Leveraging Social Media to Map Distress Calls

General Assembly DSI NYC

Preeya Sawadmanod

Nick Read

Andrew Sternick

Agenda

1. The Data Science Problem
2. The Goal of our Project and our Approach
3. Data Collection
4. Data Cleaning
5. Exploratory Data Analysis
6. Modeling
7. Mapping
8. Limitations, Next Steps, and Conclusion

The Data Science Problem

Can we use social media to map and identify locations where survivors of a disaster need assistance?

- ❑ Social media is resilient, via cell tower infrastructure
- ❑ Power outages, flooding can disable other communication channels
- ❑ Useful supplement to existing disaster response infrastructure

Choosing our Social Media Platform

Platform	Considerations
Twitter	Accessible API with 3rd party support
Facebook	Privacy restrictions limit usefulness
Instagram	Increased API restrictions 12/2018 and 10/2019

Our Goal

- Twitter is our social platform, due to ease of use of API, widespread adoption, and geotagging
- Turn an unsupervised problem into a supervised one for further analysis
- We measure our success by whether we can turn historical tweets into a usable map

Our Approach

Geographic focus is the big US east coast cities heavily impacted by Hurricane Sandy

Web Scraping by using APIs to collect Twitter data

NLP to categorize urgent and non-urgent tweets

Mapping the count of urgent tweets in selected cities through time, to indicate where best to deploy disaster response resources

Data Collection

Twitter API

- Accessible API with generous terms of use
- We targeted specific twitter accounts (SandyAid, FEMA, government offices) which allowed us to gather historical tweets from the period of the hurricane
- Scraping without account filter is limited to 7 days

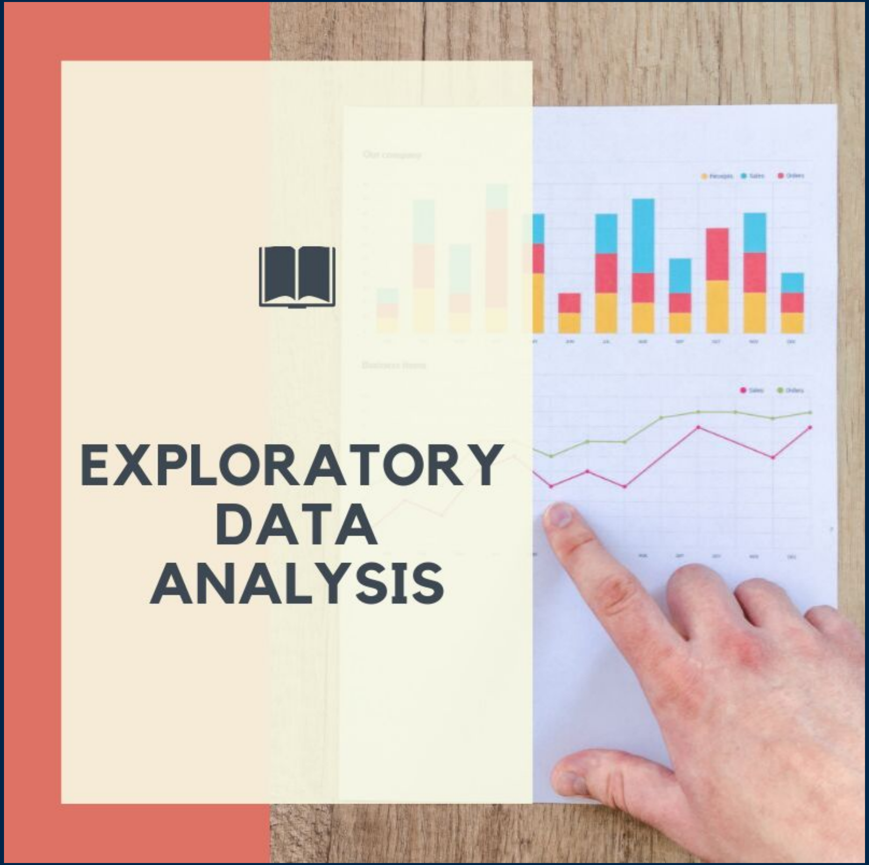
Data Collection

TwitterScraper

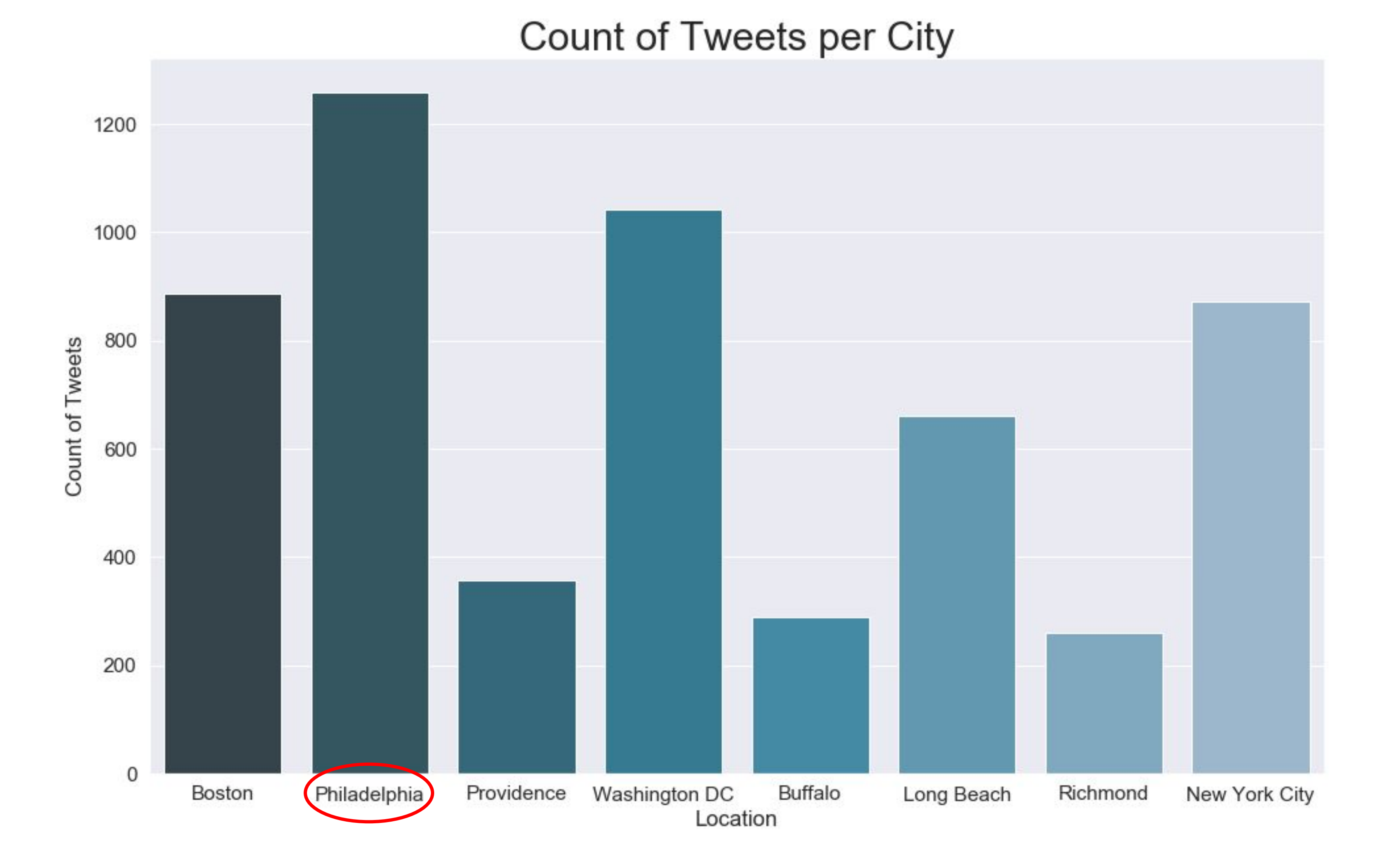
- Geocode feature allows filtering within 10 mile radius of major US cities impacted by Hurricane
- Restrict query to keywords “rescue”, “help”, “urgent rescue”, “urgent help”, help needed”, #HurricaneSandy, or #HurricaneSandyHelp
- Initial scrape date range was months of October and November 2012

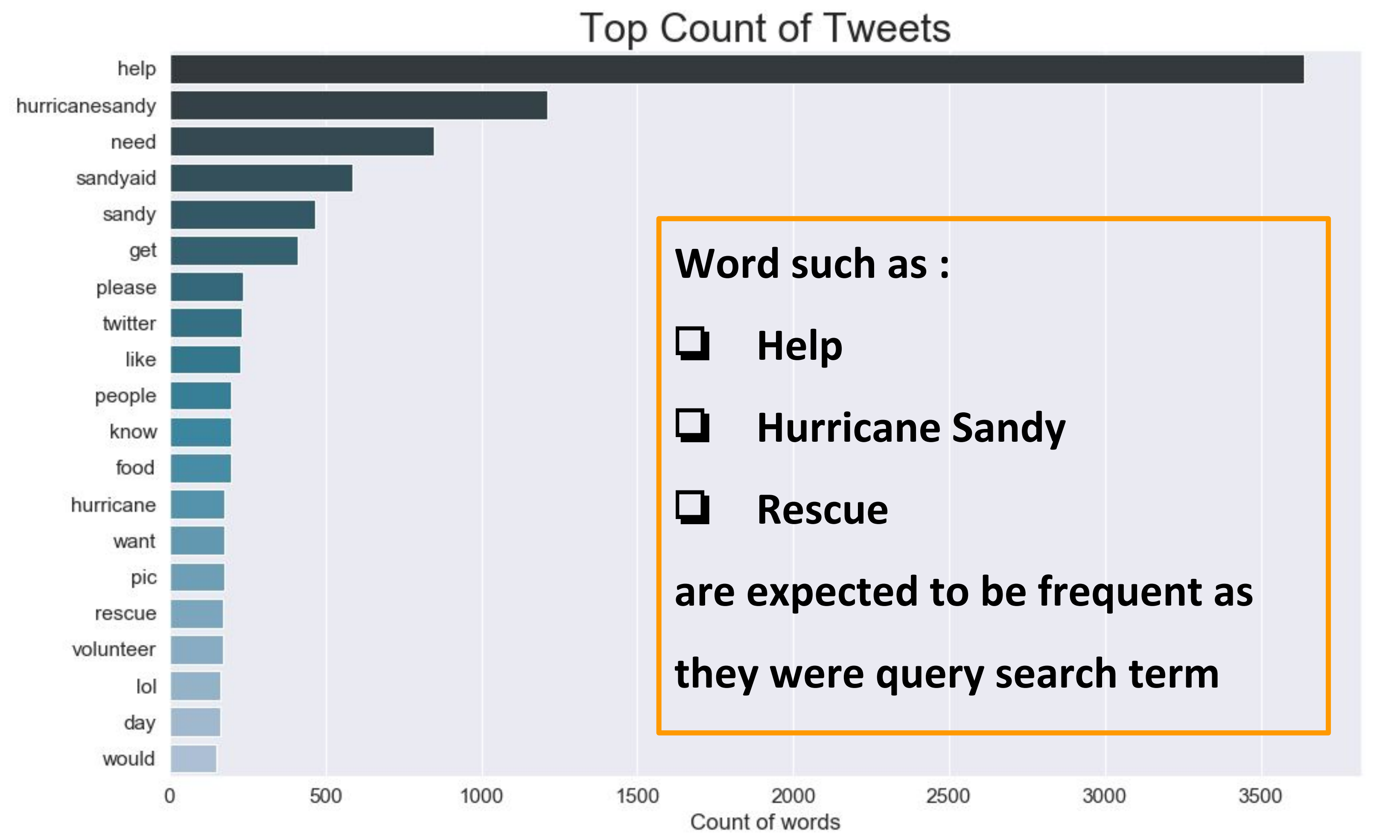
Data Cleaning

- **Total of 25,000 tweets**
  - More than 6000 duplicates
- **Limited two weeks range**
  - 10/27/12 to 11/13/12
  - remaining tweets of ~ 7000
- **Removal of URLs, numbers, and stop words**



EXPLORATORY DATA ANALYSIS

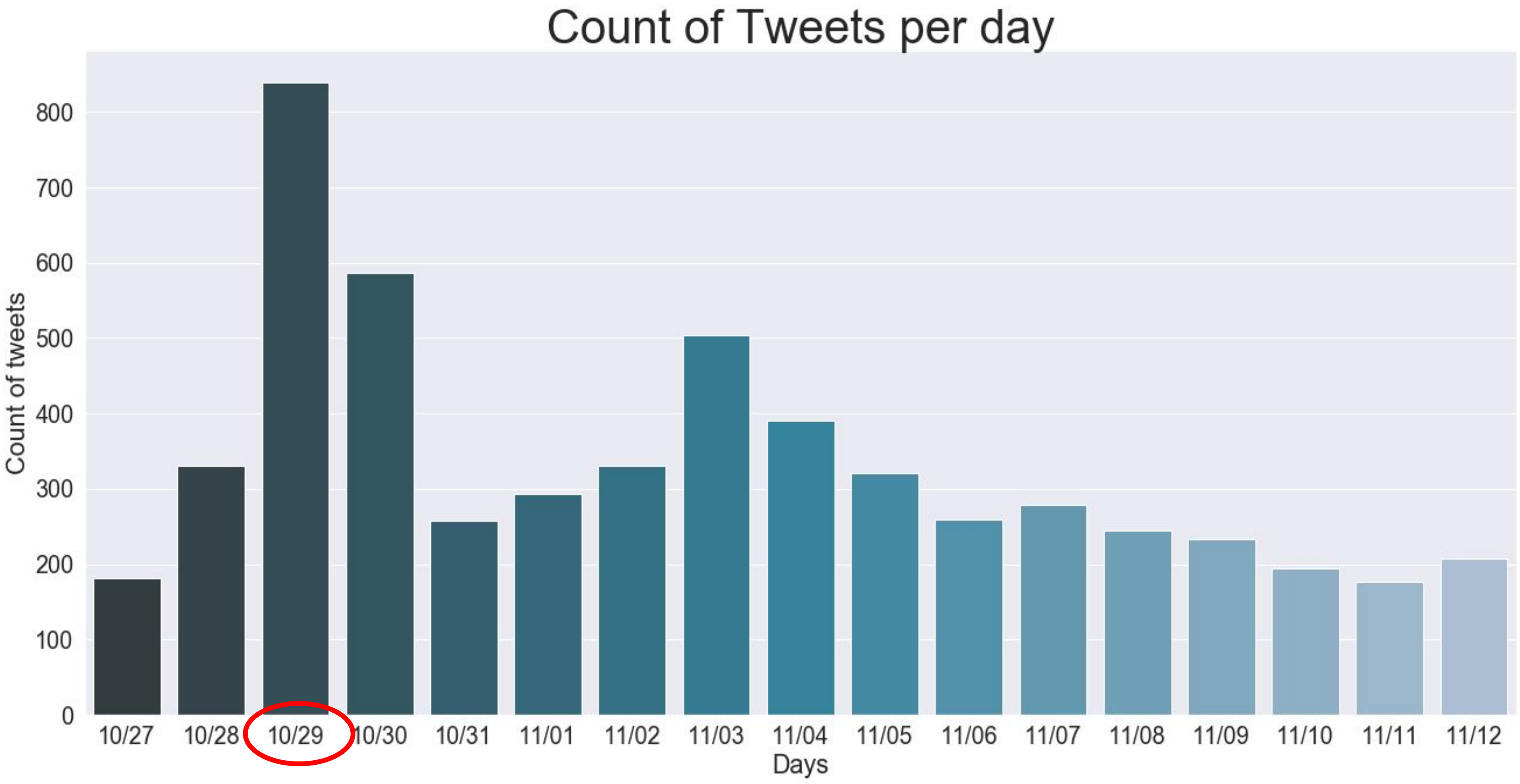

Count of Tweets per City
Count of Tweets
1200
1000
800
600
400
200
0
Boston
Philadelphia
Providence
Washington DC
Buffalo
Long Beach
Richmond
New York City
Location



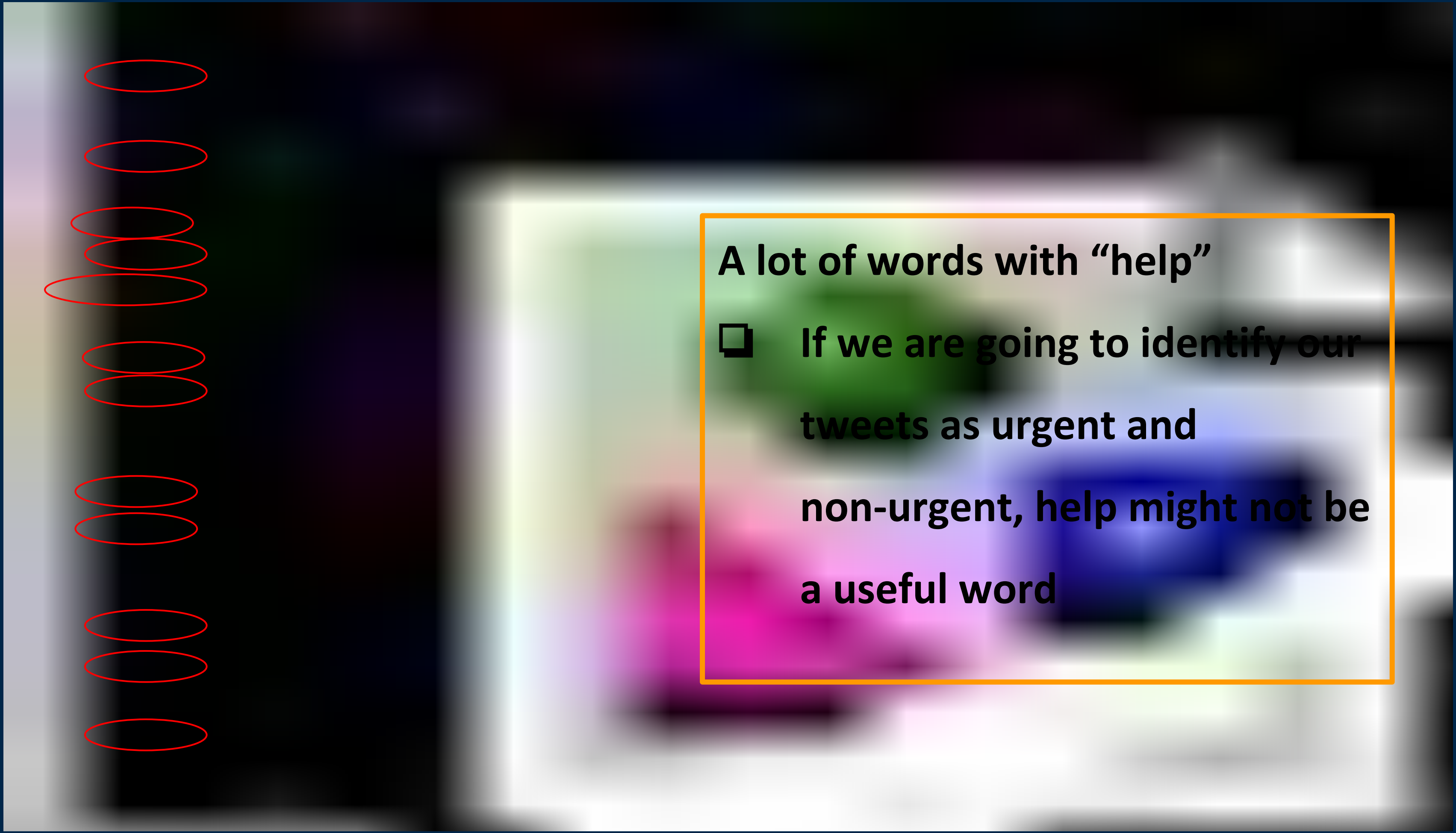
Word such as :

- **Help**
- **Hurricane Sandy**
- **Rescue**

are expected to be frequent as they were query search term


Count of Tweets per day
Count of tweets
800
700
600
500
400
300
200
100
0
10/27
10/28
10/29
10/30
10/31
11/01
11/02
11/03
11/04
11/05
11/06
11/07
11/08
11/09
11/10
11/11
11/12
Days

Looking at top 20 frequent words in Tweets (2 ngrams)

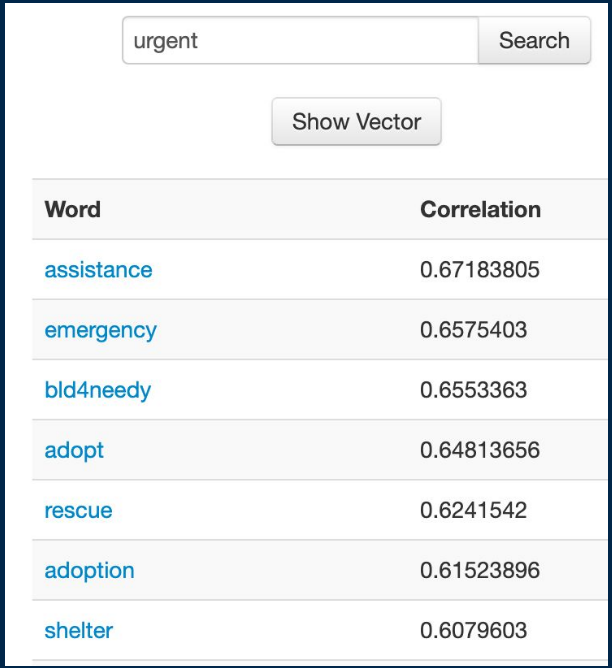


A lot of words with “help”

- **If we are going to identify our tweets as urgent and non-urgent, help might not be a useful word**

Identify urgent and non urgent words

Sample case:
Identifying related words to “urgent”
with [Word2Vec model]


urgent Search

Show Vector

Word	Correlation
assistance	0.67183805
emergency	0.6575403
bld4needy	0.6553363
adopt	0.64813656
rescue	0.6241542
adoption	0.61523896
shelter	0.6079603

List of “urgent” words

- ❑ Emergency
- ❑ Assistance
- ❑ Rescue
- ❑ Shelter
- ❑ Food
- ❑ Medical

List of “non-urgent” words

- ❑ Twitter
- ❑ Picture
- ❑ LOL
- ❑ Like
- ❑ School

Preprocessing: Tokenization

Principle is breaking the body of the text down to its constituent words (token)

Example: **Tweet Original: “urgent volunteer needed hurricane sandy shelter**

Tokenized: [“urgent”, “volunteer”, “needed”, “hurricane”, “sandy”, “shelter”]

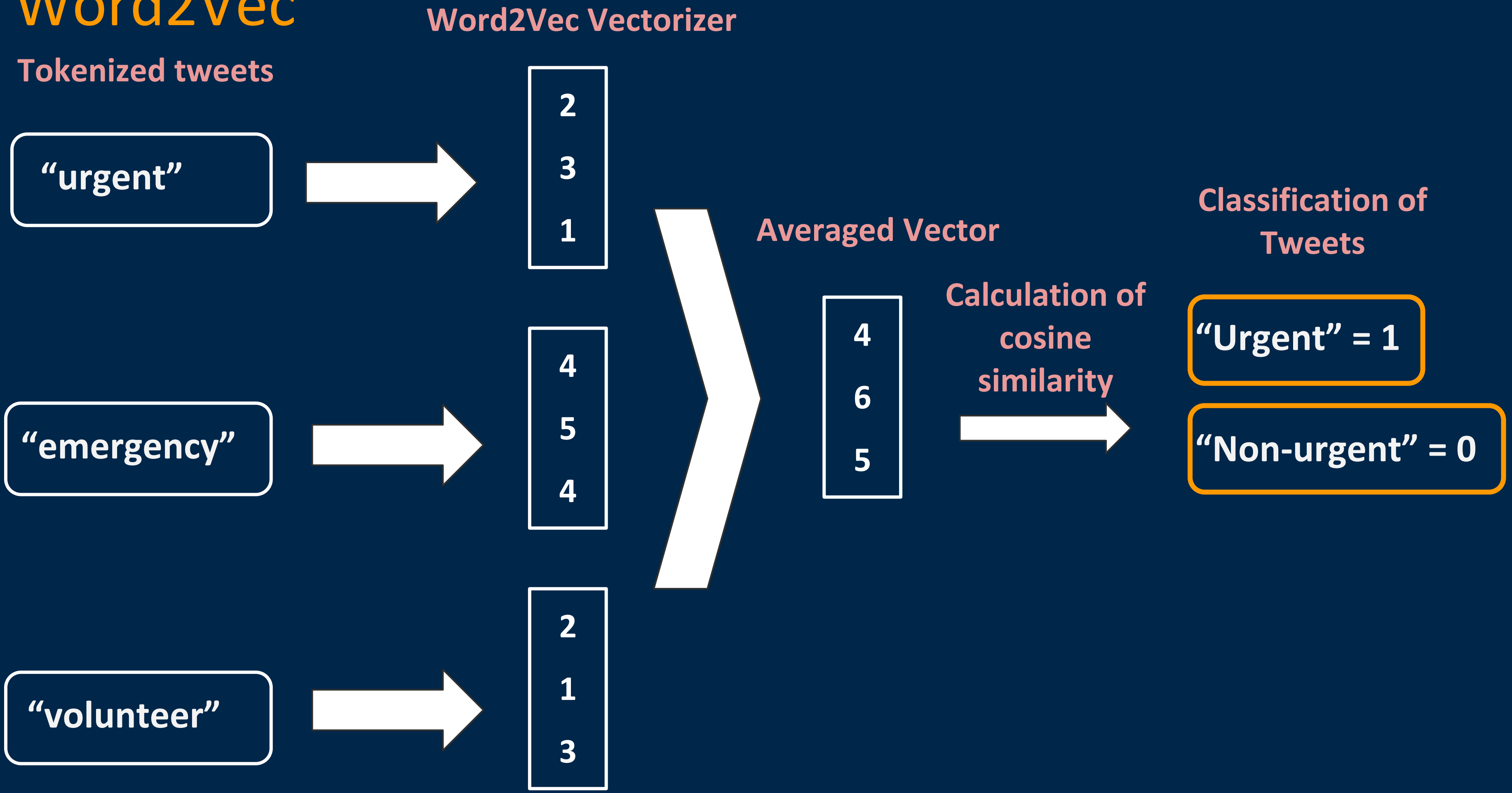
Word2Vec Model

Concept:

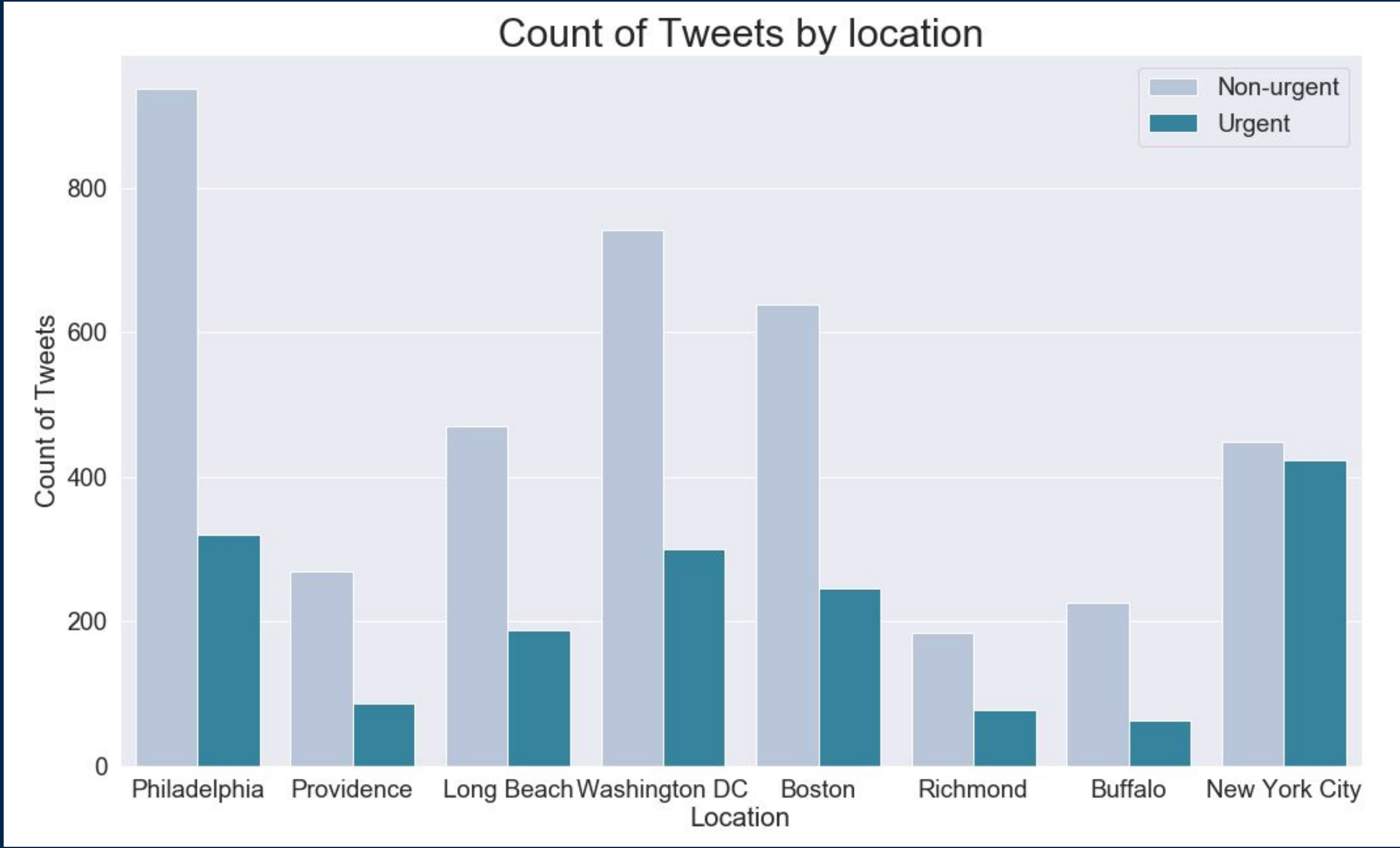
Each word is assigned a vector positioned in the space such that words in similar contexts will be positioned closely together

1. Training on Google News corpus (3M words)
2. Define our bag of words list (Urgent & Non-urgent)
3. Vectorize of list of tokenized tweets
4. Using cosine similarity to classify urgent (1) and non-urgent (0).

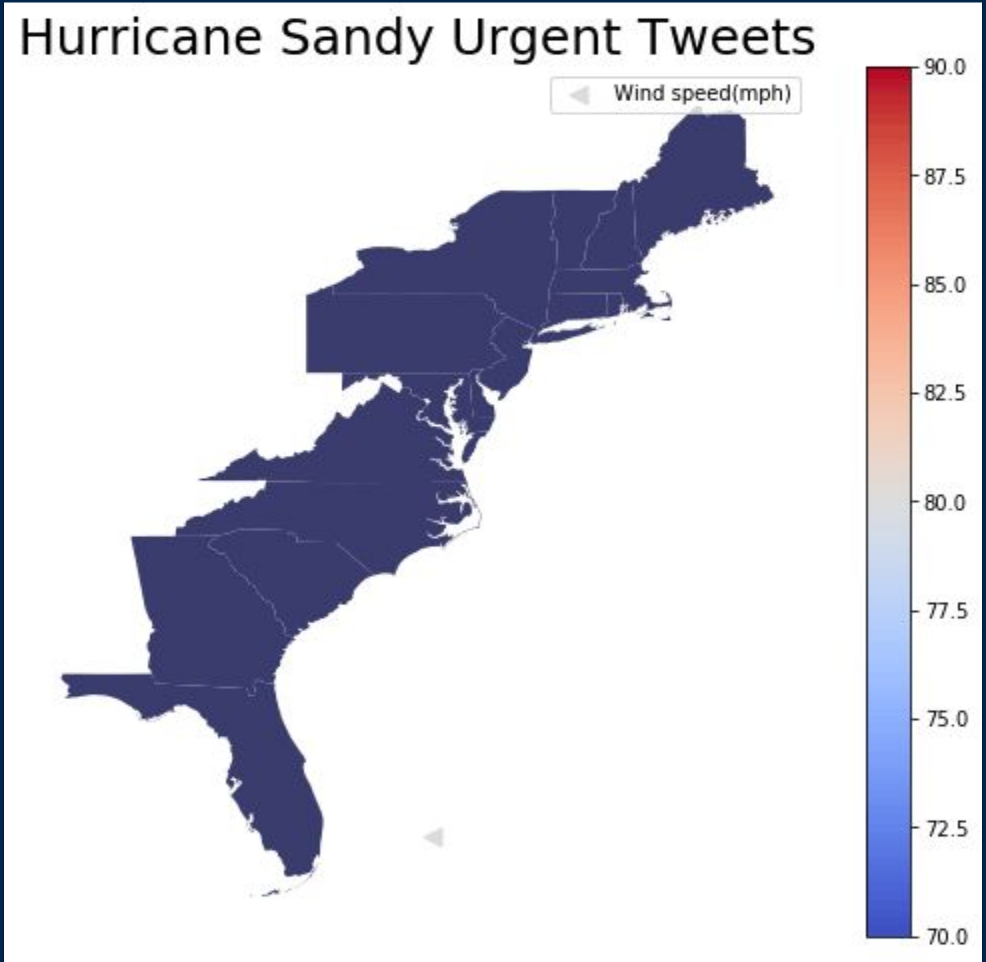
Word2Vec

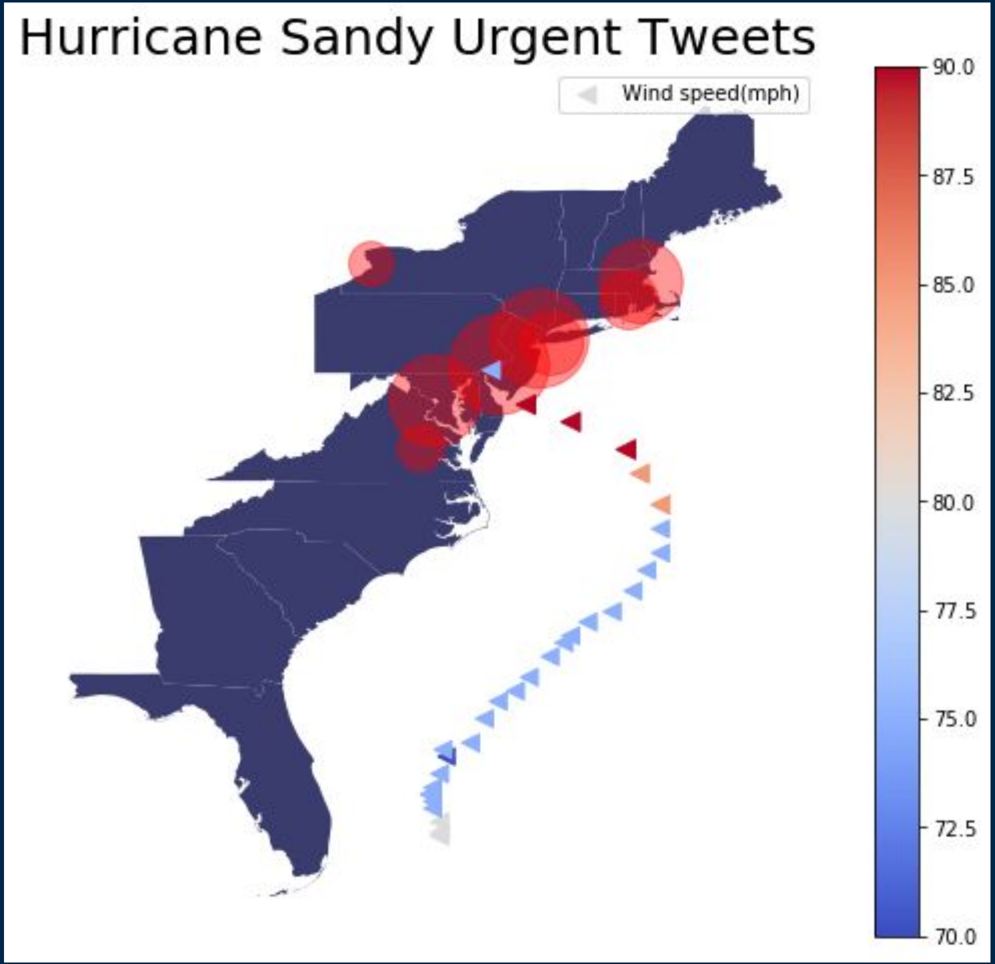


Word2Vec Vectorizer
Tokenized tweets
"urgent"
"emergency"
"volunteer"
2
3
1
4
5
4
2
1
3
Averaged Vector
4
6
5
Calculation of cosine similarity
Classification of Tweets
"Urgent" = 1
"Non-urgent" = 0



Count of Tweets by location
Non-urgent
Urgent
Count of Tweets
0
200
400
600
800
Philadelphia
Providence
Long Beach
Washington DC
Boston
Richmond
Buffalo
New York City
Location


Hurricane Sandy Urgent Tweets
Wind speed(mph)
90.0
87.5
85.0
82.5
80.0
77.5
75.0
72.5
70.0


Hurricane Sandy Urgent Tweets
Wind speed(mph)
90.0
87.5
85.0
82.5
80.0
77.5
75.0
72.5
70.0

Limitations

- Twitter API access
  - Paid access would allow to add more useful Geolocation
  - Forced to focus on major metropolitan areas
  - Limited in scope of targetable users.
- Word2vec
  - We must trust result and making assumptions (NLP)
  - Words selected could have been improved
  - Generic words selected from FEMA tweets

Limitations

- **No real time updates**
  - Our data is from flhurricane.com a hobbyist platform
  - The data is aggregated and then posted days after causing considerable lag
  - Our deliverable is contingent on lagged date and is not yet useful for a real time disaster scenario
- **Content is aggregated from the internet**
  - We must trust that the people tweeting are sincere about their posts
  - We collect a large amount of noise in the process
  - Useful data is more difficult to come by for training out models

Next Steps

- **Working with a better API**
  - Allows for faster turnaround time for mapping disasters
  - Allows to better Identify HotSpots on our map
- **Run model on different historical data sets**
  - We would like to make use of different natural disasters
  - Account for differences to build a better model to suit all disasters
- **Add more cities to current data set**
  - Allows for a more detailed path of the disaster
  - Allows us to better deploy emergency personnel

Conclusion

- **Proof of concept**
  - Generate a map of disaster
  - Clearly visualize the areas were resources should diverted
- **Use case**
  - Ability to show hot spots where a disaster occured
  - Allows rescue crews to divert attention to large scale disasters
- **Improvements**
  - Better API
  - Increase scope

Q&A