling [10]

Once the feature map has been generated after down-sampling based on the pooling technique chosen, the intermediate output can be seen in figure 8. In the final layer, activation functions like SoftMax, sigmoid, etc., can be applied to classify the input.

3.1.3 CNN For Text

When dealing with visual data, images are converted to an array of pixel values. In a similar way, text data is converted to an array of vectors. This is then fed to CNN. While dealing with text over images, one-dimensional convolutions are used instead of two-dimensional or n-dimensional [10].

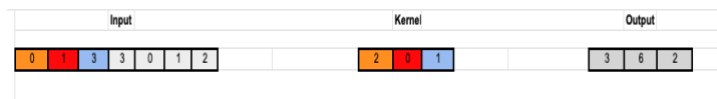


Figure 9: 1-dimensional Convolution operation

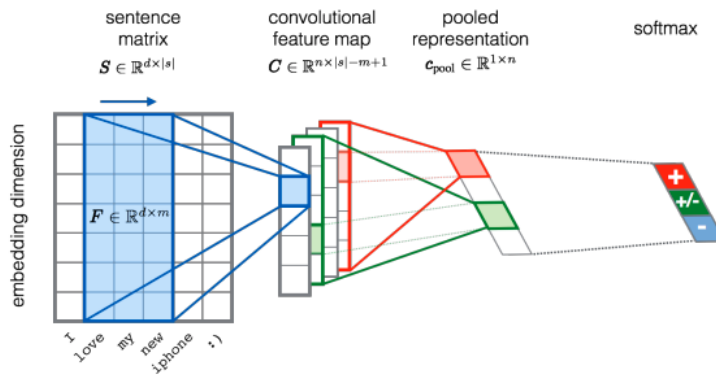


Figure 10: Working of 1Conv - CNN [11]

Using 1-D convolutions for text data is very useful and can be seen as an alternative to using RNNs, as the computations are faster due to running the operations in parallel. As seen in figure 9, the convolution operation for a single dimension is much simpler, and the kernel is slid only in one direction. The overall architecture of CNN for text data can be seen in figure 10.

3.2 Recurrent Neural Networks (RNN)

Traditional neural networks still face the issue of memory. The information is sent forward from the input layer to the output layer in feed-forward networks. It is passed through the network only once. As shown in figure 11a, feed-forward networks have no memory of the input, and due to this, they are almost inefficient in predicting the subsequent outcome. They only keep an account of the current data that is being processed.

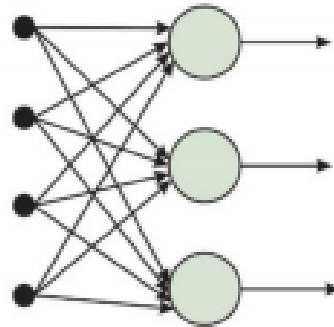


Figure 11a: Feed-forward network [11]

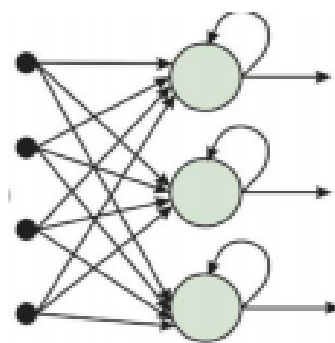


Figure 11b: Recurrent neural network [11]

For example, given the word "Covid" as input, if the word were to be processed alphabet by alphabet, by the time it reaches the alphabet 'i', it would have forgotten 'C' and 'o'. RNNs come under a class of neural networks that are traditionally good at modeling sequential data. They are used in popular voice assistants like Siri etc.

As shown in figure 11b, RNNs remember both current data and the data from the earliest inputs. Although RNN does a decent job at remembering the input, it still has memory issues, and the information is only retained for a short-term period. The RNN assigns weights to current and past inputs. The weights for gradient descent and Back Propagation Through Time (BPTT) can be modified. With RNN, one or many inputs can be mapped to one or many outputs.

In a feed-forward network, the output of the model is produced by moving forward and propagating the error. On the other hand, in backpropagation, the partial derivatives of the error concerning the weights are obtained by moving backward. This error is later deducted from the weights. The partial derivatives of the input function are the gradients. Exploding gradients is an issue with RNNs when the weights are given a lot of priority. This is prevented by neglecting the gradients whenever the gradients are minimal.

3.3 Long-Short Term Memory

In [13], Hochreiter et al. mentions how the LSTM gives importance to the essential information. Each unit, which is ideally a building block, is called an LSTM cell. From

its memory, LSTM can choose to read, update, and remove information. The diagram in figure 12 shows how an LSTM uses three different gates for this above functionality.

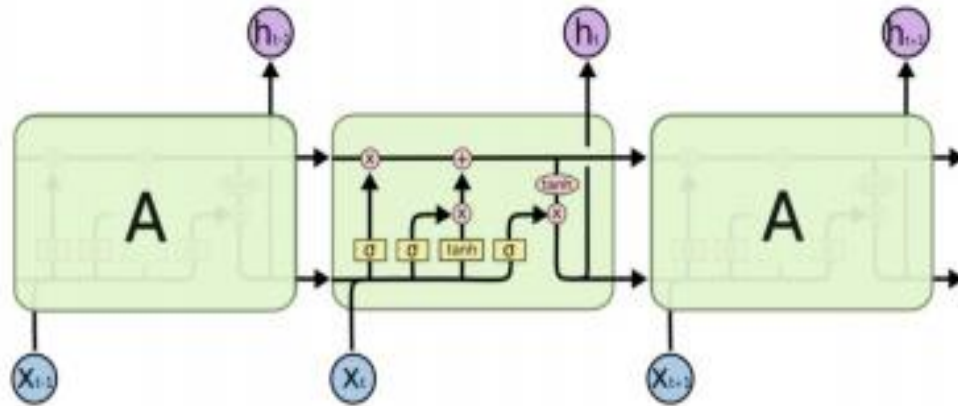


Figure 12: Single-cell LSTM [14]

3.3.1 Input Gate

The input gate is represented by $(h_{t-1} + x_t)$. It gives out an intermediate output using the sigmoid activation function. Another \tanh activation function c_t , generates an output between -1 to 1. The product of the input gate i_t , and c_t gives out a new output function.

3.3.2 Forget Gate

This gate is the most important one in the entire architecture of LSTM. It focuses on differentiating between relevant and irrelevant information and pushes forward only the critical information to the cell state $(h_{t-1} + x_t)$. Here, h_{t-1} is the previous hidden state, and x_t is the current input. The sigmoid of this sum is taken. Any information close to zero means it is irrelevant, and the higher the value, the more relevant it is.

3.3.3 Output Gate

Sigmoid of $h_{t-1} + x_t$ is processed by the output function within 0 and 1. The product of this and cell state information (\tanh activation function) gives output information o_t .

Some variations of LSTM, like Gated Recurrent Unit (GRU), don't have an output gate [14].

RNN, in general, being a model that works great with sequence data, can tackle input sentences of variable length and dependencies [14]. In [15], it can be observed that the standard LSTM was modified as a tree-structured topology to overcome the shortcomings of the LSTM model.

3.4 CNN-LSTM

Both CNN and LSTMs work in different ways and are very powerful in practice [16]. The idea is to combine both architectures and leverage the best of both. One promising approach could be to use CNN to obtain the higher-level sequence of word features and later use LSTM to capture all the long-term dependencies over window feature sequences [17].

3.4.1 Extracting features using convolution

Convolution involves sliding the filter over an input sequence and detecting the features. For d -dimensional word vectors, for every i th word in the sentence, a window vector w_j in the sentence for consecutive word vectors of length k will be generated. A filter of size n can be used and convoluted over these feature vectors to create a feature map. Experiments with different non-linear transformation functions like sigmoid, hyperbolic tangent, or the rectified linear unit can be conducted, and the best suited can be used in the convolution layer. In the down-sampling stage, the max-pooling method can be used to get a word feature representation, and this max-pooling action can help reduce the

dimensionality of data features and decrease overall complexity. For example, output from the max-pooling layer would be $\lambda_k = \max [\text{vector } 1, \text{vector } 2, \dots, \text{vector } n]$. In the convolutional layer, multiple filters can be used to generate multiple feature maps. These would recursively be fed into each layer, and the high-order window representation will be the input to LSTM.

3.4.2 Combining LSTM with CNN

Figure 13 shows an example of how the two models can be combined to classify the text data [17]. Following the 1-D Convolution and max-pooling layer, the feature output from CNN can be fed to LSTM. Each of the λ_k (max-pooling outputs) can be fed as input to one LSTM at a time. The dropout technique in LSTM helps prevent data from being overfitted by dropping the irrelevant information from the model, which doesn't add value to the performance.

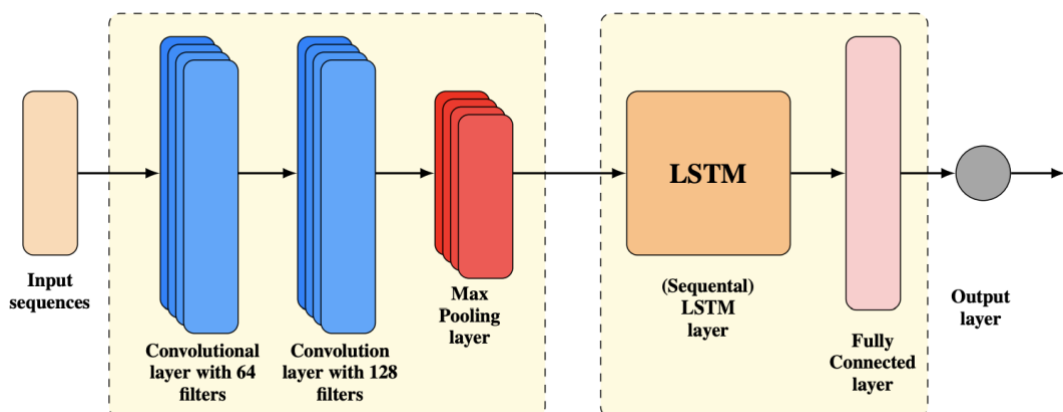


Figure 13: Combining CNN-LSTM [18]

The dense layer's purpose is to use weights and connect the input layer to the output layer. In figure 13, the fully connected output layer is a type of dense layer with SoftMax layer as an activation function. SoftMax classifier as the last stage layer helps average the random results between 0 to 1.

IV. DATASET PREPARATION

4.1 Dataset 1 – Sentiment140

The dataset that was used for model training is the Sentiment140 dataset, which is curated by Stanford University [19]. It comprises 1.6 million tweets. This dataset contains the tweet's polarity, tweet ID, date, username, and tweet text. For this project's purpose, the most important columns are the tweet's polarity and the text. Although there was a mention of a neutral class, there were no tweets with neutral labels. This dataset comprises two classes (positive and negative) equally balanced with no skewness. As the dataset was perfectly balanced, there was no need to implement any target class balancing techniques.

sentiment	id	date	query_string	user	text
0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zj - Awww, t...
1	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattyucus	@Kenichan I dived many times for the ball. Man...
3	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....
...
1599995	2193601966	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	AmandaMarie1028	Just woke up. Having no school is the best fee...

Figure 14: Raw data from the Sentiment140 dataset

Figure 14 shows the columns present in the dataset. Features like tweet ID, date of a tweet posted, and username have been discarded as they do not add value to the classification. In the sentiment column, it can be observed that 0 means negative and 4

means positive. The positive class, which was earlier denoted by 4, is now mapped to 1 as per the general labeling standards of sentimental analysis.

The experiment was performed by splitting the dataset into train, validation & test sets with a split ratio of 0.8, 0.1, and 0.1. K-fold cross-validation has been used for training the deep learning models to make the models more robust [20]. As per AY Ng, this is a good split ratio if the real-world test set size is much smaller than the train set [20]. This aligns very well with the project's system, as the test data scraped from Reddit for a few months can be close to 100,000 comments which is equivalent to the test set of the Sentiment140 dataset. The validation and test sets are of equal sizes. The best method is to take the whole dataset and shuffle it before splitting, as shuffling the dataset makes the model more robust [21].

4.1.1 Dataset 1 Cleaning

The following steps were performed for data cleaning:

- i. HTML Decoding has been done to generate the text in the tweets. It is much simpler to decode the text using BeautifulSoup, which is a python library.
- ii. A good amount of the text has a user mention '@' which is not useful for this purpose and can be eliminated as it adds no meaning to the tweet.
- iii. URL addresses are also irrelevant for this purpose and can be ignored. A pattern matching using regex can be performed to identify strings that start with 'https?:', 'http?:' and 'www.' And everything that matches these keywords can be eliminated.

- iv. Contradiction Mapping has been used to map words like "isn't" to "is not" and "doesn't" to "does not". As per Reitan et al. in [22], once the punctuation is removed, the new word with no punctuation becomes meaningless. For example, "isn't" after removing the punctuation would become "isnt". Such negative words are handled by building a dictionary, as shown in figure 15, where original words are the keys and the new words become the values. Every negative word that matches the key in the dictionary is now replaced with its equivalent values. This helps identify the negated emotions better.
- v. The punctuation (special characters) is removed, and the problem of case insensitivity is eliminated by converting the word to lower case [23].

```
#All the negative words are mapped to "*** not" as it is easy to identify if it comes under negative category
negations_dic = {"isn't": "is not", "aren't": "are not", "wasn't": "was not", "weren't": "were not",
                 "haven't": "have not", "hasn't": "has not", "hadn't": "had not", "won't": "will not",
                 "wouldn't": "would not", "don't": "do not", "doesn't": "does not", "didn't": "did not",
                 "can't": "can not", "couldn't": "could not", "shouldn't": "should not", "mightn't": "might not",
                 "mustn't": "must not"}
```

Figure 15: Dictionary built to handle negative words

Later, these sentences are tokenized, and the words are joined back. This 5-step process is implemented for all 1.6 million tweets in batches. Each batch consists of 100,000 tweets, with 16 batches in total. Figure 16 is a visual representation of the whole cleaning process. After cleaning, about 3500 null entries were dropped from the data frame as these added no value to the analysis.

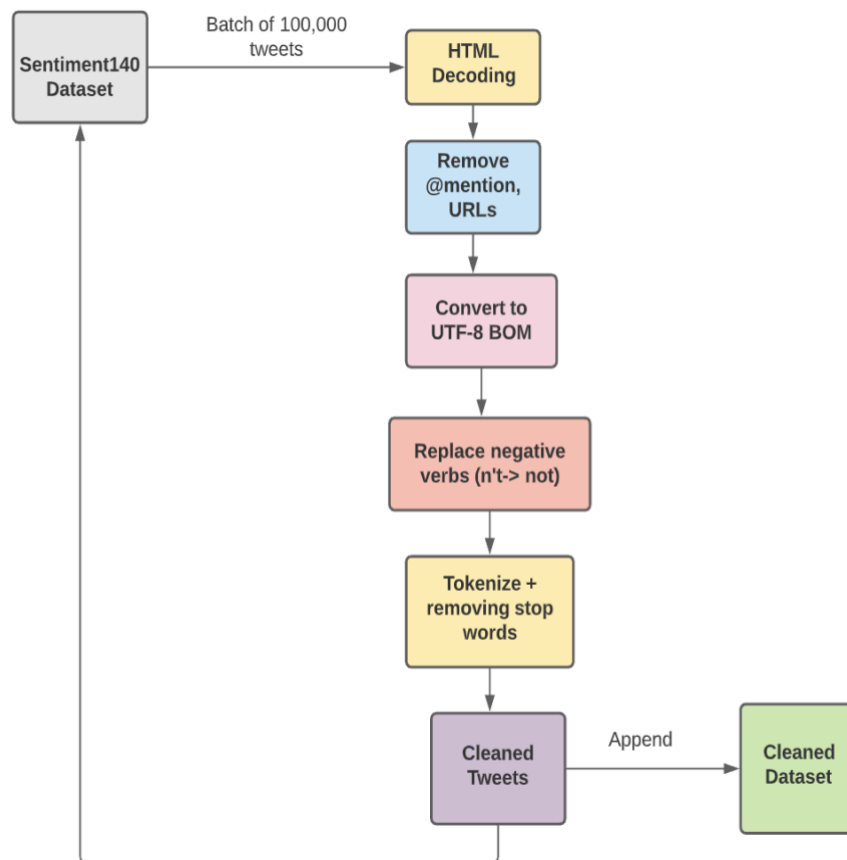


Figure 16: Data cleaning process

CountVectorizer library from scikit-learn can be used to fit the vectorizer for the entire document. When fitted to the text, the vectorizer extracted about 260000 words. 'Max_features' denotes the maximum possible number of features. A value can be assigned to 'max_features' by considering only the top features ordered by the frequency of the term across the whole corpus.

All the negatively labeled tweets are divided into batches, with a total number of batches of 100. Both negative and positive term definition data frames are built this way. The purpose was to create a 'term frequency' data frame and determine the

frequency of each term's occurrence in both positive and negative tweets. The terms were then sorted in decreasing order of frequency of their occurrence.

	negative	positive	total
to	313155	252567	565722
the	257834	265993	523827
my	190767	125957	316724
it	157444	147786	305230
and	153954	149642	303596
you	103842	198243	302085
not	194719	86865	281584
is	133429	111191	244620
in	115540	101158	216698
for	98999	117366	216365

Figure 17: Frequency of terms

```
1 custom_stop_words
↳ frozenset({'and', 'for', 'in', 'is', 'it', 'my', 'not', 'the', 'to', 'you'})
```

Figure 18: Custom built list of stop words

In figure 17, words like 'to,' 'the,' 'for,' 'in', which do not add much value to the sentiment analysis, were the highest occurring words. Figure 18 shows the list of custom-built stop words built from the dictionary shown in figure 17.

4.2 Dataset 2 – Reddit data

The dataset that was used for model testing was collected from Reddit using PushShift API. This is a RESTful API developed by the "/r/datasets" moderators' team at Reddit to help users make use of comments and submissions made on Reddit [24]. The API has two ways to use it either by (i) directly using 'api.pushshift.io/' or (ii) by leveraging

the back-end search engine using 'elastic.pushshift.io/'. For this project, the first method was used.

The three primary endpoints that the API provides are comments, subreddits, and submissions.

- a. Subreddit:** It is a subsection of Reddit, which focuses on a specific topic, can be accessed via 'reddit/search/subreddit'.
- b. Submissions:** Under a subreddit, each post by a user is called a submission. This can be accessed by hitting the endpoint '/reddit/search/submission'.
- c. Comments:** Each submission has a tree of words. The user writes text in response to the posts, which can be accessed via '/reddit/search/comment'.

The parameters that were used for scraping data from this project are:

- a. After:** All the comments after the 1st of Jan '20 have been retrieved. The date has to be in Unix timestamp format, as it tries to match with the 'created.UTC' parameter.
- b. Before:** This parameter helps set the latest date until which the data can be scraped. This also has to be in Unix timestamp format, similar to the 'after' parameter.
- c. Score:** Every submission has a score associated with it. If a comment has a high score, this means it has more upvotes and more people agree with it.
- d. Sort:** The submissions were sorted in decreasing order of comments with upvotes.

The data is returned in JSON format. An example of sample data returned is shown in figure 19.

```
{
  "data": [
    {
      "author": "SuperWonder",
      "author_flair_css_class": null,
      "author_flair_text": null,
      "body": "Definitely this would help! I have tried booking an appointment in the near by pharmacy, I was lucky to find one. My friend's in the bay haven't been able to find one, so they plan on visiting Sacramento to get vaccinated. I am glad I didn't have to deal with this..",
      "created_utc": 1614395172,
      "id": "c0nooiq",
      "link_id": "t3_b3uuu",
      "parent_id": "t1_c0ii5ux",
      "score": 2,
      "subreddit": "vaccines",
      "subreddit_id": "t5_2reo4e"
    }
  ],
  "metadata": {
    "execution_time_milliseconds": 32.2,
    "results_returned": 1,
    "shards": {
      "failed": 0,
      "successful": 36,
      "total": 36
    },
    "size": 1,
    "sort": "asc",
    "sort_type": "created_utc",
    "timed_out": false,
    "total_results": 134785,
    "version": "v3.0"
  }
}
```

Figure 19. A sample comment returned by the API

The API has a limit of retrieving only 100 submissions for every request. The comments were collected every 12 hours to ensure enough data is accumulated for the entire week. This means that the whole weeks' data is retrieved by calling the API about 14 times each week. Covid-19 related comments were collected for topics "Economy", "Layoffs", "Social-Distancing", and "Academics" with the intent to understand how these areas have been impacted.

The relevant subreddits have been grouped into the four topics shown in table 1. For example, for the topic 'Economy,' the comments would be gathered by scraping data under subreddits' #flatteningthecurve,' #flattenthecurve,' '#economy' etc. The search term will be set to 'covid' and passed as a path parameter to ensure that only Covid-19 related discussions are scraped.

	Topic	Matched Subreddits	Total Comments
1	Economy	'#flatteningthecurve','#flattenthecurve', '#economy'	223703
2	Layoffs	#layoff', '#fire', '#pandemiclayoff'	180903
3	Social-Distancing	"#socialdistancing", '#workfromhome'	204681
4	Academics	'#homeschool', '#lockdown', '#homeschoolin g', '#stayhomestaysafe'	279404

Table 1. Subreddits grouped under each topic

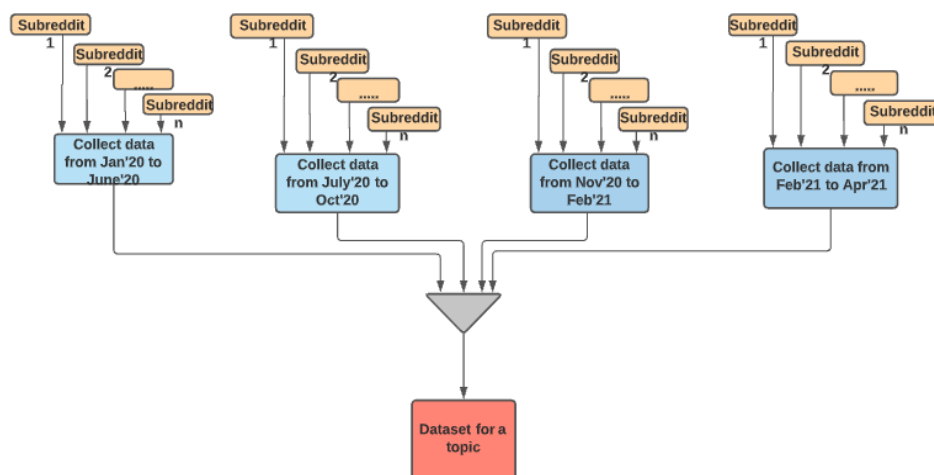


Figure 20: Data collection process

The visual representation of the data gathering approach for each topic can be seen in figure 20. The Reddit test data was cleaned and pre-processed using the techniques described in Section 4.1.1. Techniques such as HTML decoding, removing URLs, contradiction mapping, tokenization, and removing stop words have been implemented for cleaning this dataset.

V. PROPOSED MODELS

5.1 Proposed Model-A

The idea is to train a CNN-LSTM model on the Sentiment140 dataset by combining the word embeddings from both the word2vec models; CBOW and Skip-Gram. The word vector representation of each word can be obtained through a word2vec model. CBOW and the Skip-Gram models will produce word embeddings of 100-dimensional vectors each. Concatenating these would produce 200 features on the given Sentiment140 dataset. Tokenizer will tokenize each tweet into words. Word-based tokens have been used in this project.

Each sequence is composed of a set of tokens, and the longest possible sequence length observed for tweets in the dataset was 40. For the word2vec models, the maximum length was set to 45, and this ensures that any critical information is not lost. If any sentence is longer than the full length, the extra words are dropped. The sequences are padded by adding zeros at the end of each sequence to ensure the samples are all the same sizes.

For this dataset, the possible vocabulary size is over 250,000. This vocabulary size is large and can take a long time to train the data. The size of vocabulary can be limited to 100,000 by choosing the most frequently occurring words. Once tokenized, sequences are built by leveraging the 'text_to_sequences' method from Tokenizer. The model has been trained with CBOW and Skip-Gram. These features are saved to the disk and can be loaded into the model.

As seen in figure 21, an embedding matrix of 200-dimensions can be built by concatenating both the CBOW and skip-gram word embeddings. The embedding layer can take the sequential input representation of data, which is done by passing the embedding matrix as weights.

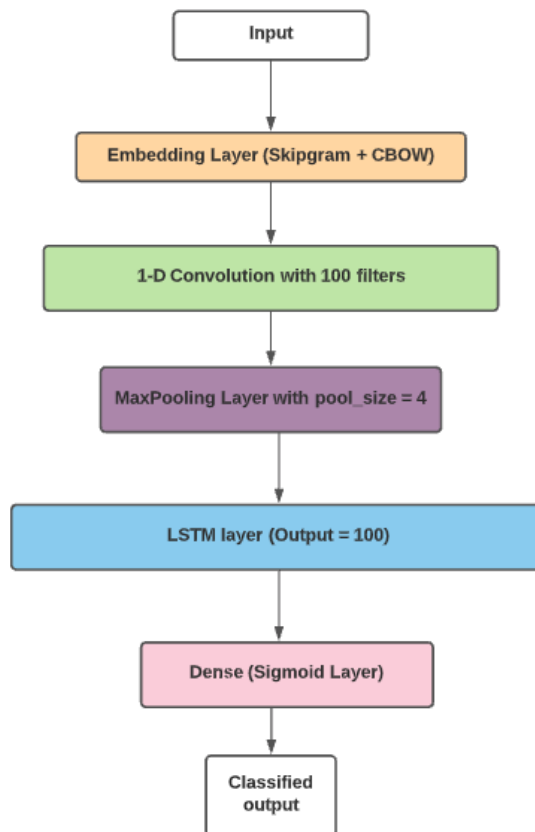


Figure 21: CNN-LSTM model with CBOW + Skip-gram features

Over this, the convolution operation is applied with 100 filters and kernel size 2. Later max-pooling function is used to down-sample the output from the convolution layer. This is followed by the LSTM layer, with 100 memory units. This is then followed by a dense output layer with a sigmoid activation function to classify the text positively or negatively.

5.2 Proposed Model-B

Figure 22 represents the model with parallelly arranged CNNs. The idea is to combine different n-gram based assumptions to make the model more robust. The n-gram features of the input are fed into each convolutional layer. This is followed by a max-pooling layer that down-samples the features. The intermediate outputs from the three CNNs are then merged and fed to the LSTM layer. The following sigmoid layer will return values between 0 to 1, which helps classify the text as positive (≥ 0.5) and negative (< 0.5).

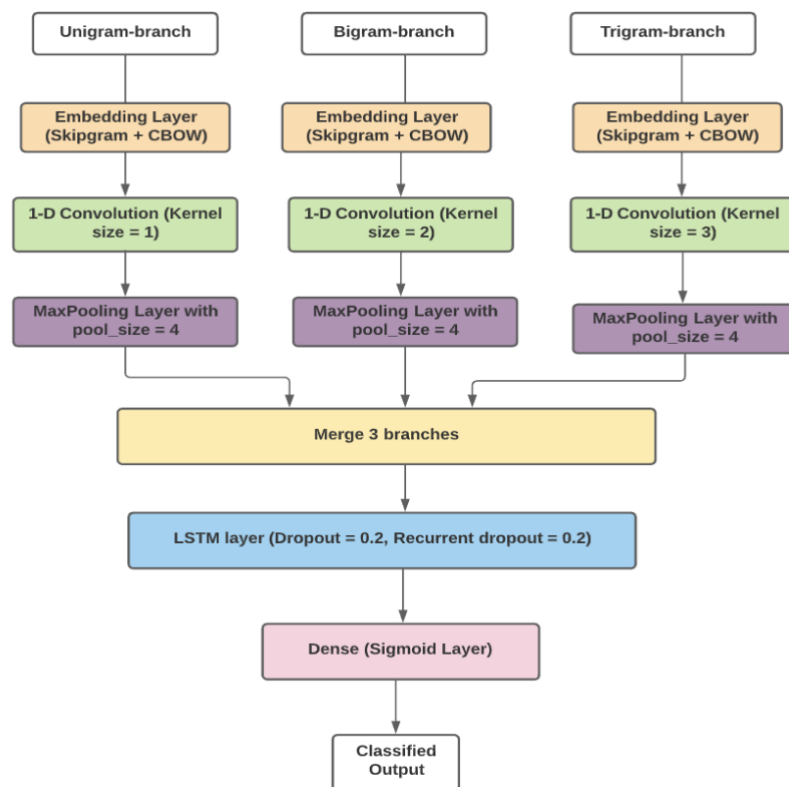


Figure 22: CNN-LSTM model with three parallelly arranged CNNs

VI. EXPERIMENTS AND MODEL RESULTS

6.1 Experimental Environment

The programming language used for this project is Python 3.7. All the experiments were conducted on Google Colab Pro, a Jupyter notebook environment that runs in the cloud. NVIDIA Tesla K80 T4 GPU and TPUv2 with 24 GB RAM have been used to train deep-learning models that are computationally heavy.

6.2 Experiments with stop words

Three different experiments were performed using:

- (i) including stop words
- (ii) excluding custom-built list of stop words
- (iii) excluding combining words from stop words library and custom-built list of stop-words

Baseline machine learning models like Naïve Bayes and Support Vector Machines (SVM) were used to evaluate the performance. In figure 23, the graph plotted explains how using stop words can improve the classifier's performance. When the stop words are present in the input data, the model performs the classification with a low accuracy of 76% that can be seen in the violet line. After removing the custom-built list of stop words, the accuracy has improved by 2.2%, as shown in the blue-line plotted. Finally, when the NLTK's stop words and the list of custom-built stop words have been combined and removed from the input data, the accuracy has increased by 3%.

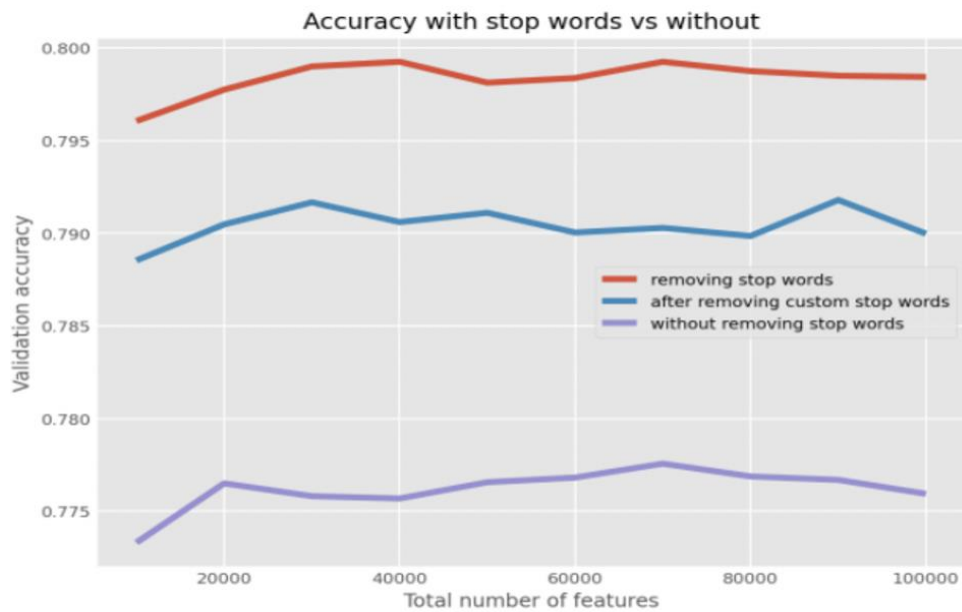


Figure 23: Accuracies plotted against the number of features

6.3 Experiments with N-gram models

Experiments were performed with the input data by combining two n-gram models. The idea is to merge the assumptions made by each n-gram technique. For analyzing each sentence, different n-gram techniques have been used, and their results have been combined.

Method	Best Validation Accuracy	Avg. train time [in seconds]
Unigram	78.84%	47.9
Bigram	78.78%	71.11
Unigram + Bigram	82.13%	91.27
Trigram	79.20%	97.3
Unigram + Trigram	82.27%	162

Table 2. Accuracies obtained with different combinations of n-grams

It can be noted in table 2, for unigram, there isn't much improvement in accuracy even with the increasing number of features. There is a steady growth in validation accuracy for bi-gram & tri-gram as the number of features increases. For a combination of unigram and bi-gram, 82.3% was the highest accuracy achieved at 90,000 features. For a combination of unigram and tri-gram, the highest accuracy achieved was 82.5% at 90,000.

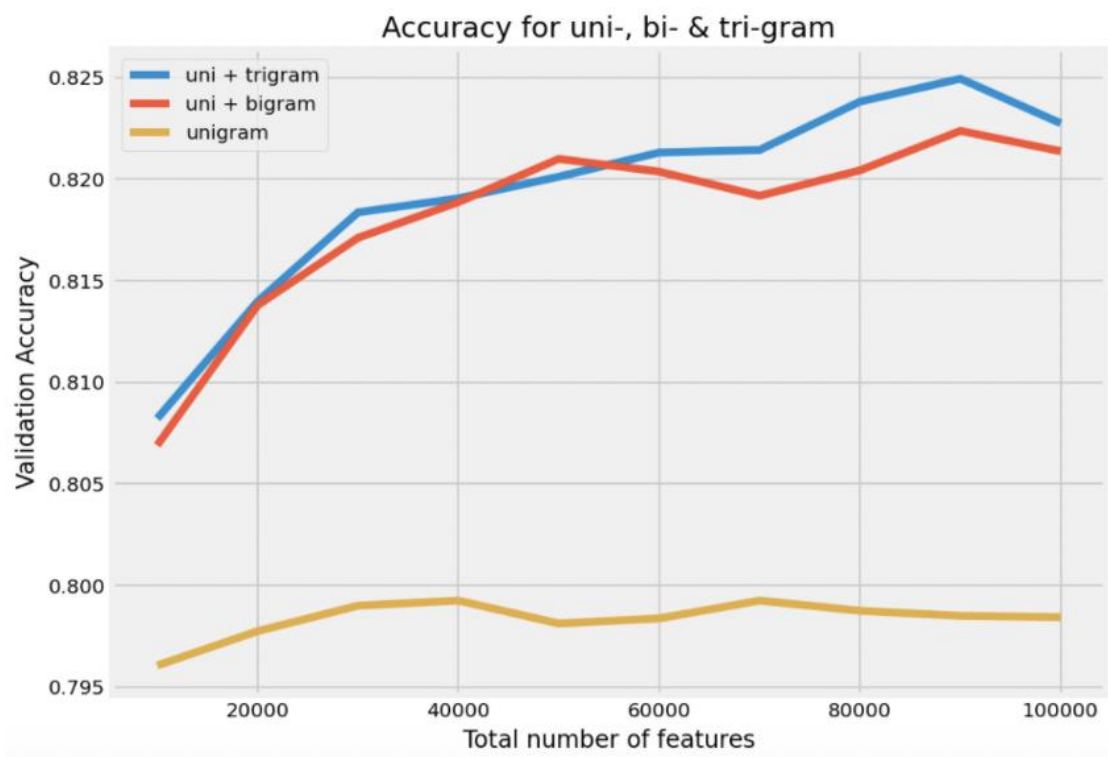


Figure 24: Graph showing increase in accuracy for a combination of 2 n-gram models

As seen in figure 24, the model with a combination of two n-grams as input outperformed a model with single n-gram input. However, a combination of two n-gram model would need a longer train time.

6.4 Experiments with Word Embeddings

Table 3 below shows the performance of different experiments conducted on the Sentiment140 dataset. Both CBOW and Skip-Gram word embedding models were trained for 50 epochs each. These trained word embeddings were then fed to a neural network. For Model 1(ANN), the experiment was performed by feeding generated word vectors into a shallow Artificial Neural Network with 3 layers.

Model Number	Model	Train Accuracy	Test Accuracy	Loss	Precision	Recall	F1-score
Model 1	Artificial Neural Network (3 layers)	83%	79.2%	0.43	0.76	0.78	0.78
Model 2	CNN with CBOW	85.62%	76.3%	0.34	0.733	0.75	0.745
Model 3	CNN with Skip-Gram	84.78%	74.8%	0.44	0.745	0.76	0.76
Model 4	CNN with CBOW + Skip-Gram	84.56%	82.1%	0.45	0.882	0.831	0.82
Model 5	CNN with pre-trained Glove2Vec	77.65%	75.48%	0.482	0.76	0.79	0.77

Table 3. Evaluation of different word embedding techniques with CNN

For models 2 to 4, experiments were performed with CNN. Model 2 (CNN with CBOW) outperformed model 3 (CNN with Skip-Gram). In CBOW training, the focus is to predict the word from the context, whereas in Skip-Gram, the focus is to predict the context. It can be observed that CBOW performed better than Skip-Gram; this could be

because CBOW works better with large training datasets as compared to Skip-Gram, which works well with smaller datasets [4]. Model 4 (CNN with CBOW + Skip-Gram), which has been built by combining both CBOW and Skip-Gram, has proven to perform better than the individual word embedding models. There is an improvement in accuracy by 6% compared to models 2 and 3.

Model 5 (CNN with Glove2Vec) shows the performance was lesser than the word2vec word embedding, and this contradicts the general perception that Glove2Vec is better than Word2Vec. In figure 25, a graph plotted shows the variation in CNN's performance for different combinations of word embeddings as input. The observation made here is that different word embedding architectures work differently, and their performance is dependent on the underlying dataset.

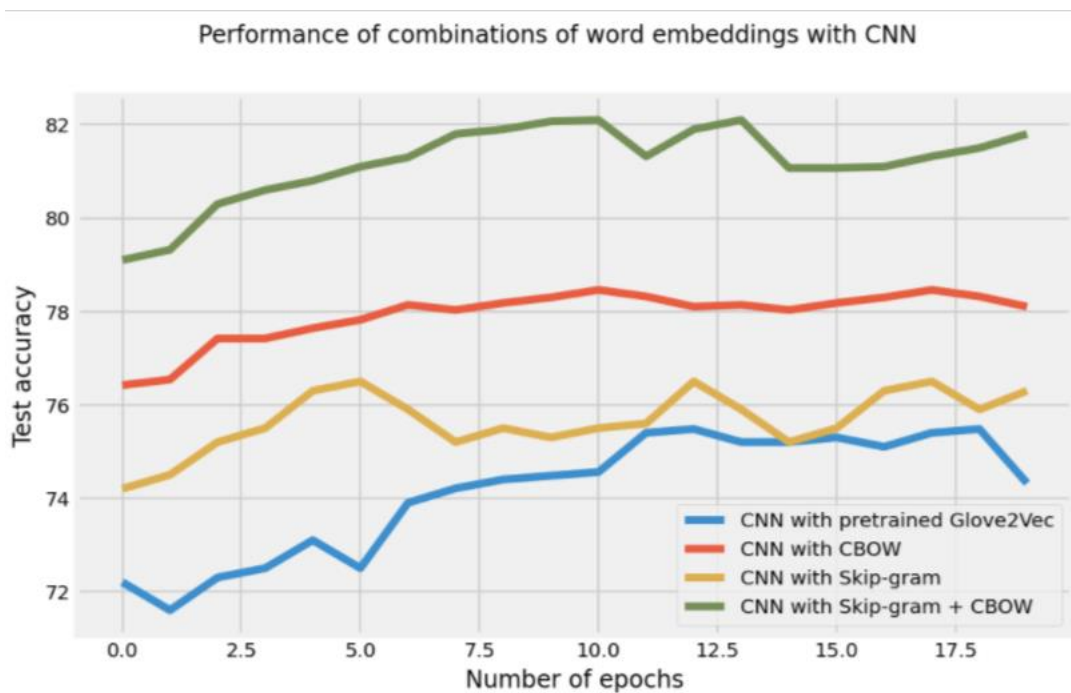


Figure 25: Performance of different word embeddings with CNN

6.5 Experiments with Proposed Model-A

Model 6 is a hybrid CNN-LSTM model with only unigram as input, and its test accuracy is lower than the accuracy of models fed with word embeddings as input. From table 3 in section 6.4, it can be observed that model 4 (CNN with CBOW + Skip-Gram) with the concatenated features outperformed CNN using a single type of word2vec architecture. Results from model 4 were the motivation behind experimenting by feeding the same input to a hybrid of CNN and LSTM models. Model 7 (CNN-LSTM with CBOW + Skip-Gram) in table 4 has shown an improvement in accuracy by 1.25% compared to model 4.

Model Number	Model	Train Accuracy	Test Accuracy	Loss	Precision	Recall	F1-score
Model 6	CNN – LSTM	89.32%	79.8%	0.22	0.78	0.79	0.78
Model 7	CNN - LSTM with CBOW + Skip-Gram	88.09%	83.26%	0.36	0.8125	0.844	0.8237
Model 8	CNN - LSTM with uni, bi, trigram	89.12%	83.7%	0.12	0.82	0.85	0.841
Model 9	CNN - LSTM with bi, tri, 4-gram	86.1%	81.5%	0.24	0.81	0.805	0.81

Table 4. Evaluation of different CNN-LSTM hybrid models

6.6 Experiments with Proposed Model-B

The graph in figure 24 in section 6.3 proves that a combination of n-gram inputs yielded better accuracy than the individual n-grams. Each n-gram generated from the data set is fed into the convolutional layer. In the embedding layer, the embedded matrix built for model 4 (CNN-LSTM with CBOW + Skip-Gram) is reused. The next layer, which is a max-pooling layer, helps in down-sampling the convoluted features. As shown in the proposed model in figure 22 in section 5.2, this technique is repeated for uni-gram, bi-gram, and tri-gram.

These down-sampled features from all three max-pooling layers are now merged. This is sent as an input to the LSTM layer. The last layer, which is a sigmoid layer, will classify each sentence into binary output, which is positive or negative. For this experiment, there has been an increase in accuracy by about 1.1%, with maximum possible accuracy of 83.7%. When the same experiment was performed by feeding an input of bi, tri & 4-gram sequences, there was no difference in accuracy; instead, model 8 outperformed model 9.

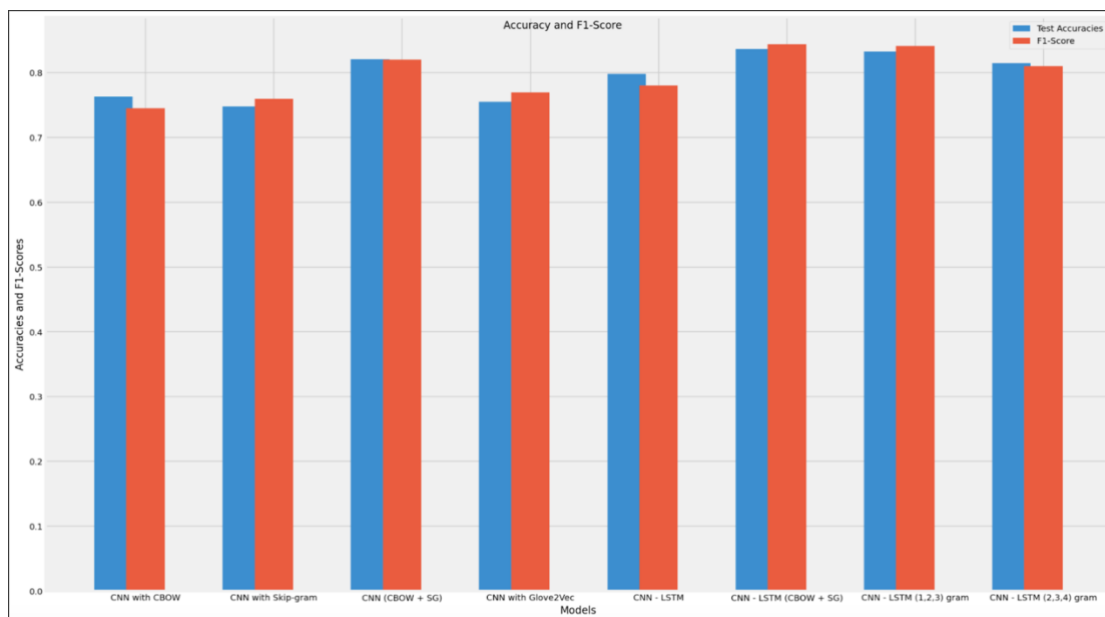


Figure 26: Accuracies and F1-scores of the experimented models

In figure 26, it can be noted that the last three models (models 7, 8, 9) have performed better than the previous models.

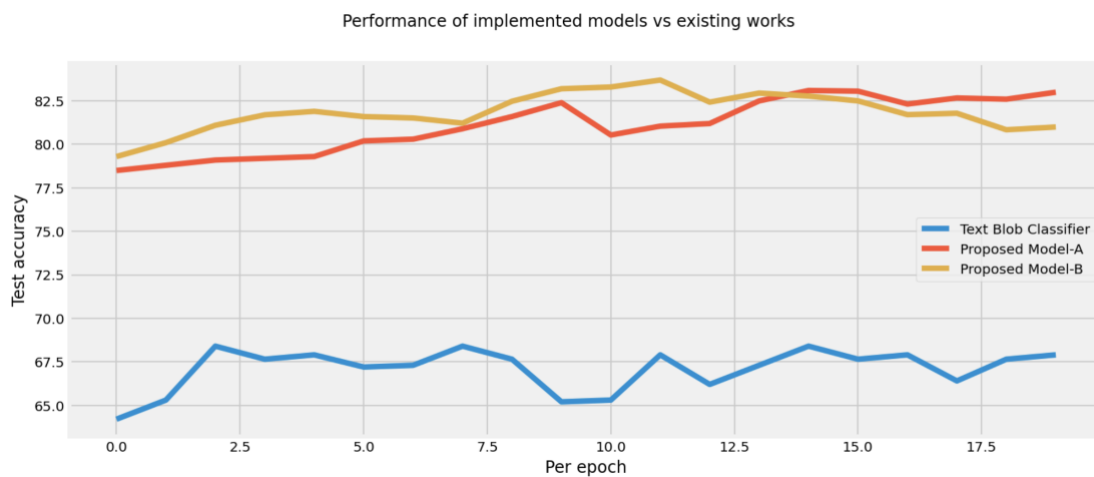


Figure 27: Performance of Proposed Models vs. Textblob classifier

The graph in figure 27 shows the performance comparison of accuracies of Textblob library [25], proposed model-A, and proposed model-B. As seen in this graph, the proposed model-A (red line) and proposed model-B (yellow line) performed better than Textblob's built-in classifier (blue line). Proposed model-B performed slightly better than proposed model-A with an improvement in accuracy of 0.5%. The proposed models (A & B) have shown better performance by approximately 1% than the highest-ranked submissions on Kaggle for Sentiment140 with CNN and LSTM and baseline Bidirectional Encoder Representation of Transformers (BERT) model [17]. Model 8 (proposed model-B), which was trained on the Sentiment140 dataset, was used to perform the analysis of the collected Reddit data.

VII. RESULTS ON TEST DATA

The test data scraped from Reddit was categorized into four topics, Layoffs, Academics, Social Distancing, and Economy. The best-performing model (Proposed model-B) was used to predict the monthly sentiment for each of these topics. It is pertinent to identify the location of the user profiles to understand the correlation between the significant events/news and the changes in users' sentiment. For example, changes made in the social distancing rules in India would not affect the social distancing sentiment of the people residing in the United States.

Unfortunately, the Reddit API does not provide access to the location of the user profiles. Due to this, it is essential to understand the demographics of Reddit users. Since more than 50% of Reddit users come from the United States, as shown in figure 28 [26], the assumption made here is that majority of the comments are made by people residing in the United States. Hence, the subsequent analysis on the four topics is performed based on this assumption.

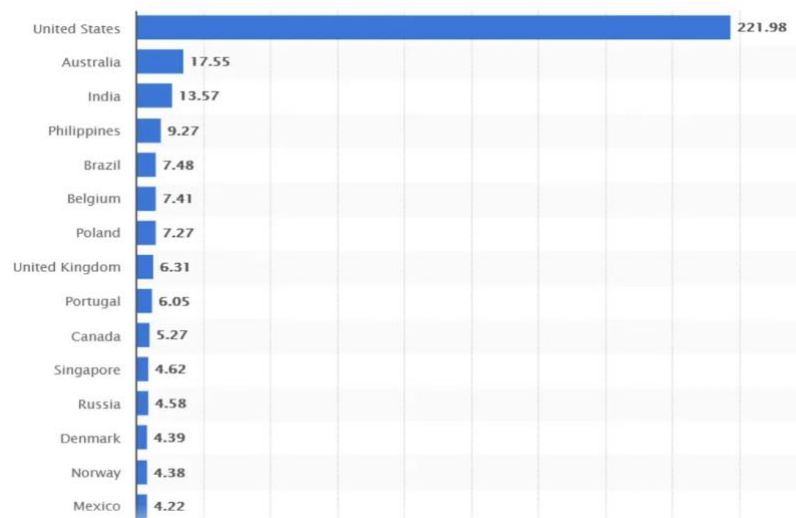


Figure 28: Graph showing the demographics of Reddit users [26]

Month, year	Number of worldwide downloads (App Store + Google Play)
January 2020	2.9 million
February 2020	3 million
March 2020	3.2 million
April 2020	3.3 million
May 2020	3.4 million
June 2020	3.3 million
July 2020	3.8 million
August 2020	4 million
September 2020	4.1 million
October 2020	4.2 million
November 2020	3.9 million
December 2020	4.6 million
January 2021	6.6 million

Figure 29: Increase in number of users through 2020 [26]

Figure 29 shows that there has been a steady increase in the number of downloads in the Reddit application. Another study in [3] indicates an increasing number of users started joining Reddit during the second half of 2020.

The data has been scraped from the start of Jan '20. The number of comments in data scraped for the four topics during Jan '20 and Feb '20 were very few. The analysis has been done for the months starting from Mar '20 as the discussions began to increase steadily from Mar '20. The percentage of negativity was captured for each topic. These graphs were used to understand how the major events in each area have affected the people's sentiment over time.

The formula for percentage of negativity can be defined as,

$$(\text{Number of Negative comments} / \text{Total number of comments}) * 100.$$

Scale	Reaction Type	Percentage of Negativity
+5 to +4	Extremely Positive	0% to 30%
+3 to +2	Moderately Positive	30% to 45%
+1 to -1	Neutral	45% to 55%
-2 to -3	Moderately Negative	55% to 70%
-4 to -5	Extremely Negative	70% to 100%

Table 5: Metrics used for scaling the reactions

These results in percentages are then mapped to a scale of +5 to -5, where +5 means extremely positive, and -5 means extremely negative, as shown in table 5. This mapping helps in better interpretation and visualization of the results. The four graphs in figures 31, 32, 34, and 35, respectively, show how the reaction has varied over a period of thirteen months. Since the number of Covid cases has increased drastically during the months May '20 – Jun '20 and Dec '20 – Jan '21 [27], we will refer to these time periods as the first wave and the second wave of Covid-19.

7.1 Test Results on Topic 1: Economy

Major countries like the United States saw a steep dip in their Gross Domestic Product (GDP) in 2020 [28]. Especially, the U.S. economy has seen a drop of 32.9% in GDP in a single quarter, as shown in figure 30. This was a driving factor for performing analysis on how the public feels about such a drastic change.

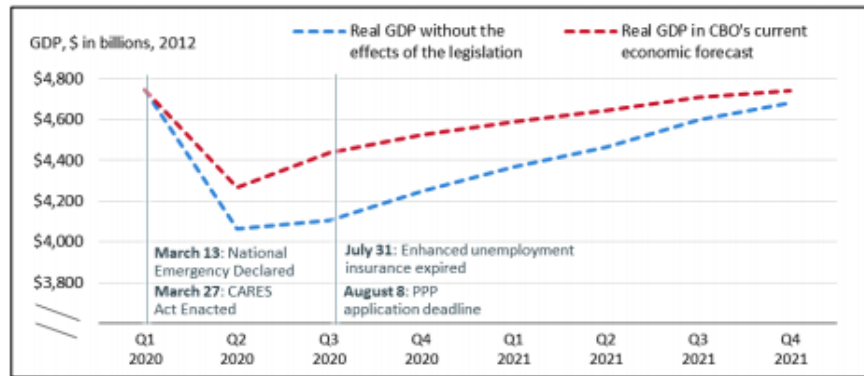


Figure 30: U.S. Economy GDP variation over 2020 [28]

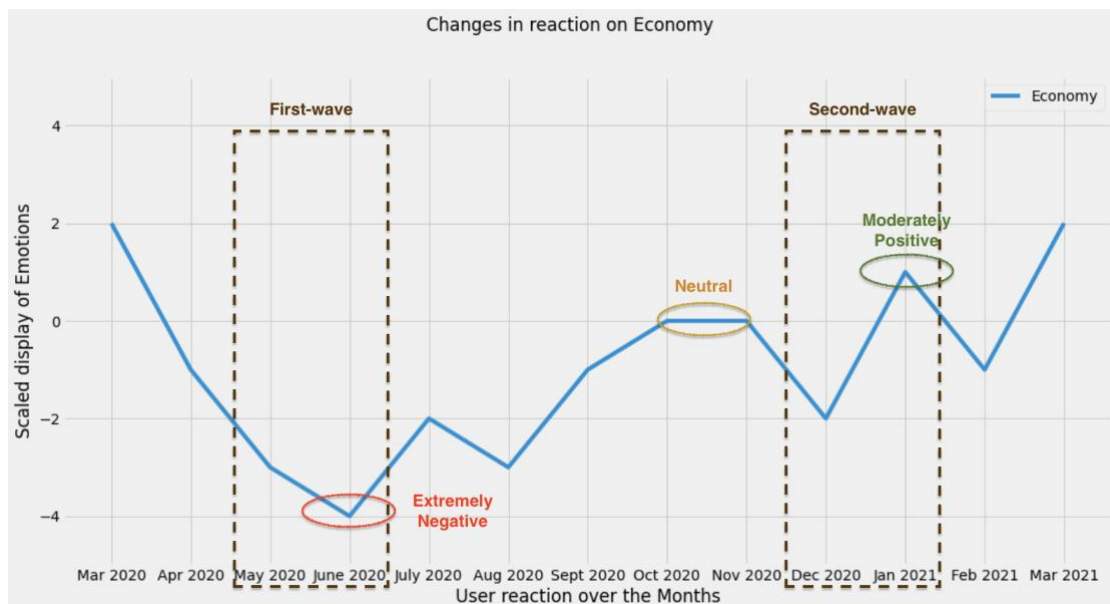


Figure 31: Scaled Reactions over 13-months for Economy

In figure 31, the first box signifies when the first wave of Covid-19 hit during the months of May '20 – Jun '20. It can be observed that there has been extreme negativity in the comments during these months. After Jul '20, there has been a rise in positivity among the comments. For Q3 (Jul '20 - Sep' 20) and Q4 (Oct '20 - Dec' 20) of 2020, and Q1(Jan '21 - Mar' 21) of 2021, the reaction has continued to remain either neutral or moderately positive. It can be observed that there is another increase in negativity

(moderately negative) between Dec '20 to Jan '21, the same time when the second wave of Covid-19 hit. However, it can be observed that the reaction remained neutral or moderately positive. This correlates to announcements by the governments of the United States and other major countries about the release of the vaccine to the public. This analysis aligns well with the current happenings in the U.S., from where a majority of Reddit users come.

7.2 Test Results on Topic 2: Layoffs

For figure 32, between Sep '20 to Nov '20, there has been a steady increase in positivity about layoffs after the first wave of Covid-19. This aligns well with the reduction in the number of unemployment insurance claims made by the public as per [29] and with the trends in the graph in figure 33 [30].

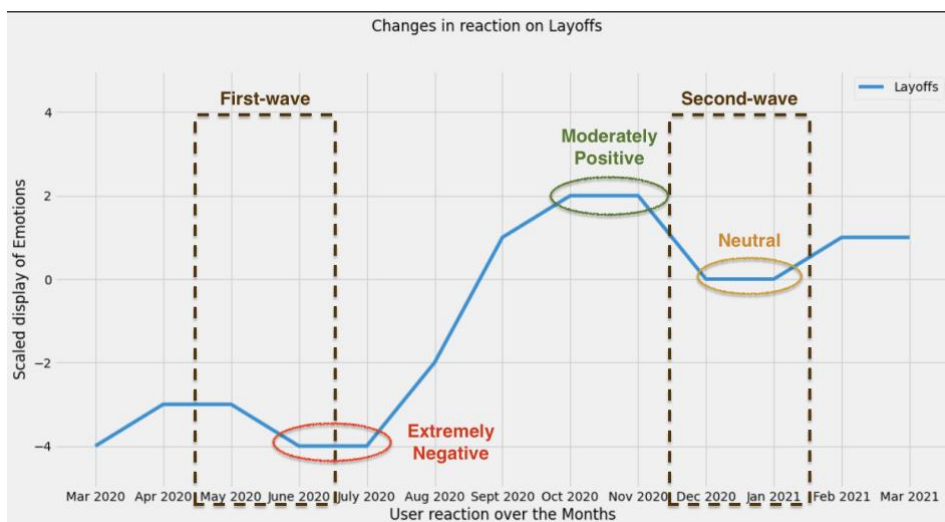


Figure 32: Scaled Reactions over 13-months for Layoffs

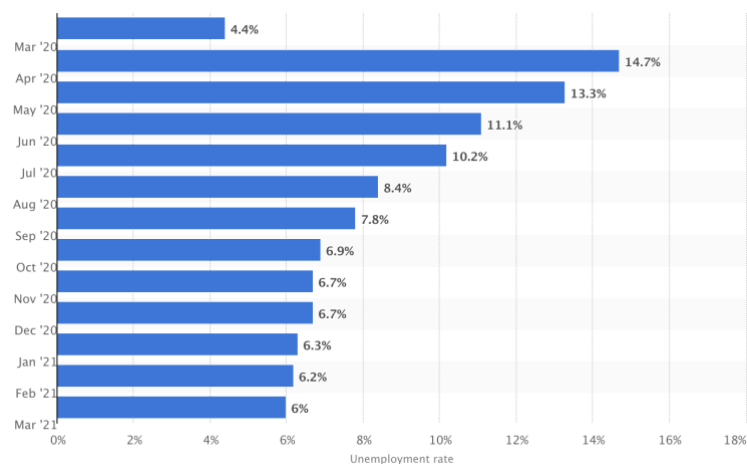


Figure 33: Variation in the unemployment rate from Mar '20 to Mar '21 [30]

As seen in figure 33, although the number of layoffs has decreased in the latter half of 2020, it can be observed in figure 32 that there still has been a significant drop in positive sentiment. The reaction became neutral during the second wave (Dec '20 – Jan' 20). This could be attributed to the tension created from the second wave. The second increase in positive sentiment can be observed in Feb '21, which can be correlated to many businesses and companies opening up their operations again.

7.3 Test Results on Topic 3: Social-Distancing

The third analysis has been performed for social distancing, as shown in figure 34. It can be observed that the reaction was extremely negative for social distancing during the first wave of Covid-19 (May '20 – Jun'20).

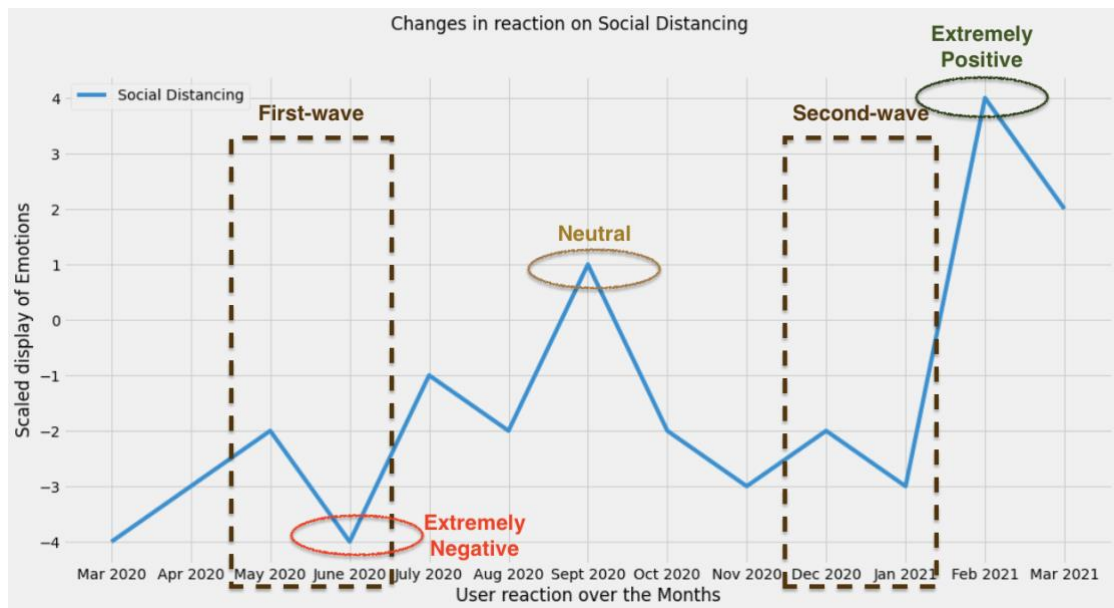


Figure 34: Scaled Reactions over 13-months for Social Distancing

The study suggests that there has been an almost equal number of both positive and negative comments during the months Jul '20 to Nov '20. The large number of negative comments in the months Nov '20 – Jan '20 can be backed by the study in [31], which mentioned that people reported depression and loneliness, which negatively affected people's mental health. After the second wave, there has again been a steady increase in positivity, and the reactions became more positive after Feb '21. This also correlates to an increased number of people getting vaccinated across the world.

7.4 Test Results on Topic 4: Academics

One important decision taken by governments in the countries affected by Covid-19 was to shut down schools and universities as soon as the first wave hit [32].

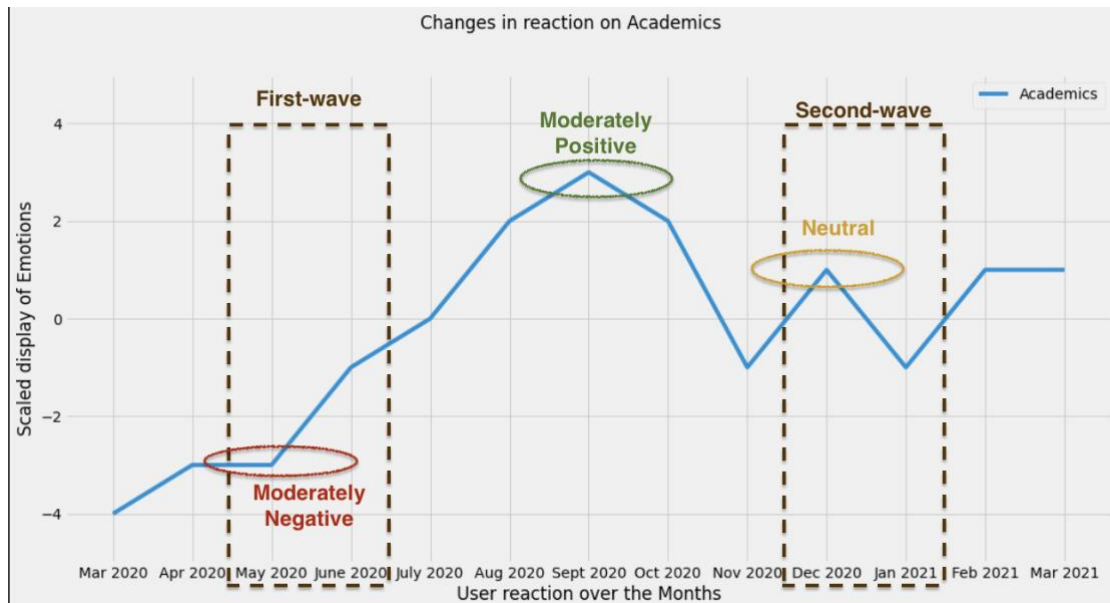


Figure 35: Scaled Reactions over 13-months for Academics

As shown in figure 35, the analysis on academic comments indicates that the sentiment among the users on Reddit has been negative from Mar '20, when the governments decided to shut down schools and universities, to Apr '20, when the schools and universities were shifting their curriculum online. This could be because students now had to rely on the Internet and electronic devices for learning. This would have been a significant burden on parents who could not afford these electronic devices for the students [33].

The positivity, however, started to grow between Jul '20 to Dec '20 among the students regarding online learning. This could be due to more organized planning by schools

and universities. In the later months (Dec '20 – Jan '20), the reactions started to become progressively neutral due to the second wave of Covid-19.

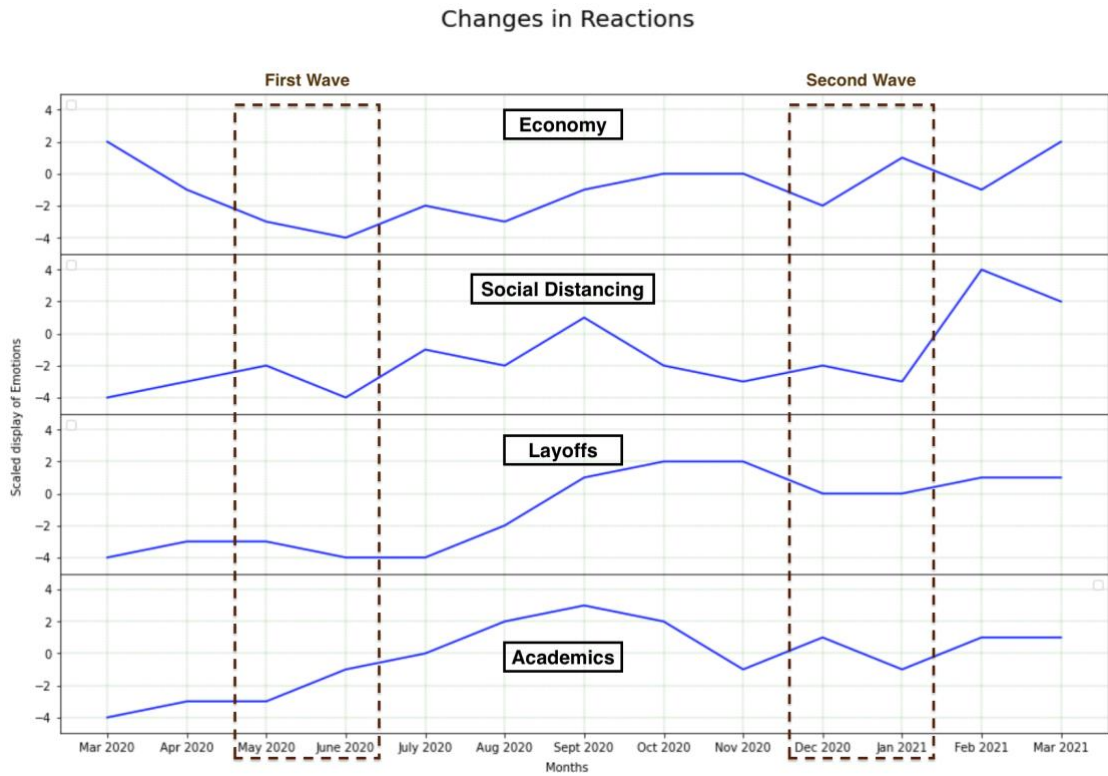


Figure 36: A graph showing a change in sentiment in all the four areas impacted

It can be observed from figure 36 that the reaction has been extremely negative during the first wave of Covid-19, whereas in the second wave, the response was more neutral. It can be concluded that the time between the months Jul '20 to Nov '20 helped people become better prepared for the second wave.

VIII. CONCLUSION

A robust sentiment classification model was built to understand the public's sentiment for social media sites such as Twitter or Facebook. The best-performing model (Proposed model-B), with word-embeddings from Skip-gram and CBOW, fed as input to a hybrid model built with 3 parallel CNNs and an LSTM outperformed the built-in sentiment classifier of the TextBlob library (test accuracy 60.65%), with a test accuracy of 83.7%.

Reddit users' sentiment was analyzed using Proposed model-B on four topics: Economy, Layoffs, Social Distancing, and Academics. Since there is no existing analysis on how the users have been reacting to Covid-19, this research helps understand how the pandemic has affected the mental health of the general population.

Correlations between change in Reddit users' sentiments and the areas that have been severely impacted have been identified. This analysis can be used by the government authorities or enterprise organizations to understand how the decisions taken by them have affected the changes in public sentiment since the start of the pandemic.

IX. FUTURE SCOPE

The analysis from the project would help in making future predictions about how the user sentiment is likely to change in the months to come. Another extension could be to capture user sentiment over other social media platforms like Twitter, Facebook, etc. Combining and analyzing data from multiple social media platforms could provide better insights into understanding user behavior. Experimenting with pre-trained models and applying transfer learning to fine-tune the model might improve the accuracy.

LIST OF REFERENCES

- [1] M. Lenzen et al., "Global socio-economic losses and environmental gains from the Coronavirus pandemic," *PLoS One*, vol. 15, no. 7, p. e0235654, 2020.
- [2] A. D. Kaye et al., "Economic impact of COVID-19 pandemic on healthcare facilities and systems: International perspectives," *Best Pract. Res. Clin. Anaesthesiol.*, 2020.
- [3] "Reddit usage and growth statistics?," *Backlinko.com*, 25-Feb-2021. [Online]. Available: <https://backlinko.com/reddit-users>. [Accessed: 07-May-2021].
- [4] Nagalavi, D., & Hanumanthappa, M. (2016). N-gram word PREDICTION language models to identify the sequence of ARTICLE blocks in ENGLISH e-newspapers. *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*. doi:10.1109/csitss.2016.7779376
- [5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv [cs.CL]*, 2013.
- [6] "Word2Vec tutorial - the Skip-Gram model · Chris McCormick," *Mccormickml.com*. [Online]. Available: <http://mccormickml.com/2016/04/19/> [Accessed: 07-May-2021].
- [7] "Word2Vec Tutorial Part II: The Continuous Bag-of-Words Model", Alex Minnaar, [Online]. Available: <http://mccormickml.com/assets/word2vec/> [Accessed: 07-May-2021]
- [8] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
- [9] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time-series," M. A. Arbib, Ed. MIT Press, 1995.
- [10] "How can filters be found for a CNN?," *Stackexchange.com*. [Online]. Available: <https://stats.stackexchange.com/questions/256056/> [Accessed: 02-May-2021].
- [11] A. Severyn and A. Moschitti, "UNITN: Training deep convolutional neural network for twitter sentiment classification," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015.
- [12] *Researchgate.net*. [Online]. Available: https://www.researchgate.net/figure/fig1_338672883. [Accessed: 05-May-2021].

- [13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [14] "Understanding LSTM Networks," Github.io. [Online]. Available: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Accessed: 07-May-2021].
- [15] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015.
- [16] X. She and D. Zhang, "Text classification based on hybrid CNN-LSTM hybrid model," in *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, 2018.
- [17] Sosa, Pedro M. "Twitter Sentiment Analysis Using Combined LSTM-CNN Models". Konukoi.com. [Online]. Available: <http://konukoi.com/2018/02/19/>. [Accessed: 05-May-2021].
- [18] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN–LSTM model for gold price time-series forecasting," *Neural Computing and Applications*, vol. 32, no. 23, pp. 17351–17360, 2020.
- [19] T. Sahni, C. Chandak, N. R. Chedeti, and M. Singh, "Efficient Twitter Sentiment Classification using Subjective Distant Supervision," *arXiv [cs.SI]*, 2017.
- [20] "Splitting into train, dev and test sets," Stanford.edu. [Online]. Available: <https://cs230.stanford.edu/blog/split/>. [Accessed: 02-May-2021].
- [21] Ofir Press, "Partially shuffling the training data to improve language models," *arXiv [cs.CL]*, 2019.
- [22] J. Reitan, J. Faret, B. Gambäck, and L. Bungum, "Negation scope detection for twitter sentiment analysis," in *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2015
- [23] C. Zhang et al., "Cleaning uncertain data with crowdsourcing - a general model with diverse accuracy rates," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2020.
- [24] J. Baumgartner, S. Zannettou, B. Keegan, M. Squire, and J. Blackburn, "The Pushshift Reddit Dataset," *arXiv [cs.SI]*, 2020.

- [25] D. Hazarika, G. Konwar, S. Deb, and D. J. Bora, "Sentiment analysis on twitter by using TextBlob for natural language processing," in Proceedings of the International Conference on Research in Management & Technovation 2020, 2020.
- [26] "Reddit: traffic by country," Statista.com. [Online]. Available: <https://www.statista.com/statistics/325144/reddit-global-active-user-distribution/>. [Accessed: 04-May-2021].
- [27] Elflein, J. (2021, May 21). *U.S. COVID-19 new cases by day*. Statista. <https://www.statista.com/statistics/1102816/coronavirus-covid19-cases-number-us-americans-by-day/>.
- [28] "COVID-19 savages U.S. economy, 2020 performance worst in 74 years," Reuters, 28-Jan-2021.
- [29] Lvb.com. [Online]. Available: <https://www.lvb.com/another-857000-workers-file-unemployment-layoffs-persist-amid-covid-19-pandemic/>. [Accessed: 07-May-2021].
- [30] J. K. Jackson, Global Economic Effects of COVID-19. Congressional Research Service, 2021. [34] D. Ji, L. Fan, X. Li, and S. Ramakrishna, "Addressing the worldwide shortages of face masks," *BMC Mater*, vol. 2, no. 1, p. 9, 2020.
- [31] K. Sikali, "The dangers of social distancing: How COVID-19 can reshape our social experience," *J. Community Psychol.*, vol. 48, no. 8, pp. 2435–2438, 2020.
- [32] R. M. Tawafak, S. I. Malik, and G. Alfarsi, "Impact of technologies during the COVID-19 pandemic for improving behavioral intention to use E-learning," *Int. J. Inf. Commun. Technol. Educ.*, vol. 17, no. 3, pp. 137–150, 2021.
- [33] C. Aldeman, "During the pandemic, 'lost' education jobs aren't what they seem," Brookings, 02-Mar-2021. [Online]. Available: <https://www.brookings.edu/blog/brown-center-chalkboard/2021/03/02/during-the-pandemic-lost-education-jobs-arent-what-they-seem/>. [Accessed: 07-May-2021].