Leveraging Twitter to Map Disasters

Aurelia Lyon, Bhupesh Kumar, Joey Romness, Tonya Chernyshova

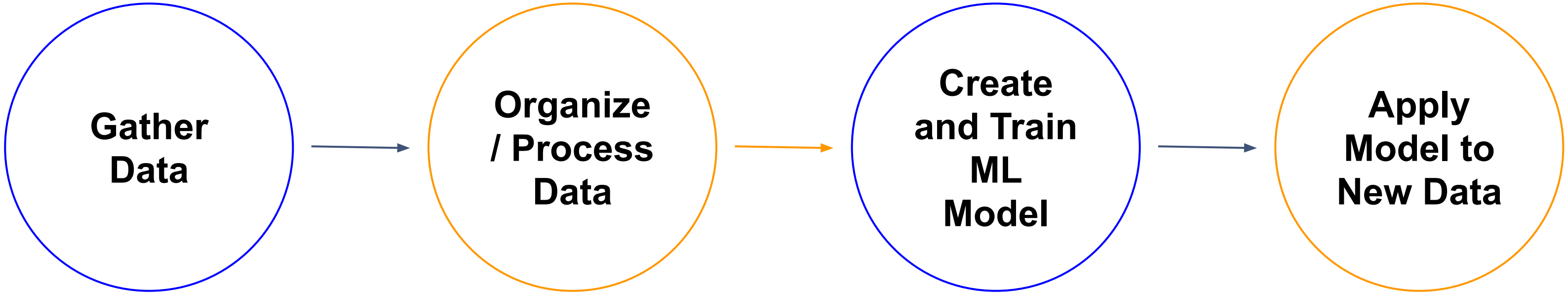
Problem Statement

The client has asked us to provide a tool that, through the usage of a customizable list of keywords that corresponds to disasters, monitors live streams of social media posts (in this case, Twitter) and then provides the geolocation of each post.

Executive Summary

1. Ultimately, we were able to create a tool that allows the client to input a customizable list of words that returns the geolocations of tweets that correspond with certain disasters.
   - Our tool should be considered a prototype at best because of the sizable amount of noise that gets labelled improperly alongside true disasters.
2. In our case, training a machine learning model to generalize to multiple different types of disaster did not work well. For best results, models should be trained on datasets that pertain to specific natural disasters (e.g fire data when looking for fire tweets, flood data when looking for flood tweets)

Approach



Gather Data

Organize / Process Data

Create and Train ML Model

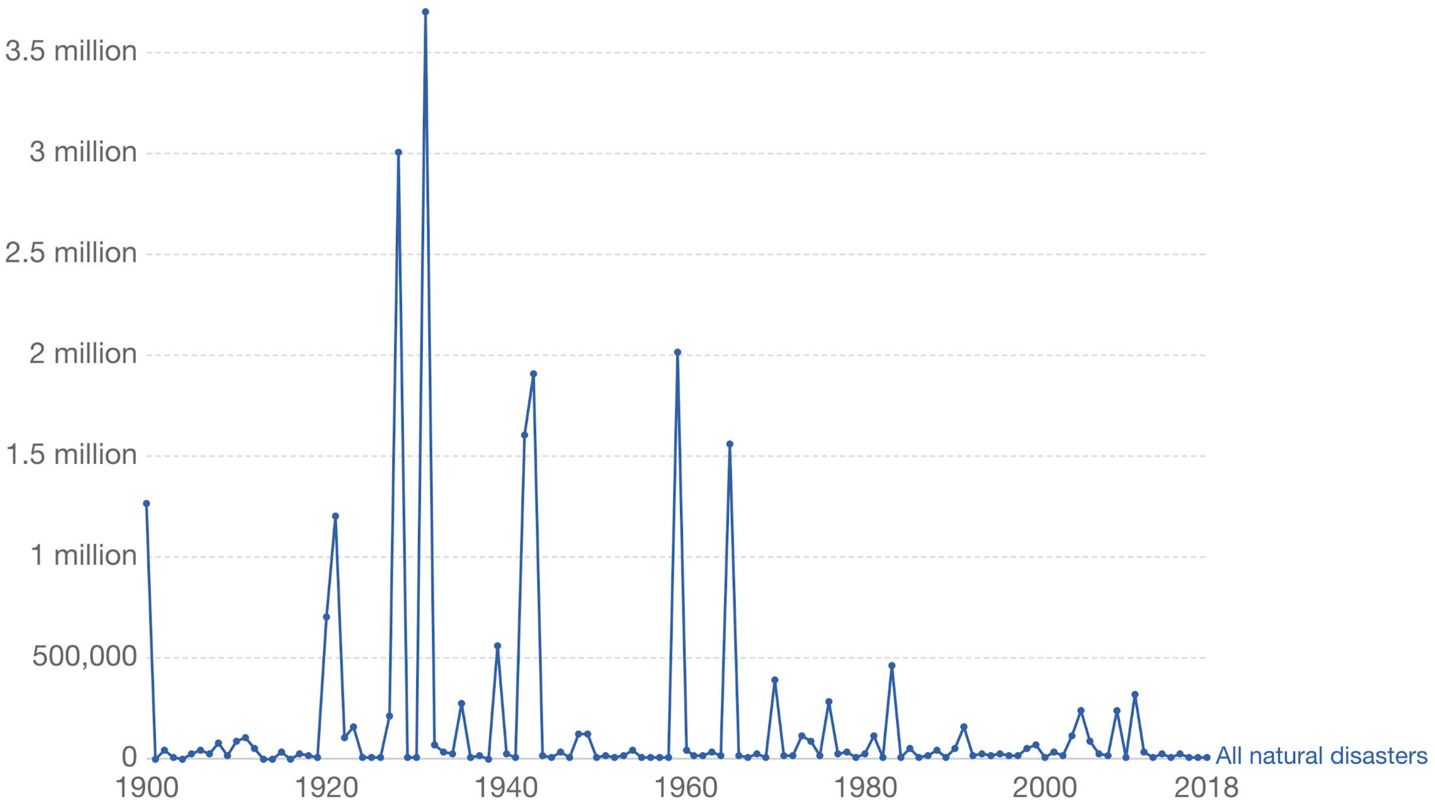
Apply Model to New Data

Background Information

- Because of external factors like Climate Change, certain natural disasters like wildfires and hurricanes/typhoons are increasing over time.
  - Due in most part to modern technology, the amount of deaths per natural disaster has gone down over time, despite the increasing frequency of such disasters.
- The goal of a project like this is to aid in further decreasing the number of deaths that occur when a natural disaster hits.
  - Ideally, by getting the geo-locations of tweets that pertain to disasters, emergency response teams can quickly ascertain where a disaster hit and where to focus their resources.

Global deaths from natural disasters, All natural disasters

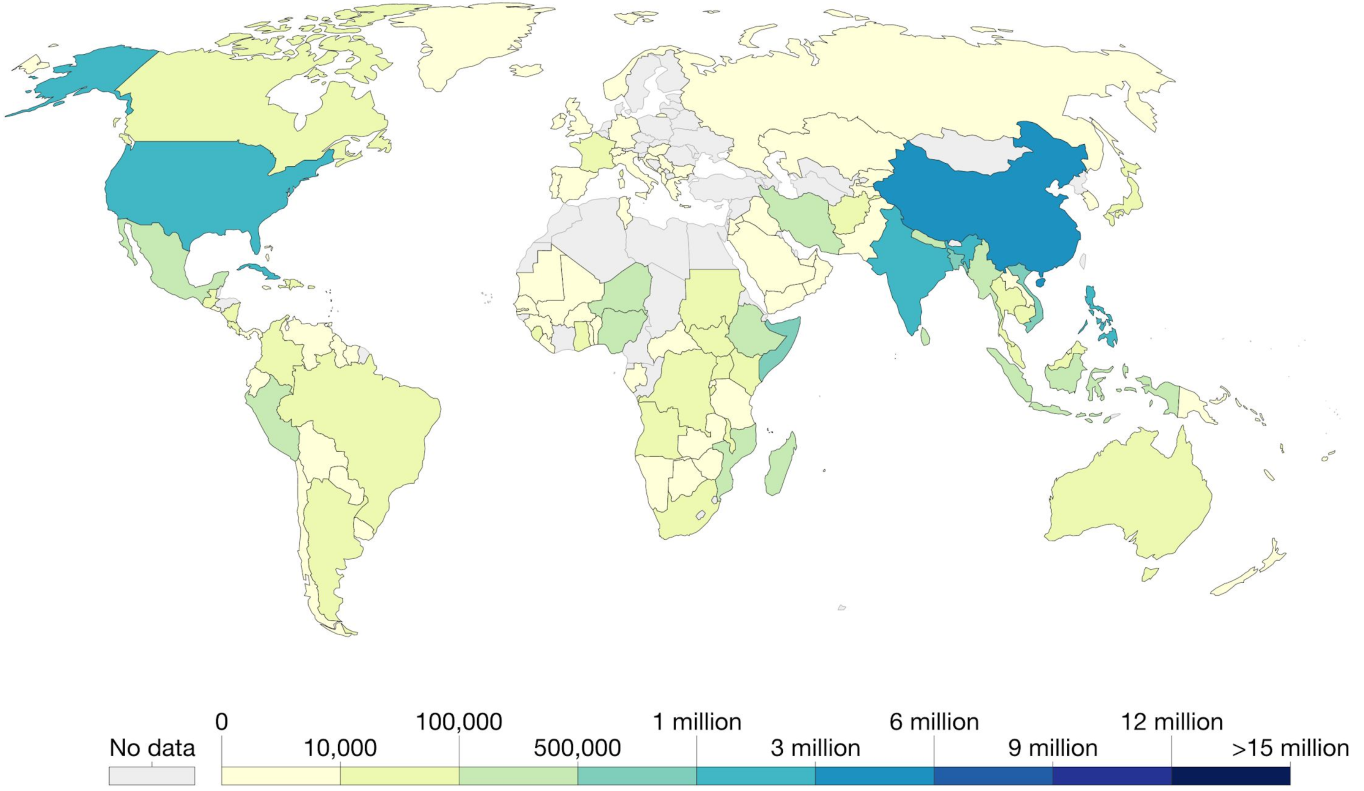
Absolute number of global deaths per year as a result of natural disasters. "All natural disasters" includes those from drought, floods, extreme weather, extreme temperature, landslides, dry mass movements, wildfires, volcanic activity and earthquakes.



Source: EMDAT: OFDA/CRED International Disaster Database, Université catholique de Louvain – Brussels – Belgium
OurWorldInData.org/natural-disasters/ • CC BY

Internally displaced persons from natural disasters, 2017

Internally displaced persons are defined as people or groups of people who have been forced or obliged to flee or to leave their homes or places of habitual residence, as a result of natural or human-made disasters and who have not crossed an international border.



Source: World Bank

OurWorldInData.org/natural-disasters • CC BY

Data Collection

- Testing data was gathered from Twitter's API, using the Python 'tweepy' library. Twitter's API returns a JSON file for each post. Lists of posts were taken from http://crisislex.org/: a website that provides repositories of crises-related social media data.

Data Collection

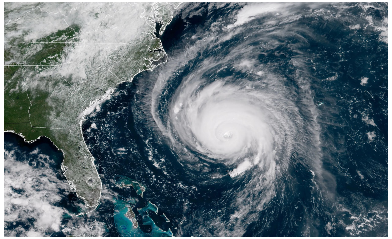
- We Defined a function for parsing data. The function accepts tweet IDs as arguments, then it sends a request to Twitter's API and saves all the information from tweet to a dictionary.
- From that a script was created that loops through list of ids and return csv.
- We automated collection using an AWS E2 instance.
- We ended up collecting about 100,000 tweets that originated around the time Hurricane Sandy hit the USA. These tweets came from all over the US.

Roadbumps - And By That I mean Giant Holes In the Road

- Initially, we were using a collection of tweets from CrisesLex that coincided with multiple different disasters. We intended to use these tweets as the training data for our ML models.
  - After training multiple models on this data, every model was making wildly inaccurate predictions as to whether or not a tweet pertained to a disaster or not when given new data, despite the models having high train-test scores initially.
- Potential Reasons for overprediction
  - Language Issues
  - Overgeneralization

The Fix: More Specific Datasets

- Much better predictions were received by training our models on data from specific disasters.
  - Through training our models on singular disaster types, our models were able to better predict disaster tweets when given new data while also getting higher Train-Test scores on our base datasets.
- For the purposes of our modeling section, all discussion will pertain to scores and methods used on single disaster datasets.
  - Specifically, we did most of our model training on Hurricane Sandy tweets.



Modelling

- Once data collection and data cleaning was done, several different pipelines were instantiated using different word vectorization and classification methods.
- Word Vectorizers
  - TF-IDF

    Count Vectorizer
- Classification Methods
  - Logistic Regression
  - SVM
  - Random Forest
  - Multinomial NB

Logistic Regression

With TF-IDF

Train: 92.96%

Test: 92.20%

Notable Best Params:

TFIDF Max Features: 500

TFIDF N-Gram Range: 1,1

Stop Words: None

With Count Vectorizer

Train: 95.11%

Test: 92.76%

Notable Best Params:

Count Vec Min DF: 6

Count Vec N-Gram Range: 1,2

Count Vec Stop Words: English

SVM

With TF-IDF

Train: 61.33%

Test: 61.31%

Notable Best Params:

TFIDF Max Features: 500

TFIDF N-Gram Range: 1,1

Stop Words: None

With Count Vectorizer

Train: 89.60%

Test: 89.68%

Notable Best Params:

Count Vec Min DF: 2

Count Vec Stop Words: English

Random Forest

With TF-IDF

Train: 98.82%

Test: 91.72%

Notable Best Params:

TFIDF Max Features: 1000

TFIDF N-Gram Range: 1,2

Stop Words: English

With Count Vectorizer

Train: 96.02%

Test: 92.36%

Notable Best Params:

Count Vec Min DF: 4

Count Vec N-Gram Range: 1,2

Count Vec Stop Words: English

Multinomial Naive Bayes

With TF-IDF

Train: 88.78%

Test: 84.88%

Notable Best Params:

TFIDF Max Features: 2000

TFIDF N-Gram Range: 1,2

Stop Words: None

With Count Vectorizer

Train: 89.90%

Test: 87.05%

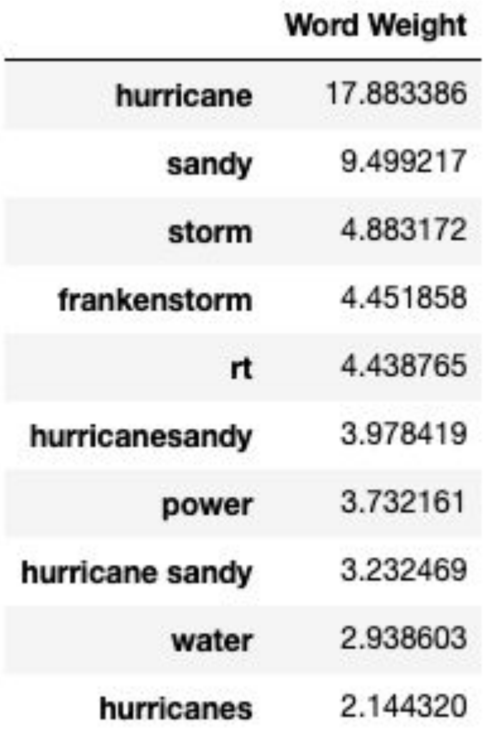
Notable Best Params:

Count Vec Min DF: 4

Count Vec Stop Words: None

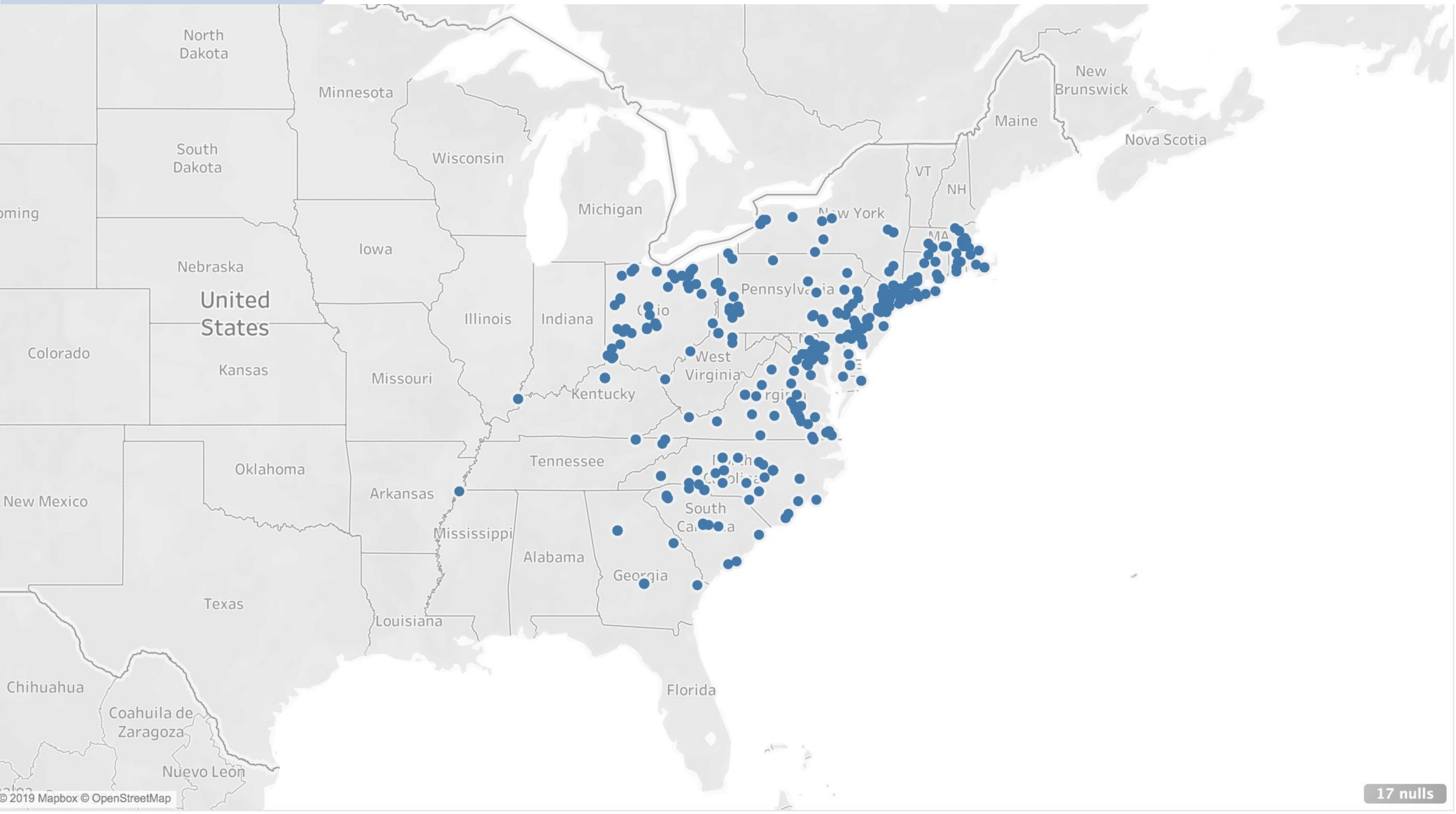
Model Used For Predictions: Logistic Regression with TF-IDF Vectorization

- Because of its high accuracy and low variance train test scores, we chose to make our data predictions using an LR model with TF-IDF vectorization. The image shown here shows the highest “weight words” for this model. Because of certain easily visualized problems that you can see in the dataset to the right, this model will likely work best with Hurricane Sandy tweets, but it should also generalize well to any hurricane.



Word	Weight
hurricane	17.883386
sandy	9.499217
storm	4.883172
frankenstorm	4.451858
rt	4.438765
hurricanesandy	3.978419
power	3.732161
hurricane sandy	3.232469
water	2.938603
hurricanes	2.144320

When making predictions on our collected tweet data, our model determined that out of the 100,000 tweets, about 527 were relevant to disaster.



© 2019 Mapbox © OpenStreetMap

Real World Application?

- In order to provide a viable and complete tool for the client, once our predictive model was developed we wrote a new set of code for interacting with the Twitter API. The new code does the following:
    a. Allows the client to input a customizable list of words to the twitter API
    b. From that customized list, it continuously collects english tweets from Twitter and returns them to the Client in a dataframe containing both Tweet Text and Geolocations of the tweets.
    c. From there, the client can run our predictive model on the gathered tweet data to identify which of those tweets are relevant to disaster.
    d. From the selection of Tweets labelled as disaster relevant, geolocations are displayed in Latitude and Longitude coordinates that can then be input into mapping software to display where the tweet originated from.

Conclusion and Future Recommendations

- Problem Statement Solved?
  - We have created for our client a tool that can take a customized list of keywords and return the tweets that correspond to those keywords. It can then identify which of those tweets likely indicate disaster and it can also give the geolocation of each tweet.
- Limitations
  - While we do have a working tool, it should be considered solely a prototype.
    - Too much noise in positively predicted tweets.
    - The model itself cannot detect the urgency of a tweet.
  - The Tool currently can only effectively predict for one disaster at a time.
- Future Recommendations
  - More Data tailored for the tool → Hand labelled disaster data
  - Removal of specific disaster words like “Katrina” and “Sandy” for better prediction ability on new data.
  - Find a way for the model to accurately predict for multiple different disaster types at once.