

ST446 Project

Twitter Sentiment Analysis of the COVID-19 Pandemic

Table of Contents

1	Introduction	Pg.3
2	Background	
	i) Stream Processing	Pg.4
	ii) Sentiment Analysis	Pg.5
	iii) LDA & Topic Modelling	Pg.6
3	Data	Pg.9
4	Methodology	
	iv) Stream Processing	Pg.10
	v) Sentiment Analysis	Pg.12
	vi) Topic Modelling using LDA	Pg.12
5	Results	Pg.14
6	Conclusion	Pg.24
7	Bibliography	Pg.25

Introduction

The coronavirus pandemic has had a pronounced effect on the lives of people around the world. It has not only raised serious economic and public health concerns but also posed challenges around the consequences of lockdowns such as home working and online learning. Social Media platforms like Twitter are a great resource to capture human emotions and thoughts. During these trying times, people have taken to social media to discuss their fears, opinions, and insights on the global pandemic. In this project, we will analyse public sentiments around tweets relating to corona virus, lockdown, vaccination, work from home and online learning to see whether the opinions of people have changed after one year of the pandemic as the world adjusted to the 'new normal'. Due to government-imposed lock-downs to contain the spread of the virus, people were forced to adapt to Work From Home scenarios, and educational institutions were asked to implement online learning. Some people might regard WFH as a necessary step, while others regard it as an inconvenience due to the lack of high-speed internet or smart devices. This project presents how some daily aspects of life have been affected by analysing people's sentiment towards online learning and WFH scenarios. We will also measure the frequency of words used in corona virus related tweets to understand the most common discussions that people are associating with the pandemic.

Further, we want to understand the major topics around the corona virus discussion. As a consequence of the pandemic, there have been major concerns around economic downfall, poor public health, breach of lockdown and restrictions, online exams, vaccinations and their side effects, new variants and second waves in the past year. We want to know if these are still points of concern now or if new topics of concern have risen over past year. For this, we will perform topic modelling using Latent Dirichlet Allocation. We will also run topic modelling with different number of worker nodes on the Google Cloud Platform cluster to check for any computational gains.

Background

Stream Processing

Stream processing is the processing of data in motion, or in other words, computing on data directly as it is produced or received. The majority of data are born as continuous streams: sensor events, user activity on a website, financial trades, and so on – all these data are created as a series of events over time. Before stream processing, this data was often stored in a database, a file system, or other forms of mass storage. Applications would query the data or compute over the data as needed.

Why do we use Stream Data Processing?

1. **Reduced Latency:** Batch processing lets the data build up and try to process them at once while stream processing process data as they come in hence spread the processing over time. Hence stream processing can work with a lot less hardware than batch processing and so are faster. They are ideal when data comes in never ending streams (like Twitter data) as the step-wise approach in stream processing handles this kind of data more efficiently
2. **Scalability:** Sometimes data is huge and it is not even possible to store it. Stream processing let you handle large fire horse style data and retain only useful bits. Computations in stream processing are done by making passes through data using a small memory footprint at any point of time
3. **Stream processing is highly beneficial** if the events you wish to track are happening frequently and close together in time. It is also best to utilize if the event needs to be detected right away and responded to quickly. As in our case, if we detect some new areas of concern around the corona virus pandemic, we can identify and find aid for it immediately rather than waiting for much more harm (and hence discussion) to happen around it

How does stream data processing work?

3 steps are followed for stream data processing:

Input: Data can be inputted into the system through various sources like Kafka, Flume, HDFS, Twitter, Kinesis etc.

Data Processing: This data is then processed in the spark system using operations such as map, reduce, join and window as well as by using machine learning and graph processing algorithms

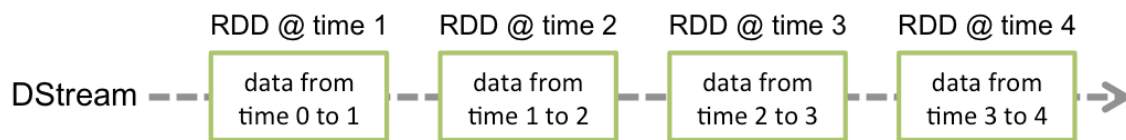
Output: After processing the data, output is produced in the form of HDFS files, databases or dashboards.



Source: <https://spark.apache.org/docs/latest/streaming-programming-guide.html>

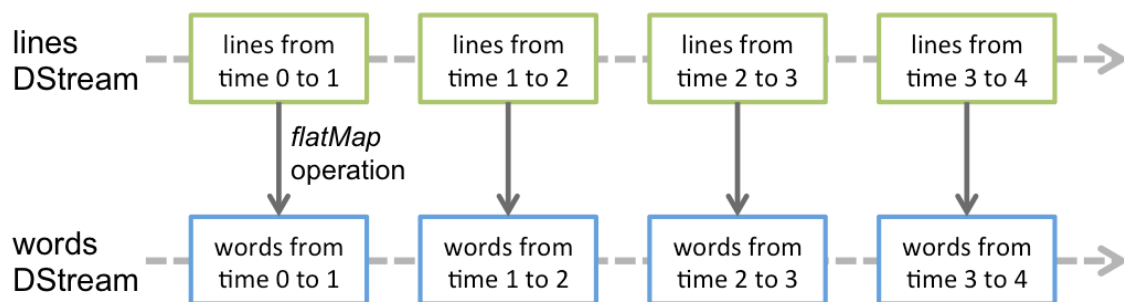
Discretized Streams (DStreams)

Discretized Stream or **DStream** is the basic abstraction provided by Spark Streaming. It represents a continuous stream of data, either the input data stream received from source, or the processed data stream generated by transforming the input stream. Internally, a DStream is represented by a continuous series of RDDs, which is Spark's abstraction of an immutable, distributed dataset. Each RDD in a DStream contains data from a certain interval, as shown in the following figure.



Source: <https://spark.apache.org/docs/latest/streaming-programming-guide.html>

Any operation applied on a DStream translates to operations on the underlying RDDs. For example, if converting a stream of lines to words, the `flatMap` operation is applied on each RDD in the lines DStream to generate the RDDs of the words DStream. This is shown in the following figure.



Source: <https://spark.apache.org/docs/latest/streaming-programming-guide.html>

Sentiment Analysis

Sentiment analysis is the gathering of people's views regarding any event happening in real life. In such situations in which the world is currently going through, understanding the emotions of the people stands extremely important. The grave scenario wherein people cannot go out of their houses demands exploring what the people is actually being thinking about the whole scenario. This information can be used by Public Health organisations, governments and welfare institutions to provide aid, make policies and prepare for the forthcoming challenges. Hence, we will work on understanding the demanding situation especially on social media. Since Twitter is one reliable source for understanding public sentiment we stream tweets from Twitter for this project.

Twitter Streaming

The social media platform at this moment which could prioritize the sharing of meaningful content is Twitter. Twitter is one of the most trendy micro blogging sites, considered as a crucial depository of sentiment analysis. Netizens tweet their expressions within allotted 140 characters. This work consists of 4 different datasets, the first one comprising of 100,000 tweets that use #coronaviruspandemic, second consisting of 10,000 tweets using #COVIDworkfromhome, third one containing 10,000 tweets

using #zoomuniversity (symbolising references to online learning) and the fourth one uses snsrape to scrape 100,000 tweets relating to keyword 'coronaviruspandemic' for topic modelling.

Packages used

For sentiment analysis we will be using the packages snsrape and NLTK.

Snsrape is a Python library that can be used to scrape tweets through Twitter's API without any restrictions or request limits. Moreover, you don't even need a Twitter developer account to scrape tweets when you use snsrape. It allows us to access more tweets than the Twitter developer account cap of 5,000 tweets a month. We are using the snsrape in this project to obtain more than 220,000 tweets over the past on year from twitter as we want to understand the changing sentiments of people over time. This wouldn't be possible using traditional packages such as Tweepy due to the cap of 5,000 tweets per month set by Twitter.

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum. We will specifically use two NLTK features stopwords and Textblob. Stopwords feature consists of a large number of stopwords in English- words that do not have a significant contribution to the meaning of the text or the message it wants to convey. Using this feature we will remove all unnecessary stopwords from the text we want to analyse as they don't add any value to them.

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

We will be analysing the subjectivity and polarity of tweets. **Polarity** helps us to understand how positive or negative a sentiment is. It ranges between -1 and +1, the closer the polarity is to +1, the more positive the message. **Subjectivity** explains how much a piece of text depends on emotions. It ranges from a score of 0 to 1 (0 being less subjective). We will also be using the package wordcloud to create a word cloud of the most common words of discussion around a specific topic.

LDA and Topic Modelling

LDA is a topic modelling algorithm used to cluster documents in a corpus into fixed number of topics. The main idea behind LDA is that each document can be described by a distribution of topics (document-topic) and each topic can be described by a distribution of words (topic-word). For example, if we have 3 sentences:

1. I love travelling through Europe
2. My favourite form of dance is Salsa
3. I would love to take a trip to Spain to learn Salsa

LDA would classify sentence 1 as 100% topic 1, sentence 2 as 100% topic 2 and sentence 3 as 50% topic 1 and 50% topic 2. LDA would give us a breakdown of which words represent the topics best:

Word	Topic 1	Topic 2
Travelling	40%	3%
Europe	20%	5%
Dance	5%	35%
Salsa	2%	25%

Table 1: Displays the distribution of topics over words. Only four words in the corpus have been included in the above table

There are 2 stages to the LDA process. First, we select the number of topics/clusters. Second, we cycle through all the words in the corpus and randomly assign words to one topic. This allows us to calculate the document-topic and topic-word distributions. Because this random allocation will not lead to meaningful topics, topics are iteratively updated.

LDA assumes the following generative process for each document w in a corpus D

1. Choose $N \sim \text{Poisson}(\xi)$.
2. Choose $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n :
 - (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$.
 - (b) Choose a word w_n from $p(w_n | z_n, \beta)$, a multinomial probability conditioned on the topic z_n .

Several simplifying assumptions are made in this basic model, some of which we remove in subsequent sections. First, the dimensionality k of the Dirichlet distribution (and thus the dimensionality of the topic variable z) is assumed known and fixed. Second, the word probabilities are parameterized by a $k \times V$ matrix β where $\beta_{ij} = p(w^j = 1 | z^i = 1)$, which for now we treat as a fixed quantity that is to be estimated. Finally, the Poisson assumption is not critical to anything that follows and more realistic document length distributions can be used as needed. Furthermore, note that N is independent of all the other data generating variables (θ and z). It is thus an ancillary variable and we will generally ignore its randomness in the subsequent development. A k -dimensional Dirichlet random variable θ can take values in the $(k-1)$ -simplex (a k -vector θ lies in the $(k-1)$ -simplex if $\theta_i \geq 0, \sum_{i=1}^k \theta_i = 1$), and has the following probability density on this simplex:

$$p(\theta | \alpha) = \frac{\Gamma(\sum_{i=1}^k \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \theta_1^{\alpha_1-1} \dots \theta_k^{\alpha_k-1}, \quad (1)$$

where the parameter α is a k -vector with components $\alpha_i > 0$, and where $\Gamma(x)$ is the Gamma function. The Dirichlet is a convenient distribution on the simplex — it is in the exponential family, has finite dimensional sufficient statistics, and is conjugate to the multinomial distribution. In Section 5, these properties will facilitate the development of inference and parameter estimation algorithms for LDA. Given the parameters α and β , the joint distribution of a topic mixture θ , a set of N topics z , and a set of N words w is given by:

$$p(\theta, z, w | \alpha, \beta) = p(\theta | \alpha) \prod_{n=1}^N p(z_n | \theta) p(w_n | z_n, \beta), \quad (2)$$

where $p(z_n|\theta)$ is simply θ_i for the unique i such that $z_n^i = 1$. Integrating over θ and summing over z , we obtain the marginal distribution of a document:

$$p(\mathbf{w}|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^N \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \right) d\theta. \quad (3)$$

Finally, taking the product of the marginal probabilities of single documents, we obtain the probability of a corpus:

$$p(D|\alpha, \beta) = \prod_{d=1}^M \int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d) p(w_{dn}|z_{dn}, \beta) \right) d\theta_d.$$

Data

#coronaviruspandemic Tweets Dataset

100,000 tweets in English using the #coronaviruspandemic, posted between 31-03-20 and 31-03-21 were obtained by using the snsrape package. We have extracted these tweets over a period of twelve months to understand whether sentiments around the pandemic, lockdowns and vaccinations changed over time, as the world adapted to the 'new normal'. The tweets are stored in a list called 'tweets_corona' with the tweet text and timestamp (the time that it was created). This list will be converted into a dataframe for ease of operations. The tweet text consists of several stopwords, @mentions and hyperlinks which need to be cleaned before conducting sentiment analysis on the data. We will be using stopwords, TextBlob and re packages to pre-process this data and make it fit for analysis.

#COVIDworkfromhome Tweets Dataset

100,000 tweets in English using the #COVIDworkfromhome, posted between 25-03-20 and 01-04-21 were obtained by using the snsrape package. We have extracted these tweets from the first time such restrictions were applied around the world until recently to understand whether sentiments around remote working changed over time, as the world adapted to the 'new normal'. The tweets are stored in a list called 'tweets_wfh' with the tweet text and timestamp (the time that it was created). This list will be converted into a dataframe for ease of operations. The tweet text consists of several stopwords, @mentions and hyperlinks which need to be cleaned before conducting sentiment analysis on the data. We will be using stopwords, TextBlob and re packages to pre-process this data and make it fit for analysis.

#zoomuniversity Tweets Dataset

100,000 tweets in English using the #zoomuniversity, posted between 31-03-20 and 31-04-21 were obtained by using the snsrape package. This hashtag has been used frequently in tweets describing online learning experience. Other hashtags such as #onlinelearning consist of a larger range of discussion including online courses, online learning platforms etc. which did not necessarily relate to the pandemic. We have extracted these tweets from the first time such restrictions were applied around the world until recently to understand whether sentiments around remote learning changed over time, as the world adapted to the 'new normal'. The tweets are stored in a list called 'tweets_list' with the tweet text and timestamp (the time that it was created). This list will be converted into a dataframe for ease of operations. The tweet text consists of several stopwords, @mentions and hyperlinks which need to be cleaned before conducting sentiment analysis on the data. We will be using stopwords, TextBlob and re packages to pre-process this data and make it fit for analysis.

Coronavirus Tweets Large Dataset

100,000 tweets using the keyword 'coronavirus' were scraped from Twitter using the snsrape package. Unlike the traditionally used twitter API, it is not limited to scraping tweets for only a duration of 7 days. It allows anyone to scrape tweets without requiring personal API keys. It can return thousands of tweets in seconds, and has powerful search tools that allows for highly customisable searches. We will pre-process this dataset using the NLTK package.

Methodology

Stream Processing

Before we perform sentiment analysis and topic modelling, let us first understand what are the most common words that are used in coronavirus related tweets? How frequently does each word appear in tweets? For this, we will perform streaming data processing to measure the frequency of words in each tweet using Kafka. We first create a GCP cluster with the following initializations:

```
gcloud dataproc clusters create cluster-name --project mystic-centaur-302320 \
--subnet default --region europe-west2 --zone europe-west2-a --master-machine-type n1-standard-4 --
master-boot-disk-size 500 --num-workers 0 --worker-machine-type n1-standard-4 --worker-boot-disk-
size 500 --image-version 1.3-deb9 \
--initialization-actions 'gs://dataproc-initialization-actions/jupyter/jupyter.sh','gs://dataproc-
initialization-actions/python/pip-install.sh','gs://dataproc-initialization-
actions/zookeeper/zookeeper.sh','gs://dataproc-initialization-actions/kafka/kafka.sh' \
```

Some flags and initialization actions we have included in this command, so we can install PySpark and get Kafka (which relies on Zookeeper) running are:

```
--initialization-actions 'gs://dataproc-initialization-actions/jupyter/jupyter.sh','gs://dataproc-
initialization-actions/python/pip-install.sh','gs://dataproc-initialization-
actions/zookeeper/zookeeper.sh','gs://dataproc-initialization-actions/kafka/kafka.sh' \
--metadata 'PIP_PACKAGES=sklearn nltk pandas graphframes pyspark kafka-python tweepy'
--metadata 'PIP_PACKAGES=sklearn nltk pandas graphframes pyspark kafka-python tweepy'
```

Apache Kafka is a distributed streaming platform. It lets you:

- Publish and subscribe to streams of records. In this respect it is similar to a message queue or enterprise messaging system.
- Store streams of records in a fault-tolerant way.
- Process streams of records as they occur.

It allows you to build real-time streaming data pipelines that reliably get data between systems or applications. It can also let you build real-time streaming applications that transform or react to the streams of data. After setting up the cluster, we will take the following steps

Step 1: Create a cluster with the required initialisations

In the SSH shell of the cluster, run the following commands:

```
cd /usr/lib/kafka
sudo -s
bin/kafka-server-start.sh config/server.properties &
```

Let this running. Click on the gear symbol and the top right hand corner of the screen and open a new connection to this cluster

Step 2: Run the Twitter producer file

In the new tab, run the following commands

Get inside the SPARK_HOME

```
cd $SPARK_HOME
```

Then run

```
sudo nano $SPARK_HOME/conf/spark-defaults.conf
```

To edit this file. Go the bottom of this document and append the following line:

```
spark.jars.packages org.apache.spark:spark-streaming-kafka-0-8_2.11:2.3.0
```

We do this step because it is important that you use this version of the JAR, otherwise Java will present errors.

Use the gear symbol at the top right hand corner of the shell to upload the project_producer.py file

Then enter ~ using cd ~ command and run

```
python project_producer.py
```

This python file uses Tweepy library to stream tweets using the Twitter Developer account. It creates a topic 'twitter-stream', streams tweets with the keyword 'coronavirus' and prints the length of each tweet.

Step 3: Run the Twitter Analyser file

Now we will run a program to see which words occur in the tweets. Open a new connection to the SSH shell, upload the the python file kafka_twitter_analyser.py and run it using the following command:

```
python kafka_twitter_analyser.py
```

This program creates word tokens from the tweets, cleans them and then prints them for each tweet. This will help us see the words that are being used along in coronavirus tweets at a glance

Step 4: Run the Twitter pyspark file

Finally, we are going to run the pyspark file to count the frequency of all the words (excluding stop words and characters) used in tweets. For this we will first upload the file project_pyspark.py into the cluster and install the nltk library using the following commands:

```
python
import nltk
nltk ("stopwords")
exit()
```

Then we will copy this into the /home directory

```
sudo cp -r nltk_data/ /home/
```

Finally, we will run the file in the SPARK_HOME directory using the following commands

```
cd $SPARK_HOME
unset PYSARK_DRIVER_PYTHON
bin/spark-submit ~/project_pyspark.py
```

Running this, we will see the words along with their frequency in descending order for each tweet.

Sentiment Analysis

We will follow a 4 step procedure to perform sentiment analysis on #coronaviruspandemic dataset

Step 1: Get the tweets

We extract the necessary tweets using the snsrape and store in a pandas DataFrame. We want to specifically test the sentiments around keywords 'lockdown' and 'vaccination' since these are two key discussions around the COVID-19 pandemic. Since the lockdown has had a major impact on the economy and people's mental health and well-being, we want to understand whether the people are still struggling with lockdown's consequences or if they have found ways around it. Recent concerns over the vaccine's side effects and availability has also raised some concerns around the world and so it is worthwhile to understand people's overall opinions about the vaccinations. We have created two separate columns 'lockdown' and 'vaccination' that have a value 1 if they contain any reference to these two key words, otherwise they take value 0.

Step 2: Data Pre-Processing

We will download the Natural Language Toolkit (nltk packages) to remove stopwords and custom stop words (set by us) from the text and make it cleaner. We will also lemmatize the text and use package re to remove hyperlinks and @mentions from the tweets.

Step 3: Calculate Sentiment

From our sentiment analysis we are going to get 2 values- polarity and subjectivity.

Polarity helps us to understand how positive or negative a sentiment is. It ranges between -1 and +1, the closer the polarity is to +1, the more positive the message. **Subjectivity** explains how much a piece of text depends on emotions. It ranges from a score of 0 to 1 (0 being less subjective).

Once we get the polarity and subjectivity for both our reference words 'lockdown' and 'vaccine', we will go ahead and find aggregate statistics for these to understand the overall polarity and subjectivity for both words.

Step 4: Visualisation of Sentiments

Once we've calculated the polarity, subjectivity and aggregate statistics, we will create visualisations for these to see patterns in data more accurately. We will create a line graph of changing polarity over time to see whether the overall trends have changed over time. We will also create a wordcloud to gauge the most commonly used words in the tweets.

We will use the same procedure for all the other sentiment analysis datasets. This whole python file was run on GCP for fast and efficient results since we are dealing with massive amounts of data.

Topic Modelling using LDA

We first create a bucket and upload the actions.sh file in that bucket. Then, we create three different clusters with different number of worker nodes using the following commands:

Cluster with 2 worker nodes

```
gcloud beta dataproc clusters create eg-cluster --project mystic-centaur-302320 \
  --bucket shruti-bucket --region europe-west2 \
  --image-version=1.4-debian10 \
  --optional-components=ANACONDA,JUPYTER \
```

```
--enable-component-gateway \  
--initialization-actions \  
gs://goog-dataproc-initialization-actions-europe-west2/python/pip-install.sh,gs://shruti-bucket/my-  
actions.sh \  
--metadata 'PIP_PACKAGES=sklearn nltk pandas numpy'
```

Similarly, we create clusters with 0 and 3 worker nodes and perform the same topic modelling task on all three. We will take the following steps to perform topic modelling on tweets:

Step 1: Get the Tweets

We extract 100,000 tweets using the snsrape and store in a pandas DataFrame. We want to understand the top topics of discussion around these tweets.

Step 2: Parse the data

Here we make use of the natural language processing module nltk. Both the module and the corresponding data have already been downloaded by our custom cluster initialisation actions.

We will have to process the messages to make them amenable to analysis. Important steps include:

- Tokenisation chops text into useful units (words).
- Lemmatisation groups together inflected words, yields their dictionary form

Step 3: Perform LDA

Once the data is ready for analysis, we perform topic modelling on it using Latent Dirichlet Allocation. Latent Dirichlet allocation (LDA) is a topic model which infers topics from a collection of text documents. LDA can be thought of as a clustering algorithm as follows:

- Topics correspond to cluster centers, and documents correspond to examples (rows) in a dataset.
- Topics and documents both exist in a feature space, where feature vectors are vectors of word counts (bag of words).
- Rather than estimating a clustering using a traditional distance, LDA uses a function based on a statistical model of how text documents are generated.

We use the LDA package from the pyspark.ml.clustering library to get 10 topics from the tweets.

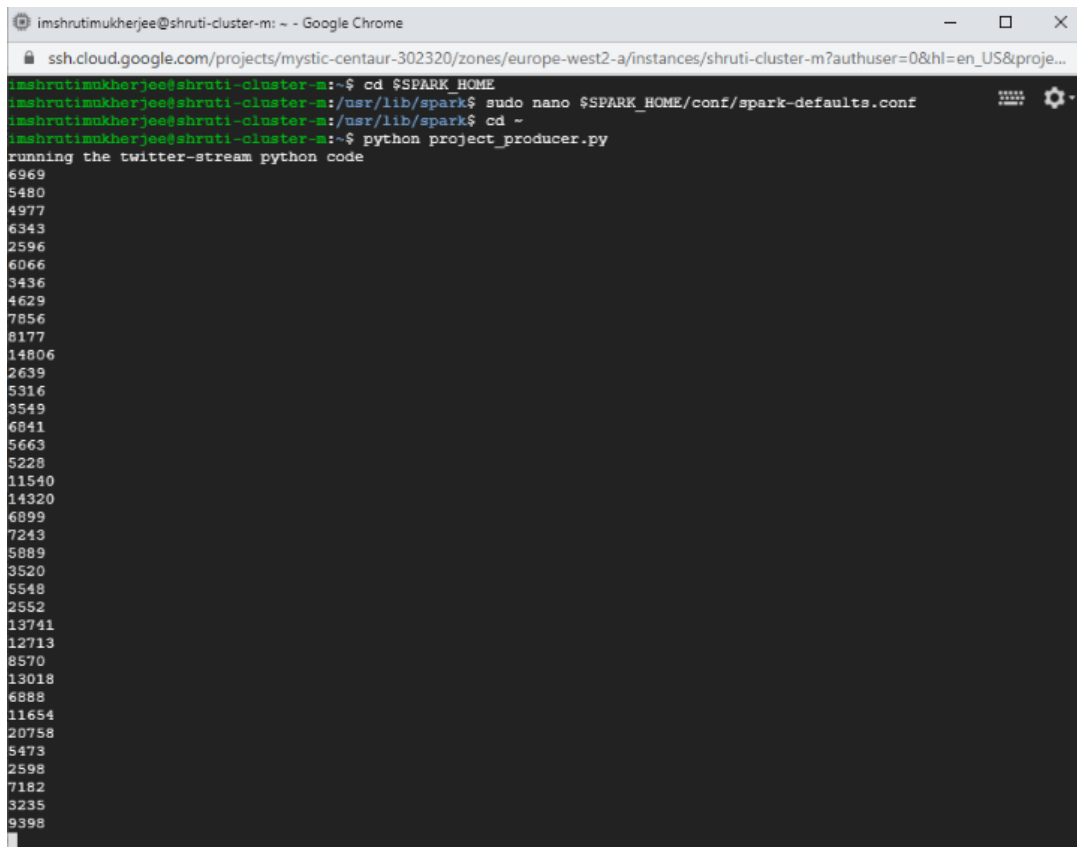
Step 4: Infer topics from the words

Finally, we go through the words under each topic to understand what topic they belong to. For eg:

Results

Stream Data Processing

We first run the `project_producer.py` file to print the length of the tweets with keyword 'coronavirus' and observe the following result:



```
imshrutimukherjee@shruti-cluster-m: ~ - Google Chrome
ssh.cloud.google.com/projects/mystic-centaur-302320/zones/europe-west2-a/instances/shruti-cluster-m?authuser=0&hl=en_US&proje...

imshrutimukherjee@shruti-cluster-m:~$ cd $SPARK_HOME
imshrutimukherjee@shruti-cluster-m:/usr/lib/spark$ sudo nano $SPARK_HOME/conf/spark-defaults.conf
imshrutimukherjee@shruti-cluster-m:/usr/lib/spark$ cd ~
imshrutimukherjee@shruti-cluster-m:~$ python project_producer.py
running the twitter-stream python code
6969
5480
4977
6343
2596
6066
3436
4629
7856
8177
14806
2639
5316
3549
6841
5663
5228
11540
14320
6899
7243
5889
3520
5548
2552
13741
12713
8570
13018
6888
11654
20758
5473
2598
7182
3235
9398
```

This list of numbers represent the length of tweets. This list keeps on growing as more and more tweets are obtained from Twitter.

After this, in the next tab we run the kafka_twitter_analyser.py file to get the list of words from each tweet with keyword ‘coronavirus’.

```
imshrutimukherjee@shruti-cluster-m: ~ - Google Chrome
ssh.cloud.google.com/projects/mystic-centaur-302320/zones/europe-west2-a/instances/shruti-cluster-m?authuser=0&hl=en_US&proje...

[nltk data] /home/imshrutimukherjee/nltk_data...
[nltk data] Unzipping tokenizers/punkt.zip.
consumer started
[{'rt', 'rrrmovie', 'patna', 'covidinfo', 'remedisivir'}]
[{'boycottchina', 'antination', 'siachen'}]
[{'arvindkejriwal', 'said', 'dearth', 'takers', 'tankers', 'imported', 'bangkok', 'mitigate', 'https'}]
[{'diana', 'dazzling', 'compone', 'una', 'canción', 'solidaria', 'contra', 'el', 'coronavirus', 'https', 'lifestyle', 'yomequedencasa'}]
[{'rt', 'nicolasmaduro', 'estamos', 'dando', 'pasos', 'certeros', 'en', 'el', 'fortalecimiento', 'del', 'sistema', 'público', 'nacional', 'de', 'salud', 'que', 'se', 'encuentra', 'atendiendo'}]
[{'rt', 'hitchensback', 'eh', 'j', 'suis', 'en', 'larmes', 'mème', 'pas', 'une', 'petite', 'crevette', 'pour', 'la', 'petite'}]
[{'rt', 'rrrmovie', 'patna', 'covidinfo', 'remedisivir'}]
[{'coronavirus', 'münchen', 'stadt', 'stoppt', 'impfaktion', 'für', 'lehrer', 'weniger', 'als', 'ein', 'drittel', 'nutzte', 'das', 'impfangebot', 'https'}]
[{'rt', 'australian', 'arrogance', 'hypernationalism', 'bureaucratic', 'incompetence', 'combined', 'create', 'crisis', 'epic', 'proportions', 'india'}]
[{'rt', 'zeenews', 'coronavirus', 'sputnikv', 'coronavaccine', 'https'}]
[{'full', 'year', 'behind', 'first', 'thoughts', 'anyone', 'paying', 'attention', 'scientists', 'full', 'fucking', 'year', 'https', 'tcolychifwhy'}]
[{'rt', 'yrdeshmukh', 'और', 'गए', 'और'}]
[{'coronavirusrelated', 'deaths', 'rising', 'fast', 'due', 'speed', 'intensity', 'pandemic', 'adding', 'healthcare', 'https'}]
[{'maintenant', 'sait', 'que', 'quand', 'ils', 'disent', 'que', 'c', 'est', 'sous', 'contrôle', 'ça', 'ne', 'l', 'est', 'pas', 'du', 'tout'}]
[{'rt', 'infonewsabc', 'las', 'televisión', 'mienten', 'ayer', 'sacaron', 'esta', 'imagen', 'de', 'gente', 'muerta', 'de', 'coronavirus', 'en', 'las', 'calles', 'de', 'india'}]
[{'rt', 'ndtv', 'covid', 'deaths', 'missing', 'delhi', 'data', 'reveal', 'civic', 'records', 'https', 'https'}]
[{'lemondefr', 'strong', 'buy', 'novavax', 'novavax', 'coronavirus', 'vonderleyen', 'andreibabis', 'elonmusk', 'https', 'tcojfdnltztum'}]
[{'rt', 'rrrmovie', 'patna', 'covidinfo', 'remedisivir'}]
[{'rt', 'gaissarjoiya', 'گائیسار جو یا', 'giving', 'details', 'coronavirus', 'situation', 'country', 'said', 'virus', 'patients'}]
[{'rt', 'evashengll', 'indian', 'soldiers', 'border', 'security', 'force', 'camp', 'use', 'pressure', 'cookers', 'steam', 'inhalation', 'fight', 'deadly', 'wave'}]
[{'rt', 'socsclences', 'anyone', 'unsure', 'boris', 'johnson', 'really', 'said', 'let', 'bodies', 'pile', 'high', 'thousands', 'remember', 'time'}]
[{'death', 'obituary', 'coronavirus', 'crisis', 'delhi', 'burial', 'grounds', 'run', 'space', 'covid', 'deaths', 'mount', 'https'}]
[{'coronavirus', 'die', 'ersten', 'hamburger', 'obdachlosen', 'sind', 'geimpft', 'https', 'via', 'welt'}]
[{'rt', 'carlzimmer', 'uk', 'deaths', 'crashed', 'since', 'january', 'peak', 'https', 'tcojcyerkfeko', 'https', 'tcowcqskcsyt'}]
[{'ireland', 'ireland', 'reported', 'new', 'confirmed', 'cases', 'coronavirus', 'new', 'death', 'data', 'last', 'update'}]
```

```
imshrutimukherjee@shruti-cluster-m: ~ - Google Chrome
ssh.cloud.google.com/projects/mystic-centaur-302320/zones/europe-west2-a/instances/shruti-cluster-m?authuser=0&hl=en_US&proje...

Linux shruti-cluster-m 4.19.0-0-bpo.9-amd64 #1 SMP Debian 4.19.118-2+deb10u1-bpo
9+1 (2020-06-09) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Apr 26 10:02:14 2021 from 35.235.242.17
imshrutimukherjee@shruti-cluster-m:~$ python kafka_twitter_analyser.py
[nltk data] Downloading package stopwords to
[nltk data] /home/imshrutimukherjee/nltk_data...
[nltk data] Unzipping corpora/stopwords.zip.
[nltk data] Downloading package punkt to
[nltk data] /home/imshrutimukherjee/nltk_data...
[nltk data] Unzipping tokenizers/punkt.zip.
consumer started
[{'三条市', '新潟県', '三条市', 'https'}]
[{'rt', 'gauvaingorges', 'un', 'variant', 'zombiedespece', 'été', 'découvert', 'au', 'guatemala', 'il', 'rend', 'les', 'gens', 'tout', 'bleus', 'et', 'leur', 'fait', 'pousser', 'une'}]
[{'rt', 'josefjg', 'mayores', 'de', 'años', 'que', 'rechazaron', 'la', 'vacuna', 'acaban', 'ingresados', 'por', 'coronavirus', 'la', 'triste', 'prueba', 'de', 'que', 'las', 'vacunas'}]
[{'rt', 'majorgauravarya', 'thank', 'saudi', 'arabia', 'help', 'unprecedented', 'times', 'indiafightback'}]
[{'covid', 'curfew', 'karnataka', 'next', 'days', 'amid', 'steep', 'spike', 'coronavirus', 'cases', 'https', 'tcooemwxfttrb'}]
[{'rt', 'santoroaleandro', 'hola', 'horacionlarreta', 'parece', 'que', 'hay', 'un', 'ministro', 'tuyo', 'que', 'sabe', 'como', 'decirte', 'que', 'te', 'equivocaste', 'fiero'}]
[{'rt', 'aajtak', 'और', 'remedisivir', 'coronavirus', 'https'}]
[{'putangina', 'philippinesrecordbreaker'}]
[{'anteil', 'der', 'hagener', 'innen', 'mit', 'erster', 'impfung', 'anteil', 'der', 'vollständig', 'geimpften', 'hagener', 'innen', 'https'}]
[{'heartbreaking', 'prayers'}]
[{'rt', 'docanocpmisra', 'important', 'information', 'meaningful', 'antibody', 'levels', 'declining', 'months', 'seropositive'}]
[{'rt', 'alberoliver', 'masmadridcm', 'monicagarcias', 'esto', 'salir', 'de', 'este', 'desastre', 'generado', 'por', 'el', 'covid', 'pasa', 'por', 'invertir'}]
[{'rt', 'tothepointnews', 'করোনা', 'অজ', 'modihaitchamkinhai', 'पर', 'पह', 'कर'}]
[{'rt', 'mippicvizia', 'CONTIGO', 'la', 'salud', 'es', 'primero', 'por', 'eso', 'debemos', 'intensificar', 'las', 'medidas', 'de', 'prevención', 'acatar', 'el', 'llamado', 'la'}]
[{'rt', 'inctelevisión', 'पर', 'coronavirus', 'कहर'}]
```

This list continues as we keep getting more and more tweets.

The list generally shows that the words that are generally used along with coronavirus tweets currently are ‘hospitals’, ‘oxygen’, ‘India’, ‘covid’, ‘relief’ etc., possibly because of the sudden surge in coronavirus cases and oxygen deficiency in India. However, we do not know the exact frequency of words for each tweet so we can’t compare between different words to understand which ones are the most common. Thus, we calculated the frequency for different words in a tweet and displayed them in the descending order of frequency for each tweet in the next section.

Finally, we ran the kafka_twitter_pyspark.py file to get the list of words along with their frequency in descending order for each tweet.

```

imshrutimukherjee@shruti-cluster-m: /usr/lib/spark - Google Chrome
ssh.cloud.google.com/projects/mystic-centaur-302320/zones/europe-west2-a/instances/shruti-cluster-m?authuser=0&hl=en_US&proje...

('coronavirus', 5)
('la', 3)
('se', 3)
('con', 2)
('hospitals', 2)
('da', 1)
('exclusive', 1)
('coronapandemic', 1)
('parent', 1)
('fault', 1)
...

-----
Time: 2021-04-27 09:46:06
-----
('needed', 2)
('help', 2)
('nt', 2)
('get', 2)
('coronavirus', 2)
('nigeria', 2)
('news', 1)
('says', 1)
('ca', 1)
('govt', 1)
...

-----
Time: 2021-04-27 09:46:07
-----
('oxygen', 3)
('coronavirus', 2)
('paytm', 1)
('cred', 1)
('trying', 1)
('tackle', 1)
('rmantri', 1)
('decentralized', 1)
('management', 1)
('best', 1)
...

```

```

imshrutimukherjee@shruti-cluster-m: /usr/lib/spark - Google Chrome
ssh.cloud.google.com/projects/mystic-centaur-302320/zones/europe-west2-a/instances/shruti-cluster-m?authuser=0&hl=en_US&proje...

('covid', 2)
('coronavirus', 2)
('diebasipartei', 1)
('ein', 1)
('stück', 1)
('ward', 1)
('coronamaasschuss', 1)
('sein', 1)
('juli', 1)
('granular', 1)
...

-----
Time: 2021-04-27 09:46:10
-----
('delhi', 2)
('india', 2)
('coronavirus', 2)
('oxygen', 1)
('new', 1)
('covid', 1)
('cases', 1)
('covidpositive', 1)
('rss', 1)
('swayamsevak', 1)
...

-----
Time: 2021-04-27 09:46:11
-----
('coronavirus', 2)
('amp', 2)
('5T', 1)
('asaduddinowaisi', 1)
('hospitals', 1)
('run', 1)
('oxygen', 1)
('country', 1)
('continues', 1)
('set', 1)
...

```


This list continues as more and more tweets are received. Overall, this word-frequency distribution shows that the most common words for each tweet are ‘need’, ‘help’, ‘oxygen’, ‘hospital’, ‘country’ etc. Most of these words relate to the new coronavirus outbreak and lack of resources to handle the surge that some countries are facing at this point.

Sentiment Analysis

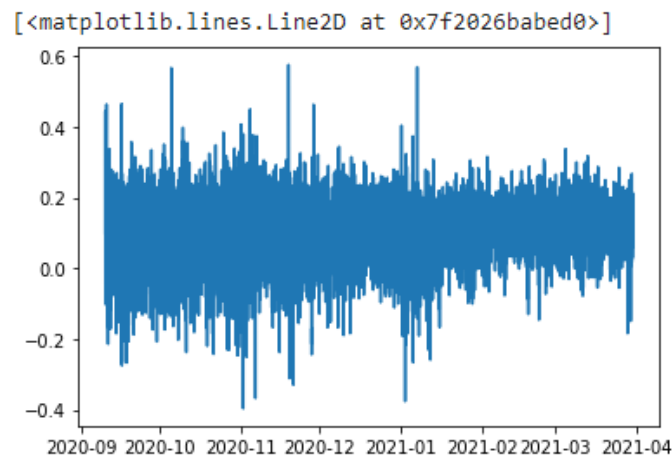
Now that we have seen an overview of the common words and areas of discussion around coronavirus currently, let us go ahead and do sentiment analysis on commonly used keywords in coronavirus tweets over time. Let’s start with the keyword ‘coronaviruspandemic’. Apart from an overview of the sentiments around this keyword, we specifically want to understand the varying sentiments around two other keywords – ‘vaccines’ and ‘lockdown’ over time. Have the opinions of people around lockdowns and vaccinations changed over time?

First we analyse the average polarity and subjectivity for lockdown and vaccination

	polarity				subjectivity			
	mean	amax	amin	median	mean	amax	amin	median
lockdown								
1	0.055205	1.0	-1.0	0.0	0.340459	1.0	0.0	0.366667
	polarity				subjectivity			
	mean	amax	amin	median	mean	amax	amin	median
Vaccine								
1	0.102739	1.0	-1.0	0.055714	0.347823	1.0	0.0	0.381818

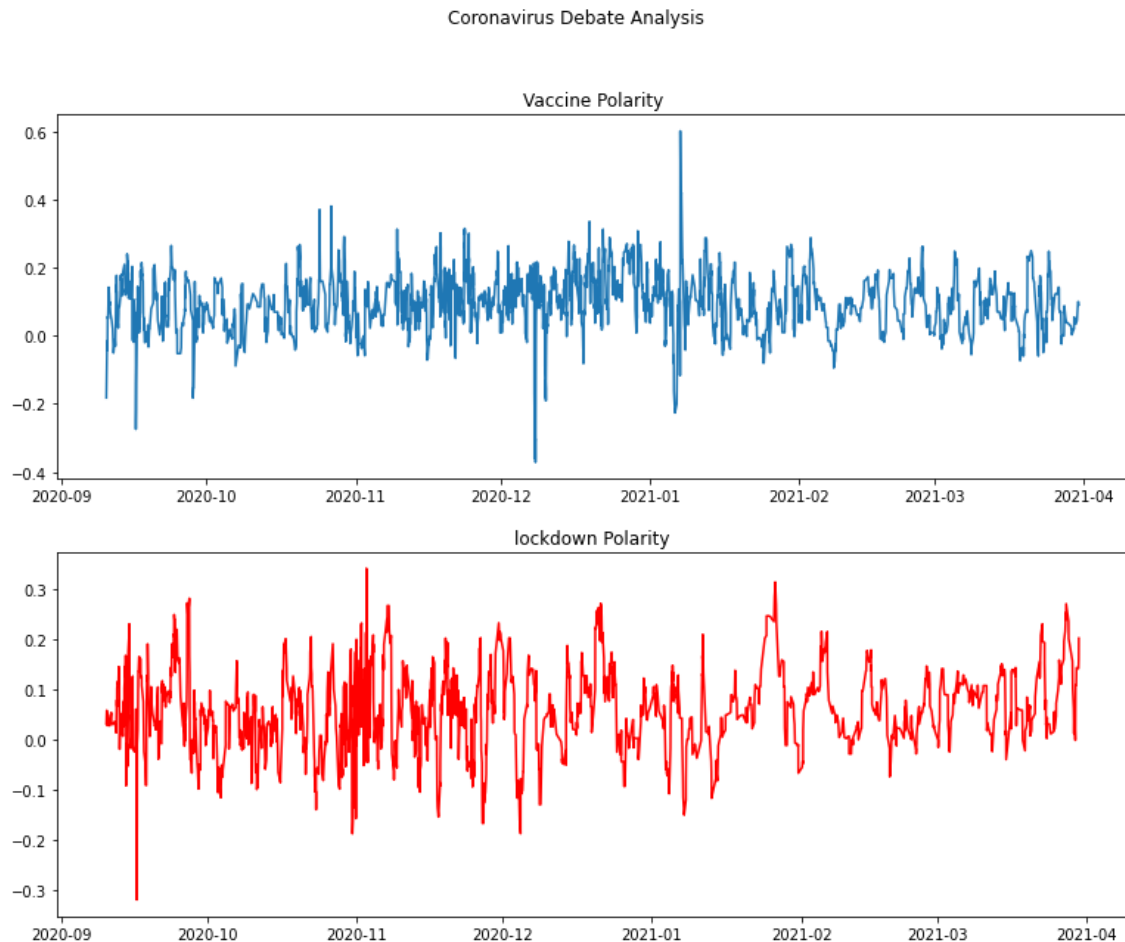
Table 2: Aggregate statistics for polarity and subjectivity of lockdown and vaccine

From Table 2 we observe that the average polarity for lockdown is 0.055205 and that for vaccine is 0.102739. This means that overall, while people are positive about both lockdown and vaccine, they are more positive about vaccines than lockdowns. This could be because most people see vaccinations as a way out of the pandemic and a better alternative than lockdowns, despite the recent concerns over the side-effects of some vaccines. The mean subjectivity for lockdown and vaccine is 0.340459 and 0.347823 respectively, meaning that people are almost as subjective for both of them. Since these values are less than 0.5, it means that people are objective (ie. Discuss facts and news rather than personal opinions) when discussing these subjects.



Graph 1: Polarity of keyword ‘coronaviruspandemic’ over time

From Graph 1, we can see that the polarity for this keyword has become less noisy since February 2021, possibly because most countries have been able to control the spread of the disease and vaccination programmes are also running in full swing across the world. There were surges in extreme opinions around November, December 2020 and January 2021, perhaps because of the ambiguity around lockdowns, vaccines, economic instability and new variants of the virus in some countries.



Graph 2 shows the varying polarity of lockdown and vaccine over time

Vaccines saw a dip in polarity around December when concerns were raised over its efficacy, availability and rollout plan. There was a very sharp incline in polarity for vaccines in January 2021 when scientists confirmed that the vaccinations are actually effective and many governments around the world starting giving it out to vulnerable people and essential care workers.

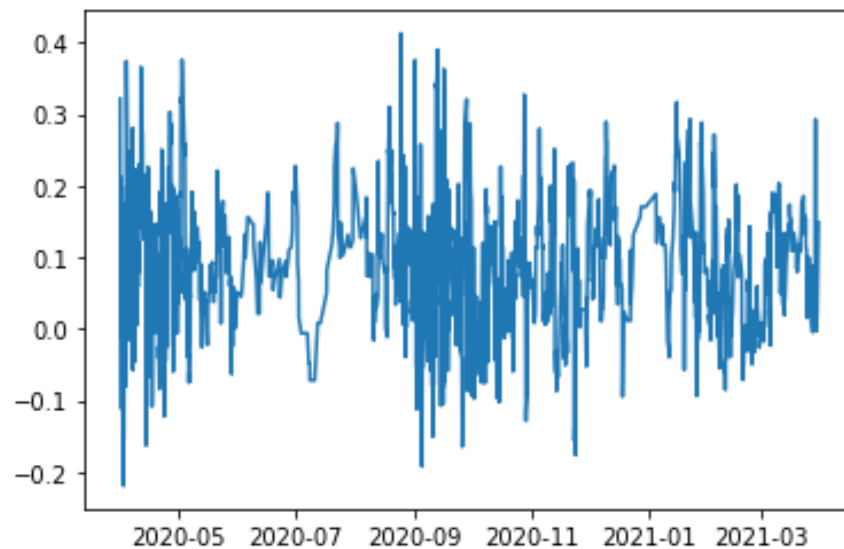
Lockdown experienced a very varying polarity in the beginning and a sharp decline between September and October 2020 when some countries were experiencing a surge in cases while others were planning to re-introduce lockdown restrictions due to rising concerns. February 2021 onwards, as most countries have started lifting lockdown restrictions, the polarity has become less noisy and more positive.

	polarity	subjectivity
mean	0.082755	0.319423
amax	1.000000	1.000000
amin	-1.000000	0.000000
median	0.000000	0.300000

Table 4: Aggregate Statistics for polarity and subjectivity of #zoomuniversity tweets

From Table 4 we can see that the mean polarity for online learning is 0.0827755, meaning that students are overall positive about it. Mean subjectivity is 0.319423, meaning that people are more objective while discussing this topic, that is, they tend to share more news and facts rather than personal opinions.

From the above tables, we can also conclude that people are, in general, more positive about online learning than remote working.



Graph 4: #zoomuniversity polarity over time

Graph 4 shows the varying polarity for the hashtag over time. Since the graph is very noisy, we can say that the opinions of people vary a lot over time. There was a small period of time between June and September 2020 when the opinions were a lot less noisy. This could be the time by which students have accepted and adjusted to this new system of learning and hence less discussions happened around it. However, due to varying policies for education around the world, the opinions have overall varied over time. The polarity was positive for a considerably long period of time around January 2021 when most countries had less cases and hence were considering shift to in-person education.

After running the LDA to obtain 5 topics from the tweets, we obtained the following the results:

```

-----+
|topic|termIndices          |termWeights
|-----+-----+
|0    |[144, 157, 177, 232, 138]| [0.0035920943412764096, 0.003481356245169105, 0.0032706372408289517, 0.0032530572206868045, 0.0031516919120893323]|
|1    |[71, 73, 79, 63, 88]| [0.003777258965710244, 0.0036712944019138263, 0.003654246041164392, 0.003595393493840895, 0.0033791157753770232]|
|2    |[17, 11, 4, 89, 36]| [0.002150201346037923, 0.002028801676022445, 0.002027831344442369, 0.001926257105976524, 0.001917372550543077]|
|3    |[48, 69, 43, 34, 0]| [0.003439122136181102, 0.003398668925510848, 0.0030924404444646262, 0.0030703225813534163, 0.0026767912251297846]|
|4    |[3, 6, 58, 86, 160]| [0.0036220885763424884, 0.003452132811633957, 0.0033162561638388007, 0.003249370335504979, 0.002609674025803824]|
-----+-----+

```

The 5 lists below the table are the list of words for each topic. Now, we will try to infer the topics from these words.

Page 22 of 25

has come into light for its wastage of COVID vaccines. Iran has also experienced record high number of cases again and Switzerland has recorded new variants of the virus. Thus, first topic can be called **Places affected by COVID**

The second list consisting of ['blog' 'covaxin' 'moderna' 'making' 'different'] seems to be a list of COVID related remedies like vaccines, trials, pharmaceutical companies. Thus, we can classify the second topic as **COVID cures discussion**

The third list consisting of ['going' 'go' 'week' 'israel' 'work'] seems to be a discussion around things returning back to normalcy as it discusses things like going to places, working and Israel which recently became COVID free. Thus, we can classify the third topic as **Returning to normalcy after COVID**

The fourth list consisting of ['gujarat' 'brazil' 'canada' 'exam' 'situation'] seems to be a list of places and things that are beginning to become causes of concern due to COVID. All the places mentioned have recently (as of 23-04-21) experienced a COVID-surge and examinations have been canned/postponed at several places because of a sudden surge. Thus, we can classify the fourth topic as **Recently affected by COVID**

The fifth list consisting of ['narendramodi' 'recovery' 'pmoindia' 'south' 'telangana'] is a list of discussions around the recent COVID-19 surge in India. People seem to be discussing politics and governance around this new wave. Thus, we can classify the fourth topic as **COVID Politics in India**

When this algorithm was run on GCP with different number of worker nodes on the cluster, following computational speeds were obtained:

Number of Worker Nodes	Computation Time
0	11.7143 seconds
2	10.6056 seconds
3	11.2349 seconds

Table 1: Computation time with different number of worker nodes

As is evident from Table 1, the LDA algorithm runs the fastest with two worker nodes in the cluster.

Conclusions

Since social media platforms like twitter are a great source to capture human emotions and thoughts, sentiment analysis of tweets can help gauge public sentiments around the most pressing issues around the globe. We use three methods- stream processing, sentiment analysis and topic modelling on tweets over the period of the past year to understand public sentiments, uncover topics of concern and gauge the change in opinions around the COVID-19 pandemic. Since stream processing has the additional advantage of reduced latency and scalability, we first use this method to see the most common words used in tweets related to the coronavirus pandemic. This method is fast, efficient and helps to see anomalies in a stream of data more efficiently.

We further go on to analyse the polarity and subjectivity around ‘vaccine’, ‘lockdown’, ‘work from home’ and ‘online learning’ over a period of one year to measure the change in sentiments as well as the overall sentiments for these words. We observe that people are positive towards both ‘lockdown’ and ‘vaccine’ but more positive about ‘vaccine’ than ‘lockdown’. The sentiments around these words have largely varied across time. We use the #covidworkfromhome to understand sentiments for remote working and observe that the sentiments for this word were initially very negative when it was first introduced, however, over time, the sentiments have become overall positive. For the #zoomuniversity (used to understand sentiments around online learning), the sentiments are overall positive but they have hugely varied over time. On an average, people are more positive about online learning than remote working.

We further perform topic modelling on tweets that have keyword ‘coronaviruspandemic’. Using the Latent Dirichlet Allocation algorithm, we observe that the main topics of discussion for this keyword are places affected by COVID, COVID related medication/ remedies and concerns and politics around the massive COVID surge in India. After using different number of worker nodes to perform the same LDA task, we observe that the cluster with 2 worker nodes had the highest computational efficiency.

Bibliography

Mansoor, M., Gurumurthy, K., U, A. and Prasad, V., 2020. *Global Sentiment Analysis Of COVID-19 Tweets Over Time*. [ebook] Available at: <<https://arxiv.org/pdf/2010.14234.pdf>>.

Boon_Itt, S. and Skunkan, Y., 2020. *Public Perception of the COVID-19 Pandemic on Twitter: Sentiment Analysis and Topic Modeling Study*. [ebook] JMIR Public Health Surveill. Available at: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7661106/>>.

Xue, J., Chen, J., Chen, C., Zheng, C., Li, S. and Zhu, T., 2020. *Public discourse and sentiment during the COVID 19 pandemic: using Latent Dirichlet Allocation for topic modeling on Twitter*. [ebook] Available at: <<https://arxiv.org/ftp/arxiv/papers/2005/2005.08817.pdf>>.

Dubey, A., 2020. *Twitter Sentiment Analysis during COVID-19 Outbreak*. [ebook] Available at: <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3572023>.

Blei, D.M., Ng, A.Y. and Jordan, M.I., 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3, pp.993-1022.

Yang, S. and Zhang, H., 2018. Text mining of Twitter data using a latent Dirichlet allocation topic model and sentiment analysis. *Int. J. Comput. Inf. Eng*, 12(7), pp.525-529.

Chakraborti, N. (2020, August 24). Easy to Play with Twitter Data Using Spark Structured Streaming. Retrieved April 28, 2021, from <https://ch-nabarun.medium.com/easy-to-play-with-twitter-data-using-spark-structured-streaming-76fe86f1f81c>.