LEVERAGING SOCIAL MEDIA TO MAP DISASTERS

Examining:
Camp Fire - Paradise, CA | November 8th - 25th 2018
Carr Fire - Redding, CA | July 23rd - August 30th 2018
Hurricane Harvey, Houston Metro Area | August 23rd - August 31st 2017

Team Members:
Patrick Wales-Dinan
Laura Luo
Andre Jimenez
Antony Paulson Chazhoor

PROBLEM STATEMENT:

When responding to disasters such as wildfires, hurricanes, floods and earthquakes, emergency workers have the critical task of being able to identify where survivors are located in order to quickly and accurately send assistance.

Given the rapid growth in wireless technology and the increasing likelihood that people all across the globe have a cell phone we want to attempt to tap into this technology in such a way that we can use social media posts asking for help and assistance to track survivors.

Social media can provide valuable information about the specifics of a disaster such as downed power lines, flooding streets, the spread of wildfire, density of smoke, etc. Additionally it can help identify isolated communities at risk, locations of survivors and areas where assistance teams should be sent for search and rescue. Social media can also provide additional information such as levels of damage, help map depths of flooding, identify where additional imagery/information needs to be collected and plan when and where resources should be allocated.

Here we attempted to specifically engage with the Twitter API in order to create tools that would allow us to geolocate people in need and/or areas in need of assistance and/or areas that should be avoided. The desired tools would have the ability to help locate problems regardless of the type of disaster.

DATA:

We specifically targeted three major disasters from the last 24 months. The first, Hurricane Harvey ravaged the gulf coast specifically the greater Houston metro area of Texas and Louisanna. The other two disasters were major wildfires here in California.

The first was known as Carr Fire. Carr Fire, was centered around the Shasta Recreation Area that encompasses Redding, CA. Carr destroyed over 229,000 acres of land and 1,600 buildings while claiming 8 lives. Finally Camp Fire, another destructive wildfire centered in Northern California. Camp Fire was not as large but was over ten times more fatal. Camp was centered in Paradise, CA and burned over 150,00 acres and claimed 86 lives.

The first challenge was to be able to pull enough geolocated data from Twitter's API. We found that only about 5% of unique tweets that we were able to procure were encoded with geolocation and only about 10% of **total** tweets were specific to people needing help or providing pertinent information about whether emergency teams needed to provide some type of assistance. The problem of obtaining relevant data was made even more challenging because Twitter only allows you to have 100 requests in a given month. Because of this we weren't able to have as many test pulls as we would have liked. Furthermore, Twitter API sends tweets to the requester from the most recent date to the farthest away date. This means that if we chose an endpoint that was too far from the start of the disaster we would not retrieve tweets that were as relevant and if we chose an end date that wasn't far enough away, we would not retrieve enough tweets for a robust data set.

It is important to be aware that Twitter does offer an enterprise level developer API that allows you to pull unlimited tweets from any time period that you desire. This API is not free to the public however and requires a subscription which we did not purchase for this project.

If these tools were being delivered to FEMA or disaster management specialists they would definitely need to invest in Twitter's enterprise level developer API service. This would give them access to all of the relevant data that they would require to locate people that were in need via Twitter.

A side note about the specific data that we observed from the tweets we pulled from each disaster: There was an incredibly high proportion of people who were tweeting about lost or found pets. It became clear that this would be very useful to help people at the humane society or the local animal shelters locate or reunite pets with their owners.

MODELING:

Our modeling process began by creating a corpus based on the text of the existing tweets we were able to pull from the Twitter API. The thought process was that we needed to use some inference and some data to decide which words would be most predictive of a tweet that was asking for help, reporting an urgent situation, or providing valuable information about where emergency personnel needed to be deployed.

To do this we first CountVectorized our tweets and looked at the most frequent words. It appeared that it was going to be quite difficult to categorize tweets that were referencing emergencies in the immediate present. For example it became glaringly obvious that someone could be referencing an emergency from days before that had already been resolved. Trying to differentiate between the signal and the noise was going to be extremely difficult.

What to do then? We explored many different options. First we decided to use a Words2Vec model to build a disaster corpus based on the words in our tweets. This would look through our corpus of words using Words2Vec and build out that corpus for us. Then we inferred that there were powerful help words or phases that we needed to add to that corpus making sure to remove any that were duplicates.

We then used Words2Vec to create a disaster corpus based on a similarity score from Wikipedia. This was our grand list containing our “help words” (words we deemed important), “disaster tweet words” (words taken directly from our tweets) “similar words” (words similar to all the words in this list based upon running Words2Vec with

Wikipedia). From there we filtered that list down to our 100 most relevant words with which to classify tweets.

Now we began the classification process. We used these words to filter through all of our tweets and classify them as emergency tweets if they contained two or more of the words that we had identified as important to displaying a need for assistance or important to alert emergency staff of a situation that needed their attention. This process yielded such a high percentage of tweets being classified as relating to an emergency that we quickly realized that we were using too many words that were not predictive. To gain some insight about what we could do to remedy this situation we determined that it was necessary to manually read through some (many) of the tweets that were specifically asking for assistance. From there we could come to a better understanding of which words would actually be playing an important role in classifying a tweet as an ‘emergency tweet’.

This process caused us to dramatically cut the number of words we used to about a dozen. While this was a little disconcerting given that we had spent so much time developing the list of words we hoped would be predictive, it was nonetheless a valuable process to go through and one that was meaningful to learning more about our dataset and the challenges it presented.

From there we were able to classify just over 100 tweets that pertained to an emergency. Those tweets were then plugged into 7 different classification models.

- Logistic Regression
- Random Forest Classifier
- Decision Tree Classifier
- Bagging Classifier
- AdaBoost Classifier
- Support Vector Classifier
- Gradient Boosting Classifier

We made sure to run each model through a GridSearch to come up with the best tuning parameters for each model and then evaluated them all based on a number of factors using the accuracy score and F1 Score for both the training and testing dataset and our number of False Positives, False Negatives, True Positives, and True Negatives.This yielded a very low F1 score and a large number of False Positives and False negatives.

What can we do to fix this? We decided that we should use oversampling techniques to fix the problem of imbalance between our minority and majority classes. After using oversampling techniques our model performed much better.

	train	test	F1-train	F1-test	true_neg	fal_pos	fal_neg	true_po
lr_class	0.998401	0.868106	0.998379	0.874715	170	31	24	192
forest_class	0.901679	0.846523	0.900566	0.848341	174	27	37	179
tree_class	0.884093	0.832134	0.886807	0.843750	158	43	27	189
ada_class	0.926459	0.846523	0.924466	0.843137	181	20	44	172
bag_class	0.599520	0.522782	0.332889	0.167364	198	3	196	20
svc	0.808153	0.633094	0.759036	0.451613	201	0	153	63
grad	0.948841	0.880096	0.947282	0.881517	181	20	30	186

The table above displays the results for our training and testing data sets. (*Note, this does not include any results from our validation set which we will discuss later). Here the ‘train’ and ‘test’ columns show the accuracy scores for training and testing. The F1 score is the harmonic mean between the precision and the recall and the true_neg, fal_pos, fal_neg, true_pos columns are self explanatory. The bagging classifier and support vector classifier clearly didn’t perform as well as the other models.

RESULTS & CONCLUSIONS:

We then used each model to aggregate results from our validation set. Our validation set was another batch of tweets that we pulled for each disaster that only contained geo-located tweets from the time period around the disaster. We ran each tweet through each model and recorded each model’s prediction for all the tweets in our validation set. If 3 or more models predicted that set to be a disaster related tweet, then we classified that tweet as true. Below is a snippet of a dataframe containing our aggregated results.

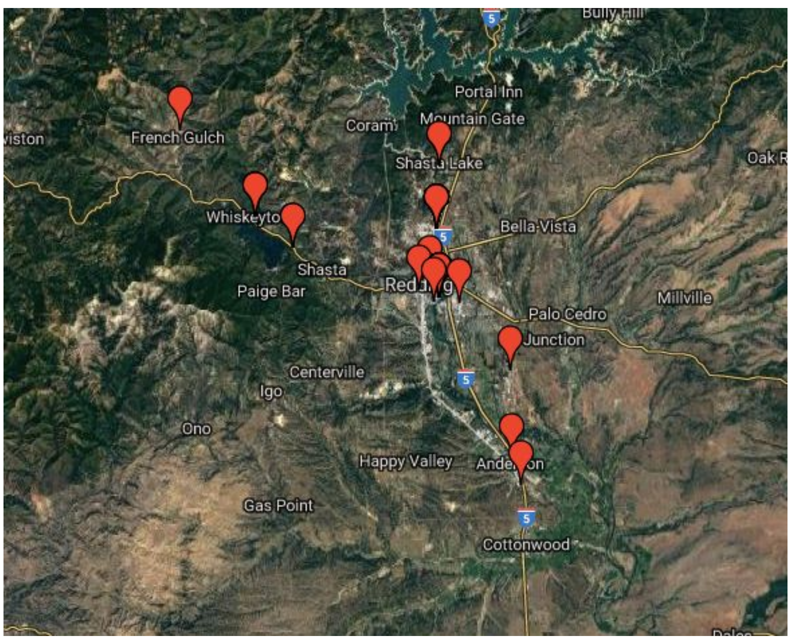
We discovered that 142 tweets of the 832 in our validation set were classified as pertaining to disasters. In doing a visual scan of the tweets it generally looked like about 40-60% of the tweets did end up being related to disasters.

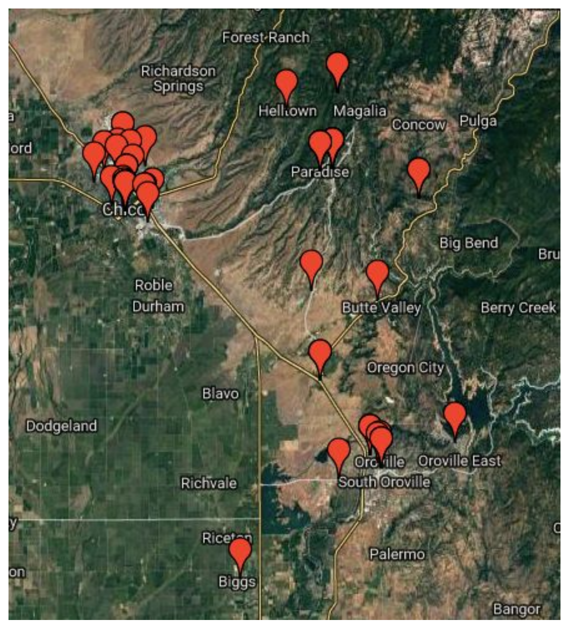
	lr_class	forest_class	tree_class	ada_class	bag_class	svc	grad	total
496	1	1	1	1	1	0	1	6
0	1	1	1	1	0	0	1	5
186	1	1	1	1	0	0	1	5
160	1	1	1	1	0	0	1	5

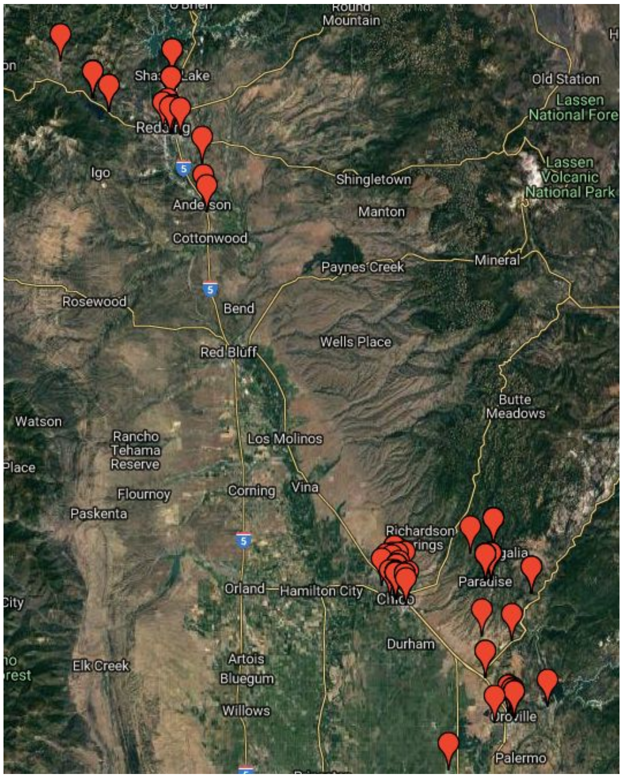
This leads us to conclude that the model would have to work in conjunction with an actual human. In this case, the model would scan the web for the tweets, return the tweets that are classified as pertaining to a disaster to a human and the human would read the tweets and make an interpretation about whether the tweet is one that needs a response. The user would then relay the geo-coordinates that were provided from the model to the emergency response people.

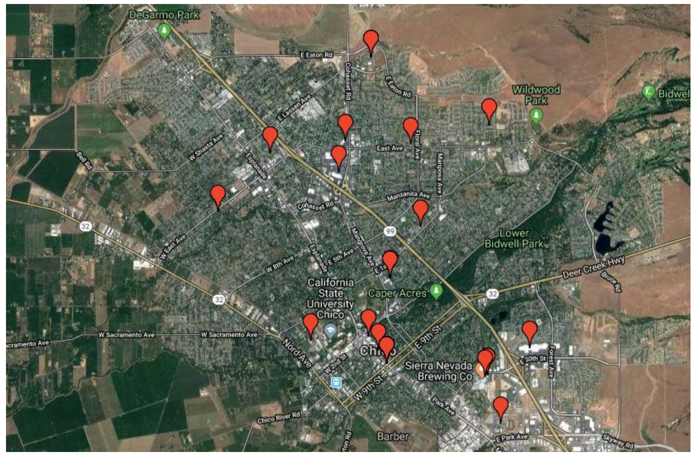
One of our initial problems was that most of the tweets did not include geo-location, so we would have to have access to some sort of disaster level twitter API in order to be sure we could respond to the tweets we received. In our case, we used tweets that **only** had geo-located coordinates attached so this wasn't a problem but in all likelihood, we would probably miss lots of people who potentially needed help.

After classifying these tweets we used [gmplot] which integrates with Google Maps and allows you to display coordinates on a map in order to visualize what is happening. Below are some examples of the maps we were able to create from the gmplot:









Geo-location thoughts: It seems important to consider that these points may or may not be the exact location. We have come across different sources that indicate that [Twitter has removed the precise location of geo-location tags](https://www.theverge.com/2019/6/19/18691174/)[1]. This would make the data that the average person could access potentially useless. This all circles back to a point we have made numerous times. If FEMA or different emergency personnel are going to attempt to implement tools such as this, then they need to be sure that the locations they are retrieving are exact and specific.

In summary, this project illuminated to us that there are opportunities to use real-time social media data to access important information during disasters or emergency situations. The key for any major company hoping to do this will be to ensure the quality of the data they are acquiring. Despite many of the challenges when attempting to leverage this information, we found the project to be useful and rewarding and it presents opportunities for further developments in the future. One final thought for companies like FEMA who might feel like this type of tool or one similar has great

[1] The Verge, Jon Porter, https://www.theverge.com/2019/6/19/18691174/

potential to be useful would be to partner with Twitter, Instagram and other companies to include an emergency feature on the app. Perhaps something that only can be activated by state and local emergency staff that sends out a warning to people's phones asking if there are in danger. They could have the option of tapping a button to instantly send their geo-location to EMS with a message about their situation. In order for these platforms to be leveraged properly it seems necessary to have coordination and cooperation between the platforms themselves and the respondents.