# Using Twitter Data to Analyze COVID-19 Cases and Attitudes in Real Time

Angela Folz
anfo9168@colorado.edu
ID: 109652753
CSCI 6502-001B

Lucas Laughlin
lula5767@colorado.edu
ID: 103626680
CSCI 6502-001B

Ksenia Lepikhina
ksle1621@colorado.edu
ID: 104404685
CSCI 6502-001B

Julie Matthias
juma8871@colorado.edu
ID:109658180
CSCI 6502-001B

## ABSTRACT

Social media platforms such as Twitter, Instagram, and Facebook play a prominent role in our society since they can influence what individuals are interested in and can inspire individuals to act in certain ways. Since these platforms are prevalent in today's society, analyzing their impacts on important events such as COVID-19 can help society anticipate the results. This project uses Twitter to study COVID-19. To do so, we utilized the Twitter API and a dataset containing COVID-19 related key words. Topics explored surround COVID-19 tweet sentiment and its relation to the number of COVID-19 cases and deaths. We built a dashboard to show (near) real-time metrics relating to these topics. We hope that our project will not only be useful for studying COVID-19, but that it can also be used in the future for other important events and can be applied to other social media platforms.

## CCS CONCEPTS

• **Applied computing → Document analysis**; • **Computing methodologies → Machine learning**; • **Computer systems organization → Cloud computing**.

## KEYWORDS

twitter, covid, sentiment analysis

## 1 INTRODUCTION

For this project we investigated the relationship between the sentiment of COVID-19 related tweets and the number of COVID-19 cases and deaths. We built a dashboard showing real-time sentiment of COVID-19 related tweets and real-time COVID-19 cases and deaths. The dashboard displays metrics such as the number of tweets mentioning the keyword "corona" today, average COVID-19 sentiment today, the number of new COVID-19 cases today, the number of new COVID-19 deaths today, and more. We also investigated whether the sentiment of COVID-19 related tweets is correlated with the number of confirmed cases and/or deaths and attempted to develop a predictive model.

This information could be useful in helping health departments prepare for an increase in COVID-19 cases, which could be critical in making sure people receive the correct care. Health departments may also be able to tweet more positive sentiment and other information that could potentially change an individual's attitude towards COVID-19, masks, and the vaccine. Additionally, individuals could use the dashboard to understand whether current public sentiment accurately reflects the actual cases and deaths. For example, a user may perceive from social media that people are feeling positive about reopening public spaces, but COVID-19 cases may in fact be increasing.

The rest of this report is organized as follows. Related work is outlined in Section 2. The data used is described in Section 3. The methods used, including models and architecture, are described in Section 4. Evaluation is discussed in Section 5. Finally, Section 6 gives an overview of changes since the project proposal, describes opportunities for future work, and reflects on what we've learned. Section 7 summarizes our conclusions.

## 2 RELATED WORK

COVID-19 has had a massive global impact, and as such there are many recent studies across a wide variety of topics related to social media sentiment analysis of COVID-19. Some studies of particular interest that helped to motivate our research questions are listed below.

For this project, we are largely following the architecture laid out in Lin (2020) [18]. The article details building a real time tweet processing application using some of the big data tools described in Section 4.5.

Sanders et al. (2020) [24] analyzed a database of more than one million tweets from January-May 2020 to determine public sentiment toward wearing masks. Natural language processing, clustering, and sentiment analysis techniques were used. The authors conclude that the number of tweets related to masks increased significantly, and find that negative sentimentality also increased.

O'Leary and Storey (2020) [20] developed a model using the number of Google searches for the term "coronavirus", the number of Twitter tweets including the word "coronavirus", and the number of Wikipedia coronavirus page views that was an effective predictor of COVID-19 cases and deaths in the U.S. They found different lag times for the different platforms: Google predicted well 2-3 weeks in advance, Twitter predicted well 1-2 weeks in advance, and Wikipedia predicted well 7 days in advance.

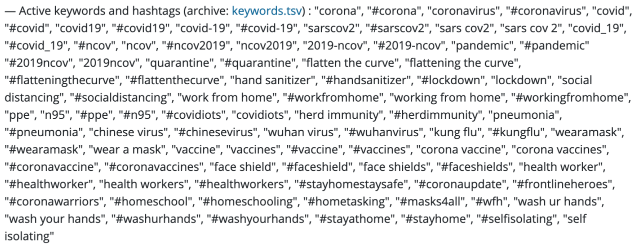
Qin et al. (2020) [21] analyzed social media search indexes for symptoms related to COVID-19 such as dry cough, fever, and pneumonia from December 31, 2019, to February 9, 2020, to predict new suspected cases of COVID-19 during that time. Methods used were subset selection, forward selection, lasso regression, ridge regression, and elastic net. Subset selection was determined to have the lowest estimation error and a moderate number of predictors.

Kaila et al. (2020) [15] analyzed tweets with #coronavirus. Sentiment analysis was performed using NRC sentiment dictionary to find 8 different emotions and their corresponding valence. Topic modelling using Latent Dirichlet Allocation was used for identifying topics in tweets. The authors conclude that Twitter was effective in spreading information related to the pandemic, with little misinformation, particularly when compared to other outbreaks such as Ebola.

# 3 DATA

For this project, our group originally planned to utilize Twitter's API through approved developer accounts to stream public tweets. However, after further research we decided to use a public dataset, the Coronavirus (COVID-19) Tweets Dataset [16] [17], because of the advantages that it offers. First, the limit for a standard Twitter developer account is 500,000 tweets per month, while the COVID-19 Tweets Dataset has over 1 billion COVID-19-related tweets collected since March 20, 2020, which allows us to analyze more data points. Second, the COVID-19 Tweets Dataset includes a sentiment score for each tweet, eliminating the need for us to implement an NLP model to compute a sentiment score.

The COVID-19 Tweets Dataset uses around 90 keywords and hashtags related to COVID-19, such as "coronavirus", "#coronavirus", "covid19", "#covid19", "pandemic", "quarantine", "social distancing", "wearamask", and "vaccine", to collect relevant tweets. For a full list of current key words, see Figure 1. All of the tweets are in English, which is an advantage over other COVID-19 tweet datasets because we don't have to filter out other languages.



— Active keywords and hashtags (archive: keywords.tsv) : "corona", "#corona", "coronavirus", "#coronavirus", "covid", "#covid", "covid19", "#covid19", "covid-19", "#covid-19", "sarscov2", "#sarscov2", "sars cov2", "sars cov 2", "covid_19", "#covid_19", "#ncov", "ncov", "#ncov2019", "ncov2019", "2019-ncov", "#2019-ncov", "pandemic", "#pandemic" "#2019ncov", "2019ncov", "quarantine", "#quarantine", "flatten the curve", "flattening the curve", "#flatteningthecurve", "#flattenthecurve", "hand sanitizer", "#handsanitizer", "#lockdown", "lockdown", "social distancing", "#socialdistancing", "work from home", "#workfromhome", "working from home", "#workingfromhome", "ppe", "n95", "#ppe", "#n95", "#covidiots", "covidiots", "herd immunity", "#herdimmunity", "pneumonia", "#pneumonia", "chinese virus", "#chinesevirus", "wuhan virus", "#wuhanvirus", "kung flu", "#kungflu", "wearamask", "#wearamask", "wear a mask", "vaccine", "vaccines", "#vaccine", "#vaccines", "corona vaccine", "corona vaccines", "#coronavaccine", "#coronavaccines", "face shield", "#faceshield", "face shields", "#faceshields", "health worker", "#healthworker", "health workers", "#healthworkers", "#stayhomestaysafe", "#coronaupdate", "#frontlineheroes", "#coronawarriors", "#homeschool", "#homeschooling", "#hometasking", "#masks4all", "#wfh", "wash ur hands", "wash your hands", "#washurhands", "#washyourhands", "#stayathome", "#stayhome", "#selfisolating", "self isolating"

**Figure 1: Current Key Words**

The sentiment score for each tweet in this dataset was calculated using TextBlob's Sentiment Analysis model. TextBlob is a Python library for NLP. It provides part-of-speech tagging, noun phrase extraction, sentiment analysis, etc.[7] This model defines sentiment scores in the range [-1,+1]. A tweet has positive sentiment if its score is in (0,+1], negative sentiment if its score is in [-1, 0), and neutral sentiment if its score is equal to 0 [17].

In addition to Twitter data, we also needed COVID-19 data to complete this analysis. We utilized the New York Times (NYT) COVID-19 data [28], which contains the cumulative number of COVID-19 cases and deaths in the U.S. from January 1, 2020, to April 12, 2021, in order to compare to the average tweet sentiment. Due to insufficient location data we were not able to restrict our tweet dataset to those only in the United States. However, since our tweets are all in English, we feel that the number of cases and deaths in the U.S. will be representative enough for our analysis. Limiting the tweet location or expanding the cases/deaths data to all English-speaking countries is an opportunity for future work.

Since we used pre-curated datasets, we simulated a stream process so data is pulled in over time as if it was coming directly from Twitter and a real-time COVID-19 data source. For further information see Section 4.5.

# 4 DESIGN AND IMPLEMENTATION

This section details how we used the data to provide a solution to the proposed problem. The models built and the architecture used are described.

## 4.1 Tweet Re-hydration

Through our data investigation, we found that Twitter's Developer terms do not allow actual tweets and their metadata to be published. Instead "dehydrated" tweets that only contain unique tweet IDs are published [26]. In order to access the tweets and the metadata for the COVID-19 Tweets Dataset, we had to hydrate the set of tweets using Twarc [29]. Twarc is an open source command line tool and Python library developed by DocNow, "a tool and a community developed around supporting the ethical collection, use, and preservation of social media content" [4]. When hydrating a tweet ID using Twarc, Twarc utilizes Twitter's API to look up the unique tweet IDs and return a JSON file containing the tweet metadata. The JSON files contain extensive information such as tweet text content, user name, user location, number or user followers/following, date/time, retweet status, etc. (for a full list visit the web page here).

## 4.2 Sampling

As mentioned in Section 3 the COVID-19 Tweets Dataset [16] [17] contains over 1 billion tweets (1,136,037,775) [16] [17]. For this analysis, we sampled from 923,605,316 tweets from March 19, 2020 - January 13, 2021. We did not incorporate the entire dataset into our analysis because of the space required on a local machine. A tweet allows up to 280 characters. If we assume each tweet is the max length, then re-hydrating all of the tweets would consume approximately 260 gigabytes of space on a local machine. Creating a stream with this amount of data would cost a significant amount of money in AWS. As a result, we chose to sample approximately 3,700 tweets from each day between March 19, 2020 and January 13, 2021 (300 days). From these samples, we saved the tweet ID and sentiment (from the dataset), text, location, keywords that appeared in the text or in the quote tweeted text, date, followers count, and following count. Of the 1,110,802 sampled tweets, 914,829 tweets contained keywords in the plain text. The tweets that did not have keywords in the main text (195,973 tweets) were quote tweets and were excluded from this analysis.

## 4.3 Time Series Analysis

We hypothesized that there may be a correlation between COVID-19 related tweet sentiment and COVID-19 cases and/or deaths. For example, a more pessimistic public attitude towards COVID-19 could be correlated with an increase in COVID-19 cases, or perhaps more positive COVID-19 sentiment (e.g. enthusiasm about the vaccine) could be correlated with fewer COVID-19 cases. So we attempted to develop a predictive model using tweet sentiment to determine how many COVID-19 cases or deaths may occur during a particular day in the future.

Exploratory data analysis (EDA) showed evidence that average tweet sentiment is correlated with the number of cases and deaths (see Figure 2). An additive decomposition of daily COVID-19 cases results in an approximately weekly seasonality (see Figure 3). The same is true for the number of daily COVID-19 deaths and the daily average tweet sentiment. We implemented an Exponential Smoothing (ETS) time series model to predict the number of COVID-19 cases using the daily average tweet sentiment as an exogenous variable. The same was done for COVID-19 deaths. EDA was conducted in Python, but the ETS models were implemented in R using the es() function from the smooth package [27] because it has a robust implementation for exogenous variables. ETS was chosen over other time series techniques such as ARIMA because our data is seasonal and therefore nonstationary and ETS doesn't require transformations for nonstationary data, while many others do. However, exploring other models is an opportunity for future work.

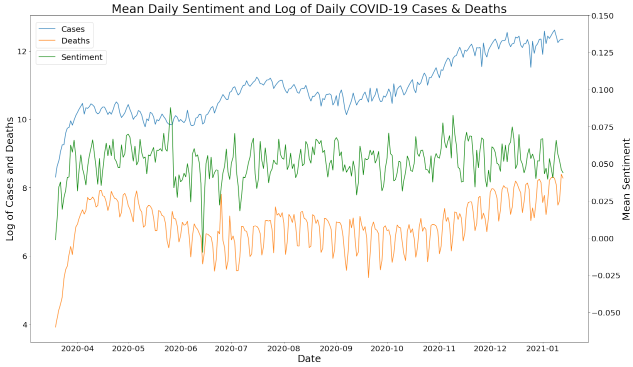


**Figure 2: Daily average sentiment increases and decreases in a similar pattern as the log of daily COVID-19 cases and the log of daily deaths due to COVID-19.**

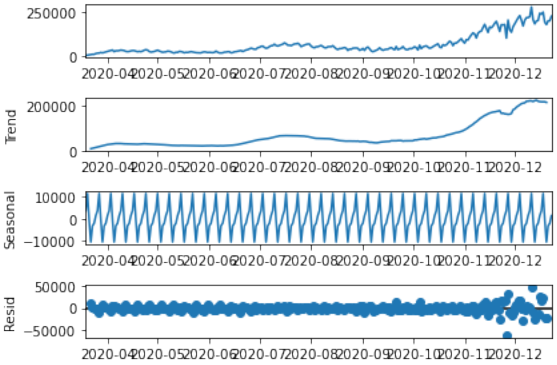


**Figure 3: Additive decomposition of daily COVID-19 cases results in an approximately weekly seasonality.**

First the daily cases data were split into training and testing sets, where the training set consisted of all but the last 14 days of data and the testing set consisted of the last 14 days of data. Then a model was fit on the training data with a frequency of 7 using the es() function, which selects the model with the lowest information criterion. The model selected was ETSX(M,Md,M), meaning that the error, trend, and seasonality were all multiplicative and the trend was damped (see [14] for more information on ETS models). This model was used to forecast the number of COVID-19 cases for a two-week period. Visually, the forecast appears to perform well, as shown in Figure 4. (It should be noted that our data did not span a full year, which may have caused some time scale issues in the model when setting the frequency. The x axis in Figure 4 has been manually set to reflect the appropriate time scale, but this may have introduced some inaccuracy. Recreating the models with data that span more than a year's time will be necessary for future work.) Comparing the forecast to the testing data results in a mean absolute percentage error (MAPE) of 14.3% and a relative root mean squared error (rRMSE) of 1.043%.

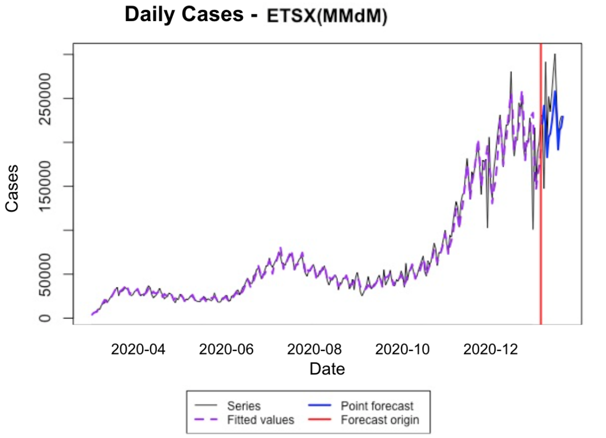


**Figure 4: Two-week forecast of daily COVID-19 cases using daily average tweet sentiment as an exogenous variable.**

The same technique was also applied to just the COVID-19 cases data (excluding the sentiment) to create a univariate model for comparison. The model selected was ETS(M,Md,M). The forecast was visually indistinguishable from the previous model and resulted in a MAPE of 14.3% and an rRMSE of 1.046%. So we concluded that including sentiment as an exogenous variable does not improve the efficacy of the forecast for daily cases.

The above techniques were applied to the daily COVID-19 deaths data with similar results. The model selected including sentiment was ETSX(M,Md,M) and the model selected excluding sentiment was ETS(M,Md,M). The model including sentiment resulted in a MAPE of 17% and an rRMSE of 0.499, while the model without sentiment resulted in a MAPE of 17% and an rRMSE of 0.5. Again, both models performed fairly well, but including sentiment did not significantly affect the performance.

These techniques were also applied to log-transformed cases and deaths data with the following results:

- Predict log of cases using sentiment as exogenous variable: Model = ETSX(M,Ad,A), MAPE = 1.3%, rRMSE = 1.111
- Predict log of cases, univariate: Model = ETS(M,Md,A), MAPE = 1.2%, rRMSE = 1.08
- Predict log of deaths using sentiment as exogenous variable: Model = ETSX(A,Ad,A), MAPE = 2.5%, rRMSE = 0.508
- Predict log of deaths, univariate: Model = ETS(A,Ad,A), MAPE = 2.5%, rRMSE = 0.507

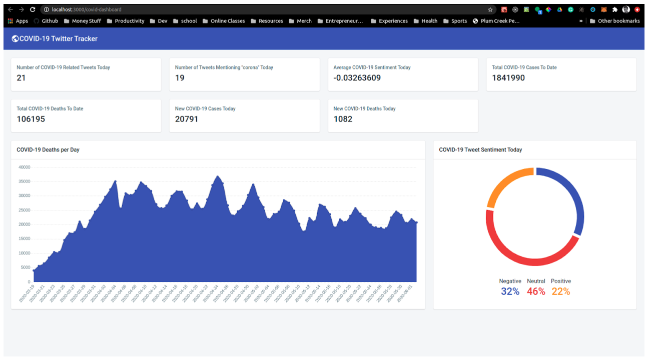
The MAPE, which is scale-independent, shows that prediction accuracy increased dramatically compared to the non-transformed data, however including sentiment still had little to no effect.

We did not end up including predictions on the dashboard because including sentiment did not improve the models and we felt that including just univariate predictive models would be less relevant to the purpose of the dashboard. In addition, including predictions would require using EMR in our architecture, which was the original plan, but when we set up an EMR cluster we experienced pricing issues. Including a predictive model on the dashboard is an opportunity for future work.

## 4.4 Dashboard

To complete our analysis, we assumed that the sentiment provided in the dataset was correct. Our main goal for our dashboard was to create a convenient and easy way to track how social media (Twitter) can influence COVID-19 cases and deaths. In order to accomplish this, we included the following items on our dashboard:

- COVID-19 deaths in real-time over time
- Number of COVID-19 related tweets in real time
- Average tweet real-time sentiment
- Total COVID-19 cases in real-time
- Total COVID-19 deaths in real-time
- New COVID-19 cases in real-time
- New COVID-19 deaths in real-time
- Distribution of tweet sentiment in real-time
- Number tweets mentioning "corona" in real-time



COVID-19 Twitter Tracker

Number of COVID-19 Related Tweets Today
21

Number of Tweets Mentioning "corona" Today
19

Average COVID-19 Sentiment Today
-0.03263609

Total COVID-19 Cases To Date
1841990

Total COVID-19 Deaths To Date
106195

New COVID-19 Cases Today
20791

New COVID-19 Deaths Today
1082

**Figure 5: Dashboard created**

We ultimately decided on including these figures into our dashboard because we thought that they would be the most helpful for helping local and national governments address COVID-19 and other events that could have a large impact on society. In particular, we thought that seeing COVID-19 deaths over time updating in real-time would be helpful in seeing how the pandemic is progressing and if deaths are higher today than they were in the past. Next, we thought that including the number of COVID-19 related tweets in real-time would be helpful because it would show the size of the sample from which statistics such as average tweet sentiment, tweet sentiment distribution, and number of tweets mentioning "corona" are being calculated. We also decided to include average tweet sentiment and distribution of tweet sentiment in real-time because although average tweet sentiment provides an idea of what the sentiment is like for a given day, it can be skewed, so seeing the actual distribution can be very helpful. As we streamed in the data, we noticed that the sentiment for the tweets on a given day tended to be slightly more positive than negative. This was surprising, as we assumed that sentiment would tend to be negative when cases were rapidly increasing. However, this information will be very useful for healthcare officials since they will be able to see how tweet sentiment for a given day impacts cases and deaths. Finally, we included statistics such as total COVID-19 cases and deaths in real-time and new COVID-19 cases and deaths in real-time so that someone using this dashboard would be able to see the bigger picture, i.e. how many cases/deaths have occurred so far and out of that many cases/deaths how many have occurred today. Overall, we think that this dashboard is really helpful for government agencies and could be easily modified to incorporate other statistics that they think would be helpful.

## 4.5 Architecture

In order to accomplish the goals detailed above, we implemented a scalable big data system. In particular, we used Python, SQL, Javascript and AWS. Python was used for processing, prototyping, and creating figures for our dashboard. It was also used for exploratory data analysis and time series analysis. Within Athena, we used SQL (Athena uses Presto with ANSI SQL) to query the Twitter and COVID-19 data. We then used serverless Javascript functions to make calls to Athena-Express in order to access data for our dashboard. Within AWS, we used five primary services: Kinesis, S3, Athena, API gateway and Lambda. See Figure 6 for an overview of the architecture.

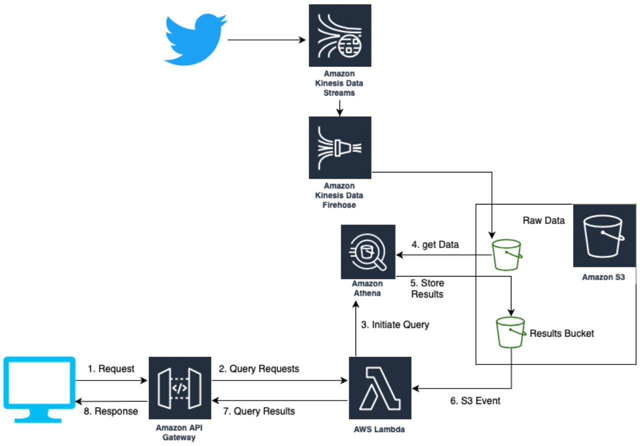


**Figure 6: Architecture**

Kinesis is Amazon Web Service's stream processing tool that "makes it easy to collect, process, and analyze real-time, streaming data so [we] can get timely insights and react quickly to new information" [30], such as incoming streaming tweets. Due to prohibitive costs associated with the amount of tweets available, we simulated streaming by using an existing data-set of COVID-19 related tweets that will continuously send tweets to the Kinesis data stream. Amazon S3 is the AWS storage service ("Simple Storage Service"). S3 can be used "to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics" [10]. Finally, Amazon Athena is a query service used to analyze data in Amazon S3 using standard SQL. Since Athena is serverless, we did not have to manage an infrastructure. [13]

These services were all combined to form a clean data pipeline. The streaming process is simulated using the rehydrated and sampled Twitter streaming data (tweets and sentiment) and the COVID-19 cases and COVID-19 deaths data. Pythons boto3 library is an AWS SDK (software development kit) that is used to simulate the streaming process. Within Python, we wrote a function that takes in a dataframe (read from the CSV's of the sampled data and NYT COVID-19 data), a stream name, and a boto3 Kinesis client and simulates a streaming process by sending data into a Kinesis data stream. Within Kinesis, we developed three input streams titled "twitter-beta-input" (data from the COVID-19 dataset), "twitter-beta-input-tweets" (data from the sampled tweet dataset), and "twitter-beta-output". We then created a couple of Kinesis Firehouse delivery streams titled "covid_data_delivery_stream" (receives data from "twitter-beta-input|) and "tweet_data_delivery_stream" (receives data from "twitter-beta-input-tweets"). Both delivery streams pause for 1 minute before sending data into S3. The delivery streams send the data into two respective S3 buckets with prefixes ("input-covid" and "error-covid" and "input-tweets" and "error-tweets"). Kinesis then sends raw tweets to S3 where they are stored.

Our client end application was a web dashboard built in React and deployed to GitHub pages. On page load, the site would query an AWS API gateway proxy which itself pointed to a single Lambda function. The proxy allowed routing of all calls through a single base URL without needing to delineate each call through its own API gateway route to its own lambda function. The lambda function then made use of an internal router to apply the correct logic based on the path appended to the base URL.

The Lambda function in turn made use of Athena-Express, a NodeJS package used for making asynchronous AWS Athena calls from JavaScript. AWS' Lambda service allows for a server-less deployment with concurrent scaling. Each additional call to the API spins up its own Lambda instance which can run concurrently avoiding the cost of compute when it is not required. This makes the deployment highly scalable, especially at smaller scales where a constant server is not needed.

To make our raw data within S3 queryable, we first needed to setup a database with Athena and define the schema for two tables (one for the tweet data and one for the COVID-19 data). When defining the schema, we needed to specify the data types of each of the incoming columns. The data type for each column was clear except for the "date" column. Intuitively, it should have come in as a date or date-time, yet due to formatting, it needed to be brought in as a string. When writing queries, we needed to cast that column to a date-time data type. By applying this schema to our raw data, we could make SQL queries on our S3 database.

In order to determine which queries were necessary for our dashboard, we prototyped the queries in the Athena console using the newly defined databases and data tables. We wrote queries for the count of total tweets on the current day, for the count of total tweets with "corona" as a key word on the current day, for the average sentiment on the current day, for the top ten key words that day, for the cumulative number of COVID-19 cases/deaths as of that day, for the number of COVID-19 cases/deaths on the current day, and the total count of positive (0<), negative (<0) and neutral (=0) sentiment. Once the queries were tested within the Athena console, they were copied into the lambda function containing Athena-Express.

Athena-Express'[1] SQL queries would store the query result in a separate s3 folder as JSON objects. This triggered a return to the original Athena-Express call within Lambda which then returned the data back through the API gateway to the React dashboard where it was displayed.

# 5 EVALUATION

## 5.1 Performance

Our three key performance indicators (KPIs) as outlined in our proposal were latency, throughput and fault tolerance. AWS S3 has built-in fault tolerance through the use of data redundancy. Each object is stored more than once in separate servers so that if one server were to lose data or go down, the network would still maintain that data.

The potential throughput of this system is limited by the Kinesis data streams. Currently our system is running with 2 data streams at one shard each for a total throughput of 1 MB/s. Each data stream has the capacity to run 5,000 shards for a total throughput of 5 GB/s which far outpaces streaming every single COVID-19 related tweet in a one month period. Our average tweet size was 0.000000291 GBs and the average number of COVID-19 related tweets in a month is approximately 80,600,000. This calculates to about 23.4546 GB of data a month to stream all possible COVID-19 tweets. This is approximately 0.00000876 GB/s to stream every tweet into our system not accounting for variation in tweet rates at different times. This is well within Kinesis data streams data streaming throughput.

Latency was well within the bounds of acceptable for our application as well. AWS Cloudwatch recorded the Kinesis data streams latency as 11 milliseconds per tweet stream record and 40 seconds per COVID-19 daily data record. At this latency for the aforementioned average tweets per month, it still only takes 886,600 seconds to upload all tweets within a month which is significantly lower that the total seconds in a month, meaning our data would never outpace our latency. Our Athena SQL queries took on average 2.3 seconds to complete, allowing our client side application to access the data very quickly.

Our preliminary data pipeline was focused on passing data through and storing all raw data from which SQL queries could be made to fetch required. A stronger and less resource-intensive design would be to only store raw data temporarily and feed that into a data processing/analyzing layer such as Apache Spark running on an AWS EMR instance. This would then store only relevant data in S3 and avoid excess storage costs. This would also improve client latency as the SQL queries would not need to run over the entirety of the raw data. This data processing layer would also allow us to run more complex models in real time such as our time series analysis or a proposed sentiment analysis network.

This was our original design but due to both cost and time constraints we could not finish this element of our pipeline.

## 5.2 Pricing

To calculate the pricing for a larger scale project, we assumed that there would be 80,600,000 tweets per month based on our dataset's reported 2,600,000 tweets per day average [16] [17]. We also assumed that the monthly cost would be based on 31 days. In addition, we found the following prices according to Amazon [10] [30]:

- Kinesis Data Stream:
  For 1 Shard: 1 MB/sec=0.06 GB/min=2678.4 GB/month
  For 1 shard: $0.015 per hr or $11.16 per month
- Kinesis Firehose:
  500 TB per month for $0.029 per GB
  Next 1.5 PB per month for $0.025 per GB
  Over 3 PB per month for $ 0.02 per GB
- S3
  50 TB per month for $0.023 per GB
  Next 450 TB per month for $0.022 per GB
  Over 500 TB per month for $0.011 GB
  $0.005 per 1000 PUT requests
- Tweets
  2,600,000 tweets a day at .000290909 MB = .000000290909 GB
  23.4472654 GB per month of tweet data

Using the data above, we derived the following equation for our price of processing every COVID-19 related tweet in a month:

$$CostperGB = (2 * 10.95) + (0.029 + 0.023) * 23.4472654 + (80600000/1000) * 0.005$$

This resulted in about **$426** in costs each month.

If we were to implement our dashboard at scale, we would process approximately $1.55 * 10^{10}$ tweets per month, which would be about 4510.5 GB. Using similar logic as above, we determined that in this case the monthly cost would be approximately $81,923.

# 6 DISCUSSION

## 6.1 Preliminary Results

Before we streamed data into our dashboard, we conducted exploratory data analysis. While conducting exploratory data analysis we found that that the total number of COVID-19 related tweets has steadily increased from the start of the pandemic (this is attributed to the dataset tracking more keywords over time) which can be seen in Section 7.

As mentioned above, we sampled 914,829 tweets that included keywords. Of these tweets, 623,920 tweets (roughly 68%) had something in the "location" field. However, some of these did not contain an appropriate or correct location which strongly influence us to not include location. We also discovered that approximately 4% contain the key words "vaccine" and/or "mask". The limited number of tweets that contain key words such as masks and vaccine convinced us to not include this in our analysis. However, these could be included in the dashboard in the future once more tweets are streamed that contain these words.

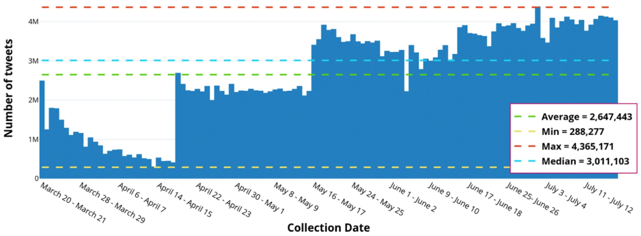


**Figure 7: Tweet growth over time [17]**

We also decided to investigate things like what was the average daily sentiment, top ten number of key words, number of cases over time, and number of deaths over time. As mentioned above, keywords were added to the dataset over time, so it is not surprising that the histogram for number of key words was heavily skewed. We found that the top five key words in the dataset were: covid, corona, covid-19, pandemic, and coronavirus. As shown in Section 8, "covid" has the highest frequency and occurs more than twice as many times as "corona."

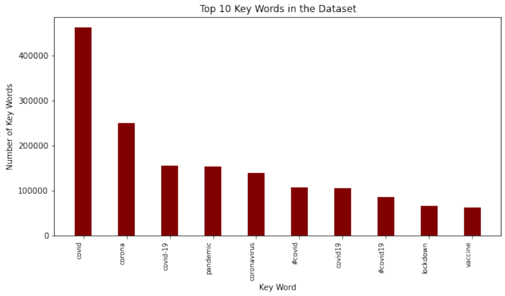


**Figure 8: Top 10 Number of Key Words**

Although we didn't conduct an analysis about masks, we decided to look at mask key words to see if there were any words that occurred more frequently than others. We found that "PPE" occurred the most and "wear a mask" never occurred. It would be interesting to look at 2021 YTD data to see if this has changed and more people are tweeting "wear a mask" instead of "PPE."

The next item that we explored using the Twitter dataset was the average daily sentiment. Figure 10 shows a scatterplot of daily average sentiment over time. Based on this plot it appears, that almost all of the dates had an average daily positive sentiment. The most negative sentiment occurred in the middle of July. Since our Twitter data was only limited to English tweets and not by location, we cannot say what was happening during that time that could have caused this drop because COVID-19 was being addressed

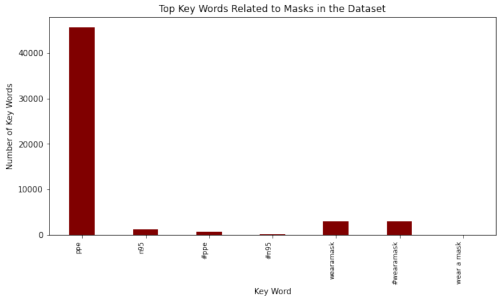


**Figure 9: Top Key Words Related to Masks**

differently throughout the world at this time. Not surprising (given Figure 10), the most common tweet sentiment was 0, i.e. neutral.

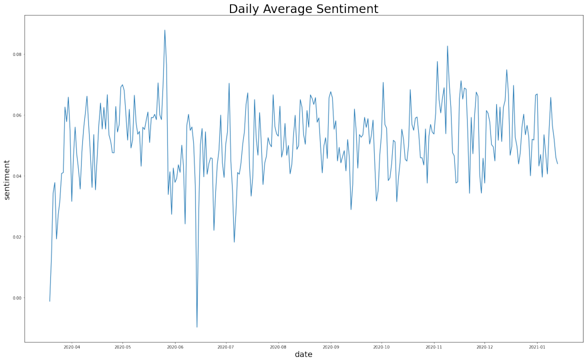


**Figure 10: Average Daily Sentiment**

Our data saw an increase in cases and deaths from March 19th, 2020 to January 13th, 2021. As of March 19th, the NYT had recorded one case and zero deaths. By January 13th, there was 23,133,938 COVID-19 cases and 384,824 COVID-19 deaths.

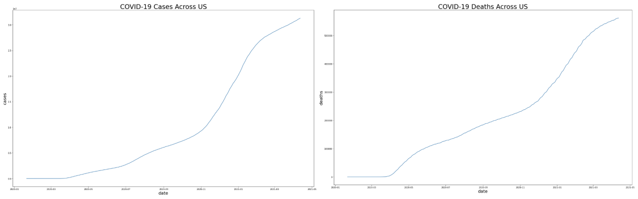


**Figure 11: US Cases and Deaths Over Time**

## 6.2 Changes

This section contains an overview of the major changes since the project proposal.

*6.2.1 Objectives.* Originally, we planned to analyze each tweet's location, content, and date/time to help determine correlation between cases and mask/vaccine sentiment in a certain location. However, with more research we learned that only 1-2% of tweets are geo-tagged [8]. Alternative methods for determining tweet location exist but didn't appear to be good options for this project. Parsing keywords for mentions of location is a possible alternative method, but this is not very accurate [8]. It's also possible to pull location from user profiles to gather location information, but many users either leave the field empty or don't include a valid location (the field is a free-form character field) [5]. Since the tweets in our dataset did not include sufficient location information, we analyzed general trends of sentiment in relation to cases over time, independent of location.

*6.2.2 Data.* We used a public COVID-19 related tweet dataset instead of pulling directly from Twitter. This also allowed us to use the sentiment score from the dataset instead of implementing our own sentiment analysis model. (See Section 3.)

*6.2.3 Models.* We also changed from a spatially oriented model to a temporal model as there was not enough location data, and for our statistical analysis we looked at overall COVID-19 sentiment rather than specific mask and/or vaccine sentiment as there was an insufficient percentage of tweets relating to these topics.

*6.2.4 Sentiment.* Our original plan was to use a publicly available pre-trained language model to determine tweet sentiment. However, the COVID-19 Tweets Dataset includes a sentiment score for each tweet in the dataset (see Section 3 for more details). We've chosen to use this instead of training our own model because it simplifies our project. However, if we were to implement this project at scale and stream tweets directly from Twitter we would need to implement our own sentiment model. The remainder of this section describes our original plan for doing so, which could be relevant for future work.

We originally planned to use a publicly available pre-trained language model such as BERT or GPT which is made available publicly by Tensorflow and Pytorch. Both pre-training methods use multilayered transformer decoder and encoder architecture to build vector representations of the words in a corpus taking into account the underlying words and their contextual meaning. We planned to pass tweets through these pre-trained models and pass the final transformer block's activation into further layers.

The original plan was to determine a tweet's sentiment in counties/states in relation to masks, COVID-19, and the COVID-19 vaccine. In particular, we intended to use a common technique in natural language processing (NLP) called sentiment analysis. In order to do so, we needed some form of pre-classified data such as a list of classified (sentiment) words from SentiWordNet in the NLTK library in Python as well as the pre-trained GPT language model. This dictionary would have classified a set number of words as positive or negative. These dictionaries could then be used to determine the sentiment of a sentence by determining the probability of a word having positive or negative sentiment and then combining the probabilities of positive and the probabilities of negative and choosing the maximum.

*6.2.5 Architecture.* AWS has cost our team money, so we waited until April (the start of a new month resets the free usage allowance) to process data. Rather than look at all the tweets we sampled approximately 1 million tweets as a proof of concept, since all of the data wouldn't fit on our machines and using AWS to stream all of the data would cost a lot of money. We simulated streaming using Kinesis data stream and Kinesis delivery stream with Python's boto3 library to create a prototype that would function similarly to pulling tweets directly from Twitter. (See Section 4.5.)

We chose to exclude EMR from our architecture as we did not have the resources (financial, mostly) to use this service. We intended to implement our own sentiment analysis model and to include our our time series analysis work on our dashboard. In the future with further resources, we hope to include these on our React dashboard.

## 6.3 Future Work

While this project effectively implemented a live dashboard, there are a number of areas to explore further and implement. One area that we did not have sufficient time or budget for is including the time series predictions on the COVID-19 Twitter Tracker dashboard. Given additional resources, including the time series analysis performed in Python on this paper would be desirable. This can be accomplished through PySpark in AWS EMR.

If a time series model is included in the future, opportunities exist for making the model more robust, including verifying the accuracy of the forecasts using cross-validation and exploring other time series models such as SARIMAX (Seasonal Autoregressive Integrated Moving Average Exogenous model).

Additionally, it would be interesting to implement/recreate the machine learning/ natural language processing model of sentiment analysis included in the tweet dataset. The dataset by R. Lamsal[17] (where we retrieved our Twitter data) included sentiment for each tweet, however the accompanying paper did not include an in-depth explanation of how the sentiment was derived. Alternatively, we could create our own sentiment model, as described in Section 6.2.4.

Another area to extend this work is by including some spatial analysis. A small subset of the tweets in the dataset contained real locations (i.e. were geotagged). If we chose to stream data directly from Twitter's API, we could collect a larger amount of tweets that included geo information. Extending the work explained in this paper into the spatial realm could help track sentiment by location and COVID-19 spread throughout the United States. Although our temporal predictive models did not improve when sentiment was included as a variable, we hypothesize that there could still be a spatial correlation between sentiment and COVID-19 cases.

This project can also be extended to a global scale. Currently, both the COVID-19 dataset and the subsetted tweet dataset contain United States data strictly. Not only can this data be broken into a more granular dataset (e.g. looking at both datasets on a state/county level) but it can also be extended globally.

These more complex models would need to be implemented in a data processing layer such as spark on an AWS EMR instance as mentioned in the previous evaluation section.

## 6.4 Reflection

Overall this project presented our team with an excellent opportunity to implement a prototype for big data architecture from scratch. We were able to successfully implement a scalable streaming big data system using a number of AWS products. We found that simulating streaming data is quite simple and that connecting each component within AWS is where the challenge of our architecture lies.

We were also able to successfully create a React dashboard that updated whenever new data was received. This was done by connecting an API Gateway to a React Dashboard. This Dashboard allowed us to display and analyzed results from big data. Although our dashboard only incorporated statistics for COVID-19 cases, deaths, and tweet sentiment, future users will be able to adjust the dashboard to include additional statistics/measures that better fit their needs. We also learned that big data has its own nuances that need to be addressed before analyzing, i.e. messy data that is hard to analyze.

Finally, we learned that there are limited resources available within a certain price point. This limitation forced us to design our project around this price point. However, we did conduct research that would allow us to implement this project using a larger budget. For example, we learned about EMR's capabilities and how this would allow us to implement a machine learning model into our dashboard. We also learned how much money we would need to spend to be able to implement this project on a larger scale which will be important for future big data projects.

# 7 CONCLUSION

In conclusion, the COVID-19 Twitter Tracker dashboard is useful for real time analysis of COVID-19. It helps individuals understand the current COVID-19 climate, conveniently displays relevant information for public health officials, and can be used to analyze information pertaining to COVID-19. Our selected architecture is sufficient for our proposed problem. We effectively created a prototype for a real time COVID-19 dashboard. Additionally, in this paper we determined that sentiment is not a useful predictor for the predictive model for COVID-19 cases and deaths, but univariate time series models for cases and deaths perform well.

# REFERENCES

[1] Athena-express. https://www.npmjs.com/package/athena-express.
[2] Cloudwatch. Available at https://aws.amazon.com/cloudwatch/.
[3] Covid-19 in colorado. https://cdphe.colorado.gov/.
[4] Documenting the now. https://www.docnow.io/.
[5] Filtering tweets by location. https://developer.twitter.com/en/docs/tutorials/filtering-tweets-by-location.
[6] Stream tweets in real-time docs twitter developer. https://developer.twitter.com/en/docs/tutorials/stream-tweets-in-real-time.
[7] Tutorial:quickstart.
[8] Tweet geospatial metadata. https://developer.twitter.com/en/docs/tutorials/tweet-geo-metadata.
[9] Tweet object | twitter developer. https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/tweet.
[10] S3, 2002. Available at https://aws.amazon.com/s3/.
[11] Cinelli, M., Morales, G. D. F., Galeazzi, A., Quattrociocchi, W., and Starnini, M. The echo chamber effect on social media, Mar 2021.
[12] Gilmour, J. B., Lui, A. W., and Briggs, D. C. Emr, 1986. Available at https://aws.amazon.com/emr/.
[13] Hoena, B. A., and Bowman, L. Athena, 2003. Available at https://aws.amazon.com/athena/.
[14] Hyndman, R. J., and Athanasopoulos, G. *Forecasting: principles and practice*, 2nd ed. OTexts: Melbourne, Australia, 2018. OTexts.com/fpp2. Accessed on 4/19/2021.
[15] Kaila, D. P., Prasad, D. A., et al. Informational flow on twitter–corona virus outbreak–topic modelling approach. *International Journal of Advanced Research in Engineering and Technology (IJARET) 11*, 3 (2020).
[16] Lamsal, R. Coronavirus (covid-19) tweets dataset, 2020.
[17] Lamsal, R. Design and analysis of a large-scale covid-19 tweets dataset. *Applied Intelligence* (2020), 1–15.

[18] Lin, C. How to build a real-time twitter analysis using big data tools, Dec 2020.

[19] Maragakis, L. L. Coronavirus, social and physical distancing and self-quarantine.

[20] O'Leary, D. E., and Storey, V. C. A google–wikipedia–twitter model as a leading indicator of the numbers of coronavirus deaths. *Intelligent Systems in Accounting, Finance and Management 27*, 3 (2020), 151–158.

[21] Qin, L., Sun, Q., Wang, Y., Wu, K.-F., Chen, M., Shia, B.-C., and Wu, S.-Y. Prediction of number of cases of 2019 novel coronavirus (covid-19) using social media search index. *International journal of environmental research and public health 17*, 7 (2020), 2365.

[22] Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training.

[23] Ritchie, R., and data: Hannah. Coronavirus (covid-19) cases - statistics and research.

[24] Sanders, A. C., White, R. C., Severson, L. S., Ma, R., McQueen, R., Paulo, H. C. A., Zhang, Y., Erickson, J. S., and Bennett, K. P. Unmasking the conversation on masks: Natural language processing for topical sentiment analysis of covid-19 twitter discourse, 2020.

[25] Shrestha, S. Fault tolerance & redundant system with seamless integration to development on aws, Nov 2020.

[26] Song, D. How to "hydrate" a tweetset?, Mar 2021.

[27] Svetunkov, I. Exponential smoothing. https://cran.r-project.org/web/packages/smooth/vignettes/es.html. Accessed on 2021-04-15.

[28] Times, T. N. Y. Coronavirus in the u.s.: Latest map and case count, Mar 2020.

[29] UNLV. Twitter data collection using twarc, Mar 2021.

[30] Walmsley, G., and Bie, J. Kinesis. Available at https://aws.amazon.com/kinesis/.

# APPENDIX

*Honor code pledge*
*On my honor, as a University of Colorado Boulder student, I have neither given nor received unauthorized assistance.*

## Author contributions

- Angela Folz researched related work and researched the feasibility of filtering tweets by location. She also researched pre-curated COVID-19 databases and selected the COVID-19 Tweets database that was ultimately used. She calculated the daily cases and deaths from the original data (which only contained cumulative cases and deaths) and trimmed the NYT data to match the time period of the tweet data. Angela also developed the time series models. (She also spent way too long setting up Spark, which we didn't end up using.)
- Lucas Laughlin handled the development of the client side application and architecture as well as contributing to the Kinesis data stream simulation using python. The client side application consisted of developing and deploying a React application and connecting it to the database through AWS API gateway and Lambda services. Lucas also calculated how the pricing would scale.
- Ksenia Lepikhina setup the original AWS account (later shut down due to increasing costs) and she setup one of the developer Twitter accounts used to rehydrate tweets. She also setup Spark and helped Angela and Julie set that up as well (Spark was not used in the end). For each day of data in the dataset, Ksenia sampled from the COVID-19 tweet dataset, rehydrated each sampled tweet and compiled a CSV to share with our team and to simulate streaming data. Ksenia and Lucas then wrote a Python script that simulated a streaming process. She then built a Kinesis input data stream and delivery stream for each data set and streamed that data into s3. She then worked on connecting Athena to S3, created a schema and made the data queryable within the Athena console.
- Julie Matthias researched Twitter developer accounts and if this would be sufficient for our project needs. She created a Twitter developer account, learned how to rehydrate tweets, and rehydrated some data. Although Spark was not used in this project, she researched how to set it up and set it up. Julie found COVID-19 datasets from the CDPHE and NYT that contained daily data about COVID-19 cases and deaths. We only used the NYT dataset for this project. Julie conducted exploratory data analysis on the COVID-19 NYT dataset and the Twitter dataset. Julie researched SQL queries to use in Athena, researched AWS, and found latency statistics that AWS provided. She also participated in investigating modifying the dashboard template. Julie also confirmed Lucas' pricing calculations.
- All four authors contributed to writing the reports and creating and delivering the presentations.

## Github

Here is the link to the Github where our dashboard can be viewed: https://github.com/LucasLaughlin/covid-dashboard