Improving Automatic Weather Observations with the Public Twitter Stream

Jeff Cox, Beth Plale

February 22, 2011

Abstract

Online social networks such as Twitter and Facebook have become a fixture in the lives of millions of people worldwide. Not only are people communicating with those in their social network, but applications like Twitter allow people to publicly broadcast information relevant to them. For the most part initial weather observations are done automatically, however, some aspects of the weather are still better observed by human eyes. In this paper, we argue that citizens report on the weather they are experiencing through social media tools such as Twitter. Citizen reporting through the Twitter stream will be less accurate than trained observers; however, we posit that the information can be accurate enough to overall improve reports of localized weather activity when contextually related through complex event processing. We develop a method to accurately mine weather events from the public twitter stream that detects primitive weather events from individual users tweets. The method will further detect clusters of all users primitive weather event tweets spatiotemporally and thus infer a real-world weather event. These real-world weather events mined from the Twitter stream are then used to improve automated weather observations within the same spatiotemporal region. We implement the proposed method using Streambase [1] and then evaluate the usefulness of the method. Unfortunately, our results indicate that the Twitter stream does not contain sufficient contextual information to be an ideal source for such spatiotemporal relationships and can not practically benefit reports of localized weather activity.

1 Introduction

Online social networks such as Twitter and Facebook have become a fixture in the lives of millions of people worldwide. Twitter alone boasts on average 65 million tweets a day [2]. Social networks are not by any means new, the notion has been around for a millennia and embodies the idea of all the people with whom one shares a social relationship [3]. However, with the emergence of applications that use the Internet as a means of social connection, online social networks tend to deviate from this traditional notion. Instead, online social networks have evolved into a constant stream of public and contextually relevant information. Not only are people communicating with those in their social network, but applications like Twitter allow people to publicly broadcast information relevant to them. Users often tweet about how their day is going, what they are eating, where they are going, what's happening where they are, etc... In general users tweet about events which are relevant to them and there is a wealth of information that can be mined from this continuous stream of data. Though, at first glance, the twitter stream is unreliable, unorganized and uncontrolled, it is still possible to obtain near-real time information about events and their locations given careful processing.

One category of events that are often tweeted about by Twitter users is the category relating to their current weather conditions. The content of these tweets range from a simple ”its raining”, to weather measurements such as temperature or wind speed. Given that some aspects of the weather are still better observed by human eyes and that human weather observations are decreasing as manned weather stations are replaced by automatic weather stations [4], tweets about the weather could prove useful.

Consider, for example, that radar based precipitation measurements are based on the reflectivity of material within the range of the radar sensor. Both hail and raindrops will reflect some amount of energy, but hail reflects a much greater amount of energy and this difference is used to detect one versus the other. However, there are a number of errors that can occur when using radar to detect precipitation. Some possible errors are the following:

Curvature of the earth: With greater distances from the radar sensor, the distance between the earth and the reflective material also increases. Hail or snow might be detected in the reflective material, but can change to raindrops if the distance to ground is great enough.

Anomalous Propagation: Atmospheric conditions can sometimes be such that the radar beams actually bend back toward the ground, and reflect off of buildings, hills, etc. and this can appear as rain where none fell.

Non-Precipitation Echoes: Birds, bats, aircraft, etc... can occasionally produce echoes that are not precipitation producing.

These errors can make it difficult to automatically detect weather events such as the precise type of precipitation. However, these errors can be mitigated by comparing them against trained human observers such as those in the Cooperative Observers Program (COOP) [4]. If the twitter stream can be mined for weather events they might provide a benefit to automatic weather observations similar to the benefit that is provided by these trained human observers.

Citizens report on the weather they are experiencing through social media tools such as Twitter. Citizen reporting of weather events can be highly accurate when carried out by trained observers, however, when carried out by untrained observers accuracy is less. Twitter weather reports are dominated by untrained experts. We posit that the information can be accurate enough to overall improve reports of localized weather activity. We evaluate the benefit that weather events reported though Twitter can bring to the overall reporting of weather patterns done through official weather sources. The approach is deemed useful if through monitoring observational data and Twitter streams together in real time we can determine a local weather event that might not have been detectable without both data sources.

We begin with a brief discussion of related work in section 2. In section 3, we describe our proposed method for improving automated weather observations with human weather observations mined from the Twitter stream. Section 5 describes the experiment and software implementation used to evaluate the proposed method. Section 6 discusses the results of the experiment and finally in section 7 is a brief consideration of future work.

2 Related Work

Previous studies of Twitter has included but not been limited to work such as [5] where the authors examine the conversational aspects of retweeting (which is a mechanism within Twitter by which users can ”forward” to their followers something tweeted by someone they follow). Java et al. investigated the motivation of twitter users in [6], messages were analyzed from Twitter users and then classified into nine different categories in [7] and finally Twitter activity during a forest fire in the south of France was analyzed to assess Twitter as a reliable source of spatiotemporal information in [8].

Twitter has also been used in the development of interesting and novel applications such as teaching english [9] and mobile learning [10]. Twitter has even been used to detect and notify registered users of earthquakes in Japan by applying Twitter to a probabilistic spatiotemporal model for events that can find the center and trajectory of event location [11].

Experiments in [12] suggest that Twitter provides a suitable open publish-subscribe infrastructure for using sensors and smart phones. A crowd-sourcing system architecture over Twitter was designed and two applications were developed (crowd-sourced weather radar and noise mapping application) as a means of evaluation. The architecture is comprised of two primary components, *sensweet* and *asktweet*. The first component is a mechanism by which to standardize the publishing of sensor observations over Twitter and the second a means to query ”crowds” for information.

A model for bridging between the physical environment and e-Science workflow systems through events processing was proposed in [13]. A proof of concept was implemented using event processing to detect and respond to severe storm patterns through sophisticated data mining algorithms. The results indicated that efficient stream mining algorithms can be used on real-time and continuous observational data streams.

Finally in [14], the reliability of using photos from the photo-sharing site Flickr was evaluated as a means to augment automatic weather observations. The authors use the coordinates and timestamps, automatically included, of photos depicting hail and collocate them with hail detected in the atmosphere. They evaluate the usefulness of this approach concluding that further exploration of Flickr photographs is warranted and that other social media sources should be considered.

3 Methodology

At the heart of the methods proposed in this paper is the idea that dissimilar and localized events can be used to detect global events if the events can be contextually related. For example, the localized event that the word ”raining” occurs within a string of 140 characters has any number of implications. If however some portion of the context in which the text was created is retained the possible implications can be narrowed as the event no longer represents the occurrence of the word, but instead represents a particular action, in a particular location and at a particular time. Furthermore, the possible implications of the original event can be yet again narrowed, if a cluster of the same particular actions, in similar locations and similar time frames are occurring.

The method described in this paper detects such events from both the the Twitter and Weather Surveillance Radar-1988 Doppler (WSR-88D) systems data streams. Twitter is an online social network (often classified as a microblogger) used by millions of people around the world and the WSR-88D data stream consists of NEXRAD Level II radar measurements made by Doppler radar sensors. These two streams will be further defined later in this section, but are very large and continuous On an average day there are around 750 tweets per minute, and each radar sensor in the continental United States produces data every 5 or 10 minutes. Such large and continuous data streams are often hard to manage, but well suited for such large and continuous streams is the complex event processing (CEP) paradigm and thus rooted are the methods described in this section.

We detect weather events on the Twitter stream by first identifying primitive Twitter events that correspond to tweeting about a particular type of weather, such as hail or rain. As these primitive Twitter events arrive they are used to derive a complex event which corresponds to the same type of weather as the primitive Twitter events, but the complex event indicates a greater degree of confidence that the type of weather is actually occurring. These complex events are derived by identifying clusters of spatiotemporally related primitive events. Complex events on the WSR-88D data stream are detected from NEXRAD Level II data using the algorithms defined in [15].

The complex events derived from both streams are used to derive a final event which indicates that a particular type of weather is occurring at a given location. These final events are arrived at by comparing the degree to which the Twitter events support the type of weather the WSR-88D events are indicating. By default, the final events will correspond to the WSR-88D events. However, a significant number of Twitter events that do not support this conclusion can sway the final event to correspond to the Twitter events instead.

To evaluate the benefit of this method in enhancing weather reporting we compare the final events and the WSR-88D events against official weather reports (having been already evaluated against human observations such as those from the COOP network) provided by the national weather service. We consider the methodology to be of benefit if the number of these final events correctly corresponding to weather that was actually occurring is greater than the number of correct WSR-88D events. The methodology was implemented using the complex event processing engine called Streambase [1]. Streambase was installed on a windows 7 VMware Fusion virtual machine configured with with 2gb of RAM and 2 processors.

The following subsections describe in more detail the above methodology. Subsection 3.1 considers event processing of the Twitter stream, subsection 4 the processing of the WSR-88D data stream and finally in subsection 4.1 the detail behind deriving the final weather events.

3.1 Twitter Stream

Ultimately the Twitter stream is a stream of status updates made by individual Twitter users as they go about their lives. The interface allows users to post up to 140 characters, that then can be read by any other twitter user by default. These posts are called ”tweets” and as of June 2010 there is on an average day 65 million tweets or 750 tweets per second [2].

For users to see the tweets of other users they must ”follow” other twitter users. The ”following” user will then receive a notification of the followed users tweet. The type of notification depends on the following users preferred client. For example, the default web client can be configured to send a text message to the users mobile phone when a new message arrives from a followed user. In addition to the default client there are now 70,000 registered applications [16] which allow users to interact with the twitter stream via a specialized client. These clients provide features in addition to the standard set available on the twitter website. Features such as mobile computing implementations and automatic geotagging of tweets based on gps receivers.

The public stream is accessible through the Twitter API and is different from the typical Twitter user access. The API is intended for applications that use Twitter and client's using the streaming API can connect to the Twitter stream and get public tweets without using the "follow" mechanism described above. The API, thus allows near-real time access to various subsets of public tweets.

Within the Twitter stream we define two primitive event types that give rise to complex events that correspond to real world weather events.

Weather Utterance Event - A weather related word is contained within the tweet. A more detailed description can be found in subsection 3.1.1

Weather Report Event - The tweet has been tagged with a weather related tag. A more detailed description can be found in subsection 3.1.2

These events do not necessarily correspond to real world weather events, but Instead they correspond to a human being in the real world "tweeting" about the weather or using words in their tweets that are often associated with the weather.

3.1.1 Weather Utterance Events

Primitive Event 1 Weather Utterance Event

```
event 'Twitter_Weather_Utterance_Event'
Longitude:
Latitude:
Timestamp:
Utterance_Word:
WeatherEventType:
NumNeighbors:
```

Tweets can be of poor grammar and contain informal language, thus making it difficult to computationally derive meaning from the tweet. However, we assume that when tweeting about the weather there are some words that must be used or "uttered" in order to talk about a particular weather event. We base this assumption on the notion that it can actually be quite difficult to indicate when it is raining via 140 characters without using the word "rain" or "raining". We derive weather utterance events from this assumption such that they occur when a particular weather related word is discovered in the context of a tweet. Other examples of weather related words that would infer a weather utterance event are: lightening, hail, thunder, snow, etc... **Primitive Event 1** describes the structure of weather utterance events.

As an example consider a tweet with the text "Its raining cats and dogs here!!". The word 'raining' within the content of the tweet would infer a weather utterance event and Table 1 shows how the tweet is mapped to the weather utterance event.

3.1.2 Weather Report Events

Weather report events correspond to tweets that are being posted to twitter to specifically 'report' on the weather. Generally these types of tweets use a hash tag so that they are easily identified

Tweet		*Twitter_Weather_Utterance_Event*	
Tweet Field	**Tweet Value**	**Event Field**	**Event Value**
Longitude	-87.627622	Longitude	-87.627622
Latitude	41.879722	Latitude	41.879722
CreatedAt	2010-09-16 10:01:05.000-0400	Timestamp	2010-09-16 10:01:05.000-0400
Text	”Its raining cats and dogs here!!”	Utterance_Word	rain
		WeatherEventType	Utterance
		NumNeighbors	0

Table 1: Example weather utterance event derived from a tweet containing the text ”Its raining cats and dogs here!!”

and contain only specific information regarding the weather. Some example weather report events correspond to the hash tags #wxreport or #weather.

#wxreport is a new and experimental project being piloted by NOAA (initiated due to Twitters new geo-tagging feature) which encourages Twitter users to post weather information in a specified format [17]. If the Twitter user does not make use of Twitters geotagging feature they are instructed to precede their weather information with location information. Location information is surrounded by ’WW”s. For example, a person might post ”WW 128 alameda st, San Francisco CA WW 88 degrees and raining”.

Primitive Event 2 Weather Report Event

```
event 'Twitter_Weather_Report_Event'
Longitude:
Latitude:
Timestamp:
WeatherEventType:
Temperature:
WindSpeed:
NumNeighbors:
```

#Weather on the other hand, is not a project, but created and accepted by Twitter users to post weather information and has no predefined format. Twitter users will simply post some status about the weather and append the hash tag #weather at the end of their tweet. For example, ”Its raining here in Bloomington #weather”. **Primitive Event 2** describes the structure of weather report events.

3.1.3 Twitter Weather Events

A single primitive event is insufficient to indicate a real world weather event. For example, consider a weather utterance event containing the word rain and that the tweet has nothing to do with the weather at all, but is instead referring to the name of a movie. Maybe the tweet is actually about a

Complex Event 1 Twitter Weather Event

```
event 'Twitter_Weather_Event'
Longitude:
Latitude:
Timestamp:
WeatherEvent:
```

weather event, but no such event is even occurring. Thus, some sort of processing must be done to insure the accuracy and reliability of the weather utterance/report events.

To increase confidence that a real world weather event is actually occurring within a specific spatial and temporal region, the primitive twitter events are aggregated into a single complex event corresponding to the uttered or reported weather event and based on the following criteria:

1. Many people are tweeting about the same weather event
2. Those people tweeting about the event are within the same spatial and temporal region
3. Similar utterance word (i.e. rain, raining, etc..)

The notion here being that if a significant weather event is actually occurring many people within the same meteorologically significant region will be tweeting about it. More precisely primitive twitter events infer a complex weather event once a cluster of primitive twitter events has been identified. This type of complex event is called a ”Twitter Weather Event” and is described by **Complex Event 1**.

When a primitive event is detected on the Twitter stream it is then evaluated against any other primitive events that have been detected. Given the new primitive Twitter events distance in time, space and utterance/report similarity from these other events, the number of neighbors to the event is determined. A *Twitter_Weather_Event* is detected when the number of neighbors of any given primitive event exceeds a predetermined threshold and thus resulting in a continuous stream of *Twitter_Weather_Events* as long as the threshold check is met.

By observing the Twitter stream for weather related tweets, we determined that setting this threshold to low could produce many false positives. So based on these observations we initially set the threshold to 10 primitive events as this seemed to reduce the overall number of false positives. However, this parameter is based on simple observation and will require a more rigorous empirical evaluation to determine the correct value. Quite possibly using a different threshold for each type of weather event.

4 WSR-88D Data Stream

The automated observation stream is a bit different than the Twitter stream in that the data flowing through the stream is only weather observation data. The automated weather observations are made by weather sensors designed to measure specific aspects of the weather such that weather patterns and events can be derived from those measurements. However, as with the Twitter stream the primitive events detected within the automated observation stream do not necessarily correspond to real world weather events. They are instead patterns detected in the base radar data of the Weather Surveillance Radar-1988 Doppler (WSR-88D) system sensors.

The data collected from the WSR-88D system is pulsed Doppler weather radar deployed throughout the United States to detect and indirectly measure meteorological and hydrological phenomena. The WSR-88D system provides real-time measurements of winds and precipitation to dramatically improve the ability to monitor and forecast weather. This data, called NEXRAD Level II data, is collected to directly support the missions of the National Weather Service (NWS), the Federal Aviation Administration (FAA), and the Department of Defense (DoD). However, Level II data is also used by university research and teaching programs throughout the U.S. in the fields of atmosphere science and climatology, hydrology, agriculture, transportation and logistics, aviation and air traffic safety, economics, air pollution and dispersion modeling, ecology, civil engineering, and many other disciplines [18].

Each WSR-88D Doppler radar station provides observations of precipitation and wind fields with extraordinarily fine temporal and spatial resolution, which are critical for understanding, monitoring, and predicting severe weather and flooding events. Each WSR-88D site continuously scans the precipitating or the "clear-air" atmosphere within 150 miles or so of the radar site and produces discrete fields of three base moments: radar reflectivity factor, mean Doppler radial velocity, and a measure of the width of the Doppler velocity spectrum [15].

> Base Reflectivity is one of the basic quantities that Doppler radar measures. This variable corresponds to the amount of radiation that is scattered or reflected back to the radar by whatever targets are located in the radar beam at a given location. These targets can be hydrometeors (snow, rain drops, hail, cloud drops or ice particles) or other targets (dust, smoke, birds, airplanes, insects).

> Base Velocity is the average radial velocity of the targets in the radar beam at a given location. Radial velocity is the relationship between the target's motion and the direction of the radar beam. Positive values denote out-bound velocities that are moving away from the radar. Negative values are in-bound velocities that are moving towards the radar

> Base Spectrum Width is a measure of velocity dispersion within the radar sample volume. The primary use of this product is to estimate turbulence associated with mesocyclones and boundaries.

Interested clients can connect to the IDD/LDM [19] data stream and receive the above measurements in near-realtime.

In the late 80's automated weather stations were built in force as well as the corresponding applications to detect weather events from these automated measurements[15].The WSR-88D system contains many algorithms which produce meteorological and hydrological analysis products derived from the base data described in the section above [15] and are made available as NEXRAD Level III data. There are a total of 41 Level-III products available in four categories.

> General: products such as baseline reflectivity and velocity, algorithmic graphic products spectrum width, vertical integrated liquid, and VAD wind profile

> Precipitation products: estimated ground accumulated rainfall

Overlay products: alphanumeric data with detailed information on identified storm cell

Radar Messages: radar system status

These algorithms are well understood and have been in use for many years, thus the primitive event in the automated observation stream correspond to the products output by these algorithms. Specifically we are interested in the Level III overlay category and some example events might be the detection of precipitation, hail or even mesocyclones. **Primitive Event 3** describes the structure of a WSR-88D event.

Primitive Event 3 Automated Observation Event

```
event `WSR-88D_Event'
    LongitudeMin:
    LatitudeMin:
    LongitudeMax:
    LongitudeMin:
    Timestamp:
    PatternDetected:
```

LongitudeMax,*LatitudeMin*,*LongitudeMin*,*LongitudeMax* and *Timestamp* are all mapped from the corresponding fields of the Level III data product and *PatternDetected* is determined by the type of product (i.e. PatternDetected = ”hail index”).

Depending on the product from which the automated observation event is mapped, the event can either correspond to a specific point or cover a rectangular region. The experiment conducted in this paper used the hail index product which correspond to single points and as such each events *LongitudedMin* = *LongitudeMax* and *LatitudeMin* = *LatitudeMax*

4.1 Deriving the Final Weather Event

Complex Event 2 Weather Event

```
event 'Weather_Event'
    Longitude:
    Latitude:
    Timestamp:
    WeatherEvent:
    TwitterWeatherEvent:
    WSR-88D_Event:
```

Finally, is the complex event described in **Complex Event 2**. This event is the result of augmenting the *WSR-88D_Event* with what is mined from the twitter stream. When both a *Twitter_Weather_Event* and a *WSR-88D_Event* are detected within the same spatial and temporal region for the same weather condition (i.e. raining, snowing, etc) then a *Weather_Event* is detected.

Twitter_Weather_Events and *WSR-88D_Events* are used to derive the final *Weather_Event* and the type of weather for which it corresponds (such as hail or rain). *Weather_Event*'s are arrived at by considering the *WSR-88D_Event* against the *Twitter_Weather_Event*'s that exist within the same meteorologically significant spatiotemporal region. The type of *Weather_Event* is then determined by comparing the confidence in the *WSR-88D_Event* with the number of *Twitter_Weather_Event*'s that do not support the *WSR-88D_Event*'s type of weather. By default the type of weather the *Weather_Event* corresponds to will be the same as the *WSR-88D_Event*, however, a significant number of *Twitter_Weather_Event*'s that do not support the *WSR-88D_Event*'s type of weather can lower the confidence in the *WSR-88D_Event* such that *Weather_Event*'s type of weather will correspond to the same as the non-supporting *Twitter_Weather_Events*.

5 Experimental Design

As mentioned in the introduction, we implemented the methodology using Streambase [1] for the specific case of rain and hail. Streambase's eventeflow CEP language was used for most of the development and java was used to implement various project specific adapters and operators. The following sections describe the resulting implementation and how it was used to determine the usefulness of the methodology.

5.1 Processing the Twitter Stream

A Twitter input adapter was developed in order to connect to the twitter stream using the tracks filter API call. Initially we had thought the input stream would be all public tweets, however, the twitter API will only stream a random one percent of all tweets if the tracks method is not used. This method requires the definition of keywords which will limit the returned tweets to only those containing the given keywords.We used the keword string "rain, raining, hail, hailing, #wxreport, #weather" to limit the input stream to only tweets corresponding to the utterances/hashes of interest.

Figure 1, 2, and 3 show the stages of processing *Twitter_Weather_Events* via the developed eventflow application. The symbols in each figure represent the following:

1. **Thick Arrow:** The represent either input or output streams
2. **F(x):** Represents a mapping of the input events to the output events, based on a defined function
3. **Funnel:** Represents a filtering of the events
4. **Fork:** Represents an ordered split of the event stream. Events are sent down a stream, when the given event has completed processing on the previous stream.
5. **Qtable:** Represents a query
6. **Triangle:** Represents a datastore of events

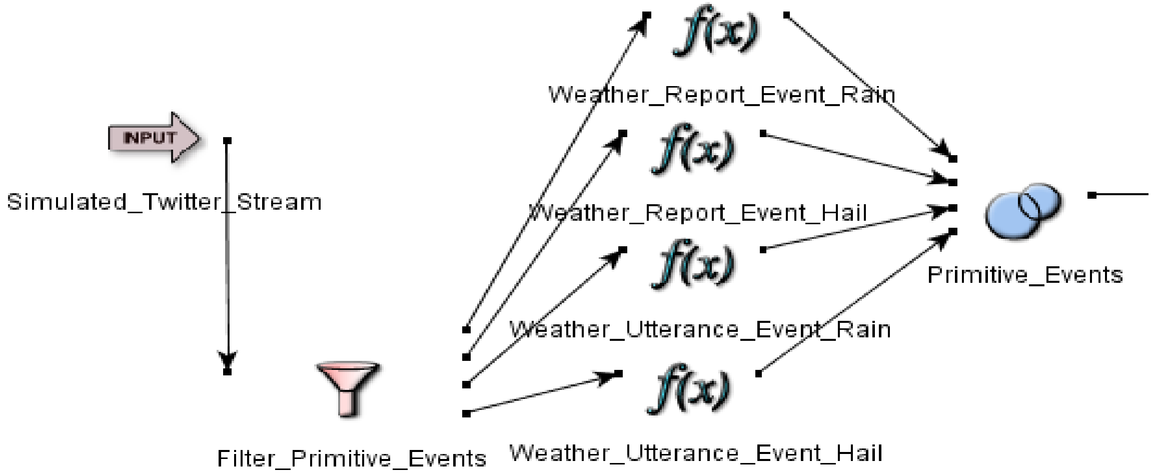

Figure 1: Determining the Primitive Events

5.1.1 Determining Primitive Events

To detect *Twitter_Primitive_Events* regular expression matching was used on the text of each incoming tweet. It might appear that by using keywords through the Twitter API regular expression matching would not be necessary. However, our experience was that there were incoming tweets that contained none of the keywords. Furthermore, we filtered out tweets with the words ”rain” and ”hail” as they produced far to many false positives. The final set of keywords used to detect *Primitive_Events* were the strings: ”raining”, ”hailing”, ”#wxreport”, ”#weather”. Once a tweet was matched the corresponding *Primitive_Event* was created and passed on for further processing.

5.1.2 Determining Tweet Location

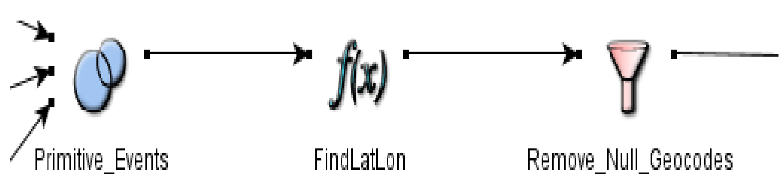

Figure 2: Determining Tweet Location

Critical to the overall method is being able to relate mined events from the Twitter stream with events in the radar data stream. However, acquiring the precise location of the tweets proved more difficult than first believed. Though Twitter provides a mechanism for the encoding of GPS data into each tweet, and most mobile clients provide this capability, very few tweets arrived with a latitude and longitude.

In order to try and increase the number of tweets with useful location information, the *Location* field of each tweet was used to derive latitude and longitude information. The location field is part of the twitter users profile and entered by the user. It is not as precise as the geo-coded information, however, it is much more prevalent. That being said, the field is a text field and contains all manner of text to identify a person's location. Some have a comma delimited string of latitude and longitude, some have city and state, while others have useless information such as "the earth". As depicted in figure 2 the information in the location field was mapped to a latitude and longitude (*Find_Lat_Lon*) if possible. This mapping was accomplished by parsing the location field for latitudes and longitudes and then if found inserted into the events Latitude and Longitude fields. In the future, a database lookup could be used to map city and state to latitudes and longitudes and potentially increase the amount of tweets with location information. However, the effects of the imprecision of city and state information with regard to the users real location would need to be considered. Ultimately any tweet with a null latitude and longitude was filtered out of the stream (*Remove_Null_Geocodes*).

5.1.3 Twitter Weather Events

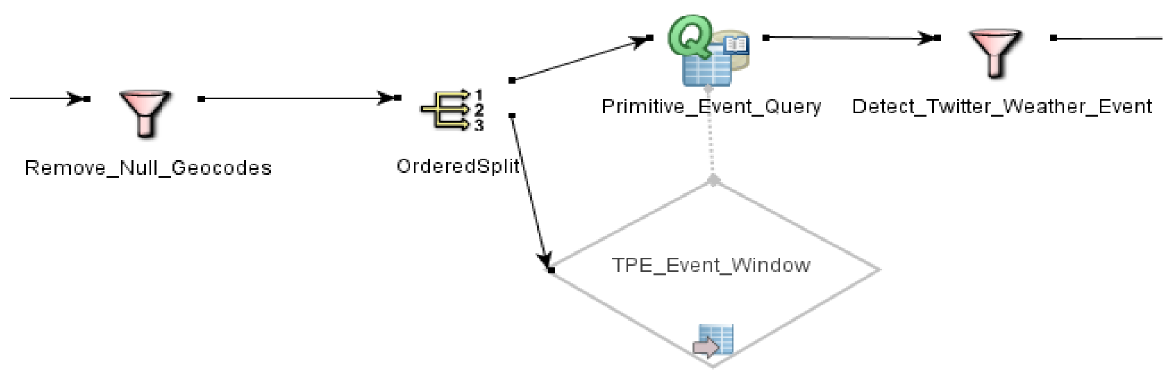

Figure 3: Determining Twitter Weather Events

Figure 3 depicts the eventflow application stage responsible for determining the existence of a *Twitter_Weather_Event*. *Twitter_Weather_Event*s are derived from *primitive_Events* that are related in space, time and weather word utterances. To implement this each incoming *Twitter_Primitive_Event* is used to query (*NumNeighborsQuery*) against a 30 minute sliding window (*TWE_Event_Window*) to find all *Twitter_Primitive_events* that would be neighbors to the current event. *Twitter_Weather_Events* are then detected by filtering for greater than or equal to 10 neighbors. The windowing mechanism used is implemented in streambase as a *materialized window* streaming operator. Parameters allow for windows to be instantiated based on a fixed number of tuples, on a time interval, or on a field value of the events contained in the stream. A 30 minute interval was chosen given that a storm cell lasts for 20 - 30 minutes on average.

5.1.4 Twitter Corpus

Historical tweets are not available from Twitter, save for a few days into the past. Even then, that data is only accessible via specific search strings on the search API. So the corpus was generated by recording tweets from the public twitter stream. Two corpus's were recorded into a comma delimited text file and then used to simulate the Twitter stream. The first corpus was recorded over a two week period of time without any keyword searching applied. During this time only $5,051,446$ tweets were recorded as it turns out that if no searching is applied, clients will only receive a 1% random sample of the full stream. Given that only 1% of the stream was being collected in the first corpus a second corpus was collected, but only for a single day. Keywords were used to collect this corpus and $109,984$ tweets were recorded.

5.2 WSR-88D Event Processing

For the purposes of this paper the WSR-88D event processing consisted of parsing the NEXRAD Level III data stream for hail and precipitation analysis products. These *WSR-88D_Events* were then sent to the eventflow application as a comma delimited text string. The specific analysis product was the hail index and derived for the same 24 hour period as the second Twitter corpus. In total there were $22,116$ *WSR-88D_Events* over the 24 hour period, and they can be seen as red triangles in Figure 4. From this map we can see that most of the *WSR-88D_Events* were in the same geospatial regions. Futhermore, if we consider figure 5 we can get a good sense of not only the data's spatiotemporal relationship, but also the probability hail will be produce

5.2.1 Improving WSR-88D Weather Events

Finally, we see how the WSR-88D events are combined with *Twitter_Weather_Events* as depicted in figure 6. As *WSR-88D_Event*s arrive they are used to query against a 30 minute sliding window of *Twitter_Weather_Events*. The query determines if there are any *Twitter_Weather_Event*s that exist within the spatial bounding box of the given *WSR-88D_Event* and then the returned set of events is used to potentially alter the weatherEventType field. A thresholding algorithm was used to determine if the *WSR-88D_Event* should change (i.e. correspond to rain rather than hail) such that if the query returned at least five equally contradictory *Twitter_Weather_Event*s the *WSR-88D_Event*'s eventType field was altered to reflect the *Twitter_Weather_Event*'s weatherEventType

6 Experimental Results

ne of the primary assumptions made by this method is that if a group of people close to each other and at relatively the same time are using a given weather related word in their tweets, the weather event corresponding to that word is likely to be happening at the given location. No such groups were detected during the experiment, in other words no *Weather_Events* were detected.

To understand why no *Weather_Events* were detected recall that any tweet without location information is thrown out. Now consider Table 2, it shows the relationship between geocoded and non-geocoded tweets (those that were thrown out). Column three is the total number of tweets that would be considered primitive events if they had not been excluded and column four the total number of tweets with sufficient location information to be considered primitive events. We see

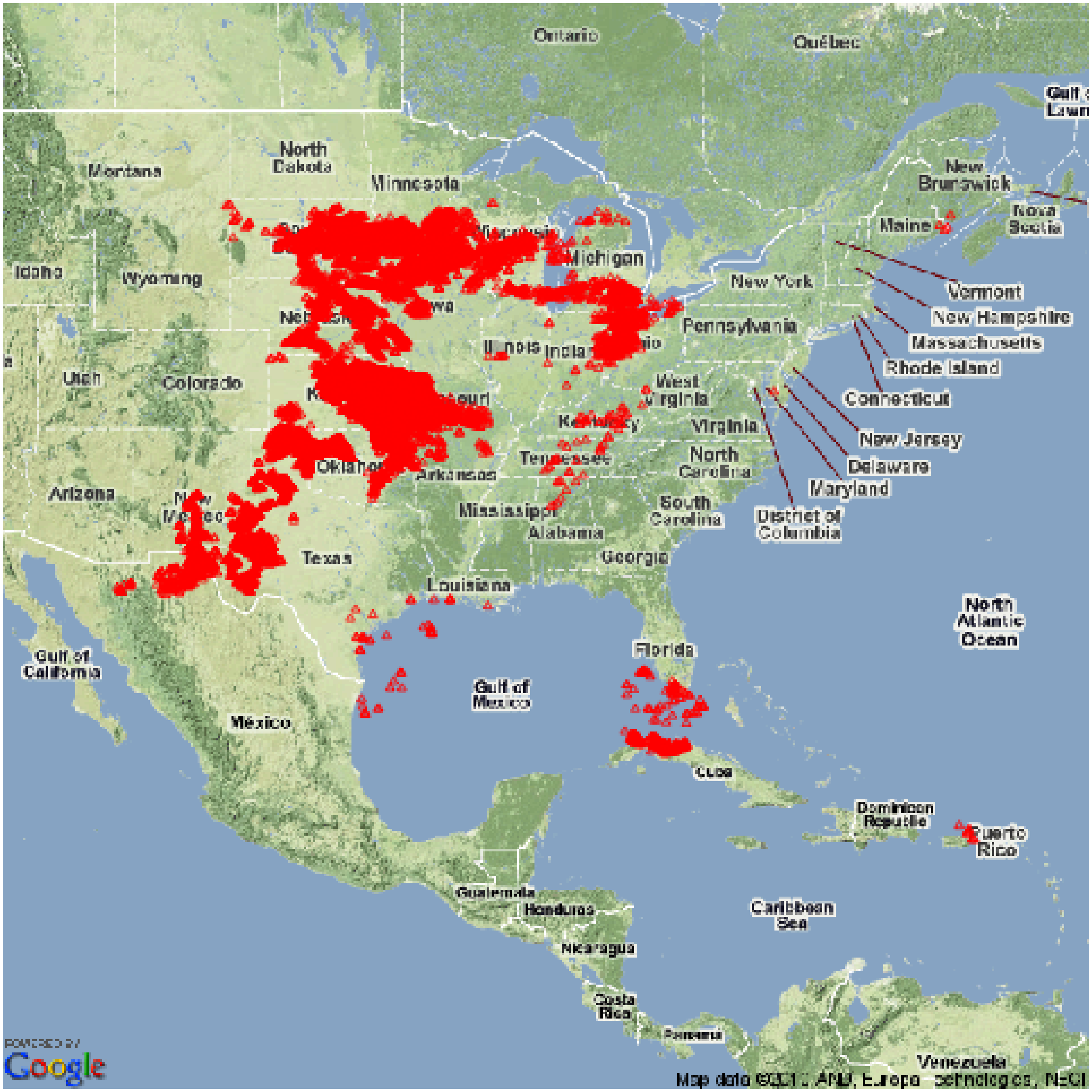

Figure 4: 24 Hour WSR-88D Data. The red triangles indicate that there was some probability of hail at the location sometime during the given 24 hour period.

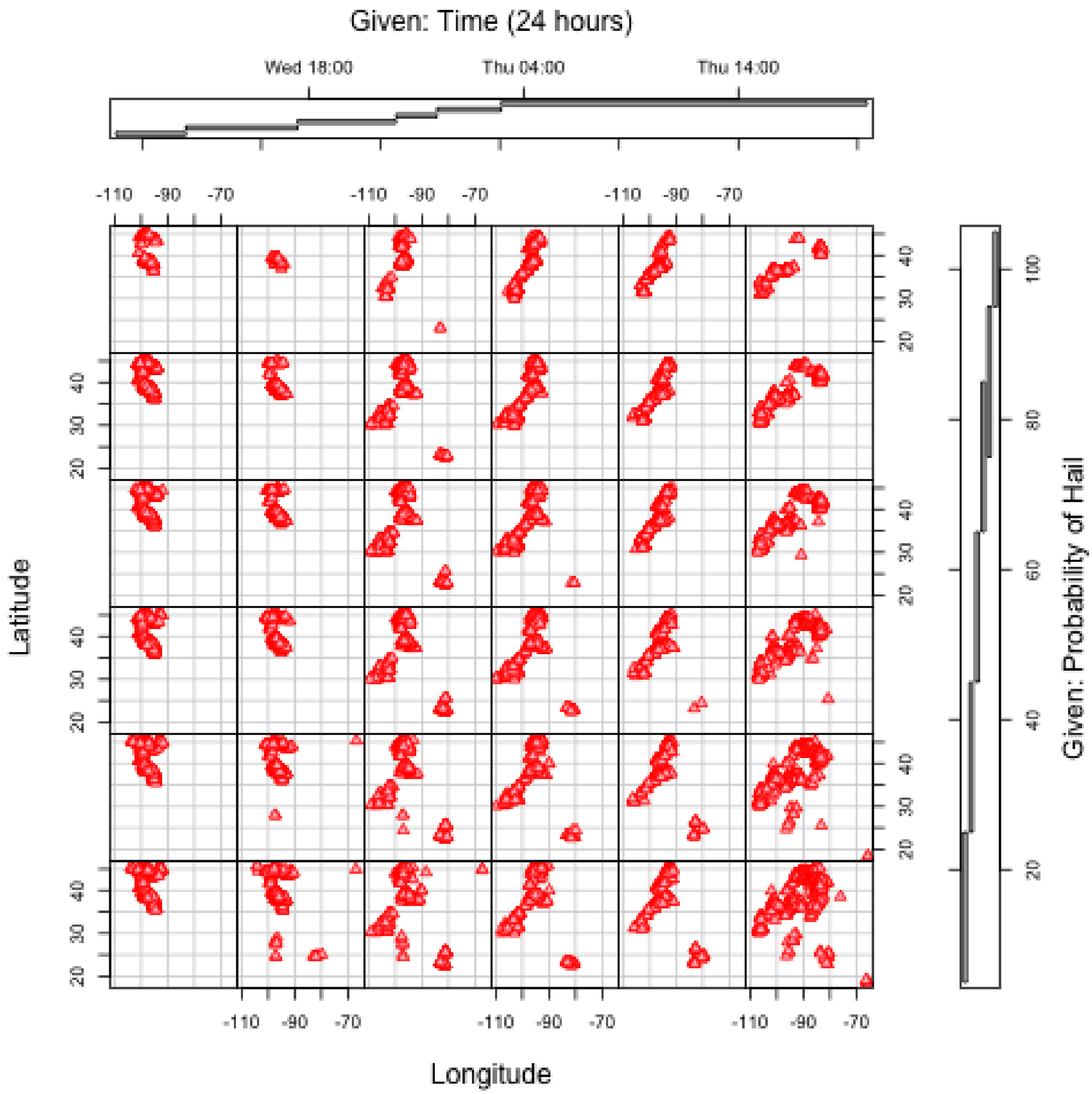

Figure 5: Relationship of the 24 Hour *WSR-88D_Events* grouped by time and probability

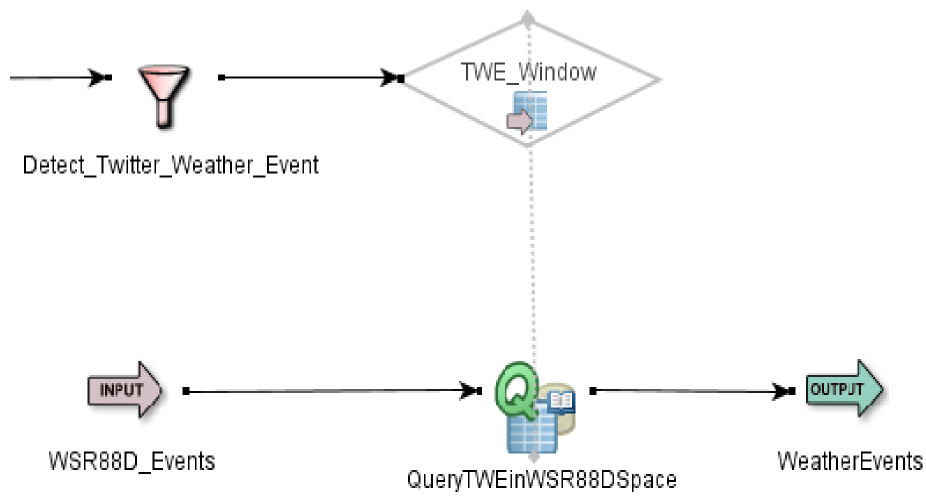

Figure 6: Detecting Real-World Weather Events

		Primitive Events	
Corpus	**Total Tweets**	**Non-Geocoded**	**Geocoded**
2 Week Corpus	$5,051,446$	6305	12
24 Hour Corpus	$109,984$	1328	33

Table 2: The number of geocoded vs non-geocoded primitive events. The number that are not geocoded was unexpected.

that in the last column the number of geocoded tweets is quite low. Though detecting such a low number of primitive events is unexpected, it does not necessarily imply that no *Weather_Events* would be detected. If the detected primitive events could be considered meteorologically significant spatiotemporal neighbors, *Weather_Events* should have been detected.

Let's consider the spatial and temporal relationship between the primitive events in each corpus. Figure 7 depicts the location of primitive events over a period of two weeks, while Figure 8 depicts the location of primitive events over a period of 24 hours. In both figures the z axis represents the time and day each event occurred. The events detected in neither corpus are particularly close in time. However, the 24 hour corpus does have a few events that are within a minutes of each other. To further investigate the spatial relationship between the primitive events let us also consider them overlaid onto a map as can be seen in figures 9 and 10. It can be clearly seen from these figures is that the spatiotemporal relationship between the primitive events is actually quite poor. A few in the 24 hour corpus are related, but not enough to have indicated a *Weather_Event*.

The lack of spatial information is certainly a serious road block for the methodology proposed and rather discouraging. However, though *Weather_Events* are not being detected, primitive events are being detected and it is possible that by trying to group the primitive evens to detect *Weather_Events* was the wrong direction. So we consider whether just the primitive events and the *WSR-88D_Events* by plotting them together in figure 11. Though this figure does not validate nor invalidate the idea of clustering primitive events together, it can be seen that the primitive events are in and of themselves insufficient.

6.1 Discussion

The lack of geocoding made it impossible to accurately relate enough tweets to derive anything meaningful from the Twitter stream. It appears that in recent weeks Twitter has begun to make changes in its geocoding policies so as to provide a greater perception of privacy. Where once Twitter users might set their location when they create their account, now the setting only lasts for 24 hours before the user is prompted to update. Moreover, when tweets arrive at the Twitter servers any geocoding added to the tweet by mobile devices is removed and the tweet's location field updated with a less granular location. In other words, a latitude and longitude indicating the north side of Chicago will be converted to the location "Chicago, IL". This behavior can be changed in the users profile, but is active by default.

Beyond the difficulty in getting useful data from the Twitter Stream and though there is a very functional API for connecting to the Twitter stream, there are a number policy driven limitations. We experienced at least one of these when trying to record tweets over a two week period of time. The number of tweets on an average day is nearly 65 million, but over the two week period of time we collected only $5,051,446$ tweets. With proper authorization Twitter will increase the amount that can be collected to a 10% random sample. This is still not likely to be sufficient for detecting weather events such that they can be used to improve overall weather reporting.

Validating the method such as the one proposed in this paper will prove difficult or at least timely without better access to historical tweets. A significant amount of time was spent in trying to acquire subsets of past tweets such that they could be processed with significant historical weather events. Twitter does not provide such data (we contacted them), though recently they donated all past public tweets to the Library of Congress. Unfortunately, the Library of Congress does not intend to make the data available, but instead use the data to enhance the public "mood" of a given

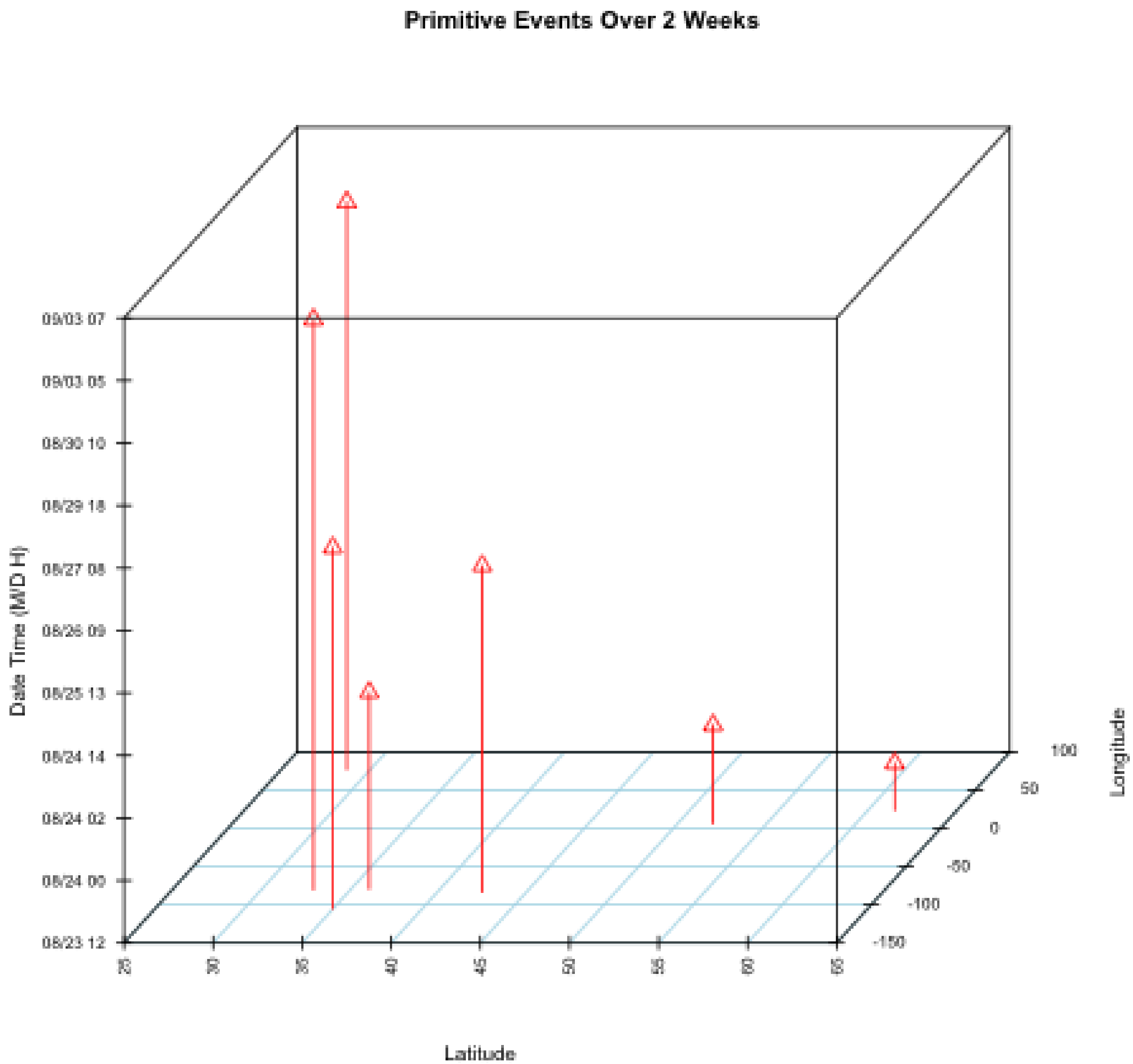

Figure 7: Temporal relationship of the 2 week Twitter corpus. Each triangle represents a detected primitive event between 12pm on 8/23/2010 and 9/03/2010. The Z axis corresponds to the date and hour the event occurred. Clearly these events are temporally distant.

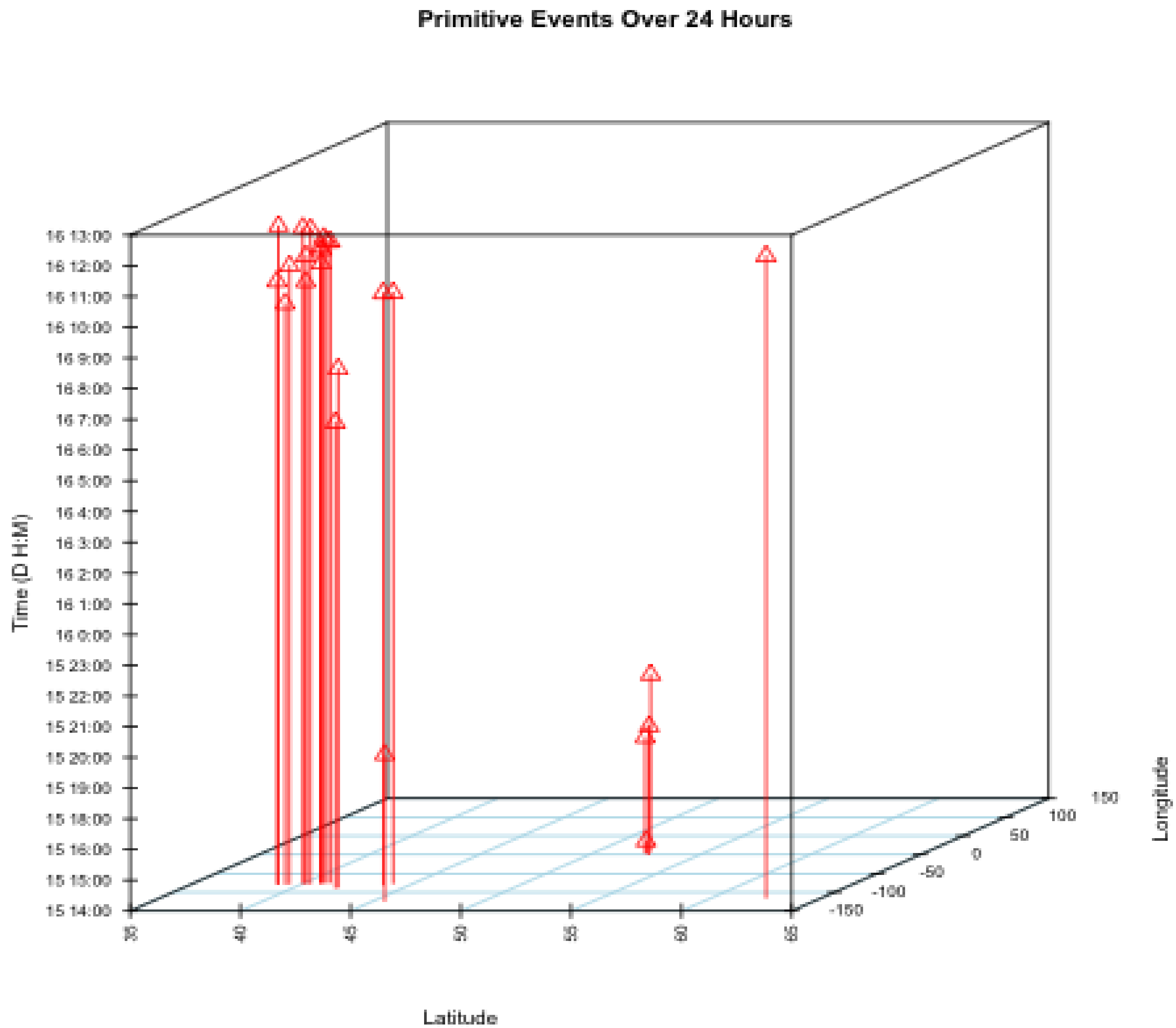

Figure 8: Temporal relationship of the 24 hour Twitter corpus. Each triangle represents a detected primitive event between 2pm on 9/15/2010 and 2 pm on 9/16/2010. The Z axis is labeled with the day, hour and minute the events occurred. There are a few events that are temporally related between 11am and 1pm on the 16th.

Figure 9: 2 week Twitter corpus location overlay. Clearly the events are spatially distant

Figure 10: 24 hour Twitter corpus location overlay

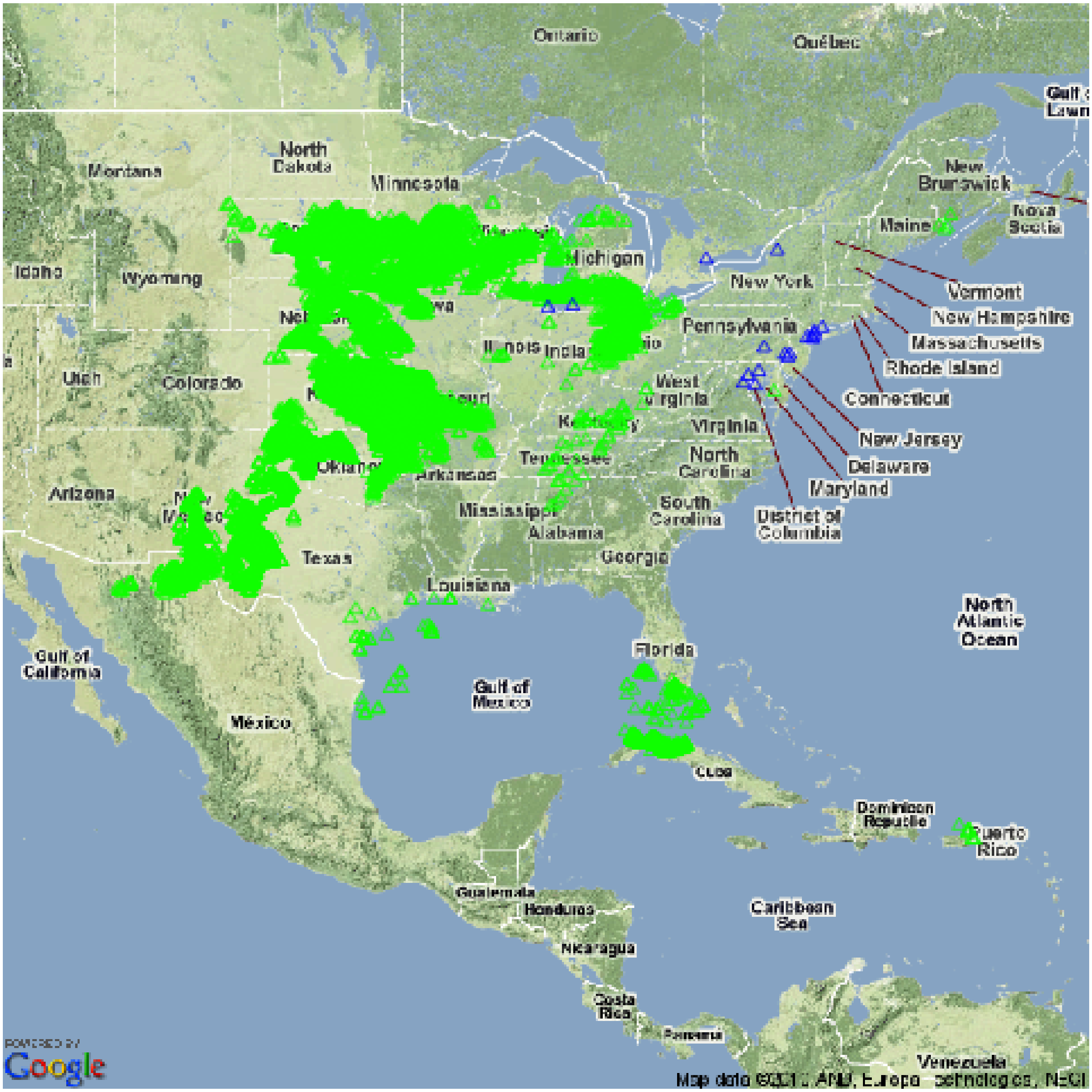

Figure 11: 24 hour twitter primitive events and *WSR-88D_Events* overlaid on the CONUS. Green triangles are *WSR-88D_Events* and blue triangles are twitter primitive events

time (i.e. what people tweeted when President Obama was elected). One source of hope is google as they have recently started recording all public tweets so that they can be made accessible through their search facilities. Location information is absent however.

Ultimately without enough geocoded tweets there is truly no way to determine how close people are to one another and in the end the methodology fails to improve on the automatic weather observations. Further investigation is warranted in order to truly invalidate the method proposed in this paper. As a practical matter the Twitter stream itself provides insufficient location information, but it might be possible to mine the location information from the tweets text or even another stream.

7 Future Work

The difficulty in accessing the data combined with what is accessible lacking critical contextual information makes the Twitter stream impractical for such purposes. Though it is still possible that the Twitter stream could be of benefit to improving local weather activity reports in the future given further work. One possible path is to try and estimate the location of twitter users as was done in [11]. By considering each Twitter user as an individual ”social” sensor, they were able to employ location estimation techniques with a great deal of success. Of course an estimated location is not as good as a precise location, but it should be possible to adjust the proposed methodology to account for the imprecision. One way to do this being to increase the number of neighbors necessary to detect a *Weather_Event*.

Another path would be to forgo the use of Twitter altogether and instead use a different social media system such as Foursquare [20]. Foursquare might prove more viable given its focus on contextual information, but it also may suffer from data accessibility issues. Even more interestingis to include other forms of social media and potentially increase the number of events with usable location information.

Finally. it would be intriguing to evaluate this method during a significant weather event. The number of tweets increase considerably during more significant events. For example during earthquakes, blizzards or hurricanes. Evaluating the methodology during these events might provide information useful in determining if better geocoded information is even a worthwhile pursuit.

References

[1] S. Systems. (2010, September) Streambase complex event processing platform. [Online]. Available: http://www.streambase.com/about-home.htm

[2] S. Garrett. (2010, June) Big goals, big game, big records. [Online]. Available: http://blog.twitter.com/2010/06/big-goals-big-game-big-records.html

[3] B. Huberman, D. Romero, and F. Wu, “Social networks that matter: Twitter under the microscope,” *First Monday*, Jan 2009.

[4] C. A. Fiebrich, “History of surface weather observations in the united states,” *Earth-Science Reviews*, Jan 2009.

[5] D. Boyd, S. Golder, and G. Lotan, "Tweet, tweet, retweet: Conversational aspects of retweeting on twitter," *2010 43rd Hawaii International Conference on System Sciences*, pp. 1–10, Jan 2010.

[6] A. Java, X. Song, T. Finin, and B. Tseng, "Why we twitter: An analysis of a microblogging community," *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, Jan 2007.

[7] M. Naaman, J. Boase, and C.-H. Lai, "Is it really about me?: message content in social awareness streams," *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, Feb 2010.

[8] B. D. Longuefille, R. S. Smith, and G. Luraschi, "Omg, from here, i can see the flames!: a use case of mining location based social networks to acquire spatio-temporal data on forest fires," *Proceedings of the 2009 International Workshop on Location Based Social Networks*, Jan 2009.

[9] G. Grosseck and C. holotescu, "Microblogging multimedia-based teaching methods best practices with cirip. eu," *Procedia-Social and Behavioral Sciences*, no. 2, pp. 2151–2155, Jan 2010.

[10] M. Ebner and M. Schiefner, "Microblogging-more than fun," *Procceding of IADIS Mobile Learning Conference 2008*, pp. 155–159, Jan 2008.

[11] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," *Proceedings of the 19th international conference on World wide web*, Jan 2010.

[12] M. Demirbas, M. A. Bayir, C. G. Akcora, Y. S. Yilmaz, and H. Ferhatosmanoglu, "Crowd-sourced sensing and collaboration using twitter," *XIn 2010 IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks"*, pp. 1–9, Jun 2010.

[13] X. Li, B. Plale, N. Vijayakumar, R. Ramachandran, and et al., "Real-time storm detection and weather forecast activation through data mining and events processing," *Earth Science Informatics*, vol. 1, no. 2, pp. 49–57, Jan 2008.

[14] O. HyvÃrinen and E. Saltikoff, "Social media as a source of meteorological observations," *Monthly Weather Review*, August 2010.

[15] T. D. Crum and R. L. Alberty, "The wsr-88d and the wsr-88d operational support facility." *Bulletin of the American Meteorological Society*, vol. 74, no. 9, pp. 1669 – 1687, Jan 1993.

[16] N. Statesman. (2010, March) Twitter registers 1,500 per cent growth in users. [Online]. Available: http://www.newstatesman.com/digital/2010/03/twitter-registered-created

[17] N. W. Service. (2010, April) Storm reports via twitter. [Online]. Available: http://www.weather.gov/stormreports/

[18] K. K. Droegemeier, V. Chandrasekar, R. Clark, D. Gannon, S. Graves, E. Joseph, M. Ramamurthy, R. Wilhelmson, K. Brewster, B. Domenico, T. Leyton, V. Morris, D. Murray, B. Plale, R. Ramachandran, D. Reed, J. Rushing, D. Weber, A. Wilson, M. Xue, and S. Yalda, "Linked environments for atmospheric discovery (lead): Architecture, technology roadmap and deployment strategy," in *1st Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology*, 2005.

[19] B. Domenico, "Unidata internet data distribution: Real-time data on the desktop," in *Science Information Systems Interoperability Conference*, 1995.

[20] Foursquare, "What is foursquare?" September 2010. [Online]. Available: http://foursquare.com/about