Leveraging Twitter to Inform Disaster Response Time

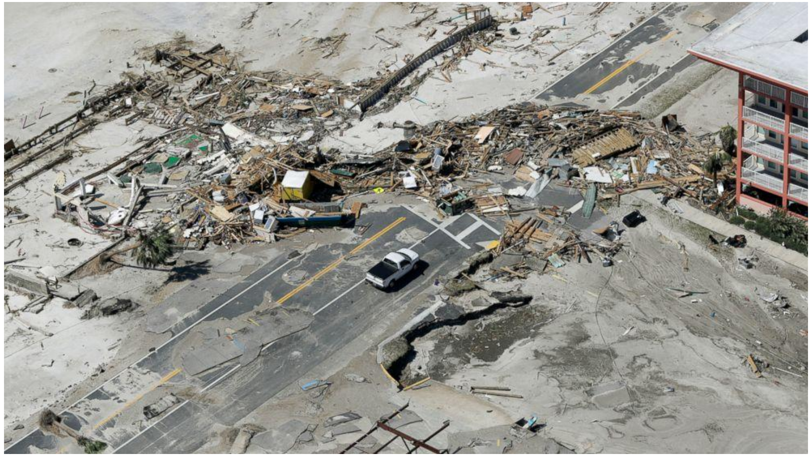
Jillian Berman,
Sree Gollakota, Zhan Yu

Problem Statement

In order to better improve emergency response times during disasters, this project aims to leverage social media to identify real time road closures or damaged roads, and other blocked routes that may affect travel time, traffic lights, safety, etc.

Case Study: Hurricane Michael

- Made landfall October 10th, 2018
- 74 deaths were attributed to the storm.
- Caused an estimated $25.1 billion USD in damages.



Workflow/ Methodology

1.) Data Collection:
    - Scraping tweets from GetOldTweets3 library
2.) Data Analysis/Preprocessing:
    - NLP cleaning
3.) Classification between traffic incident tweets and normal tweets:
    - Use old tweets to train model for live classification
4.) Pull live tweets to feed into the classification model:
    - Scraping live tweets with Tweepy library and Twitter API
    - Run classification model on live tweets
5.) Map emergency tweet locations

Data Collection

- Used GetOldTweets3 Python library
- Date Range:
  - 10/01/2018 - 10/15/2018
- Twitter accounts scraped:
  - @BayCountyTMC , @fl511_panhandl & @WJHG_TV
- Keyword query:
  - ‘HurricaneMichael’ and ‘850strong’ for example
- Training dataset contained 4,350 tweets.

Gathering Live Tweets

- Using the Tweepy Python library and a Twitter API we were able to scrape live tweets for the past 7 days.
- Scrapped the usernames and the same keywords we'd scrapped from our historical data, we were able to gather a dataset of 1014 tweets.

Data Analysis / Preprocessing

Cleaning tweets:

- Remove website links
- Remove non-letters
- Convert to lower case
- Remove basic stopwords

Options:

- Customize more stopwords
- Stemming/Lemmatizing

Modeling

- Corpus Vectorizers
  - CountVectorizer() or TFIDFVectorizer()
  - Stop_words: English
  - Monogram, bigram, and trigram ranges
- Models
  - Random Forest, AdaBoost Classifier, Support Vector Machine for Classification
  - Choose best model by comparing training, and testing accuracy score
- Validation
  - Confusion matrix: test for optimized sensitivity

Modeling

- <u>Best model #1:</u> Count-vectorized AdaBoost Model
- Low bias and low variance in accuracy scores
- Outperforms baseline

Best Parameters		**Best Train Score**	**Best Test Score**	**Baseline**
cvec ngram range	(1,1)	0.99	0.973	0.841
stop words	None			

Modeling

- Best model #2: Count-vectorized Support Vector Machine for Classification
- Low bias but slightly larger variance than AdaBoost model

Best Parameters		**Best Train Score**	**Best Test Score**	**Baseline**
cvec ngram range	(1,1)	0.995	0.951	0.841
stop words	None			

Modeling

- Should be able to predict better for false negatives
- Compare sensitivity of AdaBoost model and Support Vector Machine

Count Vectorized AdaBoost	**Actual Positives**	**Actual Negatives**	**Sensitivity**	**Specificity**
Predicted Positives	True Positives ***148***	False Positives ***4***	0.86	.996
Predicted Negatives	False Negatives ***25***	True Negatives ***909***		

Modeling

- Should be able to predict better for false negatives
- Compare sensitivity of AdaBoost model and Support Vector Machine

Count Vectorized SVM for Class	**Actual Positives**	**Actual Negatives**	**Sensitivity**	**Specificity**
Predicted Positives	True Positives ***121***	False Positives ***1***	0.699	.999
Predicted Negatives	False Negatives ***52***	True Negatives ***912***		

Modeling

- Count-vectorized AdaBoost is more sensitive and will be used for mapping
  - Minimizes false positives
  - Better able to recognize emergencies and traffic incidents without ignoring true emergencies and traffic incidents
- Model limitations:
  - Model actually classifies between tweets from specific accounts and tweets without accounts specified
  - May not perform well with live tweets

Extracting locations with spaCy

- Using the spaCy Python packages
- Training our own spaCy model by providing many examples to meaningfully improve the system, since the default spaCy model performs poorly.

New: Object on roadway in Okaloosa TWN on I-10 west INS at MM 51 MM , right lane LN blocked. Last updated at 04:45:39PM. #fl511

- With trained spaCy model, we were able to extract locations from tweets.

Getting Latitude & Longitude

- HERE.com provides mapping and location data services.
- Using the herepy python package and HERE Geocoder API we were able to get latitudes and longitudes from most of addresses we extracted from historical tweets.

Mapping

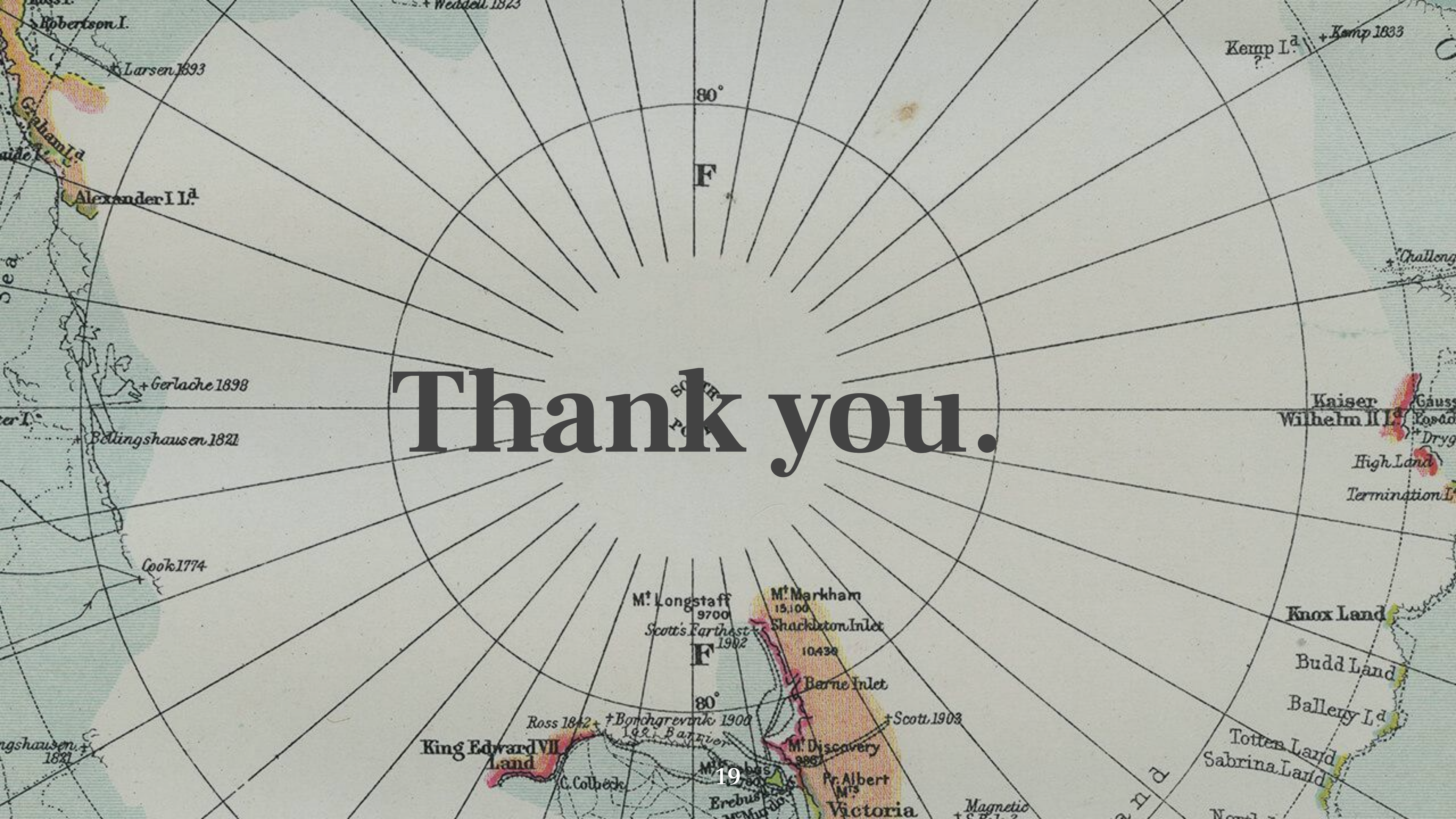
- Using shapley, GeoPandas and folium we plotted the points we were able to generate from spaCy processing of the tweets.
- Map Demonstration

Conclusions and Limitations

- Gathering tweets
  - Model will perform better if classification assumptions of gathered training data are more robust
- spaCy processing
  - spaCy performs better when we train our model with more examples
  - Training the model is time intensive though
- Coordinate extraction
  - From our training data set we were able plot 20% of the tweets.

Next Steps

- From here if you sign up for a Google Maps API you can integrate live traffic conditions, as well as point to point instructions.
- Improving classification assumptions to be more specific to the tweet context as opposed to generalizing 511 account tweets as positive class and others as negative class



Robertson I.
Larsen 1893
Weddell 1823
Graham Ld
Alexander I Ld.
Gerlache 1898
Bellingshausen 1821
Cook 1774
80°
F
Kemp Ld.
Kemp 1833
Kaiser Wilhelm II Ld.
High Land
Knox Land
Budd Land
Balleny Ld.
Totten Land
Sabrina Land
Mt Longstaff 9700
Scott's Farthest 1902
Mt Markham 15,100
Shackleton Inlet
10,430
Barne Inlet
Scott 1903
Ross 1842
Borchgrevink 1900
King Edward VII Land
C. Colbeck
Mt Discovery
Pr. Albert Mts
Victoria

Thank you.