What do people care about in the pandemic?

Rui Chen

Introduction

The outbreak was first detected in December 2019 in Wuhan, Hubei Province, People's Republic of China, and then spread rapidly to many countries around the world in early 2020, gradually turning into a global pandemic. I hope to understand what people are concerned about in this pandemic that is affecting the world by understanding exactly what people are saying when they talk about the pandemic.

Around the world, citizens in every country use Twitter to express their private thoughts, and Twitter is a fast and easy way to spread the word, as seen in the U.S. election. So I will use this platform to see what people care about in the pandemic by analyzing the tweets they post.

In general, my final project is based on tweets posted during the outbreak. I am focusing on two main things: first, what were people most concerned about during this period? Second, as manual tagging has been done, can classification and clustering algorithms perform well on this problem?

Data and Methods

The data for this project was collected from publicly available datasets on the Kaggle platform.

The tweets have been pulled from Twitter and manual tagging has been done then. The names and usernames have been given codes to avoid any privacy concerns. It contains the following six columns. 1. Location 2. Tweet At 3. Original Tweet 4. Label 5. Name 6. ScreenName. Of particular interest to me is the "sentiment" column, which contains five sentiment categories ranging from Extremely Negative to Extremely Positive. Specifically, the “sentiment” column contains Positive, Negative, Neutral, Extremely Positive, Extremely Negative. Chronologically, the tweets come mainly from March and April 2020. On March 12, the WHO officially declared COVID-19 outbreak a pandemic. Therefore, although the time period covered by this dataset is not very long, it provides me with a look at a key time point of the pandemic. Although people from different countries around the world are free to tweet their thoughts, I was limited by my technical ability to filter only English tweets as my analysis sample. In addition, due to the limitations of my personal computer's computing power, I ended up extracting 5,000 random tweets for my study.

Also, I was particularly curious about how the algorithms I learned this semester would perform on my dataset. Since the creators of this dataset put a lot of effort into classifying the

sentiment of tweets, one question I was particularly curious about was how well different classification algorithms can perform in classifying the sentiment of tweets. How do classification and clustering algorithms perform in the face of these tweets, which may have subtle emotions? So in addition to exploring what people care about most during this time, I want to know how classification and clustering algorithms perform when applied to these tweets.

Network

In this section, I explore the representation and analysis of semantic networks and networks of influence. I can analyze these networks to understand the structure of connections between words and the dynamics of how their meanings flow through a discourse system. I can define links between words as a function of their co-presence within a document, chapter, paragraph, sentence, noun phrase or continuous bag of words. I can also define links as a function of interdependent words in directed dependency parsing, or links between extracted subjects, verbs and objects, or links between nouns and adjectives modifying them. Linking words into networks or discrete topologies allows me to utilize network analysis metrics such as centrality.

Texts represent parts of in social games, so I can learn more about the relationships between people, groups, and organizations that influence each other through the analysis of my tweets. Connecting the tweets posted in the pandemic into a network or discrete topology allows me to take advantage of a wide range of metrics and models developed for network analysis. These methods allow me to visually examine the structure and complex social relationships of these tweets. In addition, I can link explicit social interactions to semantic networks to better understand how people act in the world.

In this section, I'm concerned about a few things: What are the most important keywords in these tweets? Is there anything else that people care about besides "coronavirus"? Do the results partly reflect the perception of the outbreak at the time? I will explore these questions based on different metrics and try to answer them.

I started with an initial visualization of the different tweets. I plot the bipartite network with a quick spring layout. Please see figure 1. It was clear that without filtering, this would not lead to much meaningful information. After making some adjustments, the visualization started to gradually become meaningful. After some adjustments, I drawed the graph with high and low weight edges distinguished. What you may wonder after seeing this graph is why 60 and 96 are on the outside while 78 and 28 are in the center. This is because 60 has only 30 words, while 28, a document, has 270 words, almost approaching the maximum word limit that can be sent in Twitter. Obviously, in this simple network of documents, those with the most words will undoubtedly be placed at the very center.

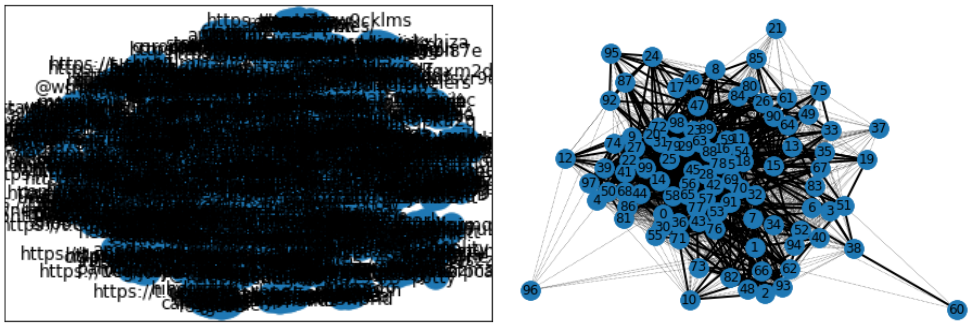


Figure 1: Quick spring layout.

Now, why not look at the word-to-word network by documents. After reducing the word count to a manageable size, dropping all the edges with weight below 25 and all the isolates, I get another figure. Please see figure 2. It's very straightforward that some central words include coronavirus, store, food. But this is not enough and I will dig into the network further later. But now, let's first look at global statistics, like the density of a network, defined as the number of actual edges divided by the total number of possible edges. Before we take a closer look at the details, we can get some idea of the overall picture. The density of the network is 0.04621337755666114. It is defined as the number of actual edges divided by the total number of possible edges. I can also calculate the average degree per node: 6.192592592592592. The average distance between any two nodes in the network is 5.

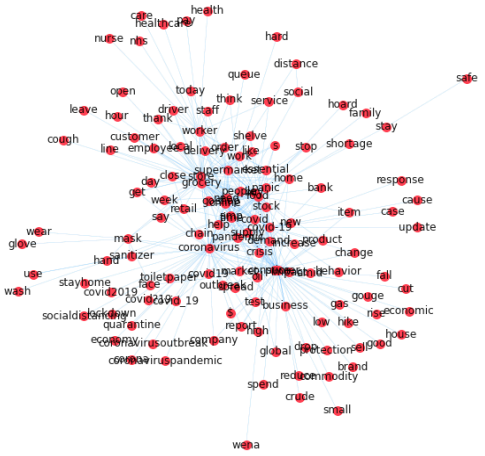


Figure 2: Simple network.

I can gain more insight into our semantic network by describing statistics about the position of words in the network. In the analysis, I color and resize nodes according to the level of centrality. The key to this part is measuring centrality. The concept of centrality is that certain nodes are more important than others in the network. The most straightforward of the various measures is "degree centrality". Its core idea is that the node with the highest number of connections is the most central. My measure normalizes the number of connections by those with the most connections. Please see figure 3. From the graph I got, the most important word in this network is of course "coronavirus". More importantly, I want to focus on what else is important to people besides this word. Other important keywords in this

graph include price, food, home, work, store, mask, hand. Next, I utilized a different measure of centrality: "betweenness centrality". Betweenness centrality distinguishes nodes that require the most shortest pathways between all other nodes in the network. High betweenness centrality nodes may not have the highest degree centrality. According to the figure 3, I would say the results are not changing much. The most important keywords include food, price, store, coronavirus. I have also explored this problem using closeness centrality. Its central tenet is the average Euclidean distance or path distance between a node and all other nodes in the network. The node with the highest distance to centrality is most likely to send the widest coverage signal to the sleep of the network. After executing the code, almost all the words in the network are colored in, so one possible problem with this visualization is that it provides too much information to my detriment of identifying the most critical ones. But I can still find the information I need in those brightest colors. The keywords I was given included price, food, stock, home, store, supermarket, work, worker, mask, help. Finally, I used eigenvector centrality as my measure, which is a way of weighting degrees based on the centrality of the people one is related to, and how much they are related to them. In fact, the results didn't change much. The most important words included: demand, supply, covid, help, stock, home, food, price, business, impact, crisis.

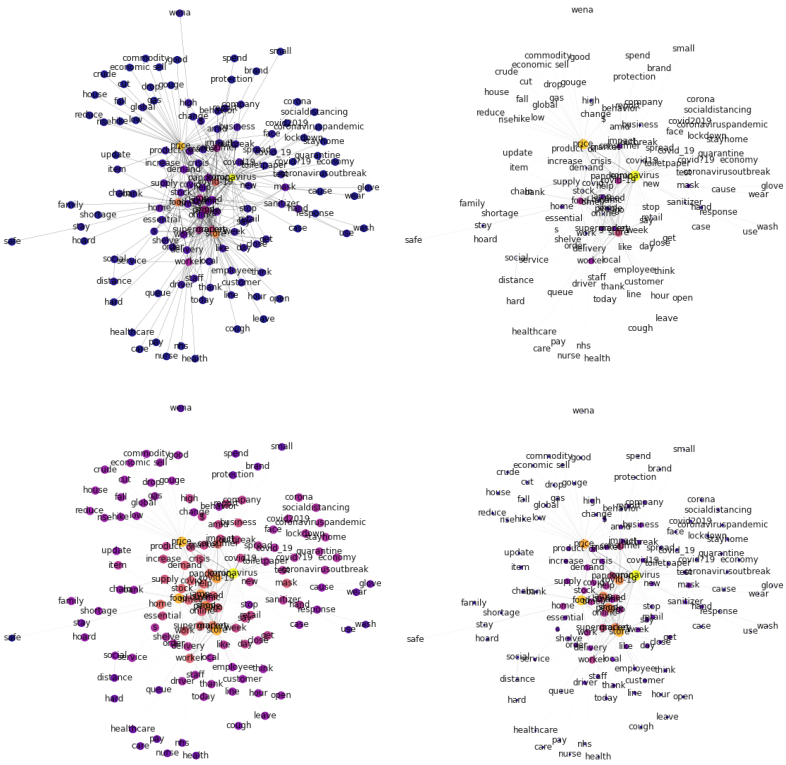


Figure 3: Centrality. First: degree centrality. Second: betweenness centrality. Third: closeness centrality. Fourth: eignvector centrality.

Each of these sub-figures in figure 3 has some differences, but more importantly, they offer some very meaningful commonalities. There is no doubt that "coronavirus" is always the most important keyword, regardless of how centrality is measured. So, given my ability to

find subgraphs in this network, I limited the keyword to "coronavirus" and focused on nodes one or two steps around the word. Now let's look at a subgraph of the network, the nodes within 1 to 2 network steps of “coronavirus”. Please see figure 4. In March and April 2020, in terms of the pandemic, users on Twitter were most concerned with such things as: consumer, price, shop, food, store, grocery. Clearly, when such a public health crisis arrives, the first concern is survival. As consumers, people seem to be anxious about how to get to the wide variety of small stores and large supermarkets to buy the food they need. However, what people are also anxious about is the price of these goods. The words closely related to the word price are "increase" and "high". The reason for this is obvious. In the midst of the pandemic, people were grabbing what they needed and didn't need, and this sudden demand far outweighed the supply of household goods at the time, so prices kept rising. People are certainly justifiably concerned about the price of the household items they need.

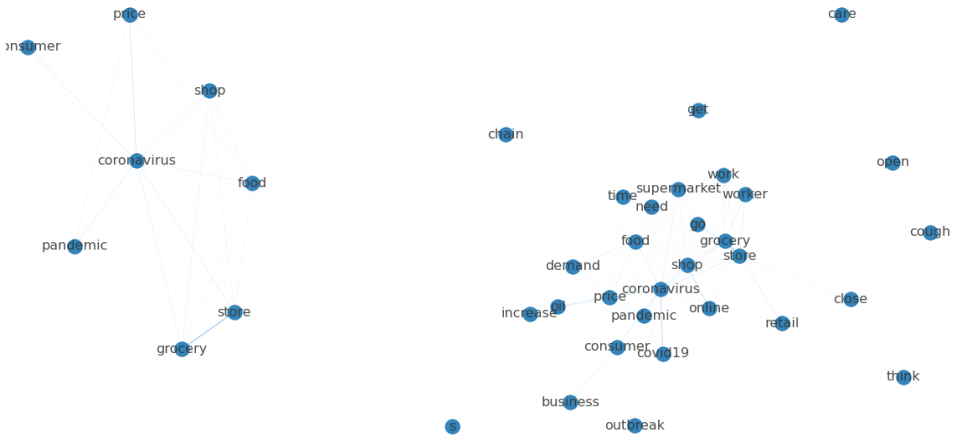


Figure 4: Left: one step of “coronavirus”. Right: two steps of “coronavirus”.

One problem with figure 4 above is that words sometimes overlap each other, which detracts from the visuals. So, next, I converted the visualization directly into a much clearer table. Please see table 1. The most important idea I learn from this table is that, in March and April, users on Twitter were most concerned about price, shop, food, store, grocery. Obviously, when a public health crisis like this comes, the first concern is survival before anything else. We can now filter our network by a centrality measure. Please see figure 5. From this figure, we can see more clearly that it is after caring about survival that people start to care about business, about hand hygiene, hand sanitizer, jobs, and workers. The interesting thing about this figure is that "stay home" and "mask" are not the most important things that people are concerned about at this time. I think this is an accurate reflection of people's attitudes towards protection measures around March 2020. There seems to be no consensus on the issues of "stay home" and "mask" at that time. But by today, there is a consensus to wear a mask to protect yourself.

Table 1: Centrality score.

Method	Keyword	Score	Method	Keyword	Score
Degree Centrality	coronavirus	0.47	Closeness Centrality	coronavirus	0.6442307692
	price	0.41		price	0.5877192982
	store	0.35		food	0.5751072961
	food	0.32		store	0.5751072961
	covid-19	0.29		covid	0.5560165975
	grocery	0.23		amp	0.5381526104
	amp	0.2		people	0.5275590551
	people	0.19		shop	0.5234375
	supermarket	0.18		grocery	0.5075757576
	shop	0.17		pandemic	0.4962962963
Method	Keyword	Score	Method	Keyword	Score
Betweenness Centrality	good	0.4701492537	Eignvector Centrality	coronavirus	0.3292773174
	covid2019	0.4104477612		food	0.2899392797
	healthcare	0.3507462687		store	0.2741101892
	protection	0.328358209		price	0.2645312995
	economy	0.2910447761		covid	0.2386993654
	low	0.2388059701		people	0.2307215429
	brand	0.2014925373		amp	0.2298740083
	house	0.1940298507		grocery	0.2255370344
	pay	0.1865671642		shop	0.2220009413
	family	0.1791044776		supermarket	0.1647037006

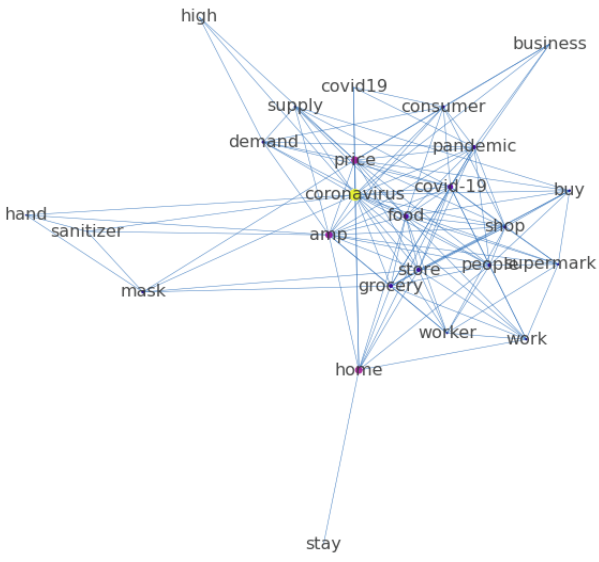


Figure 5: Filtered network by a centrality measure.

Classification

Since the creators of this dataset put a lot of effort into classifying the sentiment of tweets, one question I was particularly curious about was how well different classification algorithms can perform in classifying the sentiment of tweets. I personally sampled a sample of 200 to see if the labels in that dataset were accurate. Based on my personal observations, the categories were accurate. The samples were accurately assigned to five different categories based on the emotions they embodied: extremely negative, negative, neutral, positive, and extremely positive.

Clustering is an unsupervised learning method that helps me to steadily cluster my pandemic text data based on covariance patterns among all available text features. In contrast, classification is a supervised approach. Since my pandemic dataset is authentically labeled, I can perform this one technique. Classification will attempt to mimic and infer the features of human annotations and their variations, thus enabling the classification of textual data.

There are many ways to classify them. I will watch how they perform on this dataset one by one. In this part, I will use a variety of classification methods, including Naïve Bayes, Logistic regression, K-nearest neighbor, decision trees and random forests, support vector machines and a simple neural network. First, let's try with a logistic regression. To do that, I must turn the training dataset into a TF-IDF matrix. Second, the number of my variables cannot be more than the number of cases. So, I performed a dimensionality reduction on my dataset before conducting the formal analysis. I used PCA for this step. In logistic regression, the response variable can only be a binary variable, and my sample has five true categories. In order to apply my data set in logistic regression, I use "neutral" as my binary response variable. As we can see in the figure, after dimensionality reduction, the algorithm does not provide a clear classification pattern. There does not seem to be a clear boundary between neutral and non-neutral texts. Next, I made another screeplot to see how many principal components I needed. It is important to note that choosing how many principal components can be a very subjective thing to some extent. So I chose a different number of principal components and applied them to my logistic analysis. When I chose the first 10 principal components as our covariates, the model certainly performed well on my training data set: 0.805. However, just because a model performs well on its training data set does not mean that it is a good model. Instead, to validate the performance of logistic regression on the epidemic dataset, I needed to see how the model performed on my test dataset. In the end, the model performed well on the dataset as well: 0.802. Next, I adjusted the number of principal components and brought this dataset into the logistic model for analysis, and the results can be found in the table. The model performed slightly better than 0.800 on the test set, regardless of whether the dimensionality was 10, 40, 100, 200, or 400, which, I must admit, was very good and unexpected. Increasing the dimensionality did not result in a huge magnitude improvement in model performance, but increasing the number of covariates may have overfitted my data, so in practice, I would use a number of principal components of 10 to classify my text. I also tried logistic regression using the TF-IDF scores of each word. But unfortunately, 81% accuracy seems like the best I can get by using a logistic regression. In

addition, I also tried to analyze this problem using Naive Bayes. Although it performed very well on the training set, its performance actually dropped on the test dataset instead.

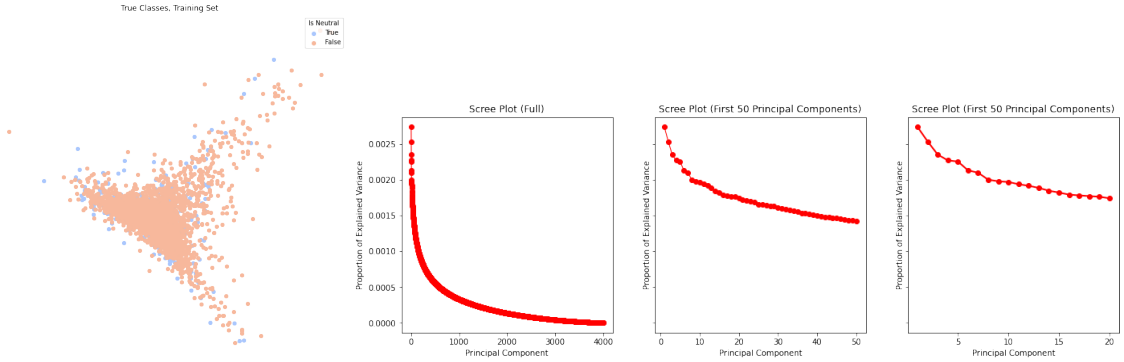


Figure 6: PCA and screeplot.

Table: 2: Logistic Regression Performance

Number of PCA	Training Score	Test Score
10	0.805	0.802
40	0.80525	0.804
100	0.80575	0.802
200	0.80625	0.803
400	0.80925	0.803
TF-IDF	0.8145	0.801
Naive Bayes	0.87975	0.793

This classification is surprisingly accurate. Now, let me see which words are the most influential in this classification. Please see table 3 for the results. This is an interesting result. Those non-neutral tweets seem to be talking more about the pandemic itself. The process of talking about the epidemic may be exactly how people are expressing their emotions, attitudes and thinking. In contrast, the tweets that were classified as "neutral" were more about things that were indirectly related to the epidemic. For example, most of these tweets focused on prices, jobs, and consumers. It seems that people are expressing a wide range of emotions about the epidemic itself, while not expressing more direct emotions about the effects of the pandemic.

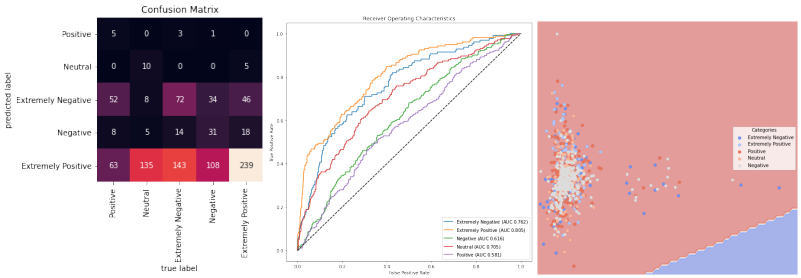
Table: 3: Most influential keywords in logistic regression.

	Neutral	Neutral_log_prob	Not Neutral	Not_Neutral_log_prob

0	good	-3.533531	socialdistancing	-3.110336
1	stop	-3.577016	outbreak	-3.298388
2	safe	-3.599489	essential	-3.369847
3	price	-3.634175	behavior	-3.574641
4	world	-3.634175	market	-3.574641
5	health	-3.670107	health	-3.721245
6	virus	-3.682377	impact	-3.721245
7	working	-3.682377	march	-3.721245
8	employees	-3.694799	quarantine	-3.721245
9	spread	-3.694799	customers	-3.775312
10	customers	-3.707378	corona	-3.832470
11	use	-3.707378	doing	-3.832470
12	staff	-3.720117	news	-3.832470
13	think	-3.720117	right	-3.832470
14	doing	-3.733021	uk	-3.832470

Multinomial Naive Bayes

However, there are actually five types of real labels in my dataset. So if I want to classify our text into many categories, I can't rely on simple logistic regression. Here, I choose to use "Multinomial Naive Bayes". I will use a TF-IDF vectorizer, which converts the document into a vector of words with TF-IDF weights. This gives high weight to words that show up a lot in a given document, but rarely across documents in the corpus. In fact, despite the good performance on the training observations with a score of 0.71275, the model performs very poorly on the test dataset: only 0.357. I generated the confusion matrix from the analysis results, see the figure 6. As we can see, the "Multinomial Naive Bayes" model classifies most of the tweets as "extremely positive". We can evaluate these per catagory. It gives me interesting results, and it is clear that the model is very accurate in classifying extreme and neutral emotions. It seems that the characteristics of those tweets with extreme sentiment are obvious.



Category	Error_Rate	AUC	Precision	Average_Precision	Recall
Extremely Negative	0.127	0.517238	0.555556	0.144701	0.039062
Extremely Positive	0.153	0.528676	0.666667	0.190194	0.063291
Positive	0.518	0.563565	0.347384	0.338561	0.775974
Neutral	0.188	0.561841	0.407895	0.215671	0.178161
Negative	0.300	0.564027	0.339623	0.265400	0.310345

Figure 6: Results generated by Multinomial Naive Bayes

Decision Tree

Decision trees can be used for classification problems as well as for regression problems. Let's see what's going on visually with the classification. Please see figure 7 for all the results generated by my basic decision tree. My conclusion here is consistent with my previous conclusion: the model classifies extreme and neutral emotions well. So after analyzing the basic tree, I further use the bagging method to deal with this classification problem. Combining multiple overfitting estimators turns out to be a key idea in machine learning. This is called bagging and is a type of ensemble method. The idea is to make many randomized estimators and then to combine them, ultimately producing a better classification. As you can see, although I used two different techniques in the decision tree approach, there is no big difference between their results. This result is similar to the result of my analysis using the "Multinomial Naive Bayes" model.

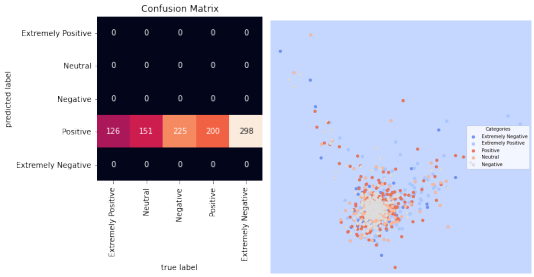
Category	Error_Rate	AUC	Precision	Average_Precision	Recall
Extremely Negative	0.129	0.499427	0.000000	0.128000	0.000000
Extremely Positive	0.152	0.593539	0.546875	0.244143	0.221519
Positive	0.671	0.495355	0.305882	0.306034	0.928571
Neutral	0.174	0.500000	0.000000	0.174000	0.000000
Negative	0.232	0.500000	0.000000	0.232000	0.000000

Category	Error_Rate	AUC	Precision	Average_Precision	Recall
Extremely Negative	0.129	0.549419	0.483871	0.169704	0.117188
Extremely Positive	0.152	0.652665	0.527273	0.293556	0.367089
Positive	0.627	0.500122	0.308063	0.308052	0.831169
Neutral	0.177	0.511793	0.400000	0.181793	0.034483
Negative	0.229	0.513986	0.615385	0.245220	0.034483

Figure 7: Results generated by two decision tree methods. First three: generated by the basic decision tree. Fourth graph: generated by the bagging method.

KNN

The K-Nearest neighbors classifier takes a simpler premise than those before: Find the closest labeled datapoint in set and "borrow" its label. When applying the KNN method to the test dataset, I get a score of 0.2. This test result is bad. Please see figure 8 for the results generated using KNN method. In addition, I generated a visualization of the PCA space. we can evaluate these per category. It is important to note that in the KNN model, it classifies all the texts as "positive". This is why we see that the columns corresponding to the other categories have a value of 0.

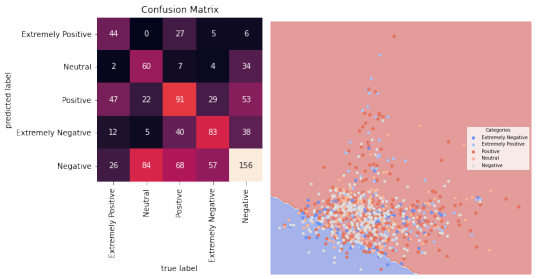


Category	Error_Rate	AUC	Precision	Average_Precision	Recall
Extremely Negative	0.13450	0.5	0.00000	0.13450	0.0
Extremely Positive	0.15600	0.5	0.00000	0.15600	0.0
Positive	0.28650	0.5	0.00000	0.28650	0.0
Neutral	0.80475	0.5	0.19525	0.19525	1.0
Negative	0.22775	0.5	0.00000	0.22775	0.0

Figure 8: Results generated by KNN.

SVM

Now we will examine Support Vector Machines, an approach that creates the partition that preserves the "maximum margin" between classes. Please see figure 9 for results based on SVM. The biggest difference brought by applying this model is that it classifies texts in a much more diverse way. Whereas the previous models I was using tended to classify texts all in the same category. So, in a practical application, I will probably use this SVM model to classify my epidemic dataset.



Category	Error_Rate	AUC	Precision	Average_Precision	Recall
Extremely Negative	0.125	0.646075	0.536585	0.267227	0.335878
Extremely Positive	0.158	0.647091	0.560748	0.307754	0.350877
Positive	0.366	0.606980	0.398977	0.347866	0.543554
Neutral	0.190	0.675360	0.466292	0.312428	0.466292
Negative	0.293	0.596844	0.376033	0.288863	0.390558

Figure 9: Results generated by SVM.

Neural Nets

Finally, I also used the Neural Nets method to classify my pandemic dataset. For the confusion matrix, PCA space and performance of the model, please see figure 10. In fact, the performance of this model is not very different from the performance of SVM.

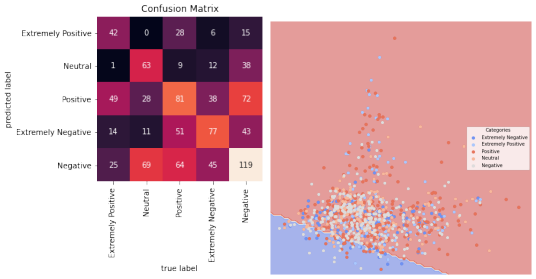


Figure 10: Results generated by Neural Nets.

Classification Summary

In general, the basic conclusion does not change regardless of the method used, except for the application of the logistic model. The most important ideas are: the classification of tweets with extreme emotions is very accurate, the classification of neutral tweets is moderately accurate, and the classification of positive and negative tweets is relatively inaccurate. The most likely reason for this phenomenon is that the characteristics of tweets with extreme emotions are distinct. This is intuitively true.

But at the same time, I must also point out that some of the other results I got were very confusing to me. For example, when I use the Multinomial Naive Bayes model, the algorithm classifies the sentiment of the majority of tweets as "extremely positive". On the other hand, the KNN algorithm gives me even stranger results, as it classifies the sentiment of all tweets as "positive".

Since using various classification algorithms doesn't perform very well, I chose to use clustering as an unsupervised way to partition my text data.

Clustering

Again, seeing that using various classification algorithms doesn't perform well, I chose to use clustering to partition my text data.

In this section, I will explore the patterns in the pandemic text data. Specifically, the work I will do is called "clustering". I will use here both the well-known flat clustering method K-means and another common hierarchical method. I will use another method called sillhouette to identify the optimal number of clusters and to evaluate the quality of unsupervised clustering on labeled data. Next, I will explore a method called topic modeling so that topics can be modeled and computed from the data. I will explore these topics and think about how they can help me understand this pandemic. To simplify my feature matrix, I have limited the word vector to 1000 words (although my pandemic dataset has a maximum of 280 words per document) with at least 3 occurrences.

K-means

I started with K-means. When using this method, I had to predetermine how many clusters I wanted. To do this I need to know how many clusters I'm looking for. Since I know that there are five real types in my text data, I set the k value to 5. However, in the more real world, this number is often unknown. That's why I need to use to identify the optimal number of clusters using another method called sillhouette. I have evaluated my clusters with a variety of metrics: Homogeneity, Completeness, V-measure and Adjusted Rand Score.

Table 4: K-means performance.

Method	Score
Homogeneity	0.010
Completeness	0.011
V-measure	0.010
Adjusted Rand Score	0.000

As we can see, the scores of "Homogeneity", "Completeness" and "V-measure" are basically around all around 0.01. The V-measure is the harmonic mean between homogeneity and completeness. Perfect labelings are both homogeneous and complete, hence have score 1.0. Therefore, by such criteria, my clustering results are not very good. Also, one can see that the results show that the value of homogeneity is very close to 0, which indicates that most of the tweets do not belong to only one single cluster. The value of completeness is also very close to 0. This result shows that I cannot assert that all tweets belong to the same

cluster. This means that the tweets belong to different clusters. The adjusted Rand Score value is very close to 0, which indicates that the clustering level of tweets is independent of the number of clusters.

We can also look at the distinguishing characteristics of each cluster. Please see the table 5. As you can see, cluter 0 is clearly about toilet paper. In the eyes of the algorithm, these tweets, which talk a lot about toilets, toilet paper, and hand hygiene, belong to a cluster. This reflects a topic that is very much on people's minds in March 2020, which is the rush for toilet paper. In addition, cluster 1 is about oil, gas, and prices. cluster 2 is about stores, workers, employment, and retail. This is because of the impact of the pandemic and the avalanche of oil prices in that month.

Table 5: K-means clustering result.

Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
sanitizer	prices	store	coronavirus	19
coronavirus	oil	grocery	supermarket	covid
toiletpaper	coronavirus	coronavirus	food	consumer
hand	19	workers	amp	food
covid19	covid	covid_19	consumer	online
toilet	gas	people	people	shopping
paper	price	going	covid19	supermarket
hands	amp	retail	shopping	demand
use	market	covid19	panic	pandemic
covid_19	pandemic	employees	online	people

After dimensionality reduction of the data using principal component analysis, I visualized the clustering results of my pandemic dataset. First, I got a graph without labels. But one advantage of PCA is that I can also make a biplot that maps the feature vectors to the same space. So, after I added the labels to the graph, I got a new graph. Note that the two graphs I have drawn so far are done on the basis of real classification. But I am more interested in how the algorithm predicts these tweets and clusters them. So let's do it again with the predicted clusters. Please see all the results in this stage in the figure.

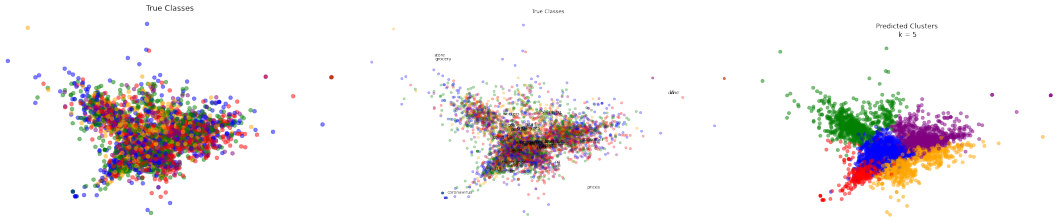


Figure 11: PCA results. Left: original. Middle: biplot. Right: predicted clusters.

What I am curious about is in the eyes of the algorithm, how many categories would it suggest to classify the clusters? Also, as I said before, in this dataset I already know in advance the true number of labels, but in fact, in the more real world I don't know the true number of categories. To solve this problem, or to determine the optimal number of clusters for the unsupervised case, I used the Silhouette method, where I set the number to 3, 4, 5, and 6, and obtained the graph based on these settings. Please see figure 11. Also, for direct

comparison, I put the scores under different settings into a table. Please see the table 6 It is obvious that there is no huge difference in their scores regardless of the number of clusters set to. But it must be admitted that the score of silhouette is the highest when the number of clusters is set to 6.

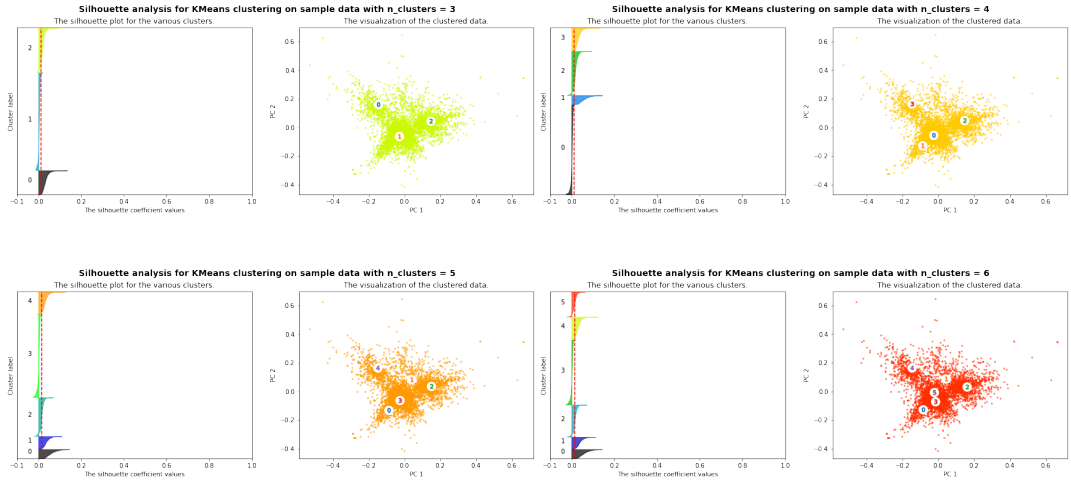


Figure 11: results generated by the Silhouette method.

Table 6: Silhouette Score.

Numbers of clusters	Average silhouette score
3	0.011
4	0.013
5	0.014
6	0.015

Hierarchical Clustering with Wald's Method

Next, I use another hierarchical clustering method: Hierarchical Clustering with Wald's Method. With the help of this algorithm, I can draw a tree of nested clusters. Due to the limitation of my personal computer's computing power and the consideration of the final visual effect, here I have filtered a part of the text for analysis. To facilitate a closer look at my tree, I cut off some branches. Please see figure 12 for the results after processing. The color of the tree shows the different clustering results. After taking this classification result back to the original dataset, I re-evaluated the performance of the model. The results are shown in the table 6. As you can see, although this result is somewhat better than the k-means method, it is not overwhelmingly better. Or to be more precise, it is only a little better than the k-means method.

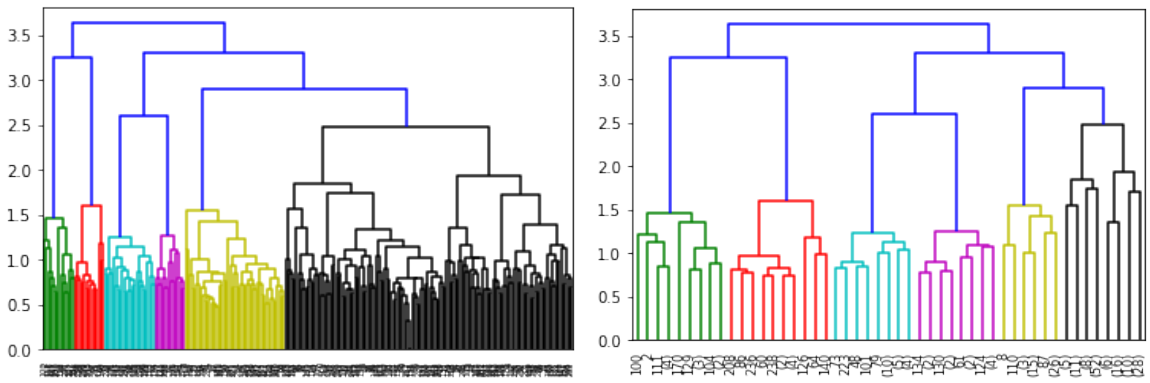


Figure 12: Hierarchical Clustering with Wald's Method.

Table 6: Hierarchical clustering performance.

Method	Score
Homogeneity	0.012
Completeness	0.013
V-measure	0.013
Adjusted Rand Score	0.002

Gensim

Next, I will do the topic modeling. I turn each theme into a separate column. I am more interested in the top words in each topic so that I can understand what is the most critical content under a particular topic. Please see table 7. We can see that there are several themes with the same top-level words, but there are definitely differences. And it is these differences that I am concerned about. First of all, the key words that appear under “topic_1" include mask, wear, face, change, high, and I think what the topic might be describing is that people are aware that the risk of being infected by the coronavirus is high and therefore need to change their behavior and wear a mask. The key words under “topic_2" are worker, economy, pay, job, project, so the topic seems to be more related to people's employment, macroeconomic environment and income. The content included under “topic_3" seems to be more complex, such as employee, company, check, job, state, march. This topic seems to reflect the story between people, companies and countries. Under “topic_4", there are many verbs, such as thank, feel, look, walk, support. But it's hard for me to tell what this topic is about.

Table 7: Top 10 words under each topic.

	Topic_0	Topic_1	Topic_2	Topic_3	Topic_4
0	order	mask	worker	report	gas
1	worker	worker	say	customer	thank
2	self	high	economy	employee	low
3	chain	wear	free	try	feel
4	share	come	change	hour	let
5	live	uk	pay	state	support
6	open	say	look	march	walk
7	increase	face	thing	company	look
8	isolate	product	product	check	news
9	item	change	job	job	year

Word2Vec

I was also using the gensim implementation of Word2Vec. But in fact, the results I obtained using this method are extremely similar to the conclusions I obtained in the network analysis. So I will not repeat them here due to space limitation. In this diagram, "food" is next to "covid-19" and "store" nad “grocery” are near "coronavirus”. “Consumer” is next to "price". This shows that besides the pandemic itself, people are most concerned with things such as food and stores that are directly related to survival during this particular time.

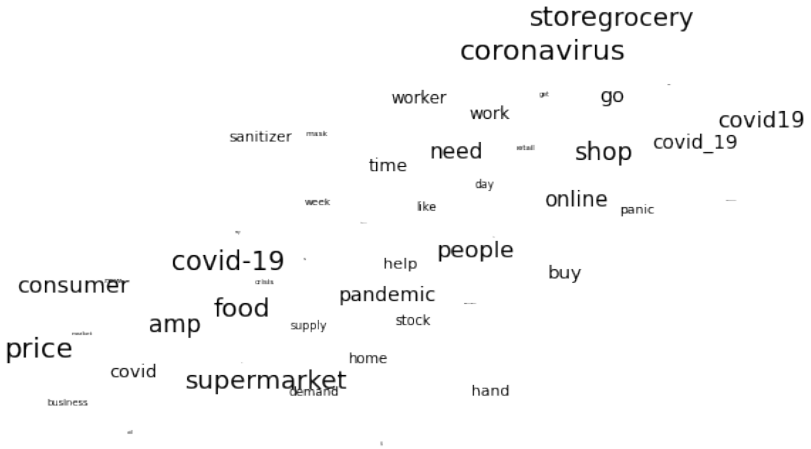


Figure 13: Word2Vec

Doc2Vec

Also, I used Doc2Vec method. As the results do not offer more insights than previous tables and figures in this paper. I will only list them here. Please see figure 14

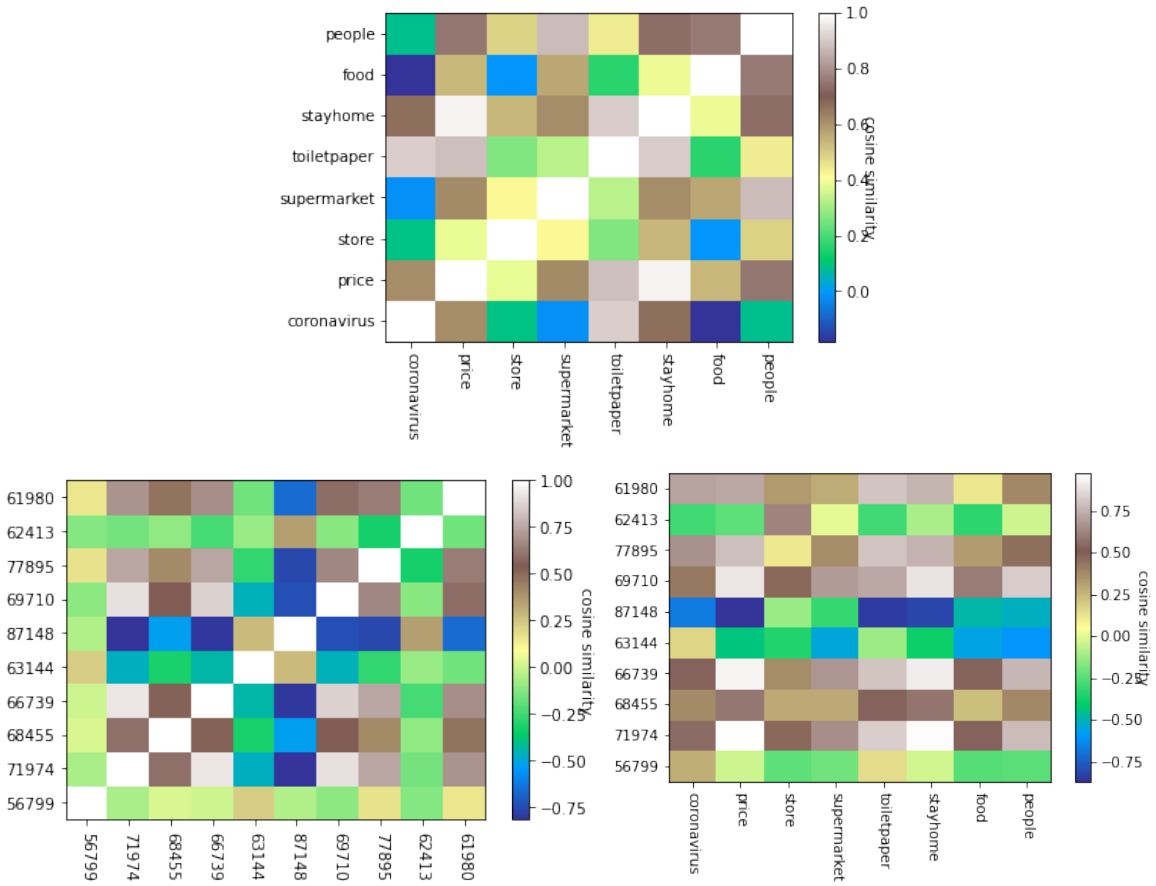


Figure 14: Doc2Vec

Discussion and Conclusion

My project is built on a public dataset from Kaggle. In this dataset, the tweets were pulled from Twitter and manual tagging has been done based on that. On March 12, the WHO officially declared COVID-19 outbreak a pandemic. Therefore, based on this dataset, I have been able to gain some insight into one of the most critical points in time in the pandemic. I began this paper with two questions that I wanted to explore in this project. first, what were people most concerned about during this period? Second, can classification and clustering algorithms perform well on this problem? Through my analysis, I think I have answered both questions well.

For the first question, I have obtained several conclusions. First, in March and April, users on Twitter were most concerned about price, shop, food, store, grocery. Obviously, when a public health crisis like this comes, the first concern is survival before anything else. Second, it is after caring about survival that people start to care about business, about hand hygiene, hand sanitizer, jobs, and workers. Third, The interesting thing is that "stay home" and "mask" are not the most important things that people are concerned about at this time. I think this is an accurate reflection of people's attitudes towards protection measures around March 2020. There seems to be no consensus on the issues of "stay home" and "mask" at that time. But by today, there is a consensus to wear a mask to protect yourself.

Since the creators of this dataset put a lot of effort into classifying the sentiment of tweets, one question I was particularly curious about was how well different classification algorithms can perform in classifying the sentiment of tweets. For the second question, I think that after using a large number of models to carry out, I got the following conclusions.

First, the logistic model performs very well. I used the “neutrality" of tweets as my binary response variable. Regardless of how I choose the number of PCAs, the logistic model performs slightly better than 0.8 on the test set. It actually far exceeds my expectation. Those non-neutral tweets seem to be talking more about the pandemic itself. People expressed a variety of emotions, either positive or negative, about the pandemic itself. In contrast, the tweets that were classified as "neutral" were more about the subsequent effects of the pandemic. For example, most of these tweets were related to price, jobs, and work and consumers.

Second, none of the other classification models performed perfectly. There are five true types of sentiment for my twitter dataset, so I applied other methods to my text data. Regardless of the method applied, the basic conclusions did not change much. These classification algorithms were accurate in classifying tweets with extreme emotions and average in classifying tweets whose emotions were neutral. The most likely reason for this phenomenon is that the characteristics of those tweets with extreme emotions are very distinct. This is intuitively true.

Third, the clustering algorithm has provided me with very meaningful new insights. Since using various classification algorithms doesn't perform very well, I chose to use clustering as an unsupervised way to partition my text data. I looked at the different distinguishing features of each cluster. For example, cluter 0 is about toilet paper. This reflects a topic that is very much on people's minds in March 2020, which is the rush for toilet paper. Furthermore, cluster 1 is about oil and gas as well as prices. This is because of the impact of the pandemic and the avalanche of oil prices in that month.

So in summary, the results of my analyze partly reflects what people were most concerned about in the pandemic in March and April 2020 and the social factors revealed behind it.