Optimizing Evacuation Routes using Real-Time Traffic Information

By Nick Vega, Dylan Lunde, & Alex Simon

Data Science Process

1. Define the problem.
2. Obtain the data.
3. Explore the data.
4. Model the data.
5. Evaluate the model.
6. Answer the problem.

Step 1: Define the Problem

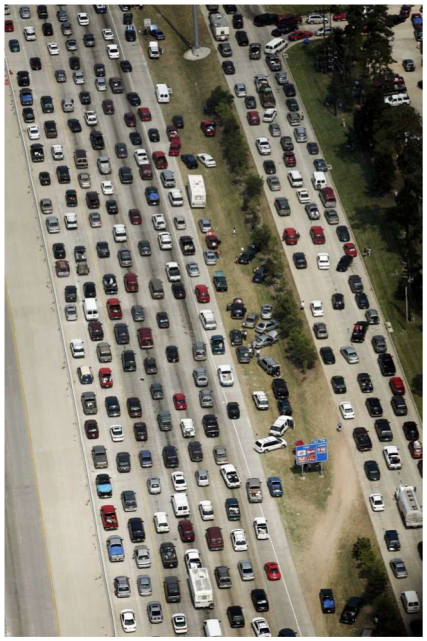
Can we use real time data to optimize evacuation routes during an emergency?

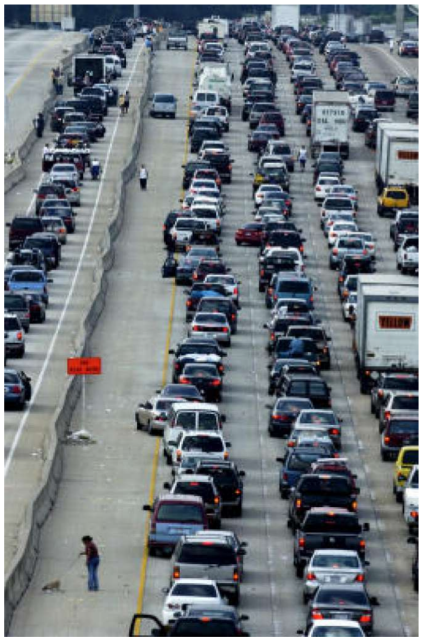
Can we provide more details beyond Google Maps simply showing an accident on the freeway?

Case Study: Hurricane Rita & Houston (2005)

- Hurricane Rita documented as the strongest Gulf of Mexico Hurricane, just weeks after Hurricane Katrina
- Forecasted direct landfall on Houston, Texas
- According to the Houston Chronicle:
  - '...an estimated 2.5 million people hit the road ahead of the storm's arrival, creating some of the most insane gridlock in U.S. history. More than 100 evacuees died in the exodus. Drivers waited in traffic for 20-plus hours, and heat stroke impaired or killed dozens. Fights broke out on the highway. A bus carrying nursing home evacuees caught fire, and 24 died."

Worst Gridlock in US History





Can we use Data Science to optimize evacuation routes in the future???

Step 2: Obtain the Data

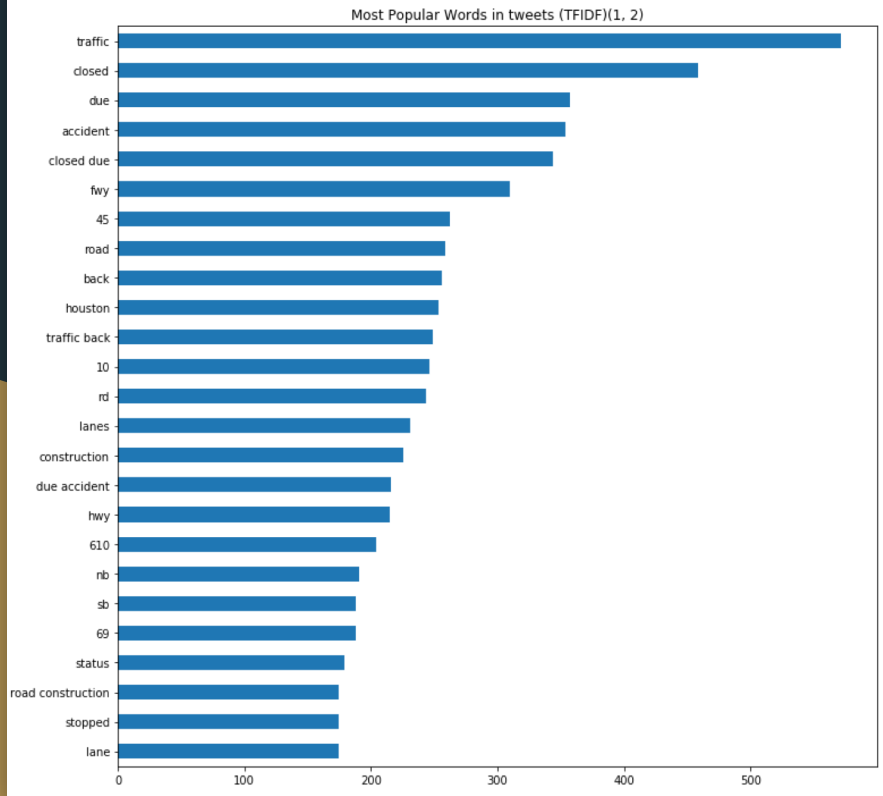
- Use real-time traffic information from Twitter!
- Selected Houston, Texas for analysis as there was a previous large scale disaster in 2017 to train model on
- Twitter API (limited acquisition) vs. Get Old Tweets Library (allows for broader historical acquistion)
- Convert acquired tweets to a dataframe
- Create a mask to identify tweets describing a road closure
- Initial pull yielded 16,000 tweets from accounts categorized as traffic/transportation, public safety and weather

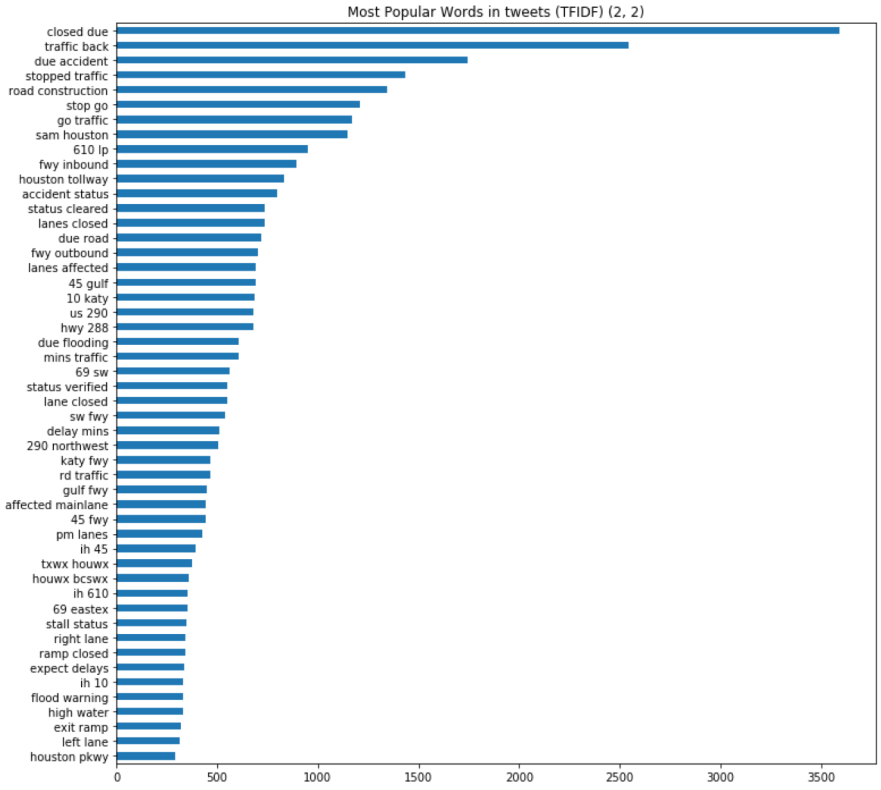
Step 3: Explore the Data

Use Natural Language Processing (NLP) to create a model that can classify whether or not a tweet is relevant to road closure

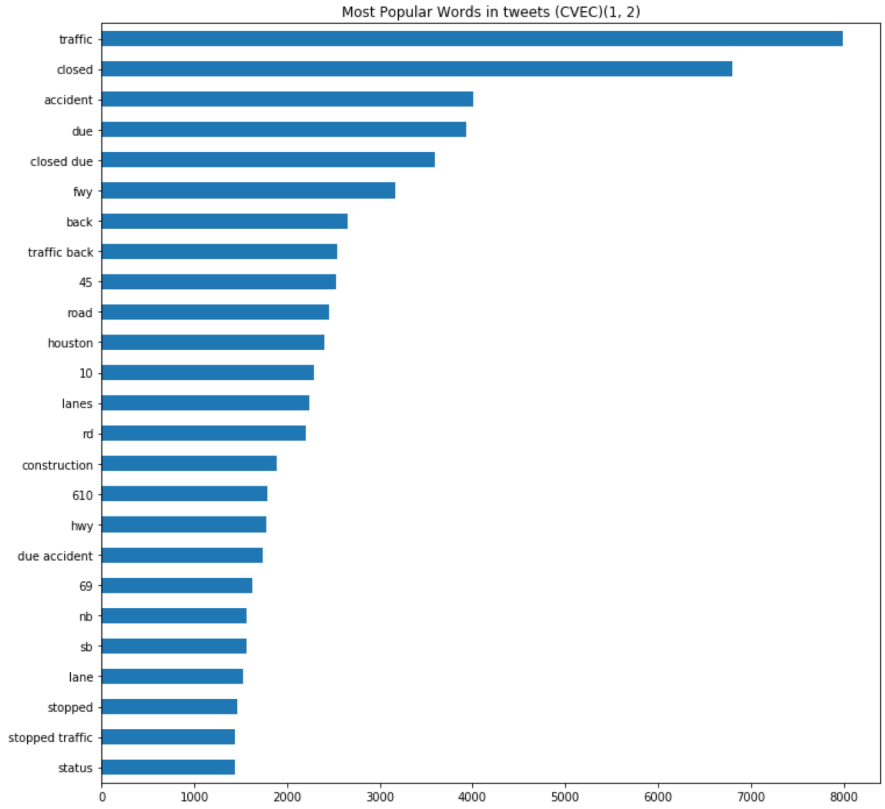
1. Vectorize Data using Count and Tfidf Vectorizers
2. Explore Vectorized Data
3. Go back and add relevant stop words (http, org, etc.)

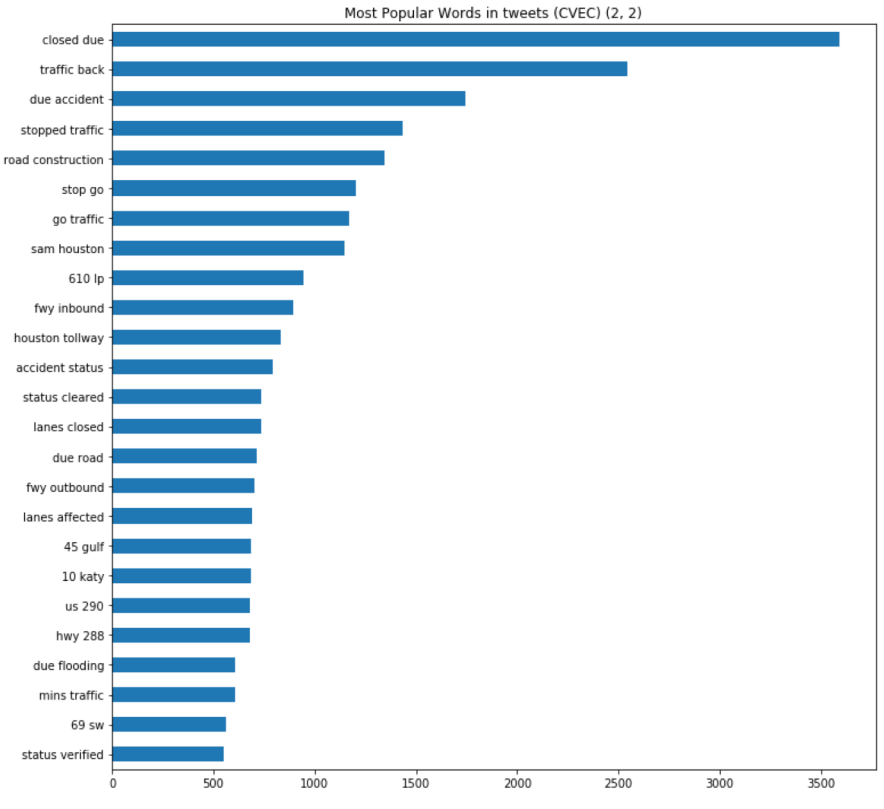
TFIDF Vectorizer





Most Popular Tweets - CountVectorizer

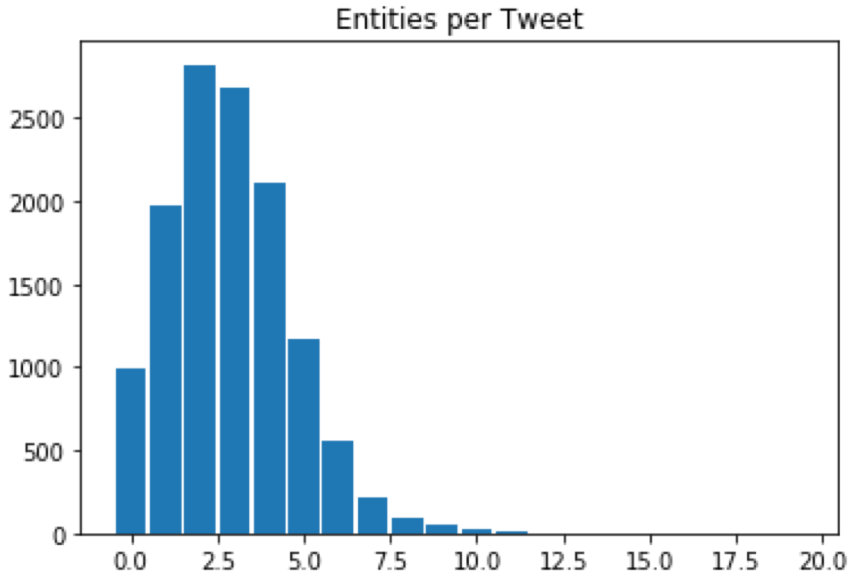




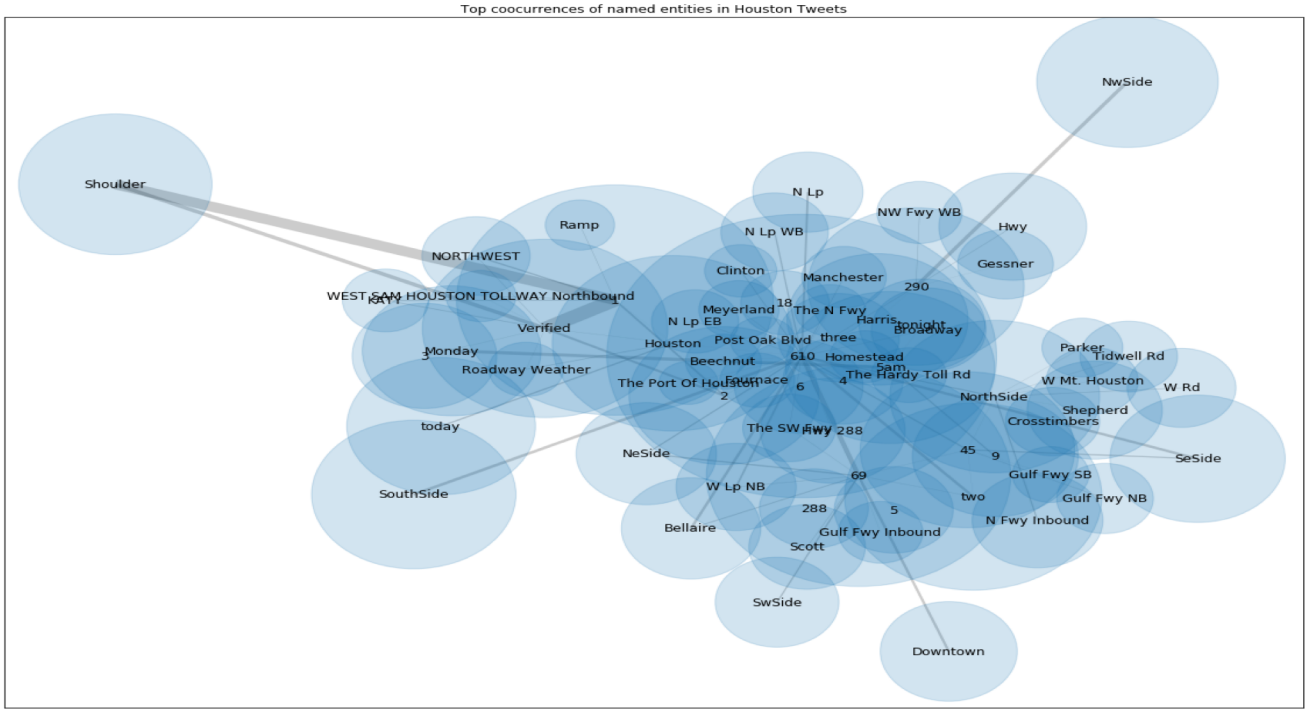
Applying Named Entity Recognition with spaCy

- Attempted to use spaCy to identify named entities in tweets
- Used networkx library and built a co-occurrence function to identify named entities in tweets
  - Analyzes tweets to identify entity pairs that occur in the corpus
  - For each entity pair that occurs together in the same tweet, records the number of times they occur together in the whole corpus
- Returns a ‘weight’ to named entity combinations which can be used to identify the relative importance of a location for mapping purposes

Using Co-Occurrence Function to Identify Number of Named Entities in a Tweet



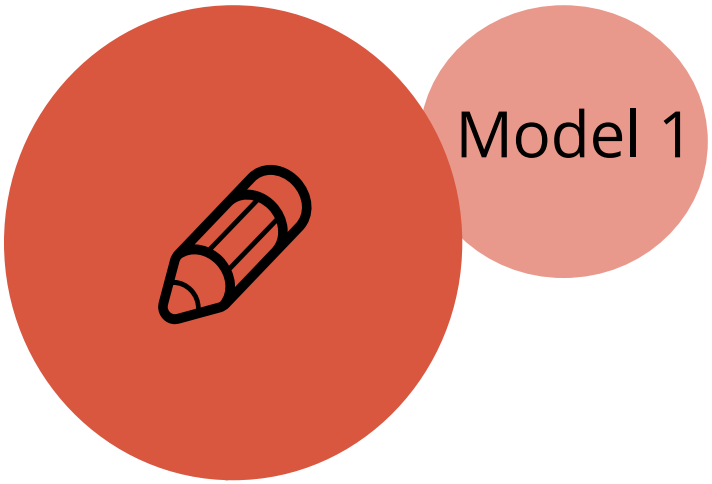
Top Co-Occurring Locations (min weight of 30)



Step 4: Model the Data

Basic Modeling for Binary Classification

Classify a tweet as relevant to road closure



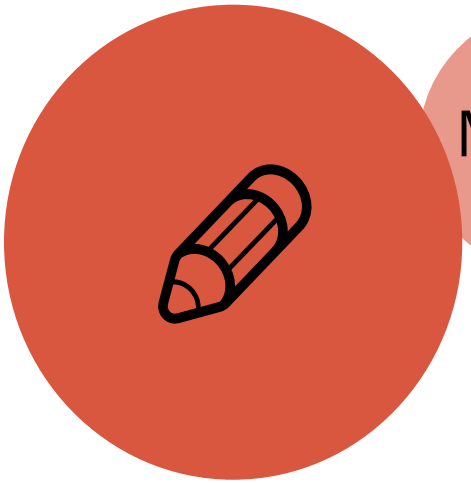
Logistic Regressions

95.14% TFIDF

96.39% CVEC

Basic Modeling for Binary Classification

Classify a tweet as relevant to road closure



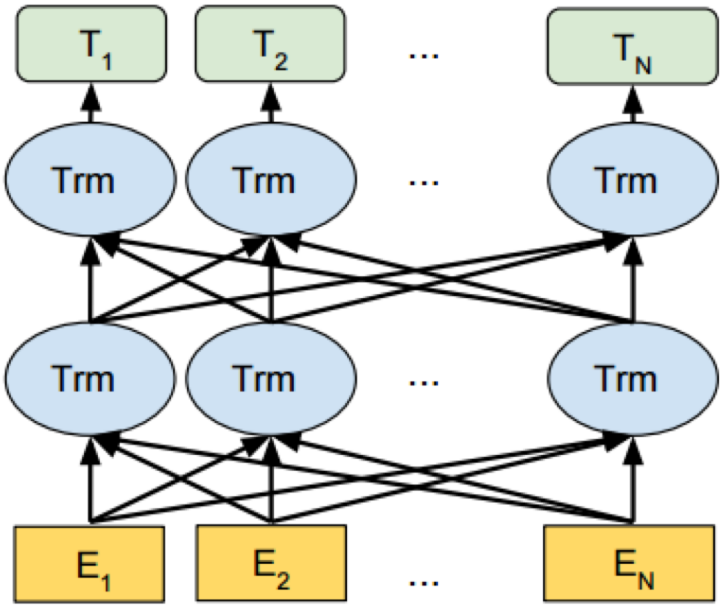
Model 2

Support Vector Machines

96.18%
TFIDF

96.28%
CVEC

Google BERT

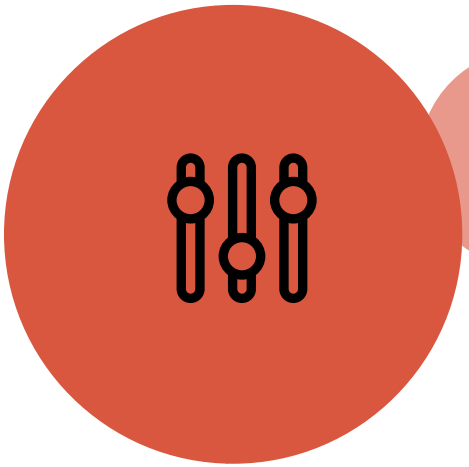


https://medium.com/@cdathuraliya/online-prediction-and-model-pipelining-for-google-bert-8ace3c85b4f6

A Little Bit About BERT

- Open Source Neural Network
- Pretrained on a large text corpus (Wikipedia)
- Although pretrained, you can add training data for your own task
- First Bidirectional Language Representation Model (Analyzes a word based on the other words before and after it in the sentence/tweet)

Advanced Modeling for Binary Classification



BERT

Bidirectional Encoder Representations from Transformers

99.8% Accuracy

Google Colab

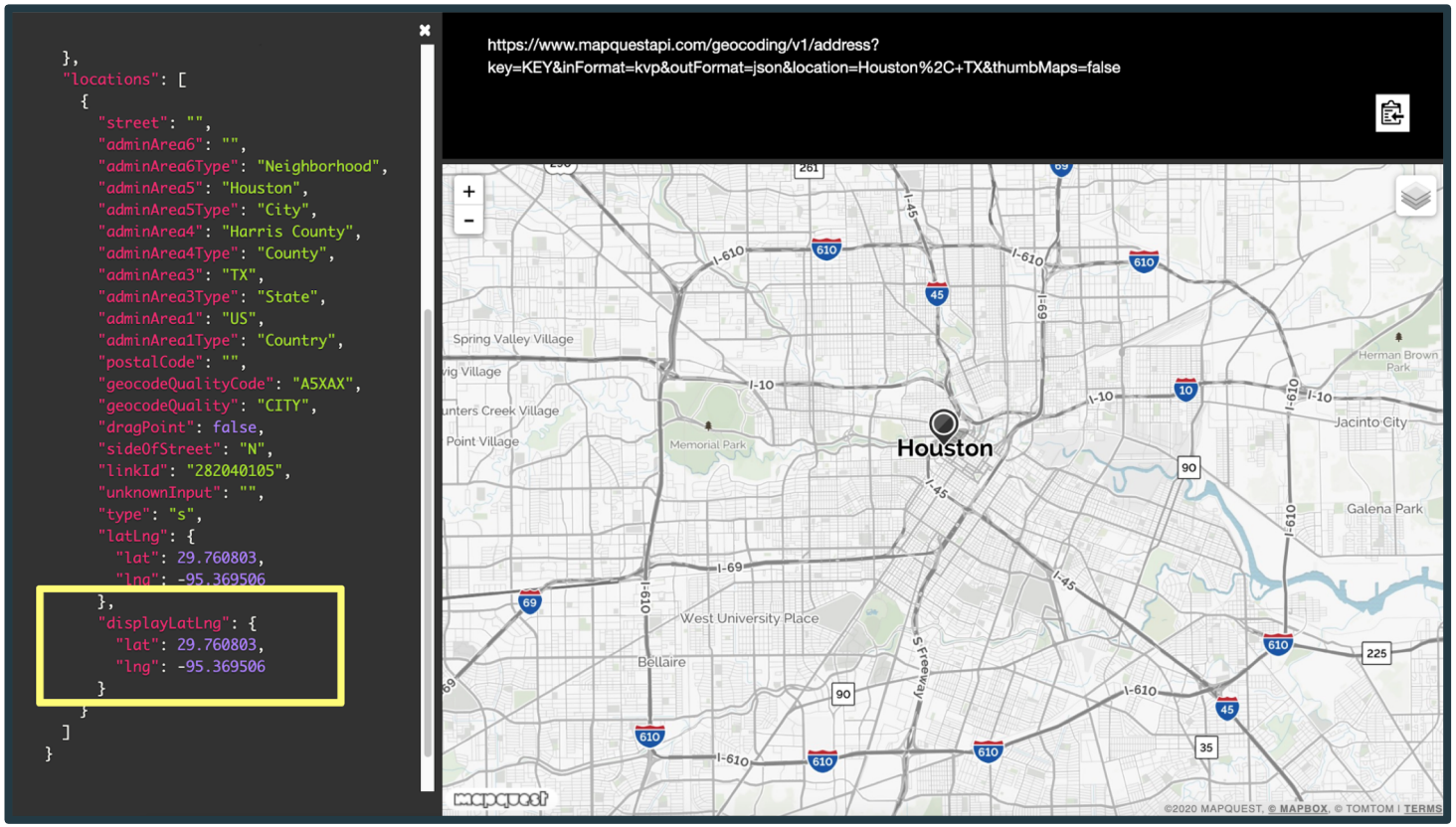
Google's Free Cloud & GPU Service

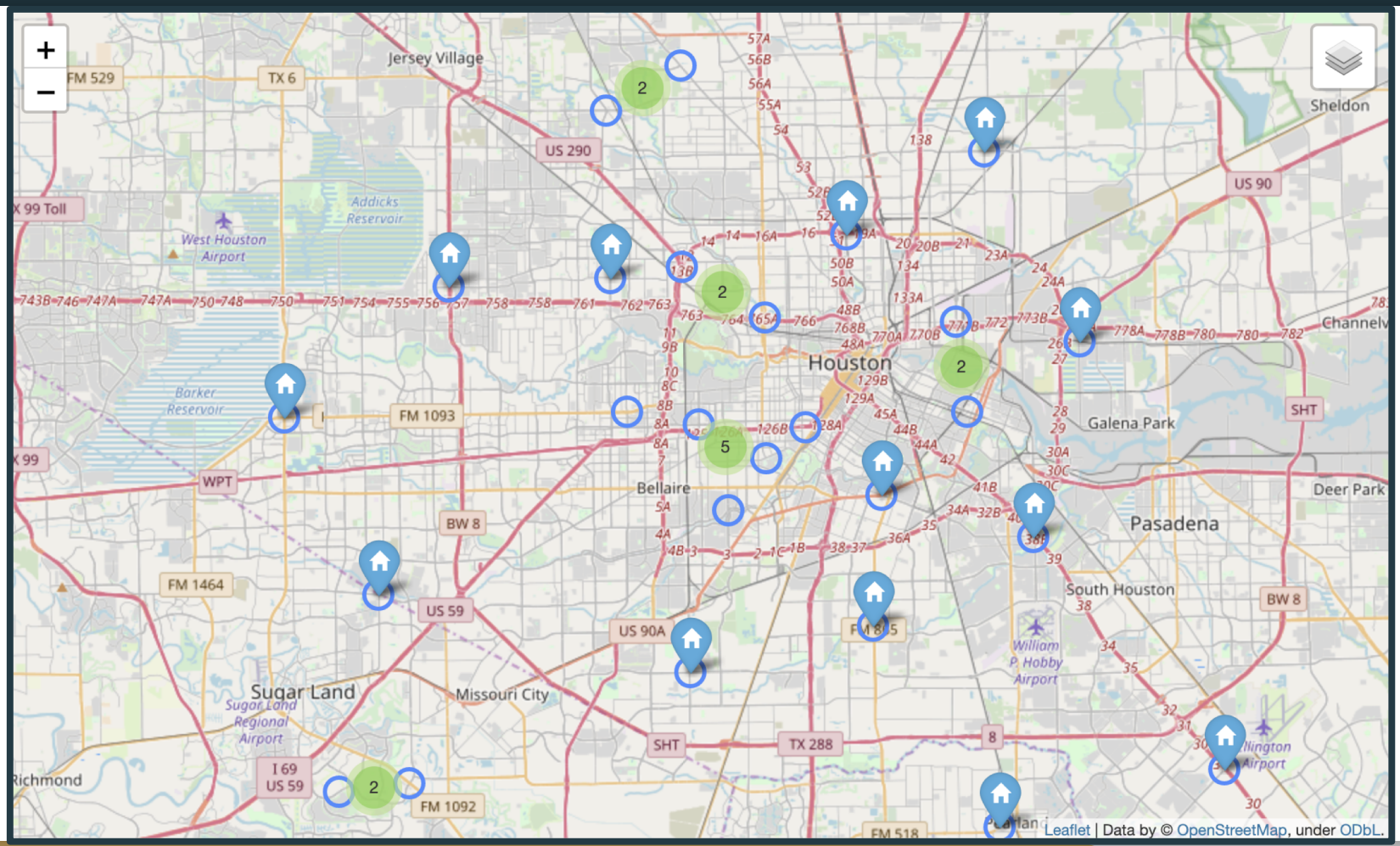
99.6% MCC

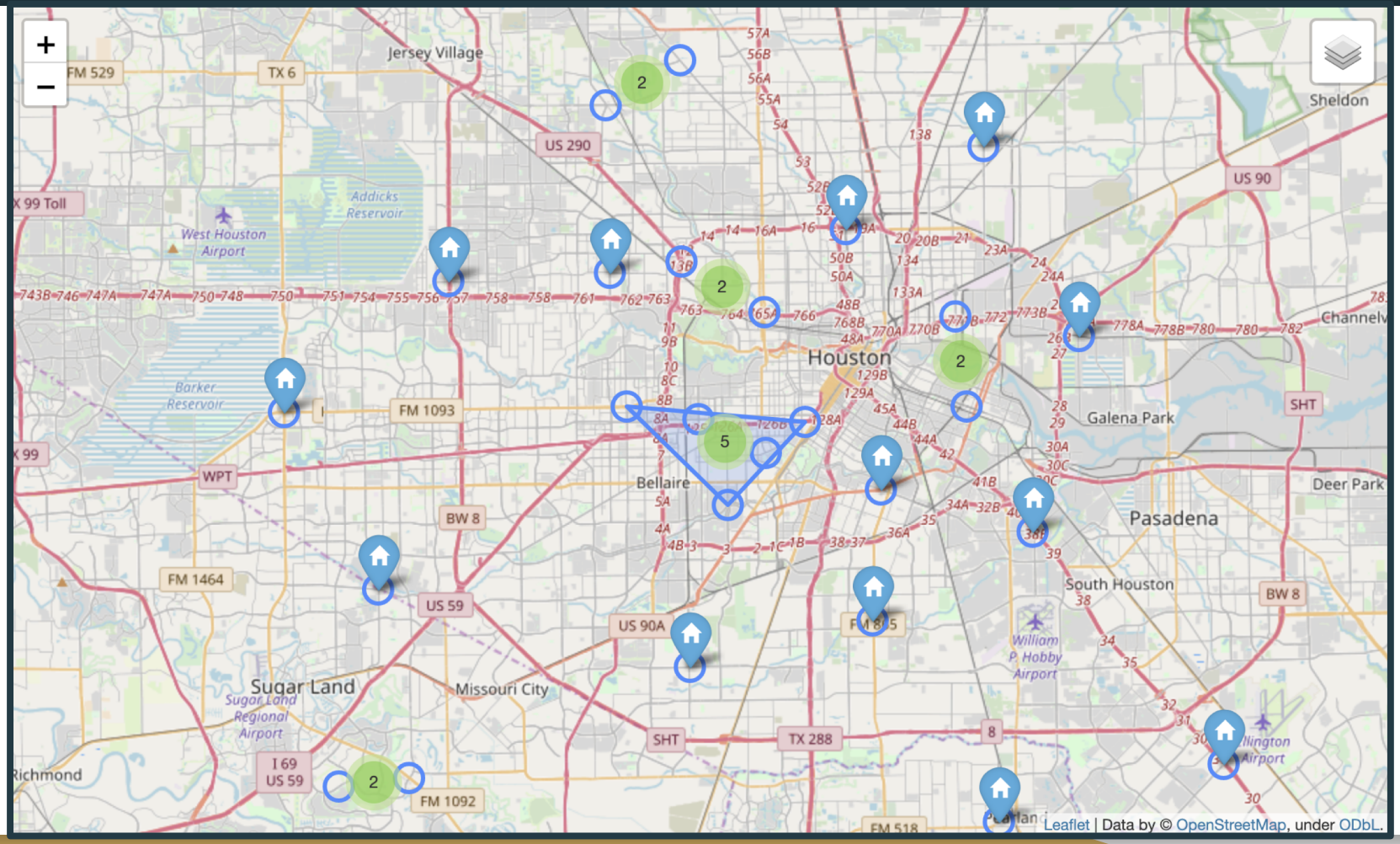
Mapping

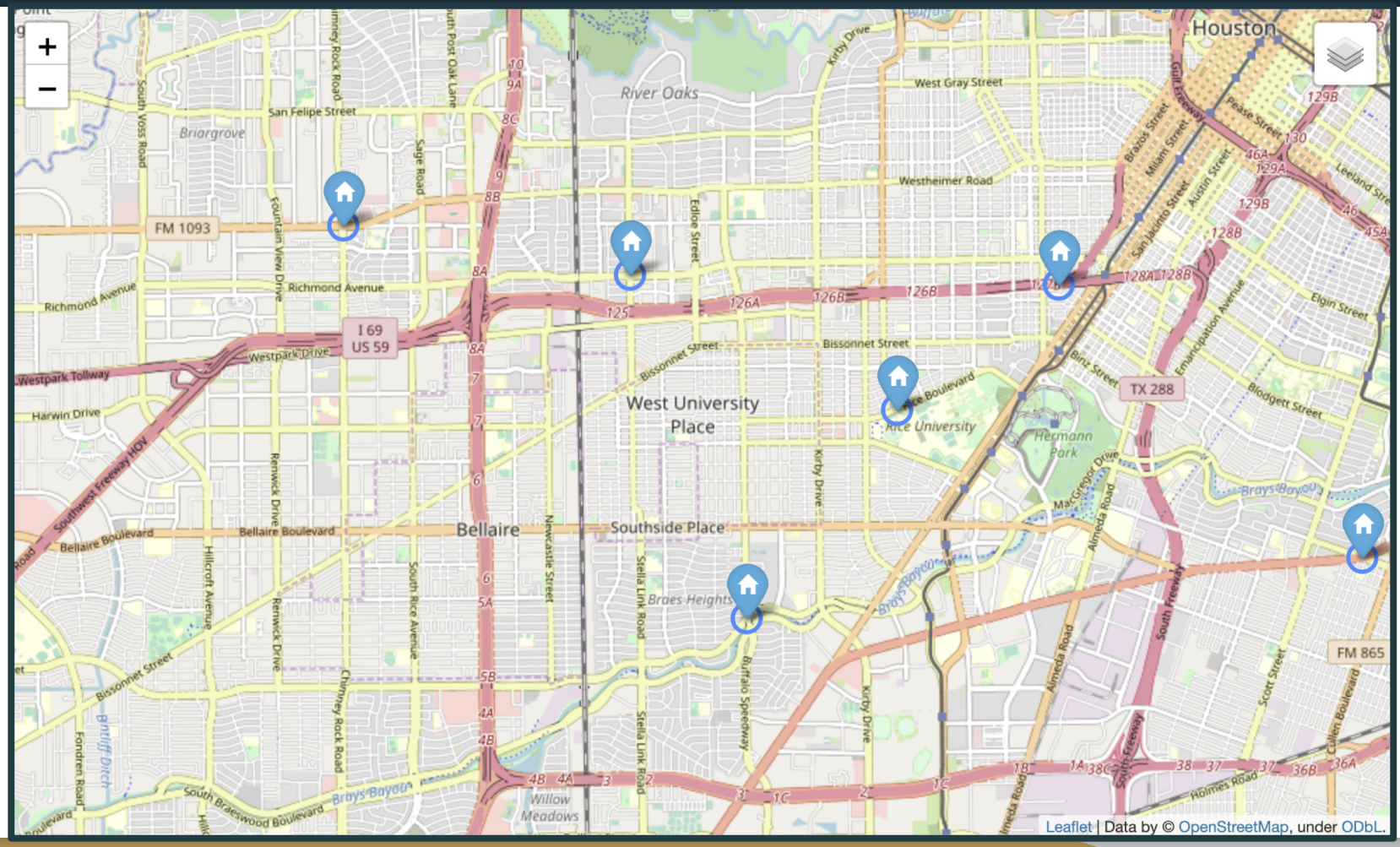
- *Formatted and converted relevant tweets into an address/location syntax compatible with Mapquest's API*
- Used API to convert tweets' location information into latitude and longitude coordinates
- Plotted locations on a map of Houston using the Folium library.

Mapquest Developer API




Jersey Village
FM 529
TX 6
US 290
Addicks Reservoir
West Houston Airport
X 99 Toll
Barker Reservoir
FM 1093
X 99
WPT
BW 8
FM 1464
US 59
Sugar Land
Sugar Land Regional Airport
Missouri City
Richmond
I 69
US 59
FM 1092
US 90A
SHT
TX 288
Houston
Bellaire
Galena Park
Pasadena
South Houston
Deer Park
William P. Hobby Airport
US 90
Sheldon
Channelv
FM 518
Leaflet | Data by © OpenStreetMap, under ODbL.


Jersey Village
FM 529
TX 6
US 290
Addicks Reservoir
West Houston Airport
Barker Reservoir
FM 1093
WPT
BW 8
FM 1464
US 59
Sugar Land
Sugar Land Regional Airport
Missouri City
I 69
US 59
FM 1092
Richmond
Houston
Bellaire
US 90A
SHT
TX 288
Galena Park
Deer Park
Pasadena
South Houston
William P. Hobby Airport
US 90
Sheldon
Channelview
Leaflet | Data by © OpenStreetMap, under ODbL.


Houston
River Oaks
West University Place
Bellaire
Southside Place
Braes Heights
Briargrove
Willow Meadows
Hermann Park
Rice University
FM 1093
I 69 US 59
TX 288
FM 865
Richmond Avenue
Westheimer Road
West Gray Street
San Felipe Street
Bissonnet Street
Bellaire Boulevard
Westpark Drive
Westpark Tollway
Harwin Drive
Southwest Freeway HOV
Kirby Drive
Almeda Road
South Freeway
Brays Bayou
Buffalo Speedway
Stella Link Road
Newcastle Street
Renwick Drive
Chimney Rock Road
Fountain View Drive
Sage Road
South Post Oak Lane
South Voss Road
Edloe Street
Rice Boulevard
Main Street
Binz Street
Emancipation Avenue
Elgin Street
Blodgett Street
Scott Street
Holmes Road
Fondren Road
Hillcroft Avenue
South Rice Avenue
South Braeswood Boulevard
Bissonnet Street
Gulf Freeway
Pease Street
Leeland Street
Austin Street
San Jacinto Street
Brazos Street
Milam Street
Cullen Boulevard
MacGregor Drive
Leaflet | Data by © OpenStreetMap, under ODbL.

Ideas for Future Development

- Fine-Tune and integrate BERT to automatically answer a question such as, “Is this road closed/damaged?” in real time and recognize named entities.
- Optimize Named Entity Recognition to identify and map relative locations of road closures.
- Pass the aforementioned information through a more sophisticated function that can convert the locations of all car accidents, traffic congestion, road closures, etc. into a mapping API-friendly format and then visually map them in real time.
- Customize map icons, colors, and visual patterns to more easily communicate the type of incident