CS 171: Project 3

Pheme 2.0:
Birds of a Feather Tweet Together

Authors:
Carl Jackson and Peter Zhang

May 2, 2013

Contents

1 Introduction

1.1 Motivation

Most current visualizations of social data require viewers to already have a subject matter in mind; they allow viewers to search for and investigate specific keywords, hashtags, or locales. However, we think that social data is interesting primarily because it allows us to discover new topics for investigation - topics that are being generated in real time by the humans who are producing the social data. Accordingly, we want to build a visualization that guides a curious but unfocused viewer in identifying undiscovered topics/events that are currently occurring in the real world.

1.2 Goals

The two overarching research questions of our visualization for projects II and II are

1. Can we identify ”events” from real-time social data?
2. If we can, what can we learn about such ”events”?

In project II, we built a visualization that tries to identify current events that are occurring in a user-specified geography (location + radius), using real time social data. Concretely, we pull a live stream of Twitter data arriving from the specified geography, and try to identify clusters of tweets; we are currently clustering primarily by geographical proximity. In effect, we've covered question 1 from above in Project II.

In project III, we have achieved two goals. First, we've worked on improving our visualization from Project II to allow users to not only identify current events, but help them learn and experience the events that they have identified. In Pheme 2.0, we've performed some basic natural language processing - word frequency analysis - and visualized this analysis to help users identify the nature and context of the event they have identified. Additionally, we allow users to explore the data within an event temporally, seeing the frequency of tweets over time and filtering events by time range. Second, we've built a significantly more involved server-side component to our visualization, which is constantly identifying events all over the world from live tweet data. Accordingly, users now have two choices for visualizing events; they can either use the live-streaming mode to identify events in real time and then learn about them, or explore pre-recorded events in more depth using the advanced visualization techniques mentioned above in Pheme 2.0.

2 Description of Data

2.1 Source

As in Project II, the ultimate source of the data for our visualizations is Twitter, an online social network and microblogging service that allows users to instantaneously send and receive short messages, called tweets, on various electronic devices. Twitter is one of the most popular, if not the most popular, sites that produces real-time social data; as of 2012, it produced more than 340 million tweets per day [1]. Given our interest high-throughput social data streams, Twitter was a natural data source to focus on.

Using Twitter's API, we pulled tweets, which contained several relevant pieces of information:

1. Geographical location: Twitter users can choose to specify a geographical location to associate with their tweet. This is particularly interesting for users who tweet on their mobile phones, and choose to expose their phone's GPS data to the service.
2. Twitter Screenname: Tweets are associated with the screen name of the user publishing the tweets.
3. Tweet contents: The actual text content of the tweet. This is also available to viewers of our visualization.
4. Hashtags: Twitter users can specify hashtags, or metadata, to associate with their tweet.
5. Tweet timestamp: The time at which the tweet was published.

2.2 Scraping Method

Typically, Twitter's API allows you to view historical tweets, at most a few hundred at a time, using its REST API. Since we wanted to see tweets as they happened, we instead chose to rely on Twitter's streaming API, which is a mechanism by which you can express interest in a particular Twitter search query, and receive push updates from Twitter as new Tweets are written that match the query. This live stream is sometimes referred to as the "Twitter Firehose," although Twitter only makes a subsampling of the full Firehose available for third party developers.

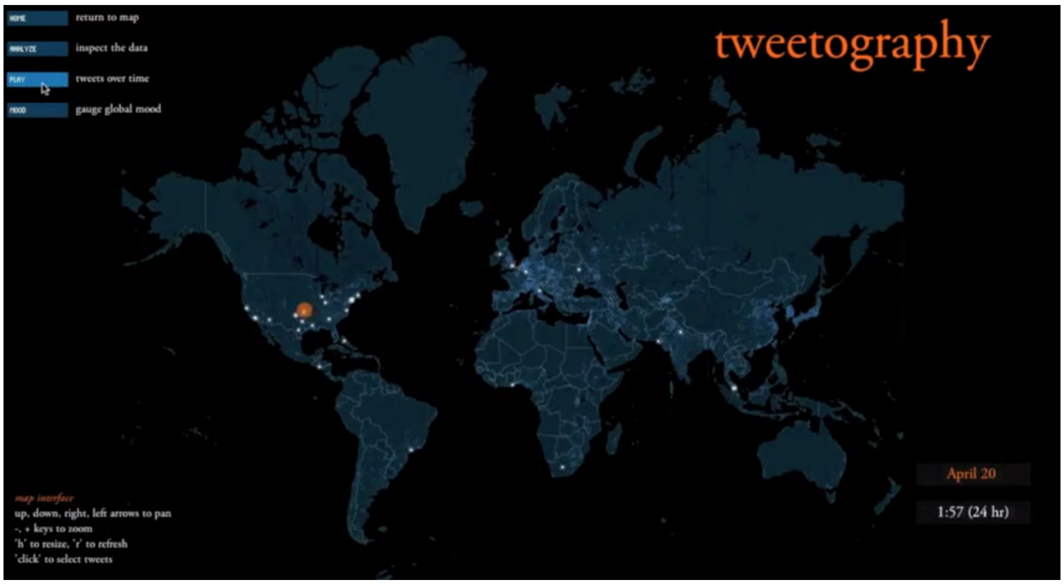
[1]http://techcrunch.com/2012/07/30/analyst-twitter-passed-500m-users-in-june-2012-140m-of-them-in-us-jakarta-biggest-tweeting-city/

Twitter's streaming API allows you to search for tweets by geographical location, so we established a connection to Twitter that requested the Tweets that would be visible on the user's screen. Each user needed to have their tweets scraped from a different live API feed, since different users can be visualizing different portions of the map at once.

One particular challenge involved with Twitter's streaming feed was that of authentication. Twitter protects their streaming API using OAuth, which is a mechanism for authenticating individual Twitter users with third party applications. In order to get access to a live stream of tweets, we must prove to Twitter that we are doing it on behalf of a valid Twitter user, and obtaining these credentials is a particularly finicky multi-step process. Only after forcing users to authenticate can we proceed with our visualization.

3 Related Work

To the extent that we've used what we've learned so far in CS 171, all the readings and lecture materials are related work; when justifying our visualization choices or visualization evolution, we will refer to this material. We consulted documentation related to the Twitter API and the Google Maps API very often while working on this project. Finally, we were inspired to investigate and study tweets and their geographical data by a past CS 171 project: Tweetography.



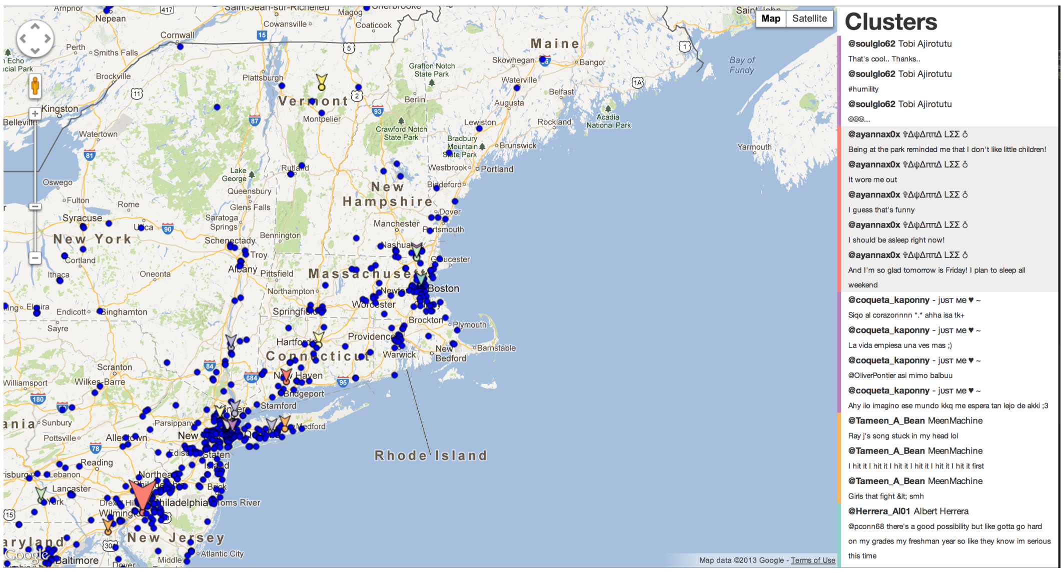
We found the concept of tweets appearing over time on a map very attractive, and based our visualization on this concept, even though the purpose and design of our visualization is very different than Tweetography. In addition to this source of inspiration for our overall visualization, there are several other sources which have guided us in specific design choices we've made (discussed in design evolution below):

1. In designing our tweet content visualization, which required some very basic natural language processing, we referred to several sources. http://www.wordle.net/ and http://www.jasondavies.com/wordcloud/ gave us guidance on how to design word clouds, although our final design was markedly different. We referred to http://norm.al/2009/04/14/list-of-english-stop-words/ in deciding how to process the text of tweets.

2. In designing the temporal aspect of our visualization - which allows users to explore the tweet data in an event by time - we made use of two convenient javascript packages, and referred to their documentation for both technical and design guidance. These two packages are http://square.github.io/crossfilter/ and http://nickqizhu.github.io/dc.js/.

4 Design Evolution

At the end of project II our visualization was basically just tweets placed on a map, a sidebar containing a list of clusters and the tweets within the clusters, and a visual representation of the clusters on the map. A screenshot of our progress at that point can be found below:



Clusters

@soulglo62 Tobi Ajirotutu
That's cool.. Thanks..
@soulglo62 Tobi Ajirotutu
#humility
@soulglo62 Tobi Ajirotutu
☺☺☺...
@ayannax0x ✞ΔψΔππΔ LΣΣ ♁
Being at the park reminded me that I don't like little children!
@ayannax0x ✞ΔψΔππΔ LΣΣ ♁
It wore me out
@ayannax0x ✞ΔψΔππΔ LΣΣ ♁
I guess that's funny
@ayannax0x ✞ΔψΔππΔ LΣΣ ♁
I should be asleep right now!
@ayannax0x ✞ΔψΔππΔ LΣΣ ♁
And I'm so glad tomorrow is Friday! I plan to sleep all weekend
@coqueta_kaponny - jᴜsᴛ ᴍᴇ ♥ ~
Siqo al corazonnnn *.* ahha isa tk+
@coqueta_kaponny - jᴜsᴛ ᴍᴇ ♥ ~
La vida empiesa una ves mas ;)
@coqueta_kaponny - jᴜsᴛ ᴍᴇ ♥ ~
@OliverPontier asi mimo balbuu
@coqueta_kaponny - jᴜsᴛ ᴍᴇ ♥ ~
Ahy iio imagino ese mundo kkq me espera tan lejo de akki ;3
@Tameen_A_Bean MeenMachine
Ray j's song stuck in my head lol
@Tameen_A_Bean MeenMachine
I hit it I hit it I hit it I hit it I hit it I hit it first
@Tameen_A_Bean MeenMachine
Girls that fight &lt; smh
@Herrera_Al01 Albert Herrera
@pconn68 there's a good possibility but like gotta go hard on my grades my freshman year so like they know im serious this time

Coming into project III, we had several specific issues we wished to address about project II, as well as several specific goals (some of which address the issues mentioned) for project III:

1. Issues:
    (a) The visualization was somewhat confusing without the process book.
    (b) Tweets arrived slowly at certain locales at certain times of day.
    (c) Some identified clusters were low quality (did not correspond to real events).
    (d) The visualization became slow and cluttered as more and more tweets arrived.
2. Goals:
    (a) Help users learn about the subject matter of an event.
    (b) Help users explore an event temporally.
    (c) Let users explore pre-recorded events.

These issues and goals presented both significant technical and visualization design challenges. We discuss the evolution of our attempts to address them in the two subsections below. We

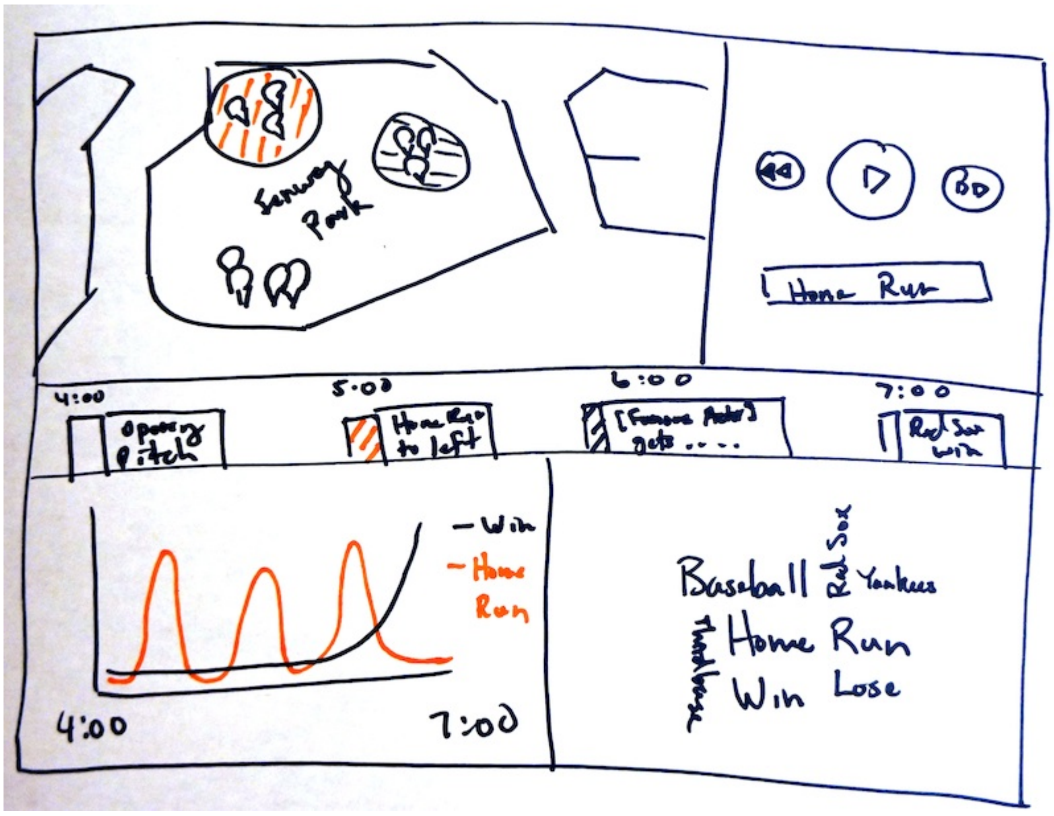
4.1 Technical Design

4.2 Visualization Evolution

4.2.1 User Flow

4.2.2 Natural Language Processing

To allow users to learn more about the subject matter of an event, we proposed a complicated visualization that would use several techniques to analyze and visualize this content:



4.2.3 Tweet Fading

4.2.4 Clustering Improvements

5 Visualizations and Discussion

5.1 Visualization

Our visualization allows the user to select a geography of interest, and to draw tweets on the map in real time. By default, our visualization is currently focused on Boston; however, users can zoom and drag the map as desired to grab tweets from any particular region in the world. As data arrives, we also dynamically detect clusters of interesting points in the current viewport/geography, and visually group them by enclosing them in a circle and coloring them similarly. In addition, we display a list of all the identified clusters, and the component tweets in each cluster, in a secondary view - a sidebar to the the right of the map. Our visualization also supports several user interactions, as described below:

1. Linking & Brushing. Our visualization has two views, the map - which displays the geographical location of tweets and clusters - and the cluster sidebar - which contains a list of all clusters currently identified and the member tweets of each cluster. We support linking & brushing by interactively highlighting the cluster marker on the map (by making it larger) when ther viewer hovers over the cluster in the cluster sidebar.

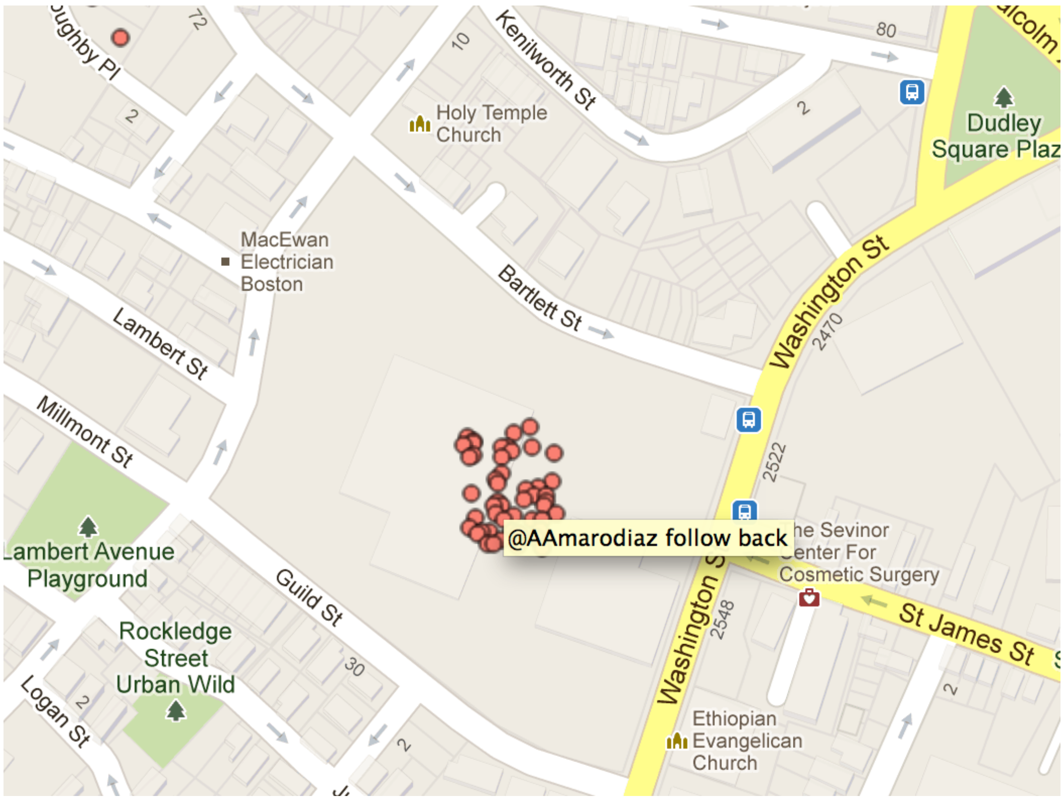
2. Details-on-Demand. Our visualization supports details-on-demand with a tool tip. Whenever a user hovers over an individual tweet on the map, regardless of whether or not it is in a cluster, a tool tip will appear after a short delay showing the text contained in the tweet.

3. Drill-Down/Filter Capability. Our visualization suuports a drill-down or filtering interaction by allowing users to drag the map viewport and zoom in or out of the viewport as desired. In this way, the user can exactly specify what geographical area they wish to pull tweets and identify clusters in; the visualization responds by drilling-down/filtering tweets to match this specification.

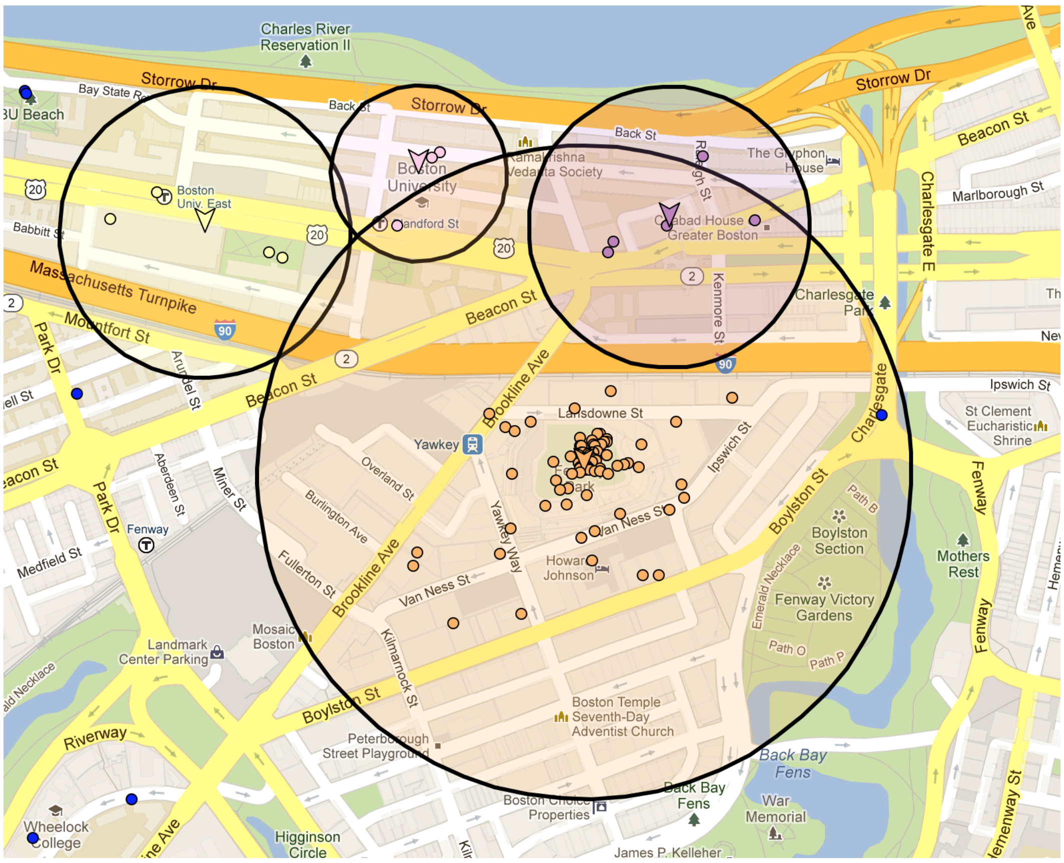
5.2 Discussion

The primary goal of our visualization was to identify events in the world as they were happening. At first, because our live twitter data stream only subsampled the true twitter "firehose", we were concerned that we were not going to be able to find any clusters at certain zooms. Fortunately, in practice, our visualization identified several clusters with only a few minutes of streaming at a wide variety of zooms, from continent-level zoom (all of Africa), to country-level zoom (all of the US), to city-level zoom (Boston), and others.

However, it is not immediately clear that the clusters we found correspond to real-world events. For example, some of the events that we found look something like this:



This cluster is just one lonely soul, tweeting from the same location, trying to convince other people to follow him on twitter. We certainly wouldn't classify this as an event. However, a significant number of our clusters did end up corresponding to real-world events; we found a baseball game at Fenway:



The fact that some of our clusters are not real events suggests that we can further improve our current visualization technique (discussed in the conclusion below). However, we did ultimately succeed in independently discovering events in the world that we were not aware of. We take this as a proof of concept; it is definitely possible to take a high-throughput social data streaming source, and identify events that are occurring in the real world.

6 Conclusion

In Project 2, we implemented Pheme, a real-time social data visualization. Pheme demostrates that it is possible to identify real events by examining the instantaneous social data generated by people participating in the events. Furthermore, Pheme allows users to experience and visualize events all over the world, through the social data these events generate, on their home computer, in real time.

6.1 Future Work

In Project 3, we hope to improve on Pheme in several ways:

1. Refine our clustering algorithm to ignore certain types of non-events, including one individual tweeting rapidly.

2. Extend our clustering algorithm to identify events using other closeness metrics, rather than just geographical closeness. We might consider tweets that we identify to be related through natural language processing, some metric related to time, etc.

3. Aggregate information about identified events, such as trying to determine what the event is about by analysing the constituent tweets, and pulling in data from that locale using other social data sources, like Instagram.

4. Implement reclustering. By doing so, we could allow users to drill down into one coarse event that they had discovered, and possibly find subevents within that event. For example, at a heavy-metal concert, one might discover a subevent in the mosh pit, and another behind the stage.