Emergency Event Understanding Assisted by Social Media

Nathaniel Cotton
Department of Computer Science
Rochester Institute of Technology
nec2887@rit.edu
Advisor: Jeremy Brown

Abstract—Over the last decade, the presence of social media in today’s society has skyrocketed to the point where platforms such as Twitter are producing nearly 500 million Tweets a day [9]. These 500 million Tweets contain a wealth of information that if understood could help to understand mass public opinion. The project is focused on using Tweets to help garner an understanding of people’s reactions to emergency events. This information could be used to answer questions such as who is saying what, where they are saying it, and when they are saying it.

Index Terms—Data Mining; Emergency Events; Clustering; Social Media; Twitter;

I. INTRODUCTION

Much of our society’s forms of communication have been altered by the advent of social media platforms, to the point where individuals will post emergency related data on social media as opposed to going through traditional channels. One example is that there is evidence that people choose to post requests for help on social media [8][11] instead of individuals using communication channels such as 911 in the United States. Evidence supports that people do choose to go to social media as their medium of contact. It is important for emergency services to be able to rapidly respond to requests posted on social media platforms.

It may be easy to say that emergency services should simply start monitoring social media platforms to answer cries for help. A few difficulties arise, with the first being the sheer quantity of posts that are created daily. For instance, Twitter has approximately 500 million Tweets daily [9]. This is simply too much information for most emergency service departments to handle [11]. Therefore, there is a need for some automated process that can sort through the social media posts and discern what content is relevant to emergency services. This would effectively create another medium of communication to which emergency services could be contacted.

II. WHY IT MATTERS

For small events, such as two girls getting lost and posting a call for help, it is easy to see why a system that can identify these social media posts is important [8]. Especially when the public’s social media posts for help, in most cases, will go unnoticed by emergency services as they simply do not have the resources to perform this kind of monitoring [11]. Therefore, in this simple, case it is clear that a system that can alleviate the manual monitoring concern for emergency services certainly would provide a benefit to society.

For large events such as the bombing in New York City, it would be very rare for an event such as this to not be processed through traditional channels [17]. Therefore, it is less of a question whether or not the event will need to be detected on social media in order to receive assistance. However, these large scale events will generally have a presence on social media. If this presence can be identified, there can be a wealth of information that can be gained from these posts. For instance, to some degree, public opinion can be identified concerning the emergency event, and how that varies with time, and location. While this knowledge does not necessarily help with the response to a particular emergency, its study can garner a greater understanding of how people react; this can be quite useful for future emergencies.

While there is a lot of good that could be gained by monitoring social media platforms, there are also numerous ethical issues. These issues will be discussed in Section VII.

III. ARCHITECTURE

Before a system can be set up to monitor social media platforms, it is critical that a system be able to identify emergency events as well as social media posts about those events.

A. DATA COLLECTION

1) SOCIAL MEDIA DATA: As mentioned earlier, one side of the problem is collecting the social media data that is pertinent to emergency events. This project is using the APIs provided by Twitter to collect real-time public Tweets sent to the platform [15]. Before the Tweets can be collected, the API needs to be initialized with a collection of terms. This project uses the terms found in Figure 1 for the initialization. Once provided with the terms, the API starts streaming Tweets to the application.

While Twitter’s APIs do perform some filtering before sending the data to the application, some issues can arise. The first being that they may filter out Tweets that the application may be interested in because a term does not match exactly. In addition, the API used for this study is free; this results in about only 1% of Twitter’s matching data being sent to the application. However, Twitter does provide APIs that will send 100% of the data, but this requires a substantial
financial contribution [15]. Due to this requirement, these particular APIs are not used.

One more concern with the data being collected through Twitter’s APIs is that they will only send Tweets that have been indexed by the platform [15]. This means that only a subset of Tweets created on the platform are accessible automatically.

For this study, it is a concern that only a small portion of all Tweets can be ingested by the application. It is understandable that only a few people will turn to Twitter to post a cry for help. With few people Tweeting requests for help, and the significant cut back on accessible data provided by the API, it may be difficult to acquire those Tweets.

2) LOCATION IDENTIFICATION: For both the social media and emergency event datasets, having location data attached is critical. Without the location data, a social media post and an emergency event could not be placed in the same vicinity. Without being placed in the same vicinity, a connection cannot easily be drawn between the emergency event and the social media post.

The emergency events dataset, fortunately, was not lacking location data. However, the social media dataset had very few Tweets that contained precise location data. This was expected as people are very cautious when it comes to revealing their location on social media. This is understandable as the location data could lead to harm for the individual [13].

Out of 19.3 million Tweets collected over a course of several weeks, only about 163 thousand contain exact location data. This results in approximately 0.85% of Tweets containing the most specific form of location data. This leaves about 99% of Tweets not tagged with a precise location; however, there are different forms of location data that are associated with individual Tweets. They are as follows:

**TWITTER PLACES:** Twitter Places, is a feature that Twitter provides to their base. This feature allows them to tag their Tweets with a particular location. The geological coordinates of this tagged place can be gathered. However, it is important to note that just because a particular location is tagged does not mean that it is the location that the Tweet originated from [15]. All the Twitter Place represents is a place that was mentioned by the Tweet.

**USER LOCATION:** For users who are setting up their Twitter accounts, there is a portion of the account that specifies a user input location. This user input location tends not to be specific; generally, its highest precision falls at the city level. Since this is user supplied information, it could easily be incorrect as well.

While this information can be useful, the origin of the Tweet may not be the city that is attached to the user. People do move around and will likely not update this information as they move from location to location. This should not rule out this data’s usage. It is indeed likely that the location origin of the Tweet is within the specified city. Additionally, knowing the location of the user’s general whereabouts can be useful in answering questions regarding differences between people of varying regions.

**TEXT ANALYSIS:** For those Tweets that do not contain any of the previously mentioned location data, a fallback can be text analysis searching for a location. This is achieved when a user writes the name of a location in their Tweet. There exists a nice library developed by Stanford, called Stanford CoreNLP that has the ability to extract locations from text using their natural language processing system [6]. However, it is limited to those entities that have already been added to their database of locations. This unfortunately limits the granularity of the Stanford CoreNLP library to popular city names. It would be too difficult to generate a library that would be able to detect street names for every city in the world. As a result, using text analysis is not going to provide significant precision.

3) EMERGENCY EVENT DATA: The other component of the data collection process is the emergency event data. This data contains information regarding the exact coordinate location and time of emergency events that take place in the

---

**Fig. 1.** Collection of terms used to interact with the Twitter API.

<table>
<thead>
<tr>
<th>emergency</th>
<th>aid</th>
<th>assist</th>
</tr>
</thead>
<tbody>
<tr>
<td>assistance</td>
<td>911</td>
<td>fire</td>
</tr>
<tr>
<td>accident</td>
<td>injured</td>
<td>injury</td>
</tr>
<tr>
<td>fire</td>
<td>stolen</td>
<td>theft</td>
</tr>
<tr>
<td>broken</td>
<td>mugged</td>
<td>attacked</td>
</tr>
<tr>
<td>call the police</td>
<td>cut</td>
<td>burnt</td>
</tr>
<tr>
<td>please help me</td>
<td>help</td>
<td>please hurry</td>
</tr>
<tr>
<td>are you ok</td>
<td>is everyone ok</td>
<td>i’m lost</td>
</tr>
<tr>
<td>we’re lost</td>
<td>please leave me alone</td>
<td>go away</td>
</tr>
<tr>
<td>kill</td>
<td>suicide</td>
<td>danger</td>
</tr>
<tr>
<td>depression</td>
<td>terror</td>
<td>terrorism</td>
</tr>
<tr>
<td>crisis</td>
<td>catastrophe</td>
<td>urgent care</td>
</tr>
<tr>
<td>evacuate</td>
<td>trapped</td>
<td>hurricane</td>
</tr>
<tr>
<td>tornado</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 2.** Percentage of Tweets containing the different forms of location data.
Rochester, NY and the Detroit, MI areas [1]. Beyond the location and time data, each instance in the dataset contains information about what the emergency event is, as well as its severity.

Additionally, the database that contains the dataset is being added to in near real-time. Therefore as new emergency events occur, they are added to this database. They can subsequently be mined and used for the exploratory research of this project.

IMPLEMENTATION

IV. ITERATION 1

Since this project is exploring the usage of the Twitter social media platform for emergency event prediction/identification, the most critical component is tying the two datasets mentioned in Section III-A.1 and Section III-A.3 together. It is intuitive to believe that the two data fields that tie the datasets together are the time in which the event/Tweet occurs and the location of the event/Tweet. By performing some fuzzy matching between the two data fields an emergency event and a number of Tweets can be placed in the same vicinity in geo-temporal space 1.

Given that the severity of the events in the emergency dataset is relatively small, the physical distance between the location of the incident and the location of related Tweets should also be relatively small. It is intuitive that someone is unlikely to Tweet about a car accident from twenty miles away. In that same respect, it is unlikely that someone would Tweet about an emergency event with useful information at times that are distant from the time of occurrence. Therefore a range of time surrounding the emergency event can also be used for the matching process.

A. Matching Process

In order to associate an emergency event and a Tweet, it has been deduced that there is a need for some geo-temporal restriction. However, it is unknown as to what the optimal geo-temporal restrictions on a particular emergency event should be. Since there is a lack of classified data, the optimal restrictions must be evaluated based on running several restriction scenarios and using common sense.

Several restriction scenarios were run on 28,175 emergency incidents that occurred while Tweets were being captured. The range of restrictions that were used was a distance of one kilometer to four kilometers from the emergency incident, and the distance metric was incremented by one kilometer for each test. Additionally, the time restriction spans from 30 minutes to 150 minutes before and after the time of the incident, and was incremented by 30 minutes for each test. These restrictions resulted in matching rates between Tweets and emergency events that can be found in Figure 3. With the most restrictive constraints there was a matching percentage of 1.44%, and at the most lenient the matching percentage was 26.3%.

From the results shown in Figure 3, it can be determined that if all variables remain the same, a fluctuation in time increases the percentage of matches far more than a fluctuation in distance. From the results of the 20 permutations of restrictive parameters, one that seems fairly reasonable is to restrict the distance to 2 kilometers, and the time to 2 hours before and after the incident. This configuration produces a matching percentage of approximately 9.718%.

Under this configuration and looking at the matched up incidents and Tweets, nearly 61% of the Tweets contained the term “accident”. Upon further exploration, all of these “accident” Tweets were posted by news agencies alerting the public that there had been an accident on a major roadway and that delays should be expected. Nearly 61% of the matches made did not go toward the project goal of being able to assist those who turn to social media for help. When looking at all emergency events in total those that have matching Tweets that do not contain the term “accident” results in a matching of 3.79%. With further processing, this number would surely decrease to a value close to 0%.

For a clearer picture of the lack of data to accomplish this task Figure 4 shows all of the exact locations of people who have sent out Tweets that use the terms in Figure 1. In comparison, Figure 5 indicates all the emergency events that have occurred in the Rochester, New York area. It is pretty clear to see that in most cases people do not turn to social media for assistance in these smaller emergency events. This is to be expected as most people generally do turn toward traditional channels to communicate a need for help.

\[ \text{Geo-Temporal Matching} \]

\[ \text{Fig. 3. Percentage of matching between emergency events and Tweets given geo-temporal constraints.} \]

V. ITERATION 2

The lack of precision location data on the Tweets prevents necessary matching in the geospatial dimension, so the use of the data needed to be altered. Instead of looking at using
the data for smaller events, such as an EMT call to a local resident, it will be used to examine larger scale events that receive national attention. With this change of scope, the need for location information, while important, is no longer critical, as location is not necessary for understanding public opinion of a major event.

A. Matching Process

One issue that arises with the change of scope to larger events is that location is no longer a restrictive force. As a result, any Tweet from anyone could be related to the emergency event. This means that the previous approach of simply collecting the Tweets that occur within a particular distance and time frame does not make sense. Therefore, a new approach for Tweet selection needs to be constructed.

1) Understanding Hashtags: Fortunately, the way that Twitter operates is based on the concept of a hashtag, a hashtag being some term that is prefixed with a "#". These hashtags play a critical role in the inner workings of Twitter, as they add searchability to the Tweets. This is a concept that the users of Twitter have a rather firm grasp of, and will use hashtags appropriately. This is evident when 4,355,037 out of 19,355,383 (22.5%) Tweets contain a hashtag of some sort.

These hashtags can be used in a rather interesting way since they are designed to be searchable terms. They give some information about what the attached Tweet is about. In a sense a Tweet with a hashtag is placed in a bucket with all the other Tweets that contain the same hashtag. Since 22.5% of tweets contain more than one hashtag, Tweets can also belong to more than one bucket. Since a Tweet can exist in multiple buckets, unidirectional connections between buckets can be formed. The strength of these unidirectional connections can be determined by the amount of cross-over. This can be calculated using the equation in Figure 6.

\[
\text{strength}(\#A \rightarrow \#B) = \frac{\text{size}(\#B \cap \#A)}{\text{size}(\#A)}
\]

Why unidirectional? The reason for the unidirectional relationship between Tweets can best be explained with an example. Take two hashtags #breakingnews and #earthquake. It is reasonable to believe that a large portion of the #earthquake Tweets are going to contain the hashtag #breakingnews as well. As a result, a large proportion of the #earthquake Tweets contains #breakingnews. However, the same cannot be said in the opposite direction, as there are many reasons why a Tweet would contain #breakingnews, as for instance, in a presidential election. Therefore, both the presidential election and the earthquake have a large proportion of Tweets in the #breakingnews bucket. However, the #breakingnews bucket does not hold the same proportion with respect to the presidential election or the earthquake.

2) Tweet Selection: Since the system is not yet analyzing the incoming Tweets on the fly and determining the important hashtags that should be analyzed, a bit of human intervention is needed. The intervention is supplying a starting point, which is represented as a collection of hashtags (could be a single hashtag).

With this collection of hashtags, a new active bucket can be formed with all the Tweets containing at least one of the provided starting hashtags. The system does not stop there when it comes to Tweet selection. It will analyze the
connection strength between the active bucket and all other hashtags. This may seem to be a rather time-consuming process, but a few heuristics can be used to speed up the process. The first step is to find the counts of all hashtags within the active bucket that were not used in the creation of the active bucket. Instead of using the formula in figure 6, the realization that the counts of the hashtags in the active bucket can be used in place of the formula greatly decreases the number of calculations. If the counts of the hashtags are sorted in descending order, the active bucket has the greatest unidirectional connection strength with those hashtags in descending order as well.

The next step is to automatically see which of those hashtags has the closest similarity to the active bucket. To perform this operation, the buckets associated with each hashtag going down the list are fetched, and two metrics are collected from them. The first is the centroid\(^2\) of the bucket, and the sum of residual squares. With these two metrics the following calculations are performed, the first being a cosine similarity test on the centroids of the active bucket and the newly calculated centroid. This cosine similarity will produce a single number of how similar the two centroid vectors are, 0 being completely dissimilar, and 1 being identical. If the produced metric fails to surpass some set threshold, usually 0.60, that bucket will be rejected. However, if the similarity metric does surpass the set threshold, the next step is to use the sum of residual squares to ensure that the new bucket is not too diverse. If the sum of residual squares does surpass some threshold (indicating it’s too diverse) the bucket will be rejected. Otherwise, the bucket will have passed the similarity test and the diversity test at which point the bucket will be unioned with the active bucket and the process will repeat. That is until it has executed for a specified number of iterations, or it has failed to expand the active bucket.

Why is this collection growing process complex? The reason is that it is certainly possible for Tweets that contain more than one hashtag to have hashtags not related to an event. That is where the similarity metric comes into play, as it will reject hashtags if the corresponding bucket is not similar enough. There is a flaw in the calculation of similarity using the cosine similarity; that is, it relies on the centroid of each collection of Tweets. It is certainly possible that the centroids can be very similar, but one collection could be very diffuse, which could mean it could contain a number of Tweets that are not associated with a particular concept. However, this diffusion can be captured by the sum of residual squares, and if it is determined to be too diffuse there is a good chance that concept drift could occur.

The flaw in this approach is that it is very focused on avoiding concept drift. However, if the algorithm is not properly initiated it will collect all the Tweets that are talking about one side of an event because it avoids concept drift to another perspective. To overcome this potential issue, the terms used to initiate the algorithm are critical.

\footnote{A centroid is a point where each dimension of the vector is the mean of that dimension of for all other points in the cluster (this centroid point may not actually exist in the data).}

B. Instance Curation

The next phase is to turn that collection of Tweets into instances that can be used for the clustering phase. The first part that needs to be done is to normalize the data. For the most part, the data is partially normalized as it was needed for the Tweet selection process. However, further normalization is needed, along with global term expansion.

The text normalization process is as follows. All the terms in the text of a Tweet are identified, and if the term exists in a collection of stop words, it is removed. The remaining terms are then summed based on their frequency. In order to produce a vector that contains values between 1 and 0, each frequency is divided by the total number of terms in the text after the stop word removal. Once this process has been performed for each Tweet in the collection the next step is global term expansion. Since the clustering algorithm provided by [5] requires a dense array-like data structure, it is necessary to ensure every term that appeared is in every instance. If the term does not exist in an instance the associated term value is 0. After each instance has been expanded to include all encountered terms, a global normalization process takes place. This involves finding the highest and lowest value for each term across all instances. With these two pieces of information for each term there is one more pass over the collection of instances. This time each term value has the global minimum term value subtracted from it, and is then divided by the global maximum term value minus the global minimum term value. As a result of this process each instance has a value for every term that is between 0 and 1. The higher the value the more prominent the term is for that instance, and the lower the less prominent.

C. Clustering

Once the normalized instances for each processed Tweet are created, the next step is to run them through a clustering algorithm. The clustering algorithm used for this portion is the k-means implementation provided by [5]. This implementation requires that the data passed into the algorithm be a dense array-like structure. This provides the benefit of faster computation, at the expense of utilizing more memory. Due to the nature of the k-means algorithm, it needs to be run several times with different initialization arguments to find the ideal clustering for a particular value of k. Beyond that, depending on the data the value of k may be different, so in addition to running the algorithm multiple times for a single value of k, multiple values of k need to be tried.

How does the algorithm determine the ideal value of k and the ideal cluster formation? The answer to both lies in one metric the sum of residual squares. This metric increases in value when the data points within a cluster are further away from the centroid. The goal is to come up with a clustering that minimizes this value. Therefore for a single value of k the clustering that produces the smallest sum of residual squares is used. As the value of k is increased the sum of residual squares will naturally decrease until it reaches zero, which would likely be the case when k is equal to the number of data points. However, that is not useful, therefore
a threshold is needed to determine a smaller value of k that will still produce reasonable cluster. A simple technique has been developed where if the sum of residual squares for each value of k are plotted they will produce a knee shape. The value of k that is associated with the bend of the knee is the ideal value of k [2].

For this project, an automated process of selecting the optimal value for k is performed. The sum of residual squares is used to determine success. However, the designed algorithm will look at the slope of two adjacent values of k. If the slope falls below a particular threshold the algorithm will select the most recent value of k as the ideal k. This has the benefit of cutting off the execution of the k-means algorithm a bit quicker, but also has the downside of potentially not being the ideal k. However, the results in Section VI seem to be decent, so this methodology does work with this particular dataset.

D. Data Visualization

All this computation, while beneficial, is not particularly useful unless humans can understand it. In order to put the data into a way that humans can interpret it, three forms of visualization have been created. The first being the use of word clouds to visualize cluster centroids, temporal line plots to show the growth of clusters over time, and map plots to determine where certain things are being said [7][10]. All of these visualizations for particular emergency events can be seen in Section XII.

1) Word Clouds: In the case of visualizing clusters, the issue is the high dimensionality of the data. This, in turn, causes the centroids to have a high dimensionality. Since humans are only adept at visualizing data in three dimensions at the most, a visualization problem emerges. To overcome this issue word clouds for each cluster centroid seemed to be a natural choice. As each dimension of the data can be represented by their respective term, and the numeric value associated with the term can be used to dictate its size. For instance, if there is a very prominent term in a particular cluster, based off of frequency, it will appear larger on the word cloud, while those that do not occur as frequently, are smaller.

By looking at individual cluster’s word clouds it is easy to see what terms are being used the most. Generally, these terms paint a vivid picture of what a group is trying to say. For instance a cluster related to the Hoboken train accident, can be seen in Figure 18. The word cloud shows terms such as "leaders", "christie", and "aid" being used extensively. With these terms in mind, it is easy to deduce that people are begging leaders, such as New Jersey’s Governor Chris Christie for aid.

2) Temporal Line Plots: While the word clouds are informative as to what people are saying it is unfortunately only one dimension of understanding the event. To expand the information, a time component needs to be visualized, that is where the line plot comes in. As it displays the sum of all Tweets related to a particular cluster within a particular time frame. Following the previous example Figure 26 shows a cluster growth and decay limited to one standard deviation from the mean time. By examining the line associated with the cluster shown in Figure 18, it can be seen that there were several spikes after the initial mean time spike. From this information, one can say that there was a delay before people were asking leadership for aid.

Several line plots are created for each event that is run, with the plots being limited to one, two, or three standard deviations from the mean time. This is done to make the graphics appear less cluttered at the expense of information loss. Comparing Figure 26, and Figure 27, can help with understanding this design decision. As Figure 26 is slightly easier to read.

3) Map Plots: In Section IV, it was mentioned that one struggle with the social media data was the lack of precision locations. However, for the major events, precision location data is not critical, as a result simply knowing the city that a Tweet originated from suffices. The Google Maps API, along with the user supplied location can be used to determine the rough coordinates for a Tweet [3]. This information is then passed to Plotly. Plotly will then place cluster color-coded points on a map of the United States [7].

Figure 16 shows the locations of Tweets that are related to the Hoboken train accident. From the points, on the map, it is clear to see that the highest concentration of Tweets can be found at the origin of the accident. However, upon further examination it was possible to see how the rest of the nation was reacting to the incident.

As mentioned each point on the map is color-coded to a particular cluster. Therefore with careful observation, one could determine that different parts of the country are talking about the same event differently.

VI. RESULTS

Due to the scope of the project being expanded to large emergency events, and limiting nature of Twitter’s API there were only a few major events that were captured [15]. These events consist of the bombing in Chelsea, NY, the train accident that occurred in Hoboken, NJ, and Hurricane Matthew. All of the application output for each of these events has been placed in Section XII.

A. Bombing: Chelsea, NY

The figure results for the bombing in Chelsea, NY can be found in Section XII-A. The very first thing to notice, is that the reaction to the event is for the most part contained on the east cost of the United States, with a higher density in the north-eastern region (see Figure 8). It can be noticed that the map does not contain too many points. By looking at the cluster membership table it can be seen that not too many Tweets were collected for this event (see Figure 9). For the most part only two clusters exist, as their membership is rather low. The next step is to look at the word clouds to see what clusters 2 and 4 are saying (see Figures 11, 13). From these word clouds the only definite difference is the prominence of the hashtag. Other than that cluster 4, may focus on the terrorism aspect of the event slightly more than cluster 2 does.
B. Train Accident: Hoboken, NJ

The figure results for the train accident in Hoboken, NJ can be found in Section XII-B. Starting with the map plot, a quick realization is that the highest density of Tweets exist within close proximity to the location of the event (see Figure 16). Next, looking at the cluster membership table it can be seen that there are three major clusters (2,4,6), and three less prominent ones (1,5,8), as well as some noise (3,7) (see Figure 17). First, analyzing the three clusters with the highest membership (see Figures 19,21,23). At a quick glance it could be determined that clusters 2 and 4 are fairly similar. However, a suitability between the two does exist, cluster 4 appears to be talking more about 100 people being injured, than cluster 2 does. However, cluster 6, the largest of all of them is drastically different (see Figure 23). This word cloud is showing a heavy focus on terms such as "crash", "injured", "100", and "dead". These terms can be used to determine that this group is more focused on the effects of the accident, and less so on the details.

Moving onto the three less prominent clusters, clusters 1, 5, and 8 (see Figures 18,22,25). These clusters tend to be more focused on specific reactions to the event, as compared to the major clusters. As mentioned earlier cluster 1 uses terms such as "leaders", "christie", "aid", and "begged". The group of people that belong to this cluster are more so focused on the response of the government. Along those same lines cluster 5 features the term "ntsb", this is a reference to the national transportation safety board. This group is focused on the investigation of the accident. Finally, cluster 8 does not appear to contain terms that are significantly, different from the major three clusters.

Adding in the time component, which can be seen in Figure 26 an interesting aspect of data is picked up. Between the x-axis values 30 and 35, there is a spike in Tweets related to cluster 1. Cluster 1, being the one that was begging New Jersey leadership for aid.

C. Hurricane Matthew

The largest event that was captured was Hurricane Matthew that came up the eastern side of the United States. The figure results for this event can be found Section XII-C. Opposed to the previous two examples, where there was a single point of incident, Hurricane Matthew has several. In fact those points of incident can easily be found on the map plot (see Figure 28). By looking at this plot it is quite evident that the eastern side of the United States is the point of impact.

With the entire eastern side of the United States covered with points, it is not surprising to find this event produced the most clusters as well, twenty-five in total. Of these twenty-five clusters four of them surpassed a membership of one-thousand (7,11,13,21) (see Figure 29). However, none of these clusters are particularly interesting, as each one, excluding cluster 11, only uses the terms "hurricane" and "matthew" (see Figures 36,42,50). However, cluster 11 has a bit more detail as a major term being used is "haiti", along with numerous other terms. The notion that a location term is prominent within a cluster creates a nice relationship between that location and the other terms in the word cloud. With this notion and examining the remain clusters it is easy to see location terms pop out. Going through the clusters by hand the following some location associations were made, they can be seen in Figure 7.

When looking at the temporal line plot for this particular event it is a bit harder to interpret due to all of the different clusters. Unfortunately, due to this readability issue the temporal line plot was not particularly useful for this event.

D. Application Performance

By looking through the results of each of the three events we analyzed a few things stick out. First, the usage of k-means, and the approach for determining the value of k, while it did work, it probably over fitted. As there were few clusters that had fairly low membership, but contained terms that other larger clusters had as well. In most cases if the k were to be decreased those clusters would likely be merged. As a result of this merging some of the graphics would have become cleaner. Second, the color scheme for some of the line plot and the map plot could be improved to be a bit more distinct. This would help with extracting information from them. Finally, for the map plot it would be easier to read the locations of particular clusters if they were not placed on top of each other.

All those negatives considered, the map plot was useful in determining where particular people were Tweeting about an event. The word clouds were useful in determining what some of the clusters were trying to say. Finally, the line plot for events with fewer clusters showed when people from particular clusters were talking.

VII. ETHICAL ISSUES

With any new technology, there are also concerns for its misuse. Since this technology requires the monitoring of people on a massive scale a few ethical issues do arise. The biggest issue that arises, but is certainly not limited to is the violation of privacy. For instance, the Twitter APIs used for this project can collect nearly five-million Tweets a day. Each of those Tweets has a user’s information attached to them,

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Location Terms</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>space, station, kennedy</td>
<td>Florida</td>
</tr>
<tr>
<td>3</td>
<td>haiti</td>
<td>Haiti</td>
</tr>
<tr>
<td>4</td>
<td>haiti</td>
<td>Haiti</td>
</tr>
<tr>
<td>6</td>
<td>haiti</td>
<td>Haiti</td>
</tr>
<tr>
<td>9</td>
<td>miami</td>
<td>Florida</td>
</tr>
<tr>
<td>11</td>
<td>haiti</td>
<td>Haiti</td>
</tr>
<tr>
<td>12</td>
<td>north, south, caroline</td>
<td>North and South Carolina</td>
</tr>
<tr>
<td>17</td>
<td>jacksonville, fl</td>
<td>Florida</td>
</tr>
<tr>
<td>20</td>
<td>haiti</td>
<td>Haiti</td>
</tr>
<tr>
<td>21</td>
<td>florida</td>
<td>Florida</td>
</tr>
<tr>
<td>22</td>
<td>south, carolina</td>
<td>South Carolina</td>
</tr>
<tr>
<td>23</td>
<td>florida</td>
<td>Florida</td>
</tr>
</tbody>
</table>

Fig. 7. Location associations based off of terms in clusters.
A malicious entity could certainly use this personal information to derive some probable associations to a particular user. For instance, a Tweet from someone saying that they are going on a vacation and have location features enabled on their Tweet. Then the malicious entity could determine that the person is not going to be home, and also has the location of their empty home.

As a result many people do protect themselves from having their information being exposed by setting up privacy policies, as well as simply not using particular features [13]. As mentioned earlier this does pose a problem for this project as the location of a Tweet is incredibly important for matching them to small emergency events.

Additionally, there is the concern of differential privacy [14]. A person who is very careful not to reveal any sensitive information may still be exposed. This occurs when someone else who is rather similar reveals their data. With this information a malicious entity could produce probable attributes for the careful individual.

Nevertheless, in order to have certain monitoring technologies people need to be willing to expose more of their data, but also need to be protected from those who would seek to misuse the data.

VIII. LESSONS LEARNED

This project utilized two aspects of the social media data quite heavily. The first being location information, for gaining an understanding of where people are talking. This posed a challenge because many individuals will not provide geo-location information on their Tweets, hence the initial idea was abandoned (Section IV). It makes sense why they would not share their location, as malicious entities could utilize this information for less than noble purposes [13]. Fortunately, there was some usable location information that along with the Google Maps API could be turned into coordinates [3]. Unfortunately, the portion of the data used for the determination of the location was also user supplied data. Therefore Andrea L. Kavanaugh, et al. acknowledge not trusting the data pulled from social media as legitimate [16]. However, having no other data to work with, and having the concern that some of the information may be false is a difficult issue to overcome.

Another aspect of the data that was used was the Tweet text. The goal of using this part of the data was to understand what the Tweeter is trying to convey. Unfortunately, Twitter is a microblogging platform as such a length of the text is limited. As a result, many of the users to overcome this limitation use slang and shorthand, as to avoid reaching the character limit. However, the result is text that is difficult for a natural language processing system to understand. The workaround for this project was to focus on the terms the users used instead. Contrary to Erik Cambria, et al. belief that this form of opinion mining is the least effective [12]. Many of the approaches that Erik Cambria, et al. proposed that would work better would also be far more computationally expensive, and may not work on the shorthand text [12]. From the results of this project, it is believed that any loss of information by using the individual terms is made up for in the quantity of the data.

IX. FUTURE WORK

There are three major components to this project, excluding the data collection process. Each of these three major components could always be improved. Given more time, and more resources, these are some improvements that could be made to the project.

Starting with the first phase, the partial agglomeration process worked to grow the active collection; however, it did run into some issues. For example, when #chelseaxplosion is given to the algorithm it will also accept #chelsea during the growth process. Some of these Tweets were related to the bombing, many of the Tweets in #chelsea are related to the English soccer team. The algorithm here could be more sensitive, that instead of considering all of the Tweets in collection it could carve out subsets to add to the active collection.

The next phase is the clustering where the k-means algorithm is used to perform the clustering. As mentioned in Section V-C, the k-means algorithm is sensitive to its starting configuration and as such it needs to run multiple times to get a near optimal answer. Since each run could be independent of any other run it could be distributed across multiple machines, not just multiple threads. Additionally, due to memory constraints on a single machine the amount of data needed to be limited. An optimal solution would be to have a single k-means run distributed across several machines. That way the memory of each machine can be used, thus raising the total amount of accessible memory. Finally, different clustering algorithms could be explored to see if some performed the clustering more efficiently, or successfully. For example the DB-Scan cluster algorithm would be good at removing noise from the data [4].

The last phase is the data visualization portion. The data visualization part of this project is critical as that is what helps users determine the meaning behind the data. While the use of word clouds, temporal line plots, and map plots are useful in conveying the information presented by the data, they are not the only ways the data could be visualized. Exploring other forms of visualization could be done to discover if there is a better way of conveying the information.

X. CONCLUSION

By using the Twitter API and collecting numerous Tweets over the course of several weeks. This project was able to build up a decent sized database of Tweets could be mined for information related to particular emergency events. The usage of the k-means algorithm worked for clustering the data into groups that were in general talking about the event in different ways. This information was successfully conveyed through the usage of word clouds, map plots, and temporal line plots.
However, there certainly were some issues with the output either being over fitted, or being difficult to interpret. These issues could certainly be resolved with the usage of more advanced algorithms, and different forms of visualization.

X. BI-WEEKLY PROGRESS

a) Week 0 - Week 1: For the first two weeks of the project the main concern was truly understanding the problem at hand. In order to do this a number of news articles as well as academic papers were read to gain insights into the domain. It quickly became apparent that there would be a benefit to a system that could analyze social media and flag posts that indicated a person was in need [8][11]. However, the academic papers talked more about ideas of how to use social media to achieve the end goal of having an automated system for detecting need.

b) Week 2 - Week 3: The next step was to attain the data needed to perform the analysis to see if social media posts could be accurately matched up to emergency events. On the social media side of things, the big social media such as Facebook, Twitter, and Instagram were searched for public APIs. It was discovered that it was against Instagram’s terms and conditions to use their APIs for data mining purposes. As a result that social media platform could not be used. However, since it is an image based platform it was not too much of a loss. On the other hand, Facebook, which is the largest of social media platforms with the most diverse set of users could also not be used. That was because Facebook as of the writing of this paper removed the ability to use their APIs to query public posts. The only other option for attaining a feed to all the public posts was to be a large media company. Unfortunately, access to those APIs was prohibited. Finally, Twitter, the microblogging platform was very generous providing APIs that were explicitly designed to pull Tweets for data mining purposes.

From there an application was created to connect to the Twitter API providing a number of terms for Tweet selection. These Tweets were then collected and stored in a MongoDB. However, the one downside to the Twitter API is that it only allows Tweets in near real-time to be collected. As a result, Tweets that were posted a week prior to starting the collection process could not be captured. Therefore the collection process needed to be run as often as possible to collect as much data as possible.

c) Week 4 - Week 5: The other half of data collection process was to attain data on emergency events. Fortunately, Professor Brown had access to a collection emergency data for events that occurred in Rochester, NY, and Detroit, MI [1]. With all of the necessary data collected, the next step was to understand how the Twitter data and emergency event data could be fit together. It was determined that the best way of associating particular Tweets with particular emergency events was to place them in geo-temporal space. Those that are closest to each other could be determined to be associated.

An application was built to perform this matching process, associating Tweets with emergency events if they occurred within a certain time frame of the time of the emergency event, and within a particular distance of an emergency event.

d) Week 6 - Week 7: Sadly, it was discovered that many of the Tweets did not contain location data that was sufficient in precision. Due to this lack of data the matching process recorded dismal results. Those that did match were mostly news stations posting after the fact Tweets about accidents on particular roadways, and that traffic would be slow.

Since this matching process was a core component for the success of the project, and its failure meant that the initial idea could not be executed to a successful state. Through discussions with Professor Brown, with regards to this insurmountable problem, it was determined that a new idea needed to be developed. Instead of trying to marry the two datasets together, the idea was put forth to explore each individually to determine if meaningful information could be extracted from them.

Due to the interest in the social media data, that data was explored first. As it was known that precision location data did not exist to a necessary degree for small emergency events. The scope was expanded to larger emergency events such as the Hoboken train accident, or Hurricane Matthew. With these large scale events that may incite a national or global response, the location data did not need to be as precise. A plan was developed for how a non-complete information agglomeration process could be used to select Tweets that were related to a particular emergency event, and a clustering process could be used to determine different groups within the collection of Tweets.

e) Week 8 - Week 9: A system was developed that would take a collection of terms as inputs and grab all the Tweets that were stored from the collection process that contained those terms. Then those Tweet’s text was split into individual terms and then all of those terms would be aggregated into total counts. Since Twitter uses hashtags for searchability the top hashtags that exist in the collection of aggregated terms are looked up to get all the Tweets that contain that hashtag. From there the system was designed to compare the two collections to see if they are similar enough to be related to the same topic. If they are deemed to be similar enough they are merged. This process continues until there are no more collections of Tweets to merge.

Once a collection of Tweets is finalized all the terms in the text are normalized by their occurrences across the board. Then each Tweet is turned into an instance where each term in all of the Tweets has a value between 0 (not occurring) and 1 (only term to occur). These instances were then fed into the k-means algorithm to determine the particular groups that exist in those talking about a particular event.

f) Week 10 - Week 11: The non-complete information agglomeration process while it does work it did run into some issues that were a bit harder to overcome. For instance, if #chelseaexplosion were used as an initialization term the non-complete information agglomeration would find #chelsea in the collection. Following its routine would then collect all the Tweets that contained #chelsea. However, Chelsea, while it is the location of the bombing, is also the
name of a soccer team in England. Unfortunately, the algorithm would deem the two collections to be similar enough that they should be merged. As a result, the application was modified to allow a collection of terms not to query to be added. For this example, #chelsea could be added to that collection, and then the finalized collection of Tweets would not contain soccer-related Tweets, at least not in such a high proportion.

The next issue that arose is how to visualize the data. Each cluster centroid produced by the k-means algorithm exists in n-dimensional space; where n is the number of unique terms across all processed Tweets. This becomes a somewhat unwieldy structure for visualization. However, if there is less of a concern displaying centroids side by side a word cloud can be used for displaying the terms. The system was built to display the centroids as word clouds with the largest words being those that occur the most frequently, and the smallest being those that occur less frequently.

At this time it was also discovered that super large events such as Hurricane Matthew would take a significant amount of time to query the database, as well as process the Tweets. To combat this issue the querying portion of the process would cache the results away. This effectively removed the long periods of time need to search 20 million Tweets for the correct Tweets.

g) Week 12 - Week 13: While the word clouds are nice for easily interpreting what each cluster is saying it does not provide the whole picture. For instance it does not account for where and when people are saying particular things. To help visualize the where portion the Plotly API, along with the Google API were integrated. Google Maps API was used to get a general idea of where a particular Tweet originated from [3]. This information along with the cluster that the Tweet belongs to is sent to Plotly for being placed on a map [7].

Up to this point one question has not yet been answered, when are people talking. To help answer this question temporal line plots were created to show the growth and decay of clusters over time. These plots aggregated the number of Tweets belonging to any one particular cluster for a period of time.

h) Week 14 - Week 15: The application was run on the major emergency events throughout its creation. It was run one last time on Hurricane Matthew, the Hoboken train accident and the Chelsea bombing. All the results were collected and interpreted. The results of this interpretation can be found in Section VI.
Fig. 11. Chelsea Bombing: Cluster 2

Fig. 12. Chelsea Bombing: Cluster 3

Fig. 13. Chelsea Bombing: Cluster 4

Fig. 14. Chelsea Bombing: Cluster growth limited to one standard deviation from the mean time.

Fig. 15. Chelsea Bombing: Cluster growth limited to three standard deviation from the mean time.
B. Train Accident: Hoboken, NJ

Fig. 16. Train Accident: Map Plot

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Membership</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>38</td>
</tr>
<tr>
<td>2</td>
<td>149</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>35</td>
</tr>
<tr>
<td>6</td>
<td>151</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>47</td>
</tr>
</tbody>
</table>

Fig. 17. Train Accident: Cluster membership.

Fig. 19. Train Accident: Cluster 2

Fig. 18. Train Accident: Cluster 1

Fig. 20. Train Accident: Cluster 3

Fig. 21. Train Accident: Cluster 4
Fig. 22. Train Accident: Cluster 5

Fig. 23. Train Accident: Cluster 6

Fig. 24. Train Accident: Cluster 7

Fig. 25. Train Accident: Cluster 8

Fig. 26. Train Accident: Cluster growth limited to one standard deviation from the mean time.

Fig. 27. Train Accident: Cluster growth limited to three standard deviations from the mean time.
C. Hurricane Matthew

Fig. 28. Hurricane Matthew: Map Plot

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Membership</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>181</td>
</tr>
<tr>
<td>2</td>
<td>162</td>
</tr>
<tr>
<td>3</td>
<td>654</td>
</tr>
<tr>
<td>4</td>
<td>379</td>
</tr>
<tr>
<td>5</td>
<td>300</td>
</tr>
<tr>
<td>6</td>
<td>577</td>
</tr>
<tr>
<td>7</td>
<td>2342</td>
</tr>
<tr>
<td>8</td>
<td>382</td>
</tr>
<tr>
<td>9</td>
<td>103</td>
</tr>
<tr>
<td>10</td>
<td>113</td>
</tr>
<tr>
<td>11</td>
<td>1183</td>
</tr>
<tr>
<td>12</td>
<td>599</td>
</tr>
<tr>
<td>13</td>
<td>3305</td>
</tr>
<tr>
<td>14</td>
<td>357</td>
</tr>
<tr>
<td>15</td>
<td>82</td>
</tr>
<tr>
<td>16</td>
<td>41</td>
</tr>
<tr>
<td>17</td>
<td>235</td>
</tr>
<tr>
<td>18</td>
<td>430</td>
</tr>
<tr>
<td>19</td>
<td>61</td>
</tr>
<tr>
<td>20</td>
<td>196</td>
</tr>
<tr>
<td>21</td>
<td>1674</td>
</tr>
<tr>
<td>22</td>
<td>73</td>
</tr>
<tr>
<td>23</td>
<td>843</td>
</tr>
<tr>
<td>24</td>
<td>116</td>
</tr>
<tr>
<td>25</td>
<td>612</td>
</tr>
</tbody>
</table>

Fig. 29. Hurricane Matthew: Cluster membership.

Fig. 30. Hurricane Matthew: Cluster 1

Fig. 31. Hurricane Matthew: Cluster 2

Fig. 32. Hurricane Matthew: Cluster 3
Fig. 33. Hurricane Matthew: Cluster 4

Fig. 34. Hurricane Matthew: Cluster 5

Fig. 35. Hurricane Matthew: Cluster 6

Fig. 36. Hurricane Matthew: Cluster 7

Fig. 37. Hurricane Matthew: Cluster 8

Fig. 38. Hurricane Matthew: Cluster 9
Fig. 51. Hurricane Matthew: Cluster 22

Fig. 52. Hurricane Matthew: Cluster 23

Fig. 53. Hurricane Matthew: Cluster 24

Fig. 54. Hurricane Matthew: Cluster 25

Fig. 55. Cluster growth limited to one standard deviation from the mean time.
ACKNOWLEDGMENT

Many thanks to Professor Jeremy Brown for advising this project, and giving advice and guidance. Another thanks to Rochester Institute of Technology for providing the resources needed to complete this project.

REFERENCES