Report on

Aspect Based Sentiment Analysis

[on Covid 19 Tweets]

AFFILIATION

Corresponding Author: Pranita D
5th Semester
Department of Computer Science and Technology
Dayananda Sagar University
Bangalore, Karnataka
India

Corresponding author: P Sai Alekhya
5th Semester
Department of Computer Science and Technology
Dayananda Sagar University
Bangalore, Karnataka
India

Abstract

The outbreak of the Covid-19 pandemic has flipped livelihoods from all walks of life. This project serves as an attempt to gain insight over people's feelings and opinions affected by the ripples brought on by the onset of the pandemic worldwide. Concepts of Web scraping and Natural Language Processing were studied and implemented to pursue this project in hopes to identify, extract and quantify meaningful data to assess the public opinion in regards to the impact of the Coronavirus pandemic over a period of time (January 2020 to October 2020).

User data pertaining to their emotions in response to the Covid-19 pandemic was extracted from Twitter in a timeline ranging from January 2020 to October 2020 via Python utilising the Twitter API. The dataset was pre-processed and cleaned after which, sentiment analysis was performed on the data yielding positive, negative and neutral sentiments. Thereafter, the data was run through an aspect-based sentiment analysis in order to determine the motive behind the corresponding sentiment. These aspect terms were represented visually through a word cloud for easy comprehension.

The project has yielded a few interesting results. From the dataset that was created which consists of roughly around 21,000 tweets, it was discovered that 47% of the tweets were positive, 18% of the tweets were negative and 35% of the tweets were neutral. To derive the sentiments of the tweets, the polarity score for each tweet was calculated and it was observed that the majority of the tweets had a score ranging from -0.25 to 0.25.

The aspect terms as observed are heavily centered around the terms "Coronavirus" and "Covid 19" and for each sentiment, these terms are surrounded by other features that direct the particular tweet towards the sentiment it has been labelled with.

The main focus of this project was to use a sentiment analysis algorithm on tweets that have been collected from January 2020 to October 2020, so that a detailed analysis can be done regarding how the people have reacted to the covid-19 pandemic situation and do an aspect-based analysis, that is to find out the various reasons associated with the sentiments.

The major goal was achieved successfully and a ground work for further improvements has been established. This gives room for more accurate and detailed research in this regard and it will help in future implementation of this project on a larger scale.

I. Introduction

I.1 Importance

Aspect-based sentiment analysis is a text analysis technique that allocates the given block of text to a particular sentiment (positive, negative or neutral) and then breaks down text into aspects (the topic that is being talked about).

This technique helps researchers and businesses become more user-centric.
It's about listening to the users, understanding their voice, analysing their feedback and learning more about customer experiences, as well as their expectations.

Aspect based sentiment analysis on Covid-19 tweets takes in all the data and structures it in a way that researchers, companies and organisations using this technique are able to interpret the textual data and gain meaningful insights. Doing so will help one gain a deeper understanding and provide valuable results.

I.2 Uses and Applications

Aspect based sentiment analysis in general has quite a few applications:

- Product Feedback
- Customer Support
- It helps business to track how end-user's sentiment changes toward specific features and attributes of a service or product
- Analysis of Online reviews on restaurants, small scale firms and so on.

Aspect based sentiment analysis on Covid-19 tweets has vast application in a variety of fields:

- Helps to analyse public opinion
- Conduct researches
- Helps in monitoring the related social media interactions
- Civic body authorities can use it to take preventative and corrective measures
- Helps to detect changes and manifestations of human moods in a given period of the coronavirus

I.3 Goal of the Project

The aim of this project is to do an Aspect based sentiment analysis on the tweets pertaining to Covid-19 from January 2020 to October 2020.

The textual data from the tweets are extracted, pre-processed, cleaned and passed through a sentiment analysis model which assigns each tweet that labels it as either positive, negative or neutral. After this the tweets are separated into three categories based on their sentiment and then passed through a natural language processing pipeline (Stanford NLP) which extracts the features or the aspect terms that are responsible for the sentiment.

I.4 Overview

A complete outline of the rest of the report is provided below:

- Section II : Problem Statement - Vision, Issue Statement, Methods used
- Section III : Objective
- Section IV : Methodology - Flow chart, Pseudo Code, Code Snippets
- Section V : Software and Hardware Requirements
- Section VI : Results
- Section VII : Conclusion
- Section VIII : References

II. Problem Statement

Vision:

To create a simple machine learning model that does Aspect Based Sentiment Analysis on Covid-19 tweets with the help of Natural Language Processing that also includes processing the textual data. This is done in order to assign every tweet with a sentiment that is classifying the tweet as either positive, negative or neutral and extracting the aspect terms or the features of the tweet using a pipeline.

Issue Statement:

The Covid-19 pandemic has many government authorities, scientists and researchers looking for various reasons for the spread of the disease and how to curb this spread. The best way to help these people is by analysing how the people have reacted to the various preventative and predictive measures that have been taken, what are the factors that caused a spike in the number of cases, what methods were effective and what were the reasons for the various sentiments across a given population.

The Aspect based sentiment analysis model will act as a starting point for deriving appropriate solutions to the above-mentioned problems.

Method:

To develop the Aspect Based Sentiment Analysis model, first the tweets from a given period (in this case from January 2020 to October 2020) are extracted and stored in a csv file along with a few necessary attributes that are relevant to the tweet.

The tweet is then pre-processed and cleaned so that the result is more refined and beneficial.

After the pre-processing is done, the polarity score of the tweet is calculated and it is assigned a sentiment label (positive, negative or neutral).

The tweets are separated into 3 different groups based on the sentiment and then each group is passed through a Natural Language Processing Pipeline which gives the aspect terms associated with each group. These aspect terms are visualised according to give a more riveting output.

III. Objective

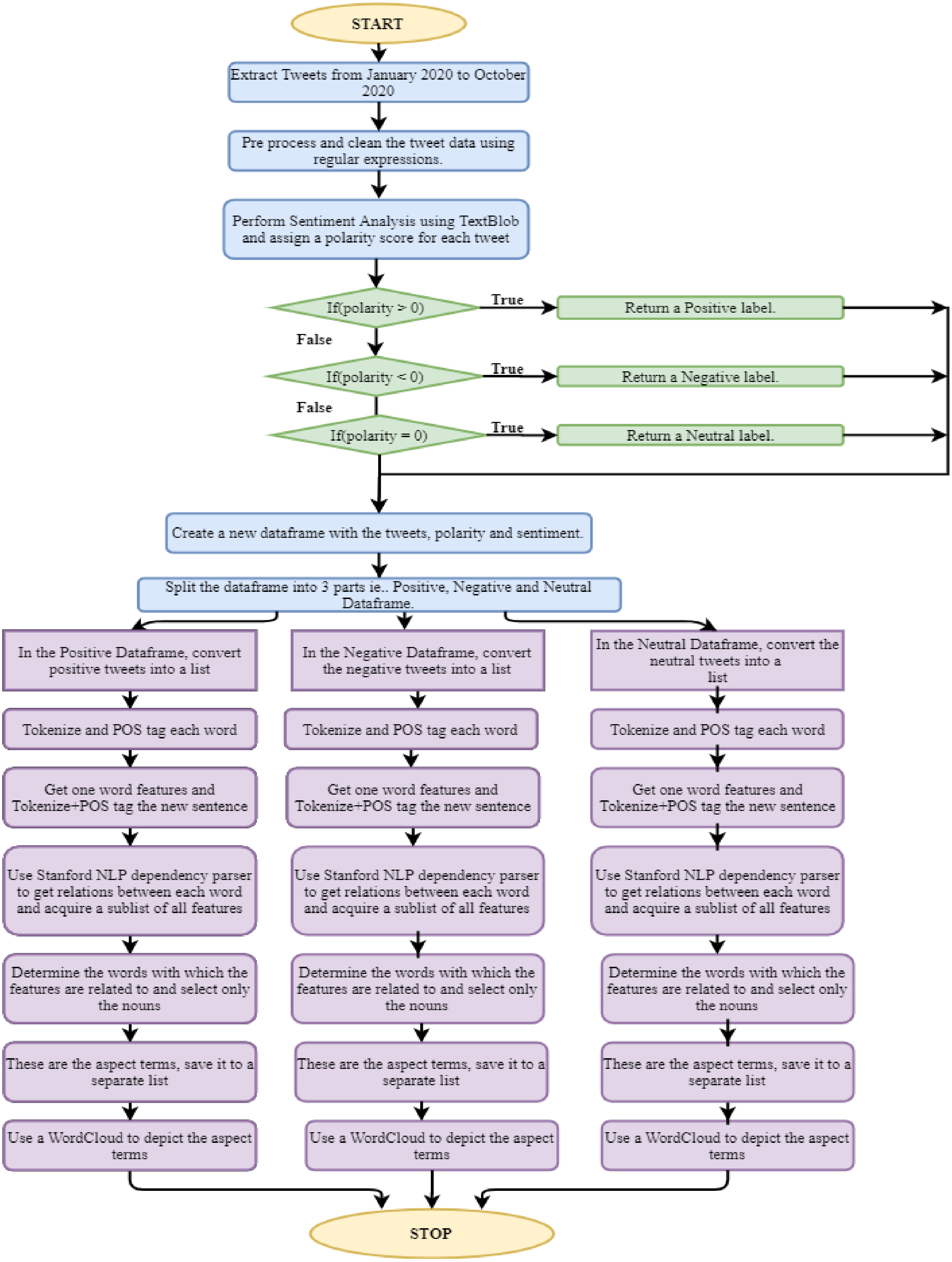
The objective of this project is to provide an initial factor which will help establish various solutions that are required to curb the spread of Covid 19 virus before its disastrous effects become irreversible. To do so, analysis of the data is crucial for scientists, healthcare and government officials so that they can take the required steps to put an end to this pandemic. Our project which is Aspect based sentiment analysis on Covid-19 tweets will help give a more helpful insight on how people have been reacting in a given timeline.

The primary outcome is to provide a list of the aspect terms that can be used to analyse the changes in the perception and mood prevailing across the worldwide population.

The deliverable of this project is putting up the code and the necessary information associated with it on an open source platform, namely GitHub where researchers, students and anyone interested in this field can use the work that has been done to develop or utilise this model for their own benefits and benefits of the community at large.

IV. Methodology

IV.1 Flow Chart


START
Extract Tweets from January 2020 to October 2020
Pre process and clean the tweet data using regular expressions.
Perform Sentiment Analysis using TextBlob and assign a polarity score for each tweet
If(polarity > 0)
True
Return a Positive label.
False
If(polarity < 0)
True
Return a Negative label.
False
If(polarity = 0)
True
Return a Neutral label.
Create a new dataframe with the tweets, polarity and sentiment.
Split the dataframe into 3 parts ie.. Positive, Negative and Neutral Dataframe.
In the Positive Dataframe, convert positive tweets into a list
In the Negative Dataframe, convert the negative tweets into a list
In the Neutral Dataframe, convert the neutral tweets into a list
Tokenize and POS tag each word
Get one word features and Tokenize+POS tag the new sentence
Use Stanford NLP dependency parser to get relations between each word and acquire a sublist of all features
Determine the words with which the features are related to and select only the nouns
These are the aspect terms, save it to a separate list
Use a WordCloud to depict the aspect terms
STOP

IV.2 Algorithm

1. Install and download all the necessary packages and modules required.
2. Initialise the keys obtained from Twitter API and verify the same.
3. Upload the extracted dataset.
4. Remove all the duplicate tweets from the dataset.
5. Convert the data into a pandas dataframe.
6. Preprocessing the Tweets:
    - 6.1. With the help of regular expressions, remove the URLs.
    - 6.2. Convert all the text to lowercase.
    - 6.3. With the help of regular expressions, remove the usernames.
    - 6.4. With the help of regular expressions, remove all the hashtags.
    - 6.5. Apply this preprocessing function to all of the tweets.

7. Doing Sentiment Analysis:
    - 7.1. Using TextBlob (used for processing textual data), find the polarity score of each tweet.
    - 7.2. If the polarity score < 0, assign it to a negative label.
    - 7.3. If the polarity score > 0, assign it to a positive label.
    - 7.4. If the polarity score = 0, assign it to a neutral label.
    - 7.5. Create a dataframe with the tweet, polarity and sentiment.

8. Split the dataframe into 3 parts according to the sentiment label.
9. For each dataframe (positive, negative and neutral):
    - 9.1. Convert the tweet data column into a list.
    - 9.2. Tokenize the list and assign a POS (Part Of Speech) tag to each word.
    - 9.3. Handle multiple word features by combining it to a one word feature with the help of the POS tags.
    - 9.4. Tokenize and POS tag the new sentence.
    - 9.5. Using Stanford NLP Dependency Parser, get the relations between each word.
    - 9.6. Select the sublists that contain the features.
    - 9.7. Using the previous output lists determine which of the words the features are related to.
    - 9.8. Select only the feature nouns from the cluster of features which represents the aspect terms.
    - 9.9. Extract all the aspect terms into a separate list.
    - 9.10. Plot a WordCloud that represents the aspect terms.
10. The aspect based sentiment analysis model is complete.

IV.3 Code Snippets :

IV.3.1 Function for Dataset Creation

```
tweets=[]
for i, tweet enumerate (snscrape.modules.twitter.TwitterSearchScraper (
"#coronavirus+#pandemic+#covid19 lang:en since: Date_as_required
until: Date_as_required").get_items()):

  if tweet not in tweets:
    tweets.append(tweet)
  if i>=1500:
    break

df=pd.DataFrame(tweets)

from google.colab import drive
drive.mount('/drive')

df1.to_csv("/drive/My Drive/Mini_Project/dataset.csv", mode="a")
```

IV.3.2 Function for Preprocessing Tweets

```
def preprocess_tweet(text):

        cleaned= " ".join(re.sub("([^0-9A-Za-z \t])|(\w+:\/\/\S+)", "",
        text).split())

         cleaned= "".join(cleaned.lower())

         cleaned= "".join(re.sub('@[^\s]+', '', cleaned))

         cleaned= "".join(re.sub('#([^\s]+)', '', cleaned))

         word_tokens = word_tokenize(cleaned)
         cleaned= " ".join(word for word in word_tokens if word not in
        stopwords.words('english'))
         return cleaned

map_object = map(preprocess_tweet, tweets)
cleaned_tweets = list(map_object)
```

IV.3.3 Function for Sentiment Analysis

```
sentiment_objects = [TextBlob(tweet) for tweet in cleaned_tweets]

sentiment_values = [[ str(tweet), tweet.sentiment.polarity] for tweet in sentiment_objects]

sentiment_df=pd.DataFrame(sentiment_values,columns=["tweet","polarity"])

def sentiment(polarity):
            if (polarity < 0):
              return 'negative'
            elif (polarity == 0):
              return 'neutral'
            else:
              return 'positive'

sentiment_df['sentiment'] = sentiment_df['polarity'].apply(sentiment)
```

IV.3.4 Functions for Aspect term Extraction

```
new_list=[]
for line in x_list:
    txt_list = nltk.word_tokenize(line)
    taggedList = nltk.pos_tag(txt_list)
    new_list.append(taggedList)
new_list

newwordList = []
flag = 0
for j in new_list:
  for i in range(0,len(j)-1):
    if (new_list[i][1]=="NN" and new_list[i+1][1]=="NN"):
        newwordList.append(new_list[i][0]+new_list[i+1][0])
        flag=1
    else:
        if (flag==1):

            flag=0
            continue
        newwordList.append(new_list[i][0])
        if (i==len(new_list)-2):
            newwordList.append(new_list[i+1][0])
finaltxt = '\n '.join(' '.join(word) for word in newwordList)
nlp = stanfordnlp.Pipeline()
doc = nlp(finaltxt)
dep_node = []
```

```
for dep_edge in doc.sentences[0].dependencies:
    dep_node.append([dep_edge[2].text, dep_edge[0].index,
dep_edge[1]])
for i in range(0, len(dep_node)):
    if (int(dep_node[i][1]) != 0):
        dep_node[i][1] = newwordList[(int(dep_node[i][1]) - 1)]
print(dep_node)


featureList = []
totalfeatureList=[]
categories = []
categoriesList=[]
for j in new_list:
  for i in j:
    if(i[1]=='JJ' or i[1]=='NN' or i[1]=='JJR' or i[1]=='NNS' or
i[1]=='RB'):
        featureList.append(list(i))
        totalfeatureList.append(list(i)) # This list will store all the features
for every sentence
        categories.append(i[0])
print(featureList)
print(categories)


fcluster = []
for i in featureList:
    filist = []
    for j in dep_node:
        if((j[0]==i[0] or j[1]==i[0]) and (j[2] in ["nsubj", "acl:relcl", "obj",
"dobj", "agent", "advmod", "amod", "neg", "prep_of", "acomp",
"xcomp", "compound"])):
            if(j[0]==i[0]):
                filist.append(j[1])
            else:
                filist.append(j[0])
    fcluster.append([i[0], filist])
print(fcluster)


finalcluster = []
dic = {}
for i in featureList:
    dic[i[0]] = i[1]
for i in fcluster:
    if(dic[i[0]]=="NN"):
        finalcluster.append(i)
print(finalcluster)
aspect_terms=list(zip(*finalcluster))[0]
aspect_terms
```

V. Software and Hardware Requirement

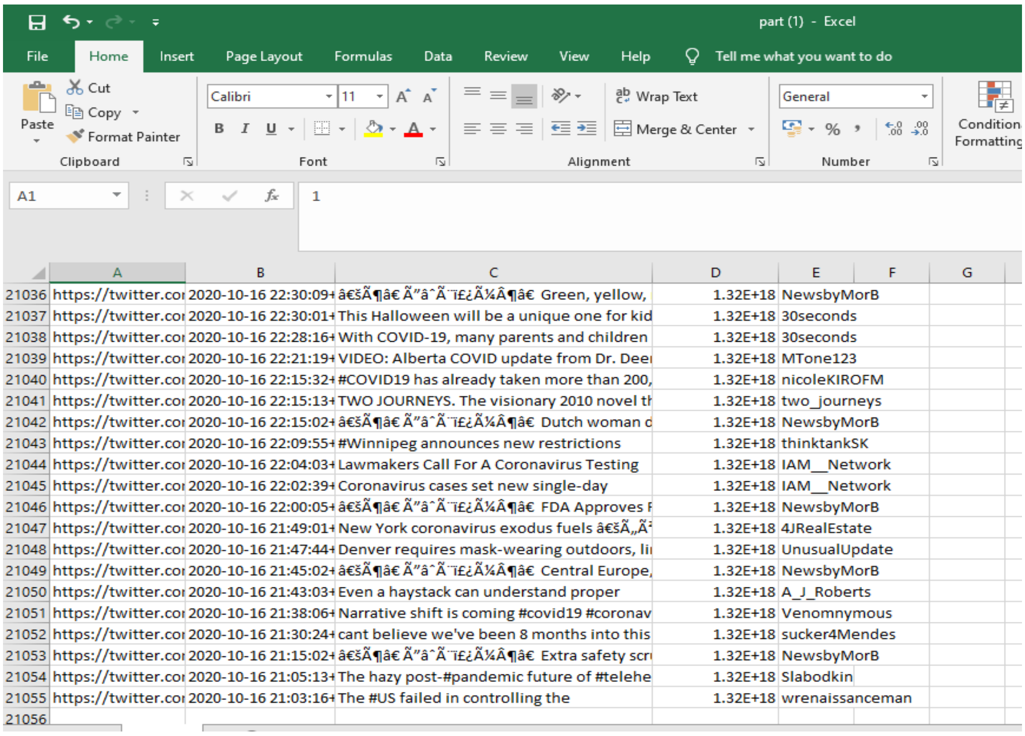
V.1Programming Language and Compiler

- The Programming language used for the project is Python 3
- The code was developed via browser using Google Colab - a free online cloud based Jupyter Notebook environment which provides free access to computer resources including GPUs and enables a collaborative work environment.

V.2Hardware Requirement

- Intel Core i5 or higher
- 8GB RAM or higher
- Windows 10/MacOS/Linux/UNIX

VI. Results



	A	B	C	D	E	F	G
21036	https://twitter.coı	2020-10-16 22:30:09+	â€šÃ¶â€ Ã”âˆ Ã¨ï£¿Ã¼Â¶â€ Green, yellow,	1.32E+18	NewsbyMorB		
21037	https://twitter.coı	2020-10-16 22:30:01+	This Halloween will be a unique one for kid	1.32E+18	30seconds		
21038	https://twitter.coı	2020-10-16 22:28:16+	With COVID-19, many parents and children	1.32E+18	30seconds		
21039	https://twitter.coı	2020-10-16 22:21:19+	VIDEO: Alberta COVID update from Dr. Deeı	1.32E+18	MTone123		
21040	https://twitter.coı	2020-10-16 22:15:32+	#COVID19 has already taken more than 200,	1.32E+18	nicoleKIROFM		
21041	https://twitter.coı	2020-10-16 22:15:13+	TWO JOURNEYS. The visionary 2010 novel tł	1.32E+18	two_journeys		
21042	https://twitter.coı	2020-10-16 22:15:02+	â€šÃ¶â€ Ã”âˆ Ã¨ï£¿Ã¼Â¶â€ Dutch woman c	1.32E+18	NewsbyMorB		
21043	https://twitter.coı	2020-10-16 22:09:55+	#Winnipeg announces new restrictions	1.32E+18	thinktankSK		
21044	https://twitter.coı	2020-10-16 22:04:03+	Lawmakers Call For A Coronavirus Testing	1.32E+18	IAM__Network		
21045	https://twitter.coı	2020-10-16 22:02:39+	Coronavirus cases set new single-day	1.32E+18	IAM__Network		
21046	https://twitter.coı	2020-10-16 22:00:05+	â€šÃ¶â€ Ã”âˆ Ã¨ï£¿Ã¼Â¶â€ FDA Approves F	1.32E+18	NewsbyMorB		
21047	https://twitter.coı	2020-10-16 21:49:01+	New York coronavirus exodus fuels â€šÃ„Ã	1.32E+18	4JRealEstate		
21048	https://twitter.coı	2020-10-16 21:47:44+	Denver requires mask-wearing outdoors, liı	1.32E+18	UnusualUpdate		
21049	https://twitter.coı	2020-10-16 21:45:02+	â€šÃ¶â€ Ã”âˆ Ã¨ï£¿Ã¼Â¶â€ Central Europe,	1.32E+18	NewsbyMorB		
21050	https://twitter.coı	2020-10-16 21:43:03+	Even a haystack can understand proper	1.32E+18	A_J_Roberts		
21051	https://twitter.coı	2020-10-16 21:38:06+	Narrative shift is coming #covid19 #coronav	1.32E+18	Venomnymous		
21052	https://twitter.coı	2020-10-16 21:30:24+	cant believe we've been 8 months into this	1.32E+18	sucker4Mendes		
21053	https://twitter.coı	2020-10-16 21:15:02+	â€šÃ¶â€ Ã”âˆ Ã¨ï£¿Ã¼Â¶â€ Extra safety scrı	1.32E+18	NewsbyMorB		
21054	https://twitter.coı	2020-10-16 21:05:13+	The hazy post-#pandemic future of #telehe	1.32E+18	Slabodkin		
21055	https://twitter.coı	2020-10-16 21:03:16+	The #US failed in controlling the	1.32E+18	wrenaissanceman		
21056							

Fig 6.1. Screenshot capturing the dataset of tweets obtained in csv file.

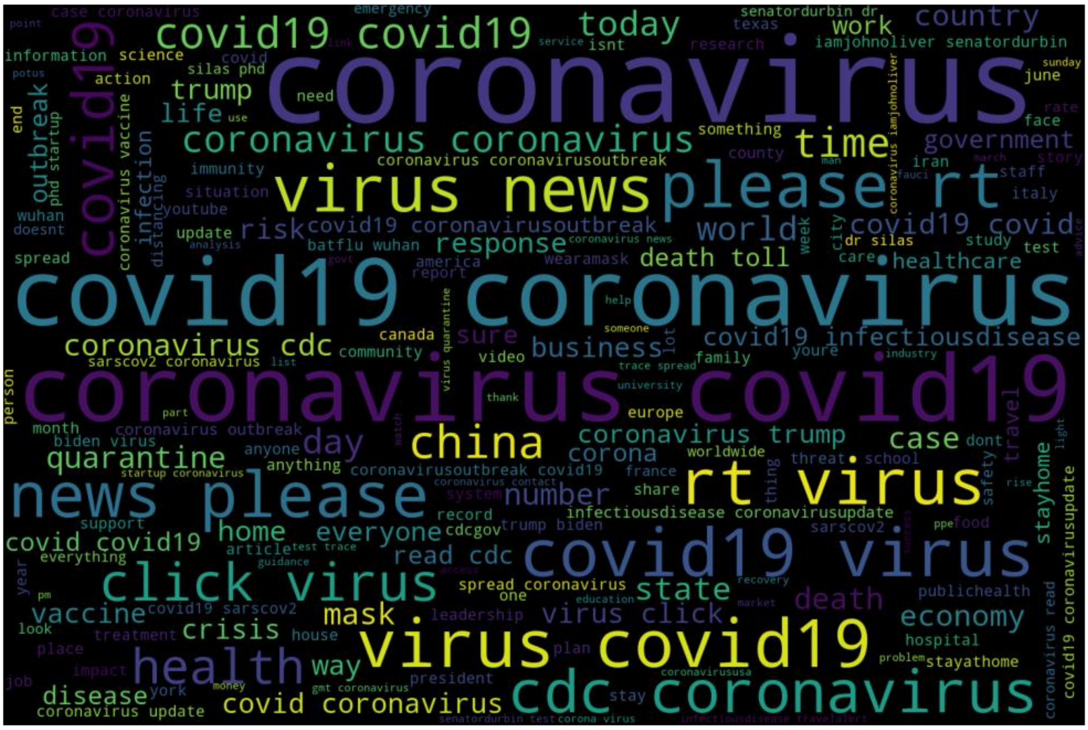


Fig 6.2. Wordcloud depicting the positive aspect.

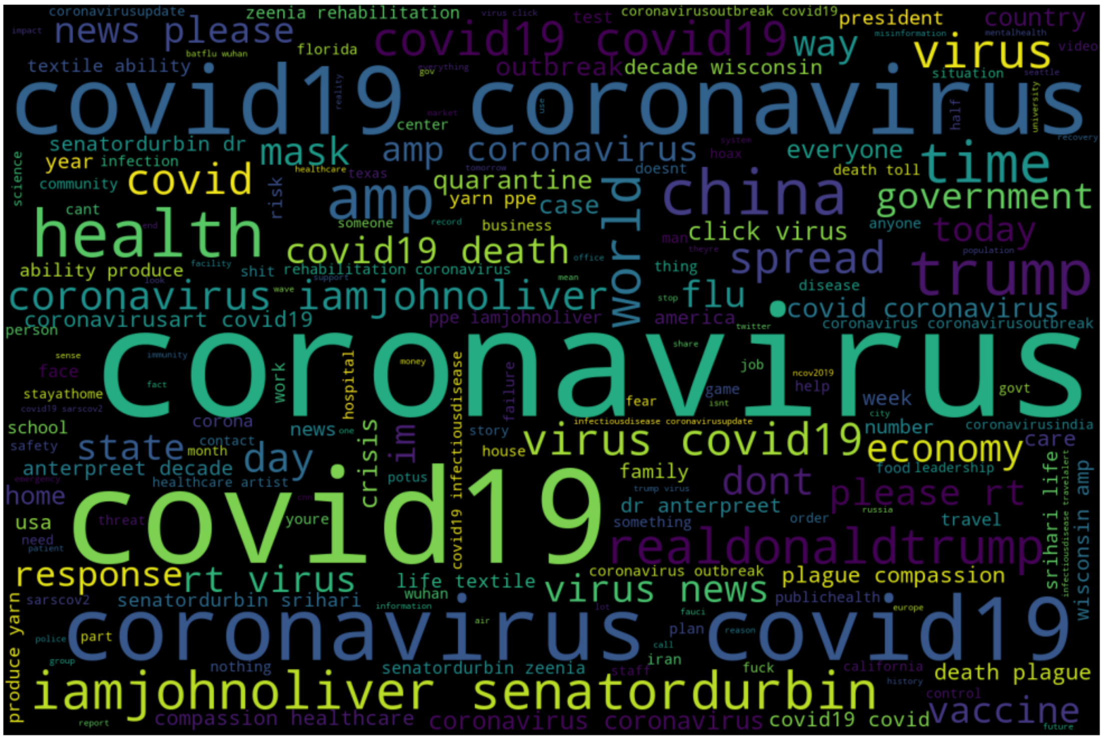


Fig 6.3. Wordcloud depicting the negative aspect.

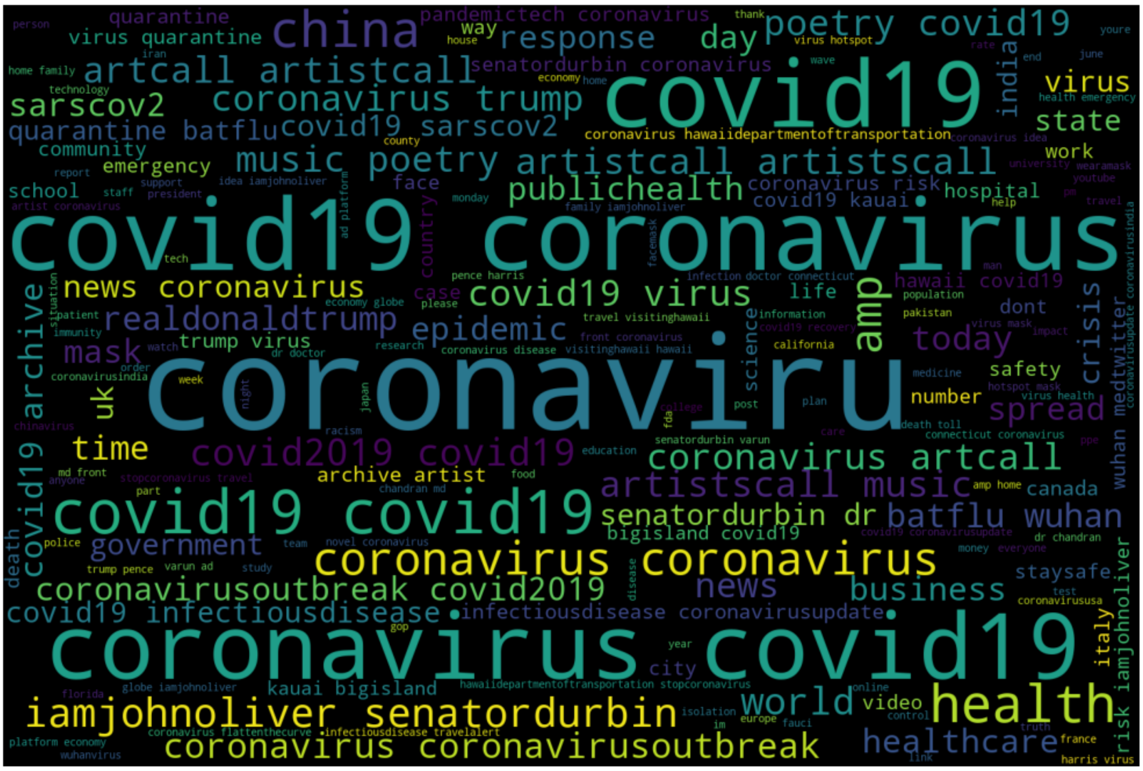


Fig 6.4. Wordcloud depicting the neutral aspect.

VII. Conclusion

- By virtue of this project, a fundamental grasp over web scraping, data cleaning and natural language processing was obtained and applied for implementation over a pressing real time scenario - the Covid-19 pandemic; to extract, identify and visualise the sentiment of people from all over the globe using sentiment analysis.
  The sentiment has been categorised as - positive, negative and neutral on which an aspect-based sentiment analysis technique was applied to deduce the reason behind the sentiment.
  This output can potentially be of help to derive feasible solutions for issues related to Covid-19 or discern the insights for further analytics and research in multiple domains.

  Ample groundwork, exploration and study was carried out over topics of interest such as - AI/ML, Python, Twitter API, Web scraping tools (snscrape), NLP, NLTK to help materialise the idea appropriately to the best of capability and comprehension.

- Owing to limited resources and time constraints, a smaller dataset was used and output data (aspect terms) was visualised by generating a WordCloud - a simple yet profound depiction.

  Given the opportunity and resources, a larger dataset would have been extracted and incorporated, and the code developed would have been refined to finer detail.

Furthermore, a more expressive and articulated data visualization would have been implemented to convey the data, which can be interpreted effortlessly.

- The project can potentially be enhanced further to gain more insight by considering a larger dataset and a notable feature to organise and depict sentiment and aspect based on categories such as - user location, age, profession, and other key features along with a visualisation of the varying sentiment based on the categories.

  Elaborate and intricate improved algorithms and methodologies can be learnt and implemented to analyse and extract sentiment and aspect with improved efficacy in order to produce finer elegant outputs.

  Sophisticated data visualisation techniques can be implemented to portray meaningful data in an engaging and insightful fashion to identify and understand underlying trends, patterns and outliers .

VIII. References

VIII.1 Package Documentations

- https://github.com/JustAnotherArchivist/snscrape/blob/master/snscrape/modules/twitter.py
- https://textblob.readthedocs.io/en/dev/
- https://stanfordnlp.github.io/stanfordnlp/index.html#get-started
- http://docs.tweepy.org/en/latest/
- https://www.nltk.org/

VIII.2 Referenced Websites

- https://www.freecodecamp.org/news/how-to-build-a-twitter-sentiments-analyzer-in-python-using-textblob-948e1e8aae14/#:~:text=the%20library%20textblob.-,TextBlob,classification%2C%20translation%2C%20and%20more.
- https://medium.com/@r.ratan/tweepy-textblob-and-sentiment-analysis-python-47cc613a4e51
- https://www.guru99.com/nltk-tutorial.html
- https://www.earthdatascience.org/courses/use-data-open-source-python/intro-to-apis/analyze-tweet-sentiment-in-python/
- https://medium.com/analytics-vidhya/aspect-based-sentiment-analysis-a-practical-approach-8f51029bbc4a

VIII.3 Paper Referenced

- https://www.polibits.gelbukh.com/2018_57/Simple%20and%20Effective%20Feature%20Based%20Sentiment%20Analysis%20on%20Product%20Reviews%20using%20Domain%20Specific%20Sentiment%20Scores.pdf