Analysis of Moving Events Using Tweets

Supritha B. Patil

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Edward A. Fox, Chair
B. Aditya Prakash
Sunshin Lee

May 13, 2019
Blacksburg, Virginia

Keywords: Twitter Analysis, Natural Language Processing, Machine Learning

Analysis of Moving Events Using Tweets

Supritha B. Patil

(ABSTRACT)

The Digital Library Research Laboratory (DLRL) has collected over 3.5 billion tweets on different events for the Coordinated, Behaviorally-Aware Recovery for Transportation and Power Disruptions (CBAR-tpd), Integrated Digital Event Archiving and Library (IDEAL), and Global Event and Trend Archive Research (GETAR) projects. The tweet collection topics include heart attack, solar eclipse, terrorism, etc. There are several collections on naturally occurring events such as hurricanes, floods, and solar eclipses.

Such naturally occurring events are distributed across space and time. It would be beneficial to researchers if we can perform a spatial-temporal analysis to test some hypotheses, and to find any trends that tweets would reveal for such events.

I apply an existing algorithm to detect locations from tweets by modifying it to work better with the type of datasets I work with. I use the time captured in tweets and also identify the tense of the sentences in tweets to perform the temporal analysis. I build a rule-based model for obtaining the tense of a tweet. The results from these two algorithms are merged to analyze naturally occurring moving events such as solar eclipses and hurricanes. Using the spatial-temporal information from tweets, I study if tweets can be a relevant source of information in understanding the movement of the event. I create visualizations to compare the actual path of the event with the information extracted by my algorithms. After examining the results from the analysis, I noted that Twitter can be a reliable source to identify places affected by moving events almost immediately. The locations obtained are at a more detailed level than in news wires. We can also identify the time that an event affected a particular region by date.

Analysis of Moving Events Using Tweets

Supritha B. Patil

(GENERAL AUDIENCE ABSTRACT)

News now travels faster on social media than through news channels. Information from social media can help retrieve minute details that might not be emphasized in news. People tend to describe their actions or sentiments in tweets. I aim at studying if such collections of tweets are dependable sources for identifying paths of moving events. In events like hurricanes, using Twitter can help in analyzing people's reaction to such moving events. These may include actions such as dislocation or emotions during different phases of the event. The results obtained in the experiments concur with the actual path of the events with respect to the regions affected and time. The frequency of tweets increases during event peaks. The number of locations affected that are identified are significantly more than in news wires.

Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor, Dr. Edward A. Fox, for his guidance throughout this research. He has been a patient and an inspiring guide. I could have not made the progress I have made, without his guidance. I would also like to extend my thanks to Dr. Sunshin Lee for his time and encouragement in understanding a crucial part of my thesis. I express my sincere appreciation to Dr. B. Aditya Prakash for his motivation and direction from the beginning of my Masters program.

Furthermore, I would like to extend my gratitude to Prashant Chandrasekar, a member of the DLRL, for his feedback and guidance. I would also like to thank my family and friends for all the moral support that they have provided.

We are also grateful to have received support through grants funded by NSF for projects: Integrated Digital Event Archiving and Library (IDEAL), Grant IIS-1319578; Global Event and Trend Archive Research (GETAR), Grants IIS-1619028 and 1619371; and CRISP Type 2/Collaborative Research: Coordinated, Behaviorally-Aware Recovery for Transportation and Power Disruptions (CBAR-tpd), Grant IIS-1638207.

Contents

List of Figures

List of Tables

List of Abbreviations

CBAR-tpd: Coordinated, Behaviorally-Aware Recovery for Transportation and Power Disruptions

DLRL: The Digital Library Research Laboratory at Virginia Tech

GETAR: Global Event and Trend Archive Research

LDA: Latent Dirichlet Allocation

LIW: Location Indicative Words

MENET: Multi-Entry neural NETwork

NER: Named Entity Recognition

NHC: National Hurricane Center

NLP: Natural Language Processing

RF: Random Forest

SVM: Support Vector Machines

Chapter 1

Introduction

1.1 Problem

There are many naturally occurring events that are spread across regions and time. Examples include floods, solar eclipses, tsunamis, and others. As soon as such an event hits, there are many people tweeting about these events. These tweets contain a wide range of information about these events, ranging from expressing condolences to mentioning the affected areas and any damages caused. Analyzing the patterns in these collections of tweets would allow efficient real-time tracking of events via social media. To be able to do that, extraction of time and location from these tweets plays a crucial role. Extracting time from a tweet is straightforward, but using it in combination with spatial information is a tricky problem. Extraction of location for a tweet when there is no direct data on the coordinates available has been an area of interest for researchers.

1.2 Motivation

There have been many attempts at finding the location of tweets.

Approaches involve contextual inference and consideration of the user's profile location. It is hard to evaluate these methods as it is difficult to find ground truth. Although there

is extensive research on this topic, there is no work that analyzes the location-time data from tweets. In the Digital Library Research Laboratory at Virginia Tech (DLRL), there are many collections that are related to events that are spread across time and location. Using existing methods to compare locations retrieved from tweets and the actual locations where the event occurred can be a form of evaluation for existing location extraction methods.

Additionally, such an analysis will aid in finding answers to some interesting questions and hypotheses I have regarding tweet collections on several moving events that the DLRL has collected over time.

1.3 Research Question

The work in this thesis is aimed at helping research projects in DLRL, which has a large collection of tweets focusing on different events. For events such as hurricanes and solar eclipses, that are spread across time and space, I would like to determine if the events can be tracked by these two parameters using collections of tweets that are related to that event. This can be used by other researchers who are trying to analyze the effects of such events on people and their behavior.

1.4 Hypotheses

My hypotheses to test regarding tweets, and their times and locations, are:

- Hypothesis: People who have moved, tweet about evacuation or preparation related plans.
- Hypothesis: The peak of the frequency of tweets matches the time of the event.

- Hypothesis: Official sources (such as the National Center for Hurricanes) on Twitter are more reliable sources for location and time information of moving data than the general population.
- Hypothesis: The (official) path of an event maps to the time-lapsed view of tweets along the path, as can be determined through our analysis and visualization for that event.

1.5 Overview

This thesis is structured as follows:

- The next chapter discusses research papers and case studies related to location, time, and movement of people in the context of tweets.
- The design and methodology chapter details the algorithms used in inference of locations and the time sequence from tweet collections. The experiments conducted and the results for each of these modules are also documented here.
- The results chapter establishes a connection between the location and time module to analyze the events as a whole and track the path of the events,
- The conclusions chapter briefly mentions the overview of the entire process and discusses the results obtained in the context of the hypotheses. It also summarizes the rest of this document.
- The chapter on future work describes various ways to extend this research to obtain more insights on moving events.

Chapter 2

Review of literature

My research concerns two components that are required for further analysis. One is location identification in tweets that do not have geo-coordinate information but might have ingrained location references. The second is the identification of the time sequence of events from tweets. For this purpose, it is important to understand what time period is referred to by tweets. This chapter discusses some papers that helped me understand the relations between events, locations, and time.

2.1 Real-Time Detection, Tracking, and Monitoring of Automatically Discovered Events in Social Media

Paper [19] aligns with my goals because it focuses on tracking events after they have been discovered, using Twitter. It introduces ReDites, a system for real-time event detection, tracking, monitoring, and visualization. ReDites can automatically detect events from Twitter streams. It accomplishes four tasks: detects new events, tracks the discovered event (finding more posts related to it and maintaining a concise summary), detects the evolving emotions about the event, and visualizes the summary so that it is easier to understand.

This system was built to identify new events that appear in traditional news wires. It uses topic modelling to categorize events by content categorization. Geolocation plays an

important role in this research as the interest lies in events that occur in a specific place. For finding the location of tweets that do not have an implicit location-tag, tweet content and meta-data (language, city/state/country name, user description, etc.) have been used. These details are used with L1 penalized least square regression (LASSO) to predict longitude and latitude.

While this paper concentrates on building a system to detect events by location, and summarize them, my research focuses on collecting event related tweets and analyzing them for location and time. My research mainly focuses on naturally occurring moving events (both with respect to time and space), and answering some questions about them. The method used for locating tweets in my research also is different.

2.2 Event Identification and Analysis on Twitter

Paper [22] states that popularity and importance of an event can be gauged by the volume of tweets covering the event. It aims at identifying events on Twitter streams, analyzing personal topics and events, and summarizing events identified from Twitter. It emphasizes separating event-driven tweets from personal-interest-driven tweets.

The authors observe that the textual content can be combined with the time patterns of tweets to obtain important insight into the general public's attention. Based on this, they suggest two models for identification of events that are extended from LDA (Latent Dirichlet Allocation) and a non-parametric model. The concept of events and users' personal interest topics are orthogonal in that many events fall under certain topics. As an extended task, they construct a unified model of topics, events, and users on Twitter.

2.3 Distinguishing Past, On-going, and Future Events: The EventStatus Corpus

Paper [10] investigates one of the aspects that plays an important factor that my research looks at, the tense that an event has been mentioned with. It aims at detecting past, on-going, and future events from a corpus of articles. The corpus contains 4500 English and Spanish articles about civil unrest. It shows that temporal status of events is difficult to classify because of three different reasons: local tense is often lacking, time expressions are insufficient, and linguistic contents have rich semantic compositionality. The authors try to classify using an SVM (Support Vector Machines) classifier and try a conventional neural net for the task. Although both techniques do not accomplish the goal, they find the semantic compositionality challenges for this task.

While this paper focuses on the above tasks for articles, my task is made simpler by the fact that tweets are short text data. We need to infer whether each tweet talks about an event in the past, present, or future using (an average of) two sentences. Therefore, the accuracy of my predictions of tense is much higher.

2.4 Resolution of grammatical tense into actual time, and its application in Time Perspective study in the tweet space

Paper [12] comes closest to my work on time analysis of tweets. Time perception (past, present, and future) is considered to have an influence on human actions, perceptions, and emotions. It aims at assessing time perception of users from their tweets by resolving gram-

matical tense into underlying temporal orientations of tweets: past, present, and future. The researchers develop a minimally supervised classification framework for temporal orientation task that enables incorporating linguistic knowledge into a deep neural network. The temporal orientation model achieves an accuracy of 78.7% when tested on manually annotated data. While doing so, they try to also categorize tweets regarding positive or negative sentiment, and perform that analysis.

While this seems like a great method to apply to my analysis, I tried to build my own model that analyses the three tenses of tweet messages and achieves an approximately equal accuracy.

2.5 Case Study: Event location detection of governments and organizations

Social media has become a common platform for reporting emergencies or disasters. The government has now started supporting the use of social media by creating accounts for its organizations. Although such platforms are now created, they need a certain amount of human intervention to gather information and identify locations. This is one place where my work can minimize the amount of manual effort.

2.6 Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors

Paper [23] aims at building a system that detects real time events from Twitter and sends out a notification to registered users about these events. To identify a target event, they build

a classifier that classifies tweets based on features such as number of words and context. Following this, they produce a probabilistic spatiotemporal model to find the center and trajectory of the event. They perform experiments on earthquakes in Japan. They show that they can identify 96% of the earthquakes with Japan Meteorological Agency (JMA) seismic intensity scale 3 or more.

2.7 Geo-Locating Tweets with Latent Location Information

The idea for my research work was sparked by Dr. Sunshin Lee's dissertation [14] on finding location from tweets. This research works towards locating tweets that do not have geocoordinate information, by identifying and applying their location indicative words. It also introduces an approach to eliminate any ambiguity in locations retrieved by using a classification model. The data-set used in this paper contains 6 million tweets from collections of water main breaks, sinkholes, potholes, and car accidents. The crucial part of this research depends on Location Indicative Words (LIWs) in tweets.

This approach to find locations works well with my research as I do not aim to find the location for all tweets in data-sets. I only try to find affected location by looking at mentions made explicitly or implicitly in tweets.

2.8 Home Location Identification of Twitter Users

Paper [16] presents an algorithm to retrieve the home location of Twitter users at different granularity such as state, city, or geographic region. They do so using the content of the

tweet and users' tweeting behaviour. The approach uses a combination of heuristic and statistical methods to predict locations, and the gazetteer to identify place-name entities.

They also build a classifier that states if a user is travelling during a particular time period. They classify a user as travelling if the geo-distance between tweets is more than 100 miles. Geo-tagged information from tweets such as mentions and hashtags are used to identify places that the user is travelling to.

They obtain an accuracy of 0.78 for the time zone prediction and an accuracy of 0.62 for city-level prediction. This work focuses on finding home location information for tweet users and eliminates the users who are travelling. In this research, I do not intend to find the home location of users, I try to find the locations spoken about by users when they mention a particular event.

2.9 Multiview Deep Learning for Predicting Twitter Users' Location

Paper [5] runs parallel to my work on geo-locating tweets. While most proposed methods follow either a content based approach or a network based approach, this paper introduces a method that combines the strengths of both approaches. The proposed model is called Multi-Entry neural NETwork (MENET).

It helps in estimating locations in cases where there is no coordinate information available and no location indicative words exist. In such cases, the user's network information can still be used to predict their home location. The drawback for this method is that, in tweets where the user is travelling away from his/her home location, the predicted location will be inaccurate. Yet this method might help better the overall accuracy of the analysis.

2.10 Analyzing Refugee Migration Patterns Using Geo-Tagged Tweets

Paper [11] runs parallel to my research. In this exploratory analysis, the authors try to identify migration patterns of people from the Middle East and Northern Africa to Europe using information from Twitter. The study involves extraction of refugee trajectories and detection of topical clusters along migration routes using the V-Analytics toolkit. They use hash-tag-based topical clustering to identify refugee routes.

This work runs parallel to ours because it is highly centered around spatial-temporal analysis of tweets. The methods used for location and time extraction in my work and this paper differ greatly. The data-sets and the type of analysis also are different.

Chapter 3

Design and Methodology

3.1 Design goals

The DLRL has many collections of tweets related to events like elections, floods, hurricanes, heart attacks, etc. There are many research projects that aim at analyzing these collections and retrieving useful information. Spatial-temporal analysis of people during prominent events will help people act immediately in case of emergencies. To gauge the impact of moving events like solar eclipses, hurricanes, floods, and other such events, location and time play a vital role in the analysis. An example is the Coordinated, Behaviorally-Aware Recovery for Transportation and Power Disruptions (CBAR-tpd) project. This project aims at analyzing the moving events in the United States in the recent past including infrastructure damage, movement of people, and other similar information.

3.2 Methodology

3.2.1 Data Preprocessing

Our Twitter data for different events are collected mostly using one of two Twitter APIs: yourTwapperKeeper [1] or Social Feed Manager [15]. Tweets are collected based on hashtags or keywords identified for different events. Unfortunately, this may lead to many nonrelevant

tweets. Suitable analysis can estimate the relative proportions of those nonrelevant tweets that are posted before, during, or after the event.

Experiments are conducted on three hurricane and one solar eclipse collection selected for analyzing moving events. I focus on Hurricane Florence (year 2018), Hurricane Harvey (year 2017), Hurricane Irma (year 2017), and the recent popular Solar Eclipse (year 2017).

Another process that is highly important in the analysis of location or time is cleaning the tweet text to remove any punctuation marks or other escaped characters. This is because the Stanford Named Entity Recognizer [7] throws errors while parsing these.

Two main modules required for my analysis of tweets are: 1) the module to identify locations affected by the movement of the event, and 2) the temporal analysis module.

3.2.2 Location identification module

There are two sources for location identification from tweets. One source is the users' geo-coordinates recorded in tweets using location services. But most Twitter users turn off the option to record location while tweeting, as they are concerned about privacy. To overcome this, I use the extraction from tweets of named entities, such as names of places or organizations, to identify the location that the user might be referring to. The assumption here is that people usually mention places, or references to places being affected in their tweets, and these can be decoded to get exact locations. Once these named entities are extracted, I send them through the Google geocoding API [8] to get an exact location. Sometimes I get more than one address as a match to the named entity being geocoded. For example, Fairfax Road is a name that is common in many towns such as Radford, Blacksburg, and Baltimore. There is an ambiguity that rises here as to which is the one the tweet is talking about. To disambiguate this, I use a machine learning model that classifies the location to

one of the states in the United States.

Using the above mentioned steps, I obtain any locations spoken about in tweets. My hypothesis is that this will be enough location information to identify most places affected by a moving event.

There are three main sub-modules in the process of location identification. The first sub-module is for extracting named entities from the tweet text. The next one is to geocode any entity retrieved in the previous step using the Google API. The final step of disambiguating locations is performed in another sub-module by the use of a machine learning algorithm. The modules and the data flow are shown in Figure 3.1.

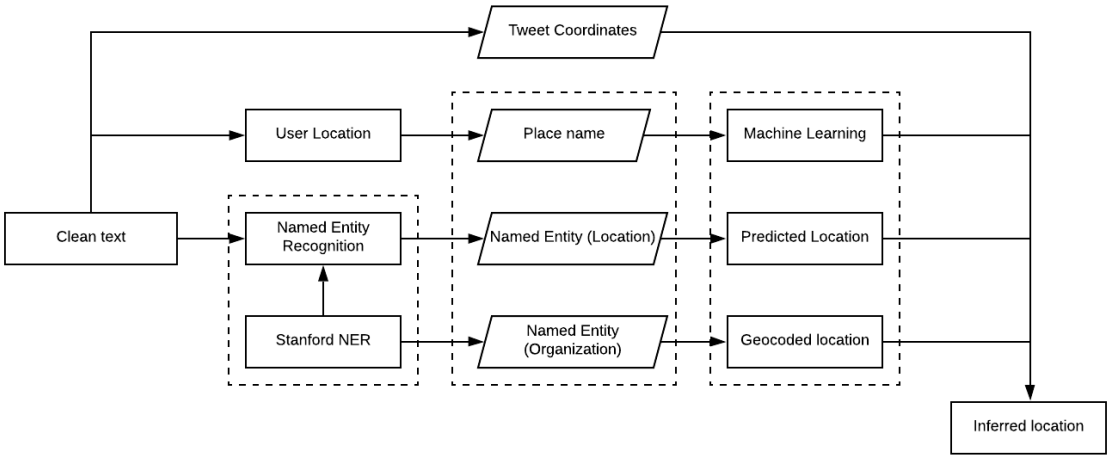

Figure 3.1: Data flow for location identification

Rather than just using an existing model [14] to locate tweets, I extend this model to work with data on moving events. This includes identifying the right features for this kind of data. The above sub-modules are described in more detail in the following sections.

3.2.3 Extracting named entities

I use the Natural Language Toolkit [20] and the Stanford Named Entity Recognizer (NER) [7] for identifying named entities like locations, organizations, and person names. The NER can identify most of these three classes of information quite efficiently. For the data considered, Stanford NER performs as well as another popular natural language processing library, spaCy [9]. Hence, I continued to use the Stanford NER as was done with the existing model [14].

After observing the results, I realized that locations and organizations that are extracted add value and provide accurate information on the locations of events. On the other hand, extracting person names as one of the features adds more noise to the locations retrieved. Hence, I choose to drop person names from the list of features.

3.2.4 Geocoding module

I make use of one of the most popular geocoding APIs existing that can be integrated with Python, the Google geocoding API. Google allows developers to choose from a set of Map based web-services such as Directions API, Places API, and Roads API. One of these is the geocoding API, to help with geocoding a given address with its full address. It also provides support for reverse geocoding, transforming a latitude-longitude combination to a formal address.

There is one issue that we face with the API, however; there is a daily limit of 2500 addresses only that it can geocode, or reverse geocode, for each API key created. I overcome this issue by creating multiple API keys.

Collections	No. of tweets	No. of tweets with recorded location
August 2018 Solar Eclipse	449284	4416
Hurricane Florence	221013	3438
Hurricane Harvey	116060	6435
Hurricane Irma	88972	18749

Table 3.1: Counts of tweets, and tweets with location

	B	C	D	E	F	G	H	I	J	K
374	Chief met @KRLDWeatherD	HurricaneHarvey	Dallas, TX	KRLD		4131 N Central E	TX,USA	2	1	TX
410	David Montes helps board wi	Harvey;Hurricane	Corpus Christi, TX		Ocean Drive	Ocean Dr, Miami	NC,USA;FL,USA	4	3	TX
440	Harris County Emergency Op	HurricaneHarvey	mlopez@kprc.co	Harris County En	Houston;Texas	6922 Old Katy R	TX,USA	2	1	TX
651	Coastal Health & Wellne	HurricaneHarvey	Galveston Count	Coastal Health &	Wellness	9850 Emmett F L	TX,USA	2	1	TX
657	.. Downtown Galvy merchant	HurricaneHarvey	Houston, TX	Post Office Ave		Post Office Ave,	MA,USA;TX,USA	8	6	TX
727	To all residents of Houston, C	HurricaneHarvey	Kissimmee, FL	Corpus Christi	Houston;Galvest	Houston Rd, Bro	TX,USA	3	1	TX
732	All of our public services, incl	HurricaneHarvey	Galveston Count	Coastal Health &	WIC	9850 Emmett F L	TX,USA	2	1	TX
740	Non-essential personnel hav	HurricaneHarvey	DC via CA + MI	Naval Air Station	Kingsville	NAS Corpus Chr	TX,USA	2	1	TX
771	Already seeing showers and	HurricaneHarvey			Texas Coast;Wes	Texas, Irvine, CA	CA,USA	4	1	YY
831	The only safe place in Texas	HurricaneHarvey	Texas		Texas:Dallas:El F	N El Paso Cir, Da	TX,USA	2	1	TX

Figure 3.2: Examples of manually labelled data from each collection, for training the disambiguating model. This is part of the manually tagged data for Hurricane Harvey.

3.2.5 Feature analysis

While I follow an existing approach to get the different locations from a tweet collection, the data-sets that the method was designed for [14] are different from the current set of data-sets. The approach was introduced with data on accidents, potholes, and water main breaks. These have more mentions of locations where the incident occurred than the data-sets that I am trying to analyze. Specifically, data-sets on hurricanes and solar eclipses have a mix of location indicative tweets and tweets on people's reaction to these events.

	Original Method Features	Our method features
Geocoordinates	Yes	Yes
User Home Location	No	Yes
Location-SNER	Yes	Yes
Organization-SNER	Yes	Yes
Person-SNER	Yes	No
Hashtags	Yes	No
Mentions	Yes	No

Table 3.2: The features used in the original model and this model

The feature-set selected in the original method is different from what we have used, since the information available about hurricanes and the solar eclipse differs from that involved in car accidents and potholes. Table 3.2 compares the features used by Sunshin Lee et al. with those in our research.

One crucial feature not considered in the original method is the home location from a user's profile Some Twitter users do not want to disclose their home location due to privacy concerns. Further, since accidents occur, and potholes are observed, while traveling, home locations may not be highly informative.

Nevertheless, users usually tend to provide state names or city names in their home location. While there are some users that put random information into this field as a part of their account, one such example being "over the rainbow", these are rare enough that this information can still be one of the features for location estimation. The statistics for the level of detail we obtain in geo-located tweet addresses is available in Figure 3.4.

While I may not be able to get accurate addresses of affected regions from home locations, I

Type of home location	Percentage of tweets (approximated)
Accurate address	4%
City-level address	62%
State-level address	30%
Country-level address	0%
Random input (does not make sense)	4%

Table 3.3: The detail level of addresses retrieved and percentage of tweets

can still use them in the process of disambiguating other feature locations. If a person living in Blacksburg, Virginia mentions Walmart, he/she is most likely talking about the Walmart in or around this area, helping me eliminate irrelevant addresses for Walmart that I receive from the Google geocoding API.

While home location is a useful feature in disambiguation, yet it can still result in incorrect location inference. This is because a person might be living in Virginia but travelling to South Carolina. A tweet from this person mentioning Blacksburg may wrongly be inferred as Blacksburg in Virginia as a result of bias from home location.

I eliminate hash-tags from the list of features used as they introduce much noise in the results obtained. This can be attributed to the fact that in a large-scale event, people frequently use the name of the event as a hash-tag. These hash-tags may be associated with a default address from the Google geocoding API; use of this repeatedly in tweets results in incorrect location inference for many tweets.

Similarly, person names and mentions do not give accurate results from the geocoding API. This is because the Google Maps geocoding API will not be able to retrieve addresses when

people's names are provided. It is most likely that a person mentioned in the tweet is from the network of the person tweeting, and will not have an address attached to them in Google. These features also cause more noise when used for disambiguation rather than help in the process. Therefore, these have been disregarded when inferring locations.

3.2.6 Geocoding and reverse geocoding

A geocoding API can either convert a place name to a formal address or convert a latitude-longitude pair into its accurate address. There are several geocoding services available that can be used with Python such as Python geocoder [3] and geopy [21]. But most of these use Google geocoding APIs to retrieve addresses. Therefore, I choose to use the Google Maps API directly for this algorithm. I have to create an API key to access the services by Google Maps.

I aim at retrieving latitude-longitude level data for locations using the current method for most tweets. There are two types of data that help in location inference. One is the direct location information retrieved from the location-tagged tweets; these are included using the users' location recorded on their devices. They are usually in the format of latitude-longitude coordinates. Thus, I can use these fields directly to map in my analysis. The other location inference is from location indicative words in tweet text. For these, the words are retrieved, sent through the geocoding API, disambiguated, and stored as results. The geocoding API provides the full-address along with several fields such as city, state, country, zip code, latitude, longitude, and so on, according to the location sent for geocoding. This information is provided to us in the JSON format. Most addresses come back with a latitude and longitude, so I can use these in location analysis.

3.2.7 Disambiguating ambiguous locations

To narrow down the list of addresses to one location when the geocoding API provides multiple addresses, I use a machine learning algorithm. I am interested only in locations that are in the affected areas for these events. Therefore, I classify the locations as either in the affected area or not. I have a list of states that fall under affected areas. I do a binary classification for each of these states and then combine the results.

The binary classification identifies if the place belongs to that class (state) or not. I conduct experiments to check if the Support Vector Machines (SVM)[25] or the Naive Bayes [17] method for classification does better with my data-sets. I notice that SVM gives better precision and accuracy on the validation set when compared to Naive Bayes. Hence I use SVM for the disambiguation of locations.

I compare SVM and Naive Bayes by using four metrics as in the original method: precision, recall, accuracy, and F1-measure. These are defined as:

$$Precision = tp/(tp + fp)$$

$$Recall = tp/(tp + fn)$$

$$Accuracy = (tp + fn)/(tp + tn + fp + fn)$$

$$F1 - measure = 2 * (precision * recall)/(precision + recall)$$

I manually labelled around 700-800 tweets, correcting the ambiguity in each of these data-sets, for training the model. I break these into training and validation sets as 70% and 30%.

Method	Precision	Recall	F1
Naïve Bayes	0.944	0.643	0.806
SVM	0.896	0.723	0.823

Table 3.4: The comparision of SVM and Naive Bayes for precision, recall, and F1 measure

3.2.8 Experiments and Results for location analysis

I performed location retrieval for four data-sets of events that I wanted to analyze. The sizes of each of these along with the number of locations that could be retrieved are recorded in the table below.

Collections	Total number of tweets in collections	No. of tweets that were predicted for locations	Total number of tweets with Lat/Long
Hurricane Harvey	116060	1143	6435
Hurricane Florence	221013	877	3438
Hurricane Irma	88972	1078	18749
Solar Eclipse	449284	945	4416

Table 3.5: Tweet and location counts

A part of Florence data contains tweets only from official sources such as the National Hurricane Center. I noticed that when analyzed independently, most tweets from such official sources contained locations and location indicative words. Also, considering home location for disambiguating tweets from these sources made the results more inaccurate. This is because considering the home location for these official sources loses meaning when talking about non-local events.

3.2.9 Time analysis of events

I have the created-time of each tweet as a part of my data-set. The time recorded in this data-set is in Coordinated Universal Time (UTC) and has the date and time when the tweet was posted. Though it seems straightforward to analyze time from this, it is not enough to analyze when a tweet was written, but rather what time the tweet is talking about. A tweet in past tense would mean that it is indicating that it was posted after the event occurred. Similarly, present and future tense represent different phases of the event. Therefore, analyzing tweets for their tenses plays an important role in this research.

My work is made easier by the fact that I need to only classify tweets into three categories: past, present, and future, without worrying about finding more detailed tense such as present continuous. This is because other detailed tenses widely fall under one of these categories, and these three provide enough information to judge the time-line of the event along with the created time.

There has been a lot of work on ways to detect tense for a sentence; some of them are spoken about briefly in the literature review section. But instead of using any of these, I use a rule-based inference approach to build my own model for tense detection.

3.2.10 Method

Here I use the Stanford parser [13] and the Stanford part of speech tagging to distinguish between different pieces of the sentences to help identify tenses. Since tweets have a limit of 280 characters, the number of sentences on an average is two. This makes finding tense for tweets easier than for longer documents. Using a rule-based model works better than more complex classification techniques such as neural networks.

The rules are based on the part of speech tagging, which identifies the tense for a verb in a sentence. Just identifying all the verbs in the tweet text and their tense will not suffice. Languages are complex; even though a sentence is meant in the past tense, each verb in the sentence can be in a different tense. I resolve this problem by identifying the primary subject in the tweet. Then I apply rules to determine tense depending on part of speech of only the verbs relating to the primary subject.

An example of part of speech tagging is provided here:

```
[('Can', 'MD'), ('you', 'PRP'), ('please', 'VB'), ('buy', 'VB'), ('me',
'PRP'), ('an', 'DT'), ('Arizona', 'NNP'), ('Ice', 'NNP'), ('Tea', 'NNP'),
('?', '.'), ('It', 'PRP'), ("'s", 'VBZ'), ('$', '$'), ('0.99', 'CD'),
('.', '.')]
```

Figure 3.3: Stanford Part-Of-Speech tagging

The rules that I use are in the table below for inferring the tense of words in a sentence.

Tense	Parts-of-speech tags
Future	MD
Present	VBP, VBZ, VBG
Past	VBD, VBN

Table 3.6: Rules for tense inference

After finding the primary subject of the tweet, I only consider verbs related to this subject to determine the tense of the sentence.

"The dependency parser analyzes the grammatical structure of a sentence establishing the relationships between 'head' words and words which modify those heads." [20]

I use this dependency parsing to analyze the primary subject and the verbs relating to this subject.

3.2.11 Experiments and results

I ran the tense detection on all the four key data-sets. I manually labelled tweets from the different collections. Accuracy of the tense detection model is measured against a part of these manually labelled tweets (validation set). The number of tweets labelled from each of these collections is listed in Table 3.7.

Collections	# of tweets labelled
Hurricane Harvey	752
Hurricane Florence	1489
Hurricane Irma	348
Solar Eclipse	283

Table 3.7: Number of manually labelled tweets for each tense

Some examples of results from tense detection are provided in Figures 3.4, 3.5, and 3.6.

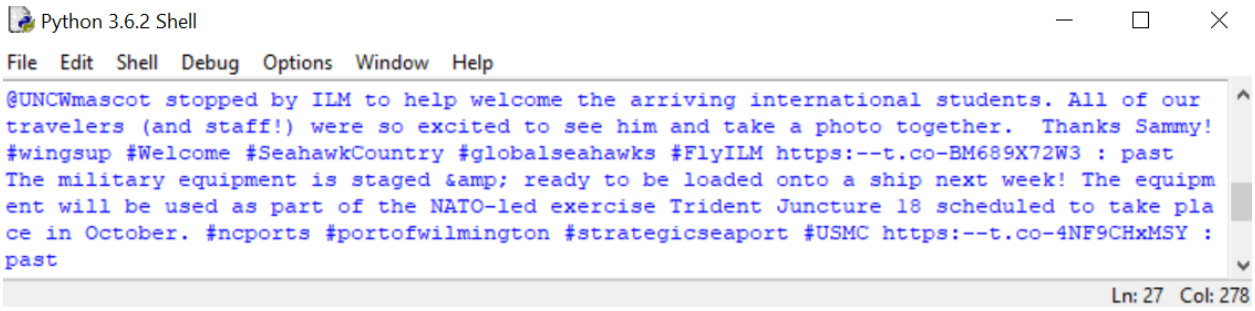

Figure 3.4: Example of tweets detected to be in past tense

The model has an accuracy of 77.43% against manually labelled data. Hence, I chose to complete my analysis using this rule-based inference model.

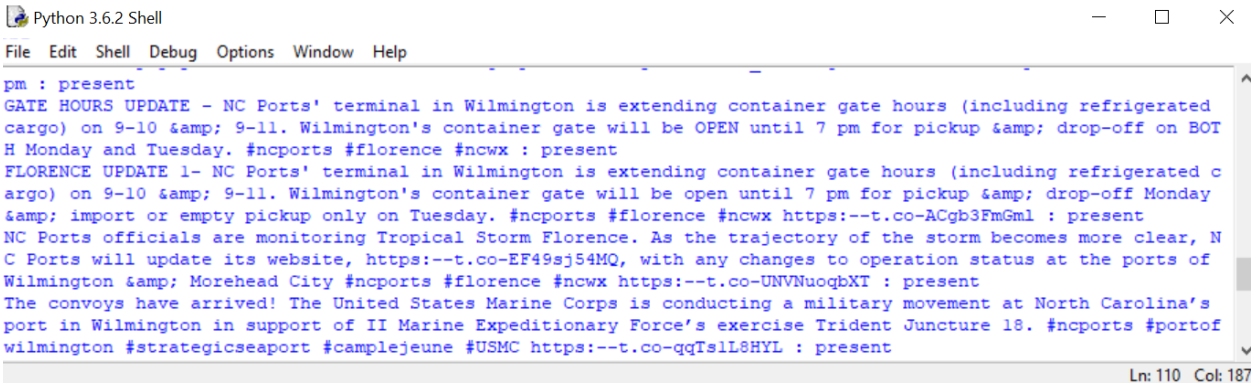

Figure 3.5: Example of tweets detected to be in present tense

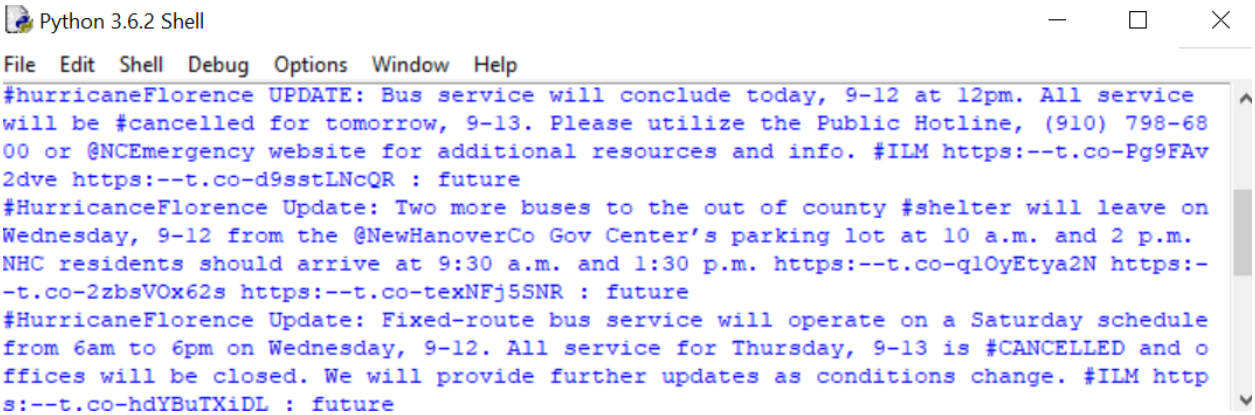

Figure 3.6: Example of tweets detected to be in future tense

Some examples that were incorrectly tagged by my model are shown in Figures 3.7 and 3.8.

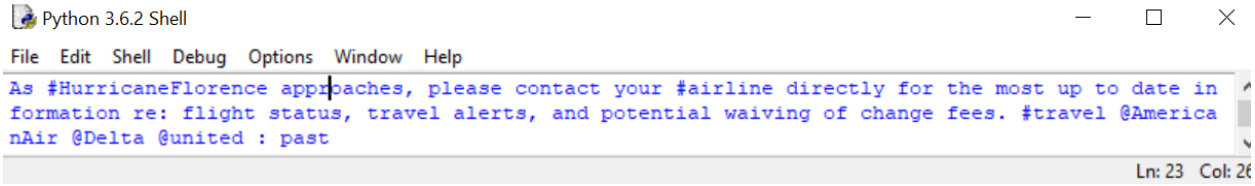

Figure 3.7: Example of a incorrectly labelled tweet

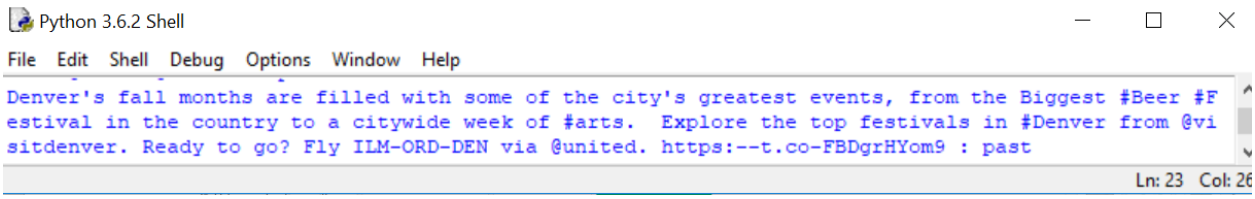

Figure 3.8: Another example of a incorrectly labelled tweet

3.2.12 Interesting results from tense analysis

The use of tense provided us with some very strong insights about tweets on events spread across time and space. I noted that the majority of the tweets were in present tense. This was an expected result. I did not remove spam tweets the first time I conducted the experiments. In one of the data-sets, most tweets in future tense were irrelevant. To verify if this was a trend across different data-sets, I perform two sets of tense analysis: one with the irrelevant tweets included, and another with all the spam tweets eliminated.

Tables 3.8, 3.9, and 3.10 show the number of tweets (including the spam tweets). The tables below show the number of tweets in each tense.

Hurricane Harvey Tweets	
Tense	# of tweets
Past	12303
Present	100576
Future	3181

Table 3.8: Hurricane Harvey: number of tweets in each tense

I also perform an analysis to see the number of tweets in present tense and compare with the time the tweet was created, to see if they are dependent on the time of the event.

Since the number of tweets in present tense is much greater than the ones in past or future, another question that arises is, while collecting tweets on different events on Twitter, can we

Hurricane Florence Tweets	
Tense	# of tweets
Past	24303
Present	191665
Future	5045

Table 3.9: Hurricane Florence: number of tweets in each tense

Hurricane Irma Tweets	
Tense	# of tweets
Past	9658
Present	77551
Future	1763

Table 3.10: Hurricane Irma: number of tweets in each tense

August 2018 Solar Eclipse Tweets	
Tense	# of tweets
Past	41720
Present	16121
Future	391443

Table 3.11: Solar Eclipse: number of tweets in each tense

Hurricane Harvey	# of tweets
Present tense before the event	2761
Present tense during the event	93434
Present tense after the event	4381

Table 3.12: Before, during, and after posts on Hurricane Harvey, and number in present tense

Hurricane Florence	# of tweets
Present tense before the event	24380
Present tense during the event	101657
Present tense after the event	65628

Table 3.13: Before, during, and after posts on Hurricane Florence, and number in present tense

Hurricane Irma	# of tweets
Present tense before the event	332
Present tense during the event	76882
Present tense after the event	337

Table 3.14: Before, during, and after posts on Hurricane Irma, and number in present tense

Solar Eclipse	# of tweets
Present tense before the event	6019
Present tense during the event	9672
Present tense after the event	430

Table 3.15: Before, during, and after posts on solar eclipse, and number in present tense

only focus on the tweets in present tense and discard ones in past or future tense without losing valuable information?

I also look at the number of tweets in present tense before, during, and after the the event and note down the counts in Table 3.12. Here, before, after, and during indicate the time period when the actual event occurred.

3.2.13 Location and time analysis

An important aspect of the analysis is to be able visualize this large set of times and locations on a map so I can make comparisons between the actual location and time of an event with the ones I inferred from the tweets. For visualizing the data, I use the Tableau public tool [24] as it provides flexibility of creating visualizations integrating multiple dimensions.

In order to study the locations from the tweets against time, I plot the locations retrieved from tweets and use color range to represent the time the tweets were created. The most evident color range that helped us comprehend time better was using the gold to red range with a step size of 25. If the step size is too low, it is hard to distinguish between the smaller differences. When the step size is too large, it makes it harder to differentiate between each step. There needs to be a proper balance between the two.

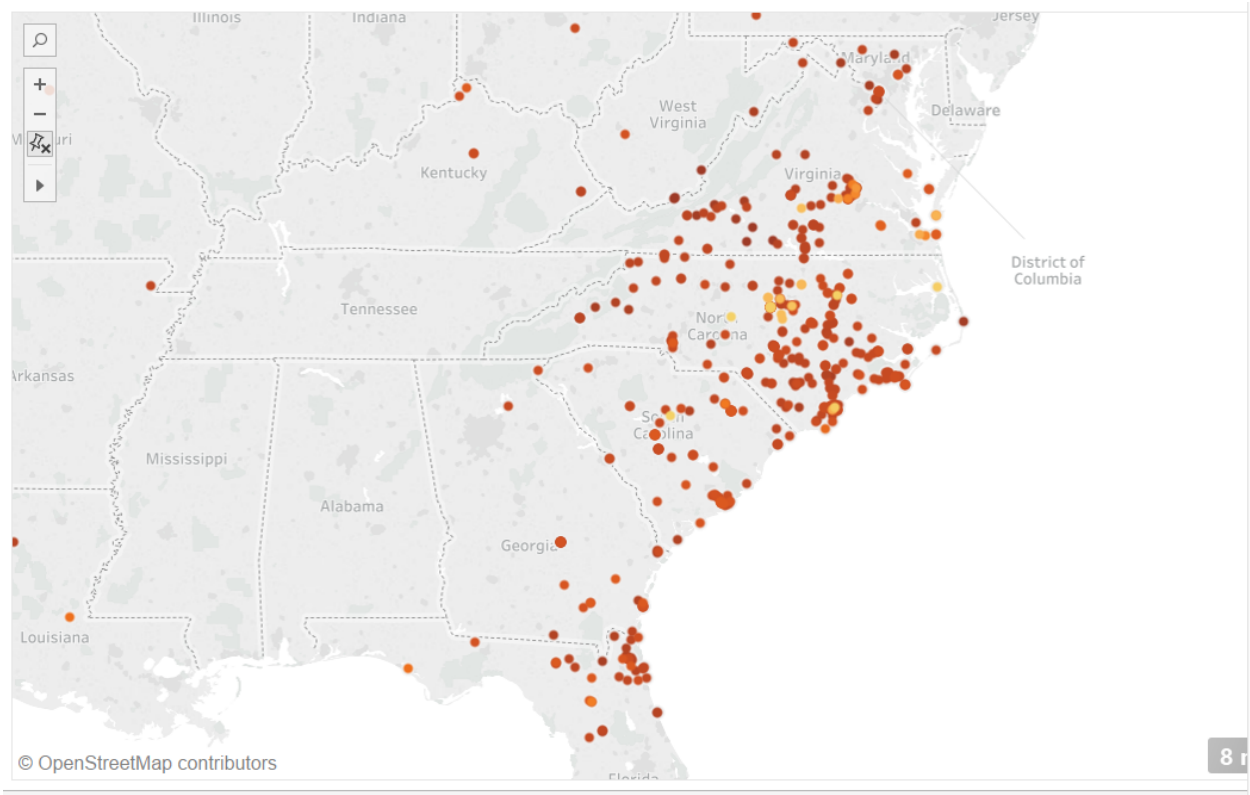

Figure 3.9: Locations represented by color range on the basis of created time

The figure above shows the color range variation with time but does not provide much clarity regarding the dates and times. Therefore, I use a time slider for viewing tweets with respect to time. This helps with the study of tweets in areas affected over a period.

I also create visualization in different colors for tweets in past, present, and future tense to see if there is any pattern in the data for the three data-sets.

Chapter 4

Results

Now that I have the different components required to perform my analysis, I investigate the results from location-time analysis of the four data-sets to get answers to the questions posed.

I created the time versus locations visualization for the four data-sets.

The most obvious observation from the visualization is that the locations retrieved are concentrated highly in the region of the event. For Hurricane Florence, the locations retrieved are in North Carolina, South Carolina, Virginia, and the northern region of Florida. When compared to the information on the internet about Florence, this accurately depicts the states in the path of the Hurricane. The visualization in Figure 4.1 shows the places affected by Hurricane Florence as of September 19 at 6:33 AM.

There are some locations detected away from the expected region. These are either a result of spam tweets or some incorrect results from the ambiguity resolution sub-module.

The tweets range from the dates August 15 to September 22, 2018. The number of tweets increases suddenly in the North Carolina and South Caroline region on September 13th and 18th (refer to Figure 4.4). This is when Hurricane Florence took a toll in these regions.

Similarly, the results for Hurricane Irma were compared with that of the path of the hurricane as in Figures 4.5 and 4.6. The tweets represent the locations quite accurately as the locations that were affected were the Caribbean Islands and the Virgin Islands. While the hurricane

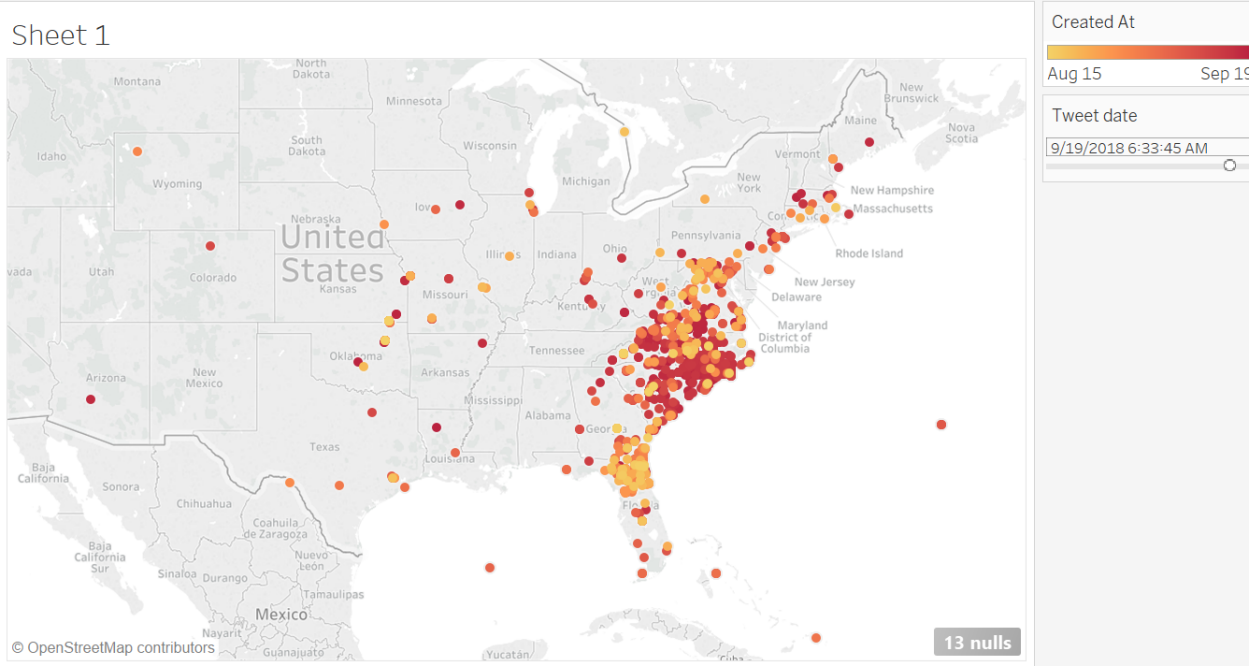

Figure 4.1: Visualization of tweets from Hurricane Florence

Figure 4.2: Hurricane Florence path till Friday [18]

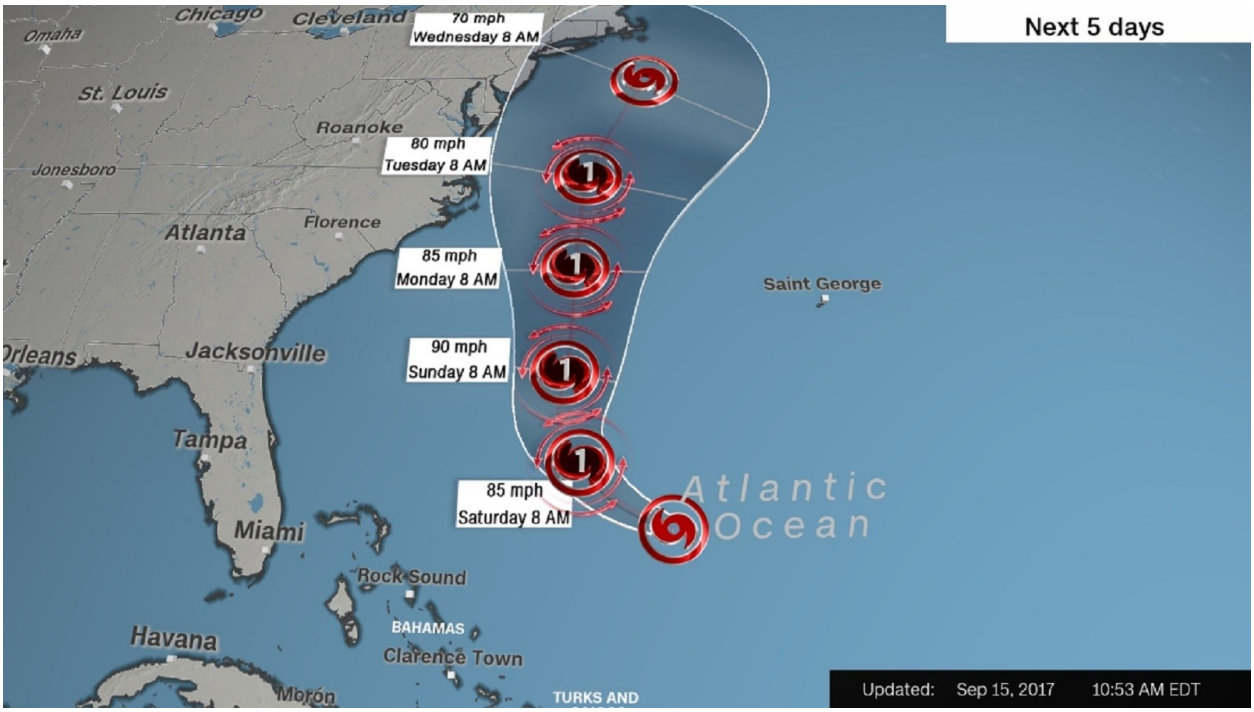

Figure 4.3: Path by Hurricane Florence in the next five days [4]

did not hit Florida directly, it had effects in this region too. From the predictions made, most tweets and locations are marked in Florida. The next dense region of prediction is the islands below Florida.

The tweets are in the range of dates August 27 to September 10. The number of tweets peaks on the 9th of September as depicted in Figure 4.7. Unfortunately, the data I had tweets created for the dates August 27 to September 10. Therefore, we do not see when the peak drops.

I also compared the visualizations for Hurricane Harvey. The actual locations (Figure 4.9) affected by Hurricane Harvey are Houston and Southeast Texas. Most tweets in the predictions (Figure 4.8) come from Houston. Below are the comparison between the ground truth and the ones we predicted.

Another important observation (Figure 4.10) is that the hurricane hit the regions from

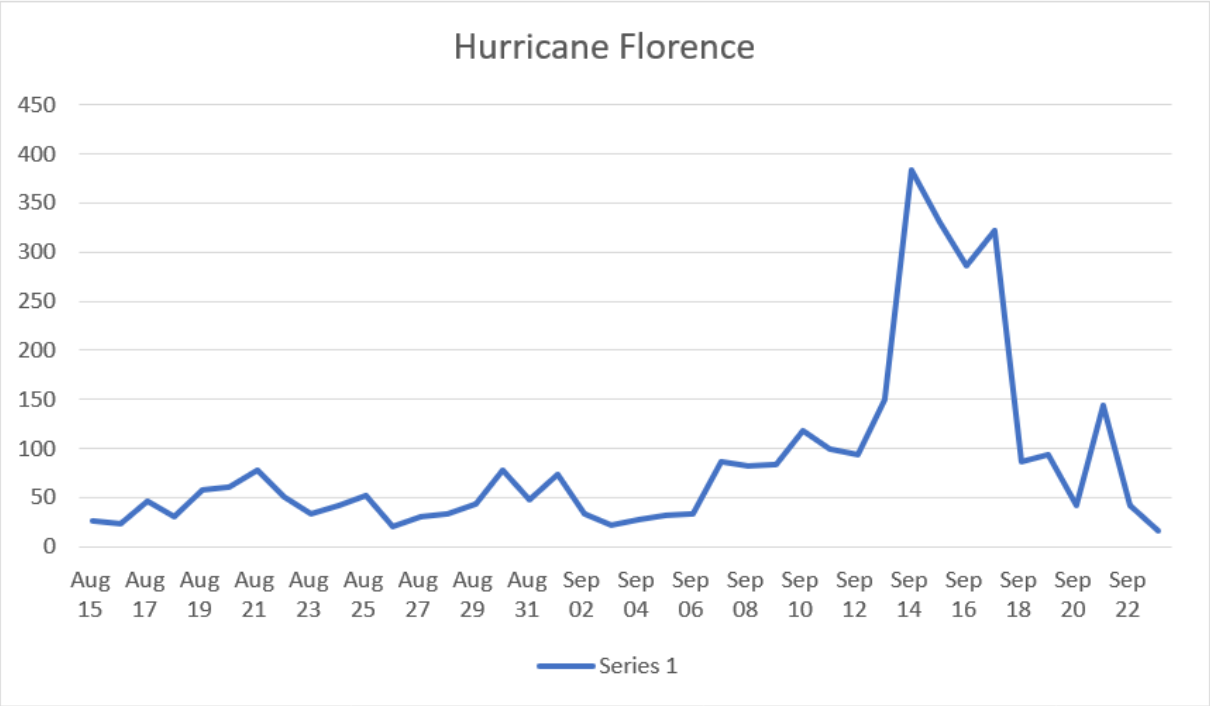

Figure 4.4: Histogram for Hurricane Florence representing the date when number of tweets peaked

August 17 to November 23. During the end of this period, August 25 is when the number of tweets peak.

The final predictions that I compare are of the August 2017 solar eclipse. From the predictions, I notice that the tweets in the solar eclipse collection have many more locations and do not have a proper path. Nevertheless, the tweets are more concentrated in the regions noted for total solar eclipse. The regions where total eclipse was visible are dense with tweets.

Time analysis of the solar eclipse was very interesting. The tweets range from dates July 23, 2017 to March 1, 2018. While a couple of locations were retrieved from tweets on September 20 to September 22, all of the other locations were from tweets on September 22 and September 23. This aligns with my assumption because after the solar eclipse people tweeted about it immediately. But we notice another slight peak in the number of tweets

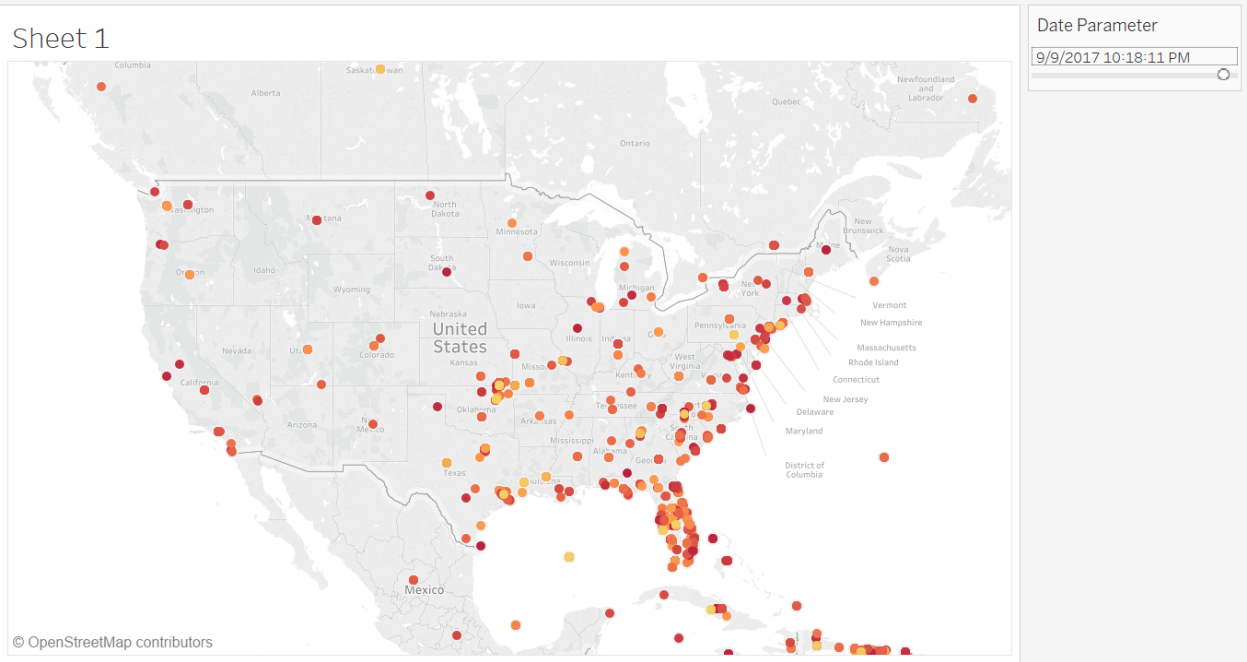

Figure 4.5: Visualization of tweets from Hurricane Irma

from August 31 to September 3. However, when I investigated the tweet content between these dates manually, I could not find any pattern in tweets between these dates. Since the path of totality of the eclipse is visible, I can gather information about the locations of view as well. (Please refer to Figure 4.11 and Figure 4.12.)

While I was able to tell apart the order of events by day, tracking the event by hour is a hard task, and I noted that this could not be accomplished using tweets. For example, it would be more useful to know at what exact time the solar eclipse could be viewed from different locations. But we could not achieve this level of detail.

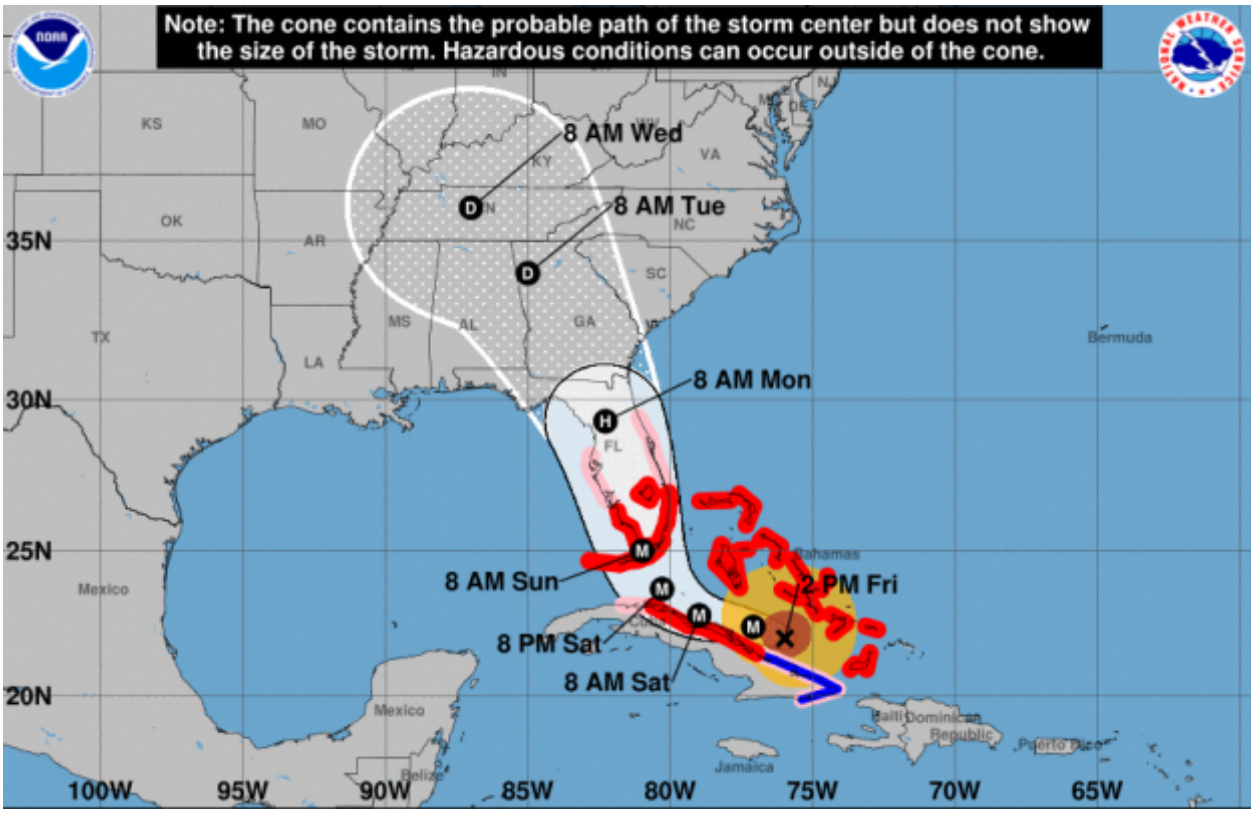

Figure 4.6: Hurricane Irma actual path [6]

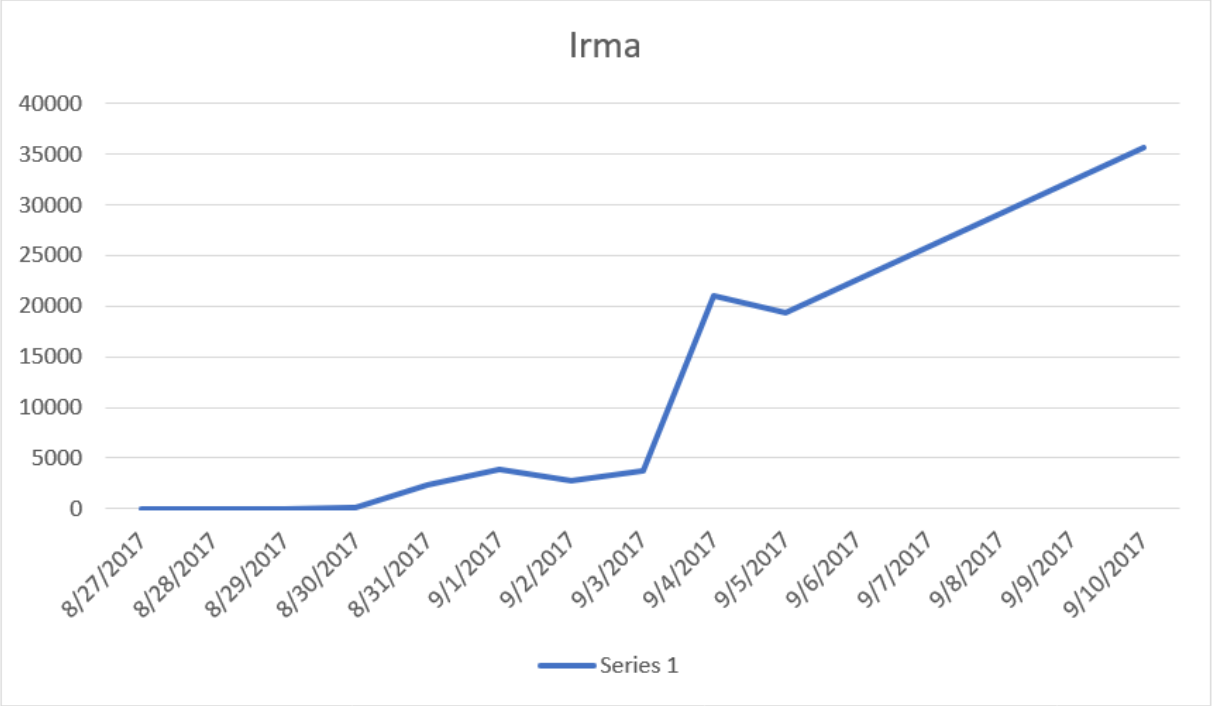

Figure 4.7: Histogram for Hurricane Irma representing the date when number of tweets peaked

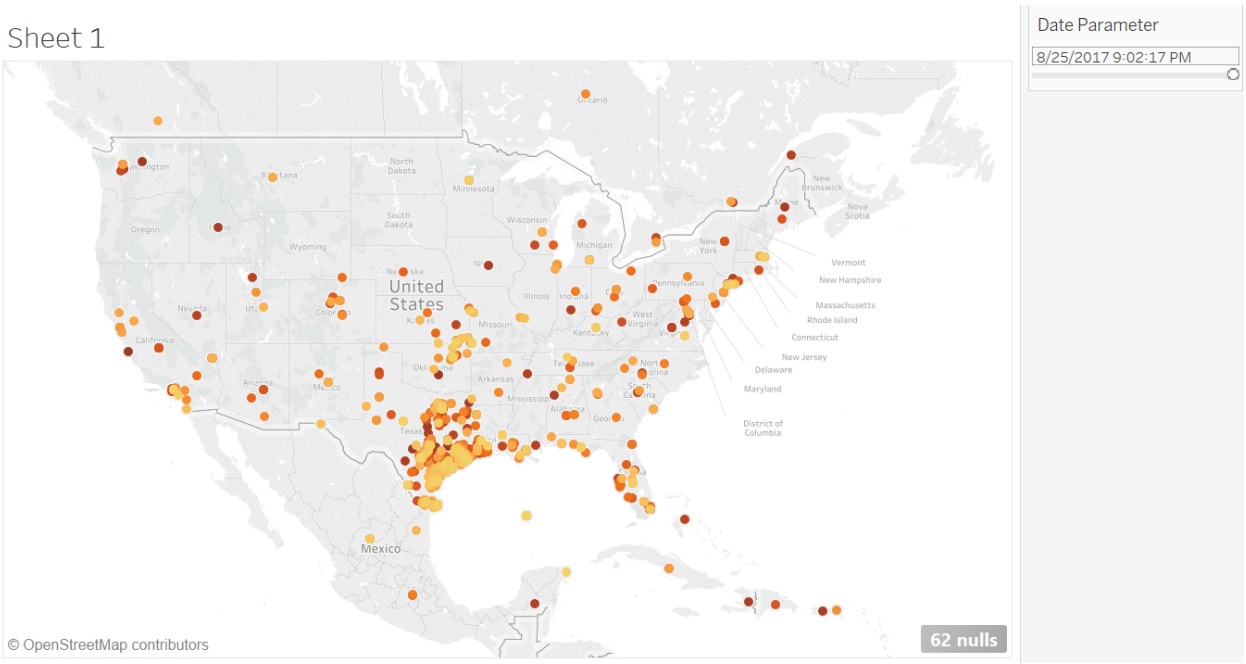

Figure 4.8: Visualization of tweets from Hurricane Harvey

Figure 4.9: Hurricane Harvey actual path [26]

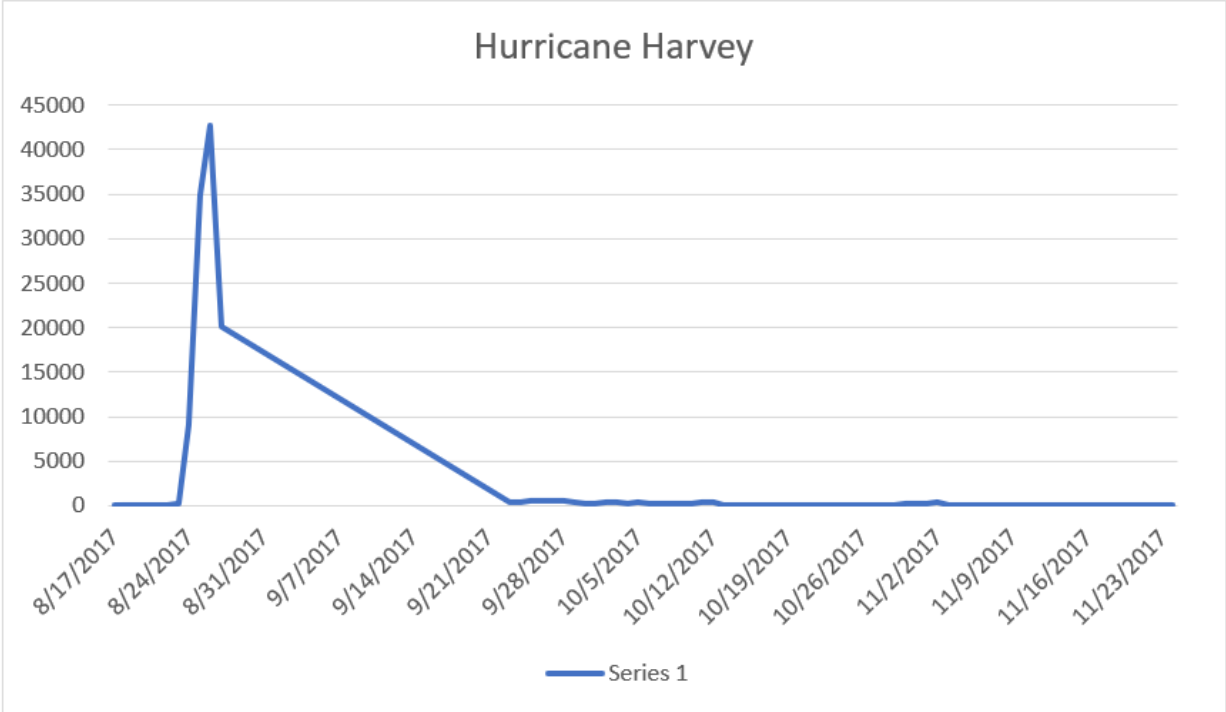

Figure 4.10: Histogram for Hurricane Harvey representing the date when number of tweets peaked

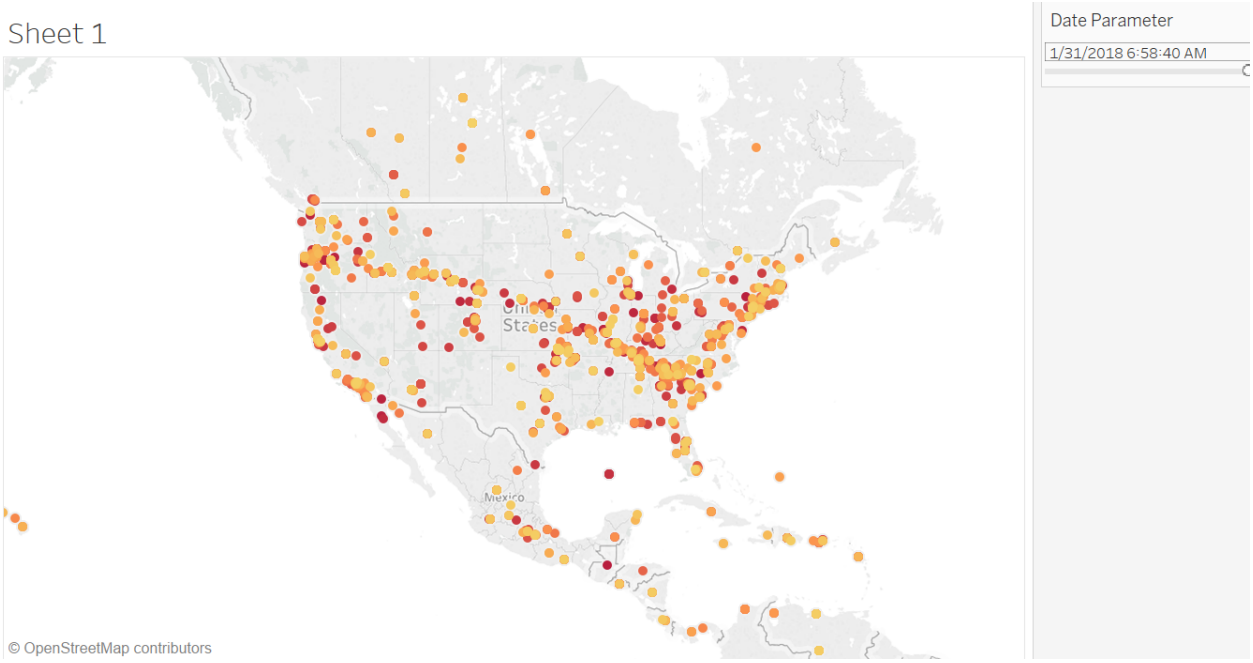

Figure 4.11: Visualization of tweets from solar eclipse

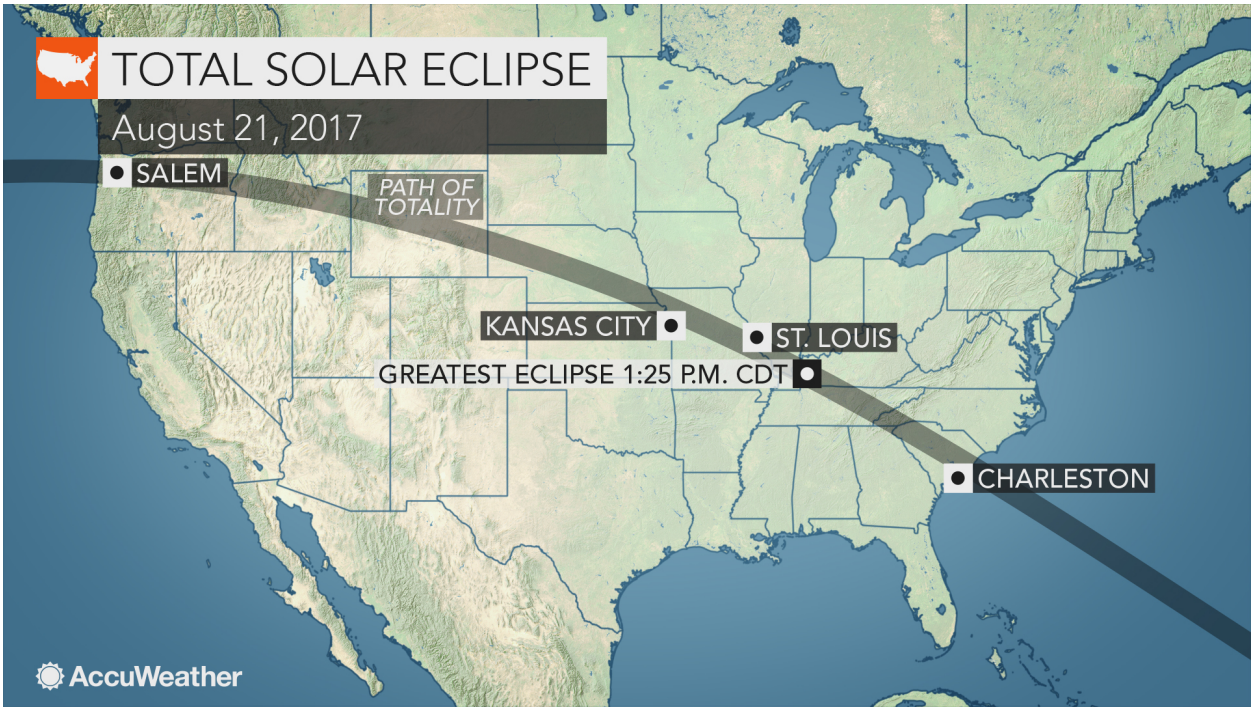

Figure 4.12: Solar eclipse actual path [2]

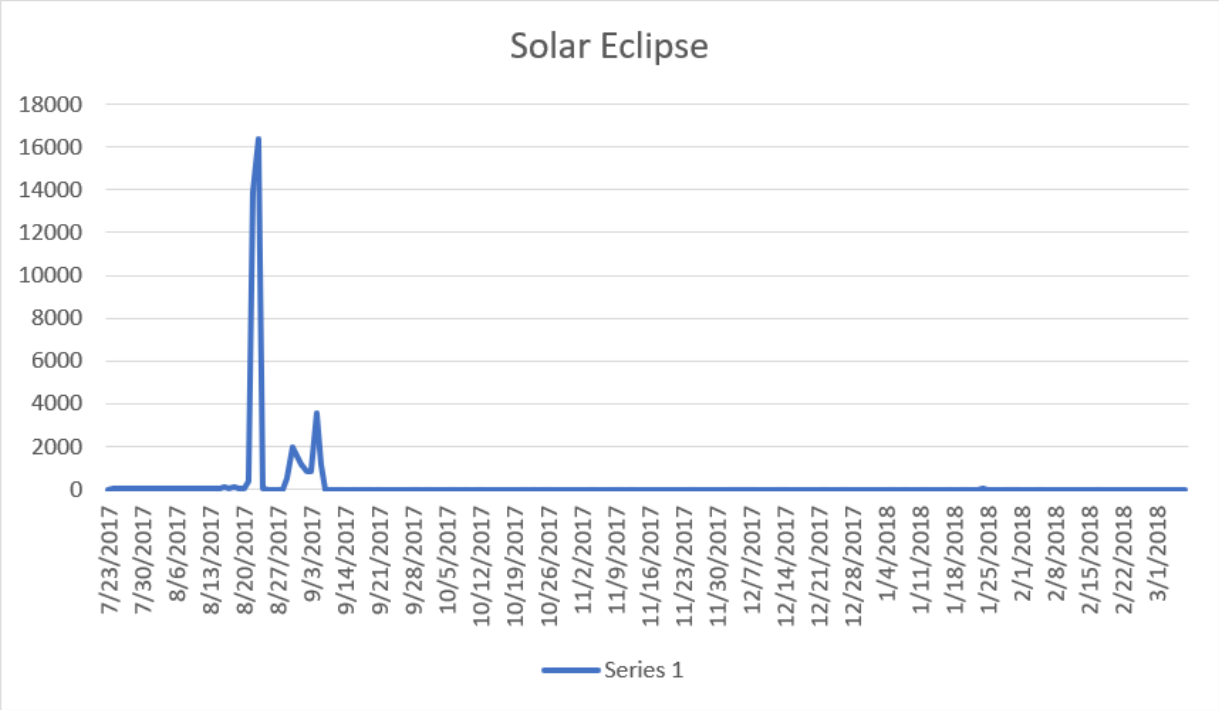

Figure 4.13: Histogram for August 2017 solar eclipse representing the date when number of tweets peaked

Chapter 5

Conclusions

I modify an existing method to extract locations that people talk about in tweets and extend these to successfully perform better in these data-sets. I use four features to do so effectively: named entity recognized locations, named entity recognized organizations, user's home location, and geo-coordinate information recorded. There are three parts to this existing method. The first one is extraction of features, the second one is geocoding with these extracted features, and the final step is using machine learning to disambiguate multiple locations. I used a manually labelled set from my collection to train the model. The model achieves an accuracy of 72.61% on the validation set.

I extract the temporal information based on the time recorded in the tweet and the tense of the tweet as a whole. When I used the rule-based model to detect tense, I obtained an accuracy of 77.43% on average.

I have seen how information retrieved from Twitter can be used to track moving events. From the results obtained, I see that the frequency of tweets during various events peak on the day of the event and the next day. The locations extracted from tweets can help in identifying the path of the moving events and also the affected regions. I also notice that for tweets from official sources, 40% of the tweets contain location indicative words, while from non-official sources, 12% of tweets contain location indicative words.

While I use data-sets of events that have already occurred, these methods can be used in

future events to track them in real time, and identify affected locations and act immediately. I also identify how to relate time of the tweets and the phase of the event, as well as how people relate the two in these collections.

For the four data-sets, the period during which the tweets talk about that respective event is on an average of 28 days around the time of the event. The solar eclipse was a anomaly here. Though with a much lower frequency, it was spoken about for 5 months after the event.

While news-wires talk about some of the most seriously impacted areas, I can see that using tweets I can find the most affected areas, even obtaining zipcode-level information.

Chapter 6

Future work

I have studied the use of tense in the four events, as well as its relation to the phases of events that people talk about in tweets. Extending this work, sentiment analysis of tweet content can help analyze people's reaction to these events in different phases. For example, people may express excitement while talking about the solar eclipse, but sadness when talking about the after effects of hurricanes, and then relief after a hurricane has passed.

Other new and more accurate methods could be used to locate the affected areas in similar events. [5] could be used to identify locations in tweets which may not contain any location indicative words. A comparison could be made between using location indicative words from tweets' context and the MENET architecture for extracting locations. Both of the results can be mapped too, to identify the method that provides more accurate locations for moving events.

Performing a classification on tweets to identify classes that tweets belong to, such as damage, evacuation, and political event, could help make this analysis complete.

Appendices

Appendix A

First Appendix

In this section, I add the important pieces of code in location analysis.

The first part shows the extraction of named entities from tweets.

```
1         def clean(text):
2             text = text.replace('"', ' ')
3             text = text.replace('`', ' ')
4             text = text.replace('\\', ' ')
5             text = text.replace('\n', ' ')
6             text = text.replace('\r', ' ')
7             text = text.replace('@', ' ')
8             return text
9
10        # will be deprecated
11        def get_sner(text):
12
13            # Tag using SNER server
14            tagger = ner.SocketNER(host='localhost', port=9199,
                  output_format='slashTags')
15            tagged = tagger.get_entities(text)
16
17            # Extract location only
```

```
18              locations = []  # list
19              try:
20                  locations = tagged['LOCATION']
21              except KeyError:
22                  # print 'No NEs'
23                  nothing = 0
24                  # locations = {'Null'}
25
26              # return string (list->string)
27              return ";".join(locations)
28
29          # will replace get_sner()
30          def get_sner_type(tagged, type):
31              entities = []  # list
32              try:
33                  entities = tagged[type]
34              except KeyError:
35                  nothing = 0
36              # return string (list->string)
37              return ";".join(entities)
```

The next piece of code shows part of the geo-coding algorithm:

```
1           def geo_code_gmaps(connection, locs, geo_locator):
2               addresses_num = 0
3               cities_num = 0
4               geo_id = 0
5               try:
```

```
6                locations = geo_locator.geocode('%s' % locs)
7                print(locations)
8            except (GeocoderServiceError, KeyError) as e:
9                print("Google error: %s", e)
10               location = None
11           lat = ''
12           long = ''
13
14           if locations is not None:
15               addresses = ''
16               lat_lng_set = set()
17               countries_set = set()
18               states_codes_set = set()
19               states_codes_countries_set = set()
20               cities_set = set()
21               cities_state_codes_countries_set = set()
22               json_raw = '{'
23               num_locations = 1
24               for address in locations:
25                   lat = address['geometry']['location']['lat']
26                   long = address['geometry']['location']['lng']
27                   lat_lng_set.add((lat, long))
28                   json_raw += '"'+str(num_locations)+'":'+json.dumps
                         (address)
29                   if num_locations!= len(locations):
30                       json_raw += ','
31                   num_locations += 1
```

```
32                  address_str = address['formatted_address'].rstrip()
33                  if address_str.endswith('United Arab Emirates'):
34                      adress_str = address_str.replace(' - ', ', ')
35                  address_elements = re.split(r',\s*', address_str)
36                  if address == locations[0]:
37                      addresses = address_str
38                  else:
39                      addresses += ';'+address_str
40                  addresses_num += 1
41                  country = address_elements[-1]
42                  countries_set.add(country)
43
44                  state = state_plus = state_code = city = ''
45                  if address_str.endswith('USA'):
46                      if len(address_elements) >= 3:
47                          city = address_elements[-3]
48                          cities_set.add(city)
49                          state_plus = address_elements[-2]
50                          if state_plus[0:2] in STATES_PCODE:
51                              state_code = state_plus[0:2]
52                          else:
53                              try:
54                                  state_code = STATES_PCODE_DIC[
                                        state_plus]
55                              except KeyError:
56                                  state_code = 'XX'
57                          states_codes_set.add(state_code)
```

```
58                      states_codes_countries_set.add(city+','+
                            state_code+','+country)
59                      cities_state_codes_countries_set.add(
                            state_code+','+country)
60                  elif len(address_elements) == 2:
61                      state_plus = address_elements[-2]
62                      if state_plus[0:2] in STATES_PCODE:
63                          state_code = state_plus[0:2]
64                      else:
65                          try:
66                              state_code = STATES_PCODE_DIC[
                                    state_plus]
67                          except KeyError:
68                              state_code = 'XX'
69                      states_codes_countries_set.add(state_code
                            +','+country)
70              elif address_str.endswith('UK'):
71                  a=0
72              elif address_str.endswith('Canada'):
73                  a=0
74              try:
75                  print(" ----------------------------")
76                  print(" %s address: %s country: %s state: %s
                        state_code: %s city: %s" \
77                          % (addresses_num, address_str, country,
                                state, state_code, city))
78              except UnicodeEncodeError:
```

```
79                      print("  ---------------(Unicode Removed) \n  %s"
                            % addresses_num)
80                  break
81
82              countries_num = len(countries_set)
83              state_codes_num = len(states_codes_countries_set)
84              cities_num = len(cities_state_codes_countries_set)
85              json_raw += '}'
86              if addresses_num > 0:
87                  geo_id = update_geo_google(connection, locs,
                        json_raw, addresses, addresses_num, ';'.join(
                        countries_set), countries_num, ';'.join(
                        states_codes_countries_set), state_codes_num,
                        ';'.join(cities_state_codes_countries_set),
                        cities_num, str(lat), str(long), int(time.time()
                        ))
88              else:
89                  print(" No return from Google")
90                  geo_id = update_geo_google(connection, locs, '{}',
                        '', 0, '', 0, '', 0, '', 0, '', '', int(time.
                        time()))
91                  failed.append(geo_id)
92              print("Geo_id(RETURN): " + str(geo_id))
93              return [geo_id, lat, long]
94
95          def geo_coding(connection, table, geo_locator, column):
96              if column == 'geoid_sl':
```

```
97                sql = "SELECT id, sner_locations FROM %s_ne2 WHERE
                      sner_locations!=' '" % (table)
98            with connection.cursor() as cursor:
99                try:
100                   cursor.execute(sql)
101                   total = str(cursor.rowcount)
102                   print("# of tweets(NEs): %s rows" % total)
103                   results = cursor.fetchall()
104               except MySQLError as e:
105                   print('Got error {!r}, errno is {}'.format(e, e.
                          args[0]))
106                   return
107           for row in results:
108               tid = row[0]
109               nes_combined = row[1]
110               nes_combined = nes_combined.replace(':', '')
111               nes_combined = nes_combined.replace('...', '')
112               print("Processing id: %s" % tid)
113               with connection.cursor() as cursor:
114                   sql = "SELECT geoid, lat, longi from z_90_geo where
                          query=%s"
115                   cursor.execute(sql, nes_combined)
116                   result = cursor.fetchone()
117               if result is not None:
118                   geo_id = int(result[0])
119                   lat = result[1]
120                   long = result[2]
```

```
121                     print("REUSE (Geo_id) : " + str(geo_id))
122                     update_experiments(connection, table, column, tid,
                            geo_id, lat, long)
123                 else:
124                     temp = geo_code_gmaps(connection, nes_combined,
                            geo_locator)
125                     geo_id = temp[0]
126                     lat = temp[1]
127                     long = temp[2]
128                 time.sleep(1)
129                 update_experiments(connection, table, column, tid,
                        geo_id, lat, long)
```

The next piece of code is showing the rule based tense analysis:

```
1     def determine_tense(sentence):
2             text = word_tokenize(sentence)
3             tagged = pos_tag(text)
4             tense = {}
5             tense["future"] = len([word for word in tagged if word
                  [1]=='MD'])
6             tense["present"] = len([word for word in tagged if word[1]
                  in ['VBP', 'VBZ', 'VBZ', 'VBG']])
7             tense["past"] = len([word for word in tagged if word[1] in
                  ['VBD', 'VBN']])
8             return tense
```

Appendix B

Second Appendix

In this section, I will describe briefly the procedure involved in visualizing the locations across time in the result section.

- I choose the columns ‘tweetid’, ‘createdat’, ‘latitude’, ‘longitude’ from the results obtained from location and time analysis.
- I map the two dimensions as latitude and longitude and use the symbol map in Tableau.
- I then plot the ‘tweetid’ on the maps. The apply the filter on ‘createdat’ and apply a color range to distinguish the time periods each tweet was generated.
- I create a parameter using the ‘createdat’ and apply a filtering formula to create the slider.

Bibliography

[1] 540co. yourtwapperkeeper. Github: https://github.com/540co/yourTwapperKeeper.

[2] AccuWeather. Webpage: https://www.accuweather.com/en/weather-news/2024-total-solar-eclipse-may-rival-last-years-great-american-eclipse/70005847.

[3] Denis Carriere. Geocoder. Github: https://github.com/DenisCarriere/geocoder.

[4] cnn.com. Webpage: https://www.cnn.com/2017/09/15/us/hurricane-jose-forecast-east-coast/index.html.

[5] Tien Huu Do, Duc Minh Nguyen, Evaggelia Tsiligianni, Bruno Cornelis, and Nikos Deligiannis. Multiview deep learning for predicting Twitter users' location. *arXiv preprint arXiv:1712.08091*, 2017.

[6] Oxford Eagle. Webpage: https://m.oxfordeagle.com/2017/09/08/hurricane-irma-path-what-time-will-irma-hit-florida/.

[7] Jenny Rose Finkel and Christopher D Manning. Joint parsing and named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 326–334. Association for Computational Linguistics, 2009.

[8] Google. Google Maps API. Github: https://developers.google.com/maps/documentation/geocoding/start.

[9] Matthew Honnibal and Mark Johnson. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[10] Ruihong Huang, Ignacio Cases, Dan Jurafsky, Cleo Condoravdi, and Ellen Riloff. Distinguishing past, on-going, and future events: The eventstatus corpus. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 44–54, 2016.

[11] Franziska Hübl, Sreten Cvetojevic, Hartwig Hochmair, and Gernot Paulus. Analyzing refugee migration patterns using geo-tagged tweets. *ISPRS International Journal of Geo-Information*, 6(10):302, 2017.

[12] Sabyasachi Kamila, Mohammad Hasanuzzaman, Asif Ekbal, and Pushpak Bhattacharyya. Resolution of grammatical tense into actual time, and its application in time perspective study in the tweet space. *PloS One*, 14(2):e0211872, 2019.

[13] Dan Klein and Christopher D Manning. Fast exact inference with a factored model for natural language parsing. In *Advances in neural information processing systems*, pages 3–10, 2003.

[14] Sunshin Lee, Mohamed M. Farag, and Edward A. Fox. A study on how location information is expressed in tweets and geo-locating tweets with less ambiguity. *Journal of Spatial Information Science*, 2019.

[15] George Washington University Libraries. Social feed manager. Webpage: https://doi.org/10.5281/zenodo.597278, 2016.

[16] Jalal Mahmud, Jeffrey Nichols, and Clemens Drews. Home location identification

of Twitter users. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):47, 2014.

[17] Melvin Earl Maron. Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM)*, 8(3):404–417, 1961.

[18] Len Melisurgo. Webpage: https://www.nj.com/weather/2018/09/hurricane_florence_major_hurricane_track_path_forecast_latest_update_landfall.html.

[19] Miles Osborne, Sean Moran, Richard McCreadie, Alexander Von Lunen, Martin D. Sykora, Elizabeth Cano, Neil Ireson, Craig MacDonald, Iadh Ounis, Yulan He, et al. Real-time detection, tracking, and monitoring of automatically discovered events in social media. ACL, 2014.

[20] NLTK Project. Natural Language Toolkit. Webpage: http://www.nltk.org/, 2017. Last accessed 11/08/2017.

[21] pypi. Geopy pypi. Webpage: https://pypi.org/project/geopy/.

[22] Behnam Rahdari, Tahereh Arabghalizi, and Marco Brambilla. Analysis of online user behaviour for art and culture events. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 219–236. Springer, 2017.

[23] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World Wide Web*, pages 851–860. ACM, 2010.

[24] Tableau. Webpage: https://www.tableau.com/.

[25] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[26] TX Weather Forecast Office. Webpage: https://www.weather.gov/crp/hurricane_harvey.