5-2014

Tweet Mapper Visualization Software

Andrew W. Nash
University of Tennessee - Knoxville, anash4@utk.edu

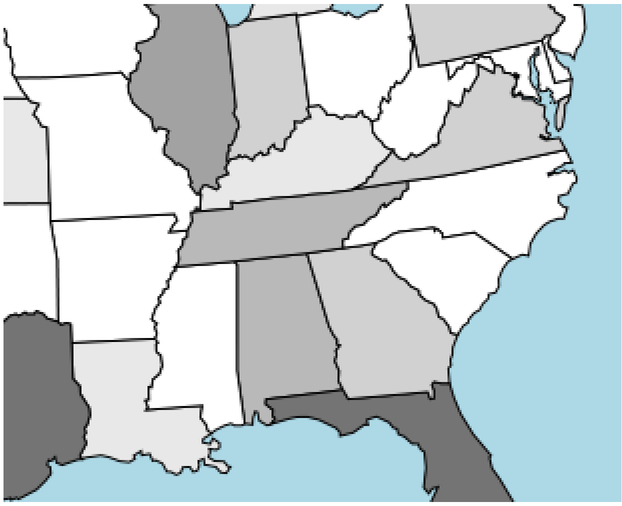
Shawn M. Cox
University of Tennessee - Knoxville, scox31@utk.edu

Michael T. Adams
University of Tennessee - Knoxville, madams44@utk.edu

Recommended Citation

Nash, Andrew W.; Cox, Shawn M.; and Adams, Michael T., "Tweet Mapper Visualization Software" (2014). *University of Tennessee Honors Thesis Projects.*
https://trace.tennessee.edu/utk_chanhonoproj/1675

Tweet Mapper
Visualization Software



Project Report for COSC 400 Senior Design
3 December 2013

Project Team

Michael Adams	BS Computer Science 2014 *(anticipated)* http://cs400weblog.blogspot.com	University of Tennessee madams44@eecs.utk.edu
Shawn Cox	BS Computer Science 2014 *(anticipated)* http://shawncoxutk.blogspot.com	University of Tennessee scox31@eecs.utk.edu
Drew Nash	BS Computer Science 2014 *(anticipated)* http://web.eecs.utk.edu/~anash4	University of Tennessee anash4@eecs.utk.edu

Project Mentor

Dr. Michael W. Berry	EECS Professor and CISML Director http://web.eecs.utk.edu/~berry	University of Tennessee berry@eecs.utk.edu

This documentation describes *Tweet Mapper*, a software program designed and built by Adams, Cox, and Nash to visualize the locations of millions of tweets sent from within the United States. These tweets are visualized by state, with shading applied so that regions with more tweets are darker than regions with less tweets sent in a given amount of time.

1. [author: Nash] How can this software program satisfy a user's need?

When Twitter.com launched in 2006, it opened a new world of communication by allowing anyone to broadcast their message to the world, as long as that message was 140 characters or less. Twitter, known by some as the text messaging service of the Internet, is a social media platform that allows users to share messages, photos, and locations. Users send their media, or *tweets*, to Twitter, which, in turn, shares the tweet with that user's "followers." Users can opt to make their profiles public or private, with the latter option requiring that followers be approved by the user. Tweets from users with public profiles are featured in the main Twitter public feed, which anyone with Internet access can view. Today, Twitter has over 500 million registered users that collectively send billions of tweets every day. The content of these tweets yields a vast amount of data, and since it is written and much of it is public, the tweets can be analyzed and used to gauge general public interest. There are numerous opportunities that Twitter offers to researchers, marketers, or anyone curious about the current state of the world. However, harvesting information for the benefit of a person or organization is a difficult task.

The following hypothetical situation describes how Tweet Mapper can be used in the medical community.

Summary of the Project Story Idea: *A medical researcher needs to determine whether the number of posts about certain diseases and infections on social media has any correlation with the actual number of diagnoses.*

Example Story: Dr. K's Analysis of Influenza Regions
Dr. K works as a researcher at an esteemed medical school. Her specialty is the ability to analyze vast amounts of data, such as the number of diagnoses of certain illnesses by geographic region, and to submit her analysis to researchers at the Centers for Disease Control, where they will review the findings in order to archive the information for historical purposes and use it as a reference for predicting future diagnoses and outbreaks. She has been assigned the task of analyzing the outbreak of influenza in the United States in 2012. She has information from the Centers for Disease Control, including the number of diagnoses by geographic region.

As Dr. K works on her analysis, she thinks about social media. Today, she realizes, people share many of their thoughts and opinions on Twitter, so she expects that they would most likely share when they are sick. Dr. K does some quick research and realizes that certain keywords, such as *flu*, *influenza*, and *vomit*, appear in hundreds of tweets each day. She meets with her colleagues, and after some brainstorming and discussion, they agree to harvest the tweets about influenza that were sent in 2012. They decide that their ultimate goal is to see if there is any correlation between the locations from where most of the tweets originate and where most of the diagnoses occurred each day of the flu season. Dr. K and her team work with some consultants to harvest flu-related tweets between July 2011 and December 2012. Their work leads to a database of approximately 47 million tweets, which is certainly an overwhelming amount. Dr. K has no idea how to compare the locations of millions of tweets with the locations of diagnoses in a practical, quick, and concise manner.

Dr. K consults with a leading researcher in the field of data mining, Dr. B, who suggests that Dr. K plot the locations of the tweets on a map in order to depict them in a manner that is easy to perceive. Dr. B states that the best way to do this is to load the tweets in a database, and to utilize a software program that is capable of querying the database and generating images depicting the locations of the tweets that are returned. An ideal map would shade the country according to the number of influenza-related tweets in certain regions, with regions containing a large number of tweets showing up darker than regions with less tweets. For example, if most of the tweets related to the flu on a given day were sent from New England, then the New England region would be shaded darker than the Midwest. Ideally, in order for Dr. K to be able to use the data accordingly, it would be necessary that the program allow her to customize her view. She should be able to examine any date with tweets available in the database. She should also be able to use a control, perhaps in the form of a slider, that would allow her to iterate through several dates. This would allow Dr. K to see the regions of greatest activity over time, in a manner similar to the way weather reporters use radar to depict precipitation in a region over time. The software should smooth the depictions in order to make the generated images easy for Dr. K to decipher.

If a software program was capable of depicting the locations of millions of tweets, the opportunities for Dr. K would be endless. She would be able to generate maps of the United States showing the regions of heaviest social media traffic in a certain period of time. This would allow for an easy comparison of diagnoses over the same time period, to see if the regions with the greatest social media traffic are the same regions with the greatest number of doctor visits. After generating these maps, Dr. K and her team at the Centers for Disease Control could thoroughly analyze the data and publish a paper documenting whether there is a likely correlation between the number of tweets about the flu being sent and the number of influenza diagnoses within a specific region.

Dr. K thinks that this software would be very useful, and there are several websites and software programs capable of plotting tweets on a map. However, none of them are actually capable of handling such a large amount of data. Millions of tweets take up several gigabytes of storage space. Furthermore, they need to be stored in a database in order to be usable. As a result, if the program relies on a database, it would need to run on a reliable server-grade system. The costs of a server and the potential need for help using such a powerful program can be prohibitive. Dr. K has a very small budget, after all, she works in academia, and she needs to generate some significant data before she can apply for grants. A powerful, yet efficient and inexpensive software program is needed.

Enter Tweet Mapper, an inexpensive software program capable of generating maps and visualizations of millions of tweets in a large database. Dr. K uses the Tweet Mapper program to easily generate the maps that she needs for comparison with data from the CDC. After she installs and uses the program, it takes her less than a week to report her social media analysis to her colleagues at the Centers for Disease Control, leading to more grant funding for Dr. K's research! Furthermore, because of Tweet Mapper's intuitive interface and ability to analyze large amounts of data, Dr. K shares the program with other leading researchers that utilize social media, leading to the expansion of the program and its ultimate success.

2. [authors: Cox, Nash] Market Analysis to Prove Why an Audience Needs Tweet Mapper

Since Twitter.com launched in 2006, the social media platform has seen unprecedented growth. In November 2013, the company debuted its initial public offering at $26 per share. The company trades its shares on the New York Stock Exchange using the symbol *TWTR*. As of Tuesday, 3 December 2013, TWTR shares were selling for over $40. As the media was gearing up for this initial public offering, research was done in order to examine just how much power Twitter held in sheer numbers. There are several interesting facts about Twitter that were just released.

There are over 200 million active Twitter users around the world. These users actively send around 500 million tweets during a 24-hour period. When examining this wholistically, it is easy to note that the amount of data that passes through Twitter's servers and is available on the website is incredible. The number of Twitter users continues to grow every day as social media becomes more widespread throughout the world. It is also interesting to note that sixty percent of these people use the mobile service. As a result, many people are tweeting several times a day, expressing their thoughts and actions in real time. This is also important from a data mining standpoint, since it is apparent that the data available on Twitter is fresh and current. Last, it is known that the average user spends 170 minutes on the website each month. This is significant, because if people spend this much time on one website over the course of a month, then the opportunities for effective advertising are vast. Furthermore, as of 3 December 2013, Twitter is ranked tenth globally for website traffic on Alexa.com. Therefore, the popularity of Twitter makes it a repository of extremely useful data. [Curtis, Sophia. "Twitter IPO: 14 fun facts." *The Telegraph*, Nov. 2013. Web. 3 Dec. 2013.]

With the popularity and massive amount of data available on Twitter, Tweet Mapper is a very practical software program that can be used to easily visualize where people are tweeting about certain topics. There are many groups and industries that can benefit from the use of Tweet Mapper.

Healthcare

Tweet Mapper was specifically developed for a medical researcher in order to visualize the locations of tweets about influenza sent in 2012. The ultimate goal of the researcher's project is to determine if there is a correlation between the locations of influenza-related tweets and the actual number of diagnoses by region within the United States. Since the idea for this software program came from this researcher, there are some features that were implemented to satisfy a specific need that may not be applicable to most users. Furthermore, this researcher will not be purchasing the program; it will be provided at no cost as part of the project. However, Tweet Mapper definitely has potential in other areas of the healthcare industry. Given how much money most hospitals and medical practices make each year, this industry would be an excellent source of buyers and users of Tweet Mapper. In 2009, a group of researchers examined the number of search queries about the flu and compared this information with the actual number of diagnoses. As a result, there is definitely a niche market consisting of medical researchers that would be interested in Tweet Mapper. [Ginsberg, Jeremy. “Detecting Influenza Epidemics Using Search Engine Query Data.” *Nature* 457 (2008): 1012-1014. Print.]

Law Enforcement

The Boston Marathon bombing was a tragic event that will never be forgotten within the United States. However, it did emphasize the importance of social media in the field of law enforcement and criminal justice. The Boston Police Department’s information officer specifically used Twitter to inform the public about the latest, most accurate information during that time of crisis. Another way that Twitter can be used is to determine where people are tweeting about crime within the United States. This would allow law enforcement agencies to examine the country as a whole and compare regions to identify causes of crime. [Bensinger, Ken. “Boston Bombing: Social Media Spirals out of Control.” *LA Times*, Apr. 2013. Web. 3 Dec. 2013.]

Politics

Social media is a crucial part of any political campaign today. Barack Obama has over 42 million followers on Twitter. Furthermore, during his reelection campaign, he had a team that worked around the clock monitoring social media. This team connected with potential voters and responded to any signs of trouble. Tweet Mapper would be an excellent tool for a political campaign that needs to be prepared. By using this program, the social media team could quickly determine where no one is tweeting about a given political candidate and immediately direct their advertising to that region. This would certainly allow for the use of limited political resources quite effectively. [Rutenberg, Jim. “Data You Can Believe In.” *New York Times*, 20 Jun. 2013. Web. 3 Dec. 2013.]

Marketing

Marketers and advertisers work hard to ensure that their product or service stays in the spotlight. Tweet Mapper would allow these professionals to easily determine where they should target their future advertising in the United States. For example, if the marketer for a specific laundry detergent used Tweet Mapper and determined that no one in New England tweets about the detergent, he could organize a marketing campaign centered around New England. This would allow for extremely effective, targeted advertising.

Insurance

Insurance companies are always trying to have the upper hand when it comes to predicting the future. They spend millions of dollars each year on research in order to minimize loss. With Tweet Mapper, insurance companies could monitor for signs of a natural disaster and could see how far away the damage occurred by examining tweets about the event. Furthermore, using Tweet Mapper could potentially allow insurance companies to stay in contact with policyholders, which is the key to exceptional customer satisfaction.

At this time, Tweet Mapper is being designed and developed to satisfy the specific needs of a healthcare researcher. As a result, Tweet Mapper already had a niche market during the development process. Since there is no timeline on when this software program will be ready to sell to the general public, it would not be ideal to spend more than $1000 during the development process. There are no expenditures with the development of the program at this stage; all libraries and tools used to generate the program are open source and available to the developers at no cost.

A working prototype has been developed without the need for funds. However, if money was needed for further research and development, it would be possible to apply for grants from the National Institute of Health. However, the software has the potential to be converted to satisfy different needs. Therefore, it could be sold in the future to marketers, political campaign teams, etc. [Stein, Sam. "NIH Losing $1.7 Billion, 700 Research Grants Due to Sequestration." *Huffington Post*, 4 June 2013. Web. 3 Dec. 2013.]

Once this software program gets to the point that it can be marketed to various users outside of the medical community, then a Kickstarter campaign could be launched. A sample Kickstarter campaign for Tweet Mapper was created with two tiers of support. For $10, a donor would receive a link to download the most recent version of the program. For $100, the donor would receive the link and a one-hour tutorial session with assistance in setting up the program for first-time use. The crowdfunding page did not go live; this is just an idea of how the developers would have raised funds, if needed.

3. [authors: Adams, Nash] Similar work; measurement of similar products

There are two products currently available that allow users to visualize the locations of tweets on a map. The first program, which is the quite popular, will be referred to as *Program X*. The second program will be called *Program Y*. These programs were compared to Tweet Mapper using various measures.

Three significant measures to compare products

The first measure is the time period of tweets that the program can visualize. Tweet Mapper can visualize tweets from any time period, as long as the tweets were harvested in advance. For example, the current version is being used at this time by the developers to visualize the locations of tweets sent in 2012. Program X can only visualize seven days of tweets. This cannot be an arbitrary week that the user chooses; it only can visualize the past seven days from the current day. Program Y does not even allow users to select a time period; users can only visualize live tweets. Therefore, Tweet Mapper is the only program of these three that gives the user full flexibility in selecting a time period.

The second measure is the number of keywords that the user can select when wanting to customize the visualization. Tweet Mapper allows the user to filter based on an unlimited number of keywords and is fully customizable. For example, if a user wants to only see the flu-related tweets containing the words “vomit” or “influenza” on a map, the user has the option to do that. Program X only allows a user to visualize current trends on Twitter. There is very limited customization in allowing users to filter based on certain keywords. Program Y only allows the user to filter tweets with one keyword at a time, which is certainly inconvenient and rather limiting.

The third measure is the cost associated to run the program and the flexibility given to the user. Program X costs $19 per month, or $190 for a year-long subscription, which is rather pricey, given its limited functionality. Program Y is free, but it must run in a user’s web browser, and the site sometimes crashes, which can be quite frustrating. Tweet Mapper is available for a fixed cost; if the Kickstarter campaign goes live, then that cost will be $10. Furthermore, it can be installed on almost any personal computer, giving the user the flexibility to utilize and customize the program many different ways.

Comparison 1: Tweet Mapper vs. Program X

Tweet Mapper

- Capable of depicting a database with millions of tweets on a map
- Can display harvested tweets from any point in time stored in a database
- Keywords fully customizable by user
- Fixed, one-time cost for software, to be determined

Program X

- Capable of displaying current trends on Twitter on a map
- Can display trends for the past seven days on Twitter only; no database input
- Keywords generated based on current trends
- Cost for Plus version is $19/month or $190/year

Comparison 2: Tweet Mapper vs. Program Y

Tweet Mapper

- Uses a map of the United States that is *draggable* and *zoomable*
- Shades states according to the number of tweets from that location
- Allows the user to *scroll* through time and *play* through the time periods
- Uses multiple keywords

Program Y

- Uses a Google map that is fully interactive and can be dragged and zoomed
- Displays locations as dots/pins, which accumulate in mass numbers
- Shows a map that depicts the current state of Twitter using the live feed
- Uses a single keyword

There is currently is no program widely available that does exactly what Tweet Mapper was designed to do: To display an extremely large number of tweets on a map for data analysis purposes. Most mapping programs are for personal use or marketing purposes. Therefore, a niche market certainly exists for this program.

Two patents related to Tweet Mapper
The first patent is about "Interactive data visualization and manipulation" (US 20130275904 A1) and was issued to Secondprism, Inc. This patent protects the inventors' ideas regarding the interaction with visualizations of data. The inventors developed a way that users could quickly comprehend data by allowing them to generate and manipulate graphs through touch. It seems as though in order to utilize this patent, a tablet computer is needed, and the graphs generated by a software program should be easily manipulated through touch. This is a great idea, and should Tweet Mapper be utilized by other industries, especially in marketing, some of the features listed in this patent may need to be licensed and implemented into the program. However, at this time, Tweet Mapper does not infringe upon this patent since there is limited interaction between the user and graphs once they have been generated.

The second patent is about "Automated social networking graph mining and visualization" (EP 2569715 A2) and was issued to Microsoft. This patent protects the inventors' ideas regarding an algorithm that generates graphs based on social media connections. This research area of graph theory is quite popular today because, by generating social media graphs, data miners can learn many different things about how people interact and can easily decipher large amounts of data. In the future, if Tweet Mapper becomes widely used and there is a need, this patent may need to be licensed, and the graph-generating algorithm may need to be implemented into the program. By generating graphs between Twitter users, this program could allow researchers to quickly see if those tweeting about the flu in a certain time period actually know each other and visualize the spread of influenza through social media. Furthermore, this technology could also be used in marketing, allowing marketers to reach a targeted audience through their acquaintances and friends. At this time, though, Tweet Mapper does not have the capability to generate graphs of social media connections and therefore does not infringe on this patent.

Since Tweet Mapper relies on a database to plot tweets on a map, there is no need to apply for a patent. The algorithm that Tweet Mapper uses is rather straightforward. However, if the program is developed so that it can analyze data on its own, rather than just generating maps and graphs for visualization purposes, then a law firm will need to be retained in order to start the patent application process and ensure that no pending or current patents are being infringed.

Three repositories related to Tweet Mapper
The first repository contains a software program known as *Twitter Ambrose*, which is capable of visualizing and monitoring data workflows. This program utilizes *d3.js*, which is also used in Tweet Mapper. This program is similar to Tweet Mapper, but it does not visualize millions of tweets. Rather, this program visualizes job statuses. It seems as though this program was solely created to satisfy a need that the developer had. It is not flexible, nor does it have a niche market, like Tweet Mapper.

The second repository contains software called SickWeather that allows the user to analyze data regarding illnesses. However, this program does not allow for the analysis of tweets; it just receives raw data as input. This program certainly is not a competitor of Tweet Mapper. It seems as though this program could be used by researchers in conjunction with Tweet Mapper for the purpose of fully analyzing illnesses and diagnoses within a certain geographic region.

The third repository contains a program called ACS Twitter Visualization that allows the user to generate a “tag cloud,” which is also known as a “word cloud,” of a passed set of tweets. This program essentially counts the frequency of words in the data set and generates an image consisting of these words. The larger a given word in the image, the more times it appeared in the data set. This program works well and could possibly be implemented into Tweet Mapper in the future, allowing a user to generate a tag cloud based off the content of millions of tweets, instead of a small dataset of tweets.

4. [author: Nash] Measurement of Prototypes and Final Product

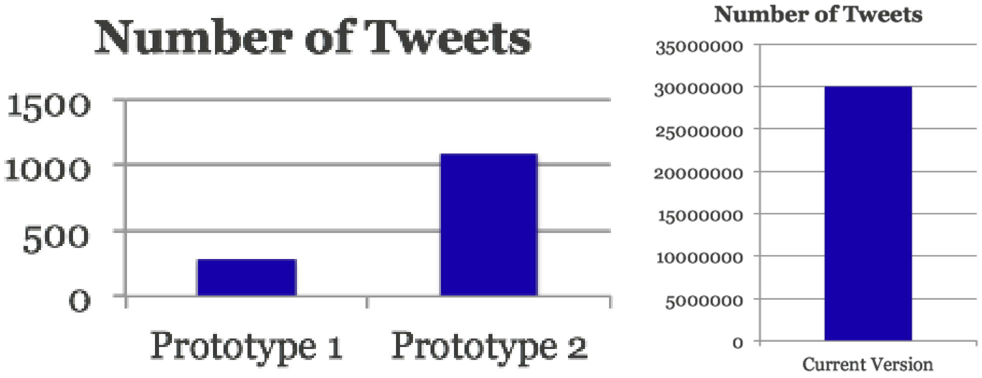
Tweet Mapper was not born overnight. It has been dramatically improved over the past few months. There are three different characteristics of the program that were observed and measured throughout the development process. The first involved the size of the dataset that each prototype was capable of handling. In other words, the number of tweets taken as input that each prototype could plot on a map was measured. The second measure was the amount of data that could be depicted on a map without being too hard to decipher and confusing to the user. There is a point where, when tweets are plotted on a map, if the number gets to be too large, it is too difficult to discern where the tweets are actually located. The third measure involved the number of steps and amount of time it takes the user to iterate through time. The more difficult it is to change the time period, the longer it takes the user to visualize any date, making the user much more inefficient.

Measurement 1: Amount of data the prototype could visualize

The first working prototype of Tweet Mapper was capable of plotting 275 tweets on a map of the United States. While this may seem like a large number, considering the fact that hundreds of millions of tweets are sent on a daily basis, it is a very small amount that is not large enough to be used for scientific research purposes. These tweets were passed to the program in the form of a text file, so there was not much behind-the-scenes processing in this program.

The second significantly updated prototype was able to visualize 1093 tweets on a map, which is a large increase in the amount of data, when compared to the first working prototype. This program also received the tweets as input in the form of a text file. The resulting maps plotting these tweets were not particularly useful to the user, since the maps did not reveal scientifically significant data that a researcher could use.

The third and current prototype is able to visualize about 30 million tweets, which is an exponential increase in the amount of data available for the second prototype. These tweets are stored in a database, and the program queries the database each time a command to visualize a certain data set on a map is given by the user. Figures 4.1 and 4.2 depict the number of tweets that each prototype is capable of visualizing on a map. Unfortunately, these graphs could not be combined into a single figure since there is too great of a difference between the numbers.



Figures 4.1 and 4.2 - Number of tweets each prototype is capable of visualizing

Unfortunately, since Program X and Program Y mentioned in Section 3 are not capable of receiving tweets as input, there is no way to use this measure to compare Tweet Mapper to these competing products. However, it is evident that the prototypes significantly improved over the course of the semester.

Measurement 2: Amount of data plotted on a map before it is undecipherable
An early prototype, shown in Figure 4.3, used dots to mark tweet locations, which became undecipherable when grouped in large quantities. For example, once more than five dots were grouped together, it was difficult for users to discern their significance since the area becomes a single, large white spot. Considering that five dots are hard to discern, imagine what would happen if all 30 million tweets are simultaneously plotted on a map of the United States in the form of dots. The entire country could be shaded white, preventing a user from determining which regions actually contain the most tweets. This is a problem because two dots overlaid on each other cannot be distinguished by a typical user. Therefore, the first prototype actually had a limit to the number of tweets that could be plotted on a map that was realistically comprehensible by a user. The second prototype used dots with a different color map. However, the change in colors did not do anything to assist the user in deciphering the data. A better solution was needed.

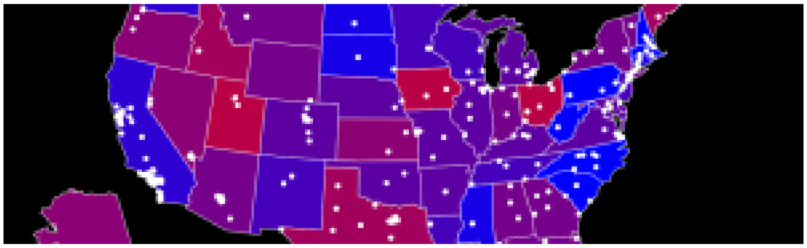


Figure 4.3 - Visualization of tweets using the first Tweet Mapper prototype

With this feedback, it was decided that it would be more effective to use shading by state so that it can be easier for a user to decipher. Darker states have more tweets about the given topic than lighter states, allowing a user to quickly determine which regions have the most activity with hundreds of tweets. Figure 4.4 shows a section of the map generated by the most recent version of Tweet Mapper, which uses shading.

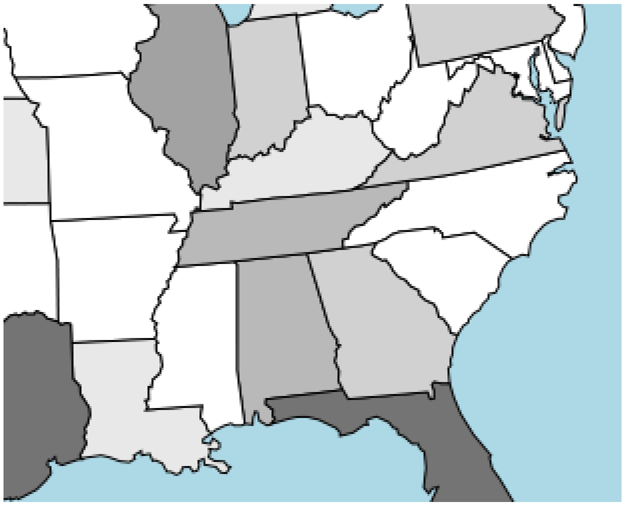


Figure 4.4 - Visualization of tweets using the current version of Tweet Mapper. Darker states have more tweets.

As a result of the decision to use shading instead of dots, the maps generated by the current version are much easier to understand than the maps generated by the first and second prototypes. Furthermore, the map generated by the most recent version has the ability to represent millions, instead of hundreds, of tweets without the whole country being overlaid by a large cluster of dots.

This measurement is a great way to demonstrate how the program has improved throughout the development process. Furthermore, it also demonstrates how Tweet Mapper is better than the competition. Programs X and Y both use points, in the form of a Google pin, to represent a tweet. When many of these points are grouped together, it becomes extremely difficult to compare regions. For example, these competing programs will usually place a lot of points in California and New York, which are typically the two states with the greatest number of tweets. With the large amount of points in these two states, it becomes very hard to determine which state actually has more tweets. With Tweet Mapper, the shades of the two states are different, allowing the user to quickly and easily determine which state has the most tweets.

Measurement 3: The process required to iterate through time

In the first prototype of Tweet Mapper, users had to manually key in a date range in order to change the time frame. This was a very cumbersome process that was time consuming. Furthermore, it only allowed a user to view a date range as a single image. This was not very helpful for research purposes. Because of these limitations, it was decided that a scroll bar would be added to the bottom of the map in the second prototype, allowing users to “scroll” through time by dragging the control through the bar. This was helpful, because it allowed users to visualize tweets day by day through a predefined date range, rather than just seeing all tweets on one map image. However, after some initial user testing, it was determined that users had trouble locating the scroll bar and understanding how it worked. Furthermore, some users requested a “play button” that automatically scrolled the map through a time period so that it would look like a time loop, similar to the way a weather radar can play through a period of time for a meteorologist who is tracking precipitation.

As a result of these tests, problems, and requests, a noticeable scroll bar was added to the bottom of the map in the current version of Tweet Mapper. This scroll bar is depicted in Figure 4.5. The black square is the control that users can drag left and right through the blue bar in order to easily iterate through time. The date inside the green box to the right is the current date being depicted on the map. Furthermore, the red square on the left is the play button. The square is red when the image is paused and is green while it automatically scrolls through time. When the user clicks the play button, the map changes to represent each day through the pre-defined date range. Smoothing is used so that the images are easy for users to comprehend.

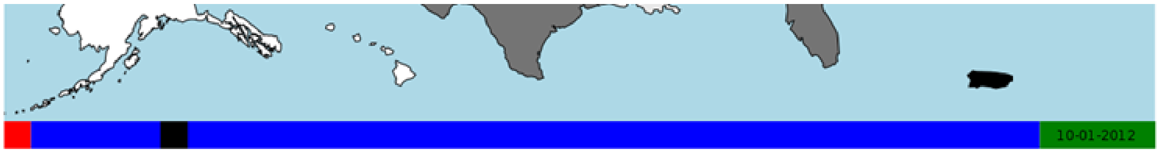


Figure 4.5 - Bottom of map in current version of Tweet Mapper, depicting the scroll bar

Since the scroll bar is easy to find and allows the user to iterate through time in one mouse click, instead of having to enter a date every time the users want to see a different time, the current version of Tweet Mapper is much more effective and efficient when compared to the previous two versions. Unfortunately, Programs X and Y do not allow a user to iterate through time in the same way that Tweet Mapper works, so this measurement cannot be used to compare this software program with the competition.

In order to improve Tweet Mapper, comments were open on the project website to allow for feedback from potential users during the development process. Furthermore, crowdsourcing has potential if a user wants to modify this program for international use. Anyone interesting in collaborating with this project is welcome to contact the developers.

5. [author: Adams] Design Decisions

According to Dr. Fred Brooks, there were three lessons to be learned after he was on a team that designed what he argues is the worst programming language ever developed. They can be read on Page 173 of *The Design of Design.* [Brooks, Fred. *The Design of Design*. Boston:

Pearson, 2010. 173. Print.]

Decision 1: JavaScript over Java: Processing vs. d3.js
Although the first prototype was made in *Processing*, *d3.js* was used instead for several reasons. The developers initially chose not to use D3 out of concern for time efficiency. JavaScript, since it runs in the browser and is interpreted, can be somewhat slow when compared to a language that is closer to the hardware, like Java. However, when Processing was used to build the first prototype, it was determined that it was not ideal for Tweet Mapper, and JavaScript with D3 was instead used as the platform.

Processing, a visualization library implemented in Java, has benefits. It is fast, easy to understand, and it has a short startup time. However, its end result is a Java Applet, which, arguably, is unpopular and considered by some as insecure. This means that, while it could (and, without undue effort, must) be displayed in a web browser on a computer with Java installed, the user must have a recent version of Java already installed and expressly allow permission for the applet to run. The benefit of developing in Processing was offset by this extra work and the system requirements for the user.

The software program should ideally be easily accessible, and so the web browser was a preferable environment. Processing, while easily configurable into an applet, is more difficult to make into a standalone program. At this point, it was decided that d3.js would be used.

This led to a radical reconstruction of Tweet Mapper as a whole and, ultimately, a fracturing of it. Before, the applet made in Processing would query a database for the data to be displayed, receive it raw, and generate the structures that would make displaying it easier. When the program was converted to a JavaScript alternative, that was no longer a viable option, which turned out to be beneficial. The decision to use D3 was made during a group meeting with all three developers and the project mentor. This decision corresponds with Brooks's Lesson 1, where the developers to learn from a failure. [Brooks, Fred. *The Design of Design*. Boston: Pearson, 2010. 173. Print.]

Decision 2: Dedicated Java server
A Java server that took requests from the JavaScript code, turned that into actual mySQL calls, received the raw data, converted that into a form more easily displayed, and packaged and sent that back was implemented. By doing this, the user does not need a very powerful computer, since most of the work is done by the server, which can be offsite. Moreover, the user can specify the server to which they would like to send a request, allowing her to choose which dataset she would like to visualize using an arbitrary configuration of options.

Since the JavaScript code is more minimal with much of the logic offloaded onto the Java server, it takes less time to load than if the user was to download an equivalent Processing applet every time she visited the page. Moreover, if the goal was to hide the implementation of the visualization, this would be an ideal first step. The only thing the d3.js code does is display the data, but the more interesting generation of that data is done via server code that the user cannot directly access. If this was a Processing applet, in which the server would have initially been embedded, a user could disassemble that code, since applets must be downloaded onto the user's machine before they can be run.

This decision came rather late in the project and only after PHP code was used, which turned out to be far too slow. This decision was also made during a group meeting with all three developers and the project mentor. This decision was important and allowed the developers to understand Brooks's Lesson 2. [Brooks, Fred. *The Design of Design*. Boston: Pearson, 2010. 173. Print.] While it was a success getting this program to work with a server, it is important for the developers to not become overly confident, in case the server crashes or has significant problems.

Decision 3: How to visualize with Dots vs. Shading
Initially, the idea was to have dots representing groups of tweets. While the exact (*x,y*) coordinates for most of the tweets were not available, a large portion had city and state specified, which could then be translated into close (*x,y*) coordinates if the location of that city was known. Since this could result in millions of individual dots spread out and placed on top of one another, the development of a clustering algorithm to gather them into a centroid with a radius somehow representative of the number of dots within a certain region was planned.

This had a couple of problems. For one, if there was more than one keyword and each dot's color represented which keyword that tweet matched, how would clustering work? One idea was to turn the representative centroid dot into a pie chart. However, after meeting with the project mentor, this idea was nixed since, in the fields of data mining and visualization, pie charts are not commonly used because they do not convey detailed information. They are useful for determining which keyword has a higher proportion but not for conveying the exact frequency of a keyword. Another problem was the processing power required for this. Visualizing 30 million tweets is very slow, if not outright improbable, to render in a browser, and even just manipulating that much data in Java would take a considerable amount of time.

In light of this, the developers decided to forgo city coordinates altogether and try to match tweets to states. Moreover, instead of dots representing the frequency, the shading of each state did that. Darker states have more tweets and lighter states have less. Using this method resulted in a visualization that conveyed more information quite effectively.

Even then, there were several questions. How should the shading be done? That is, when should a state be 100% black? Should it be dark on the day that state had the most tweets or on the day when a state had the most tweets for any state on any day? Or, perhaps, should the population of that state be considered, or, more directly, the population of that state's tweeting populace? Also, is state level clear enough?

It was determined that scaling each state's tweets by the largest number of tweets seen on any day for any state was the best option. Scaling by the population of a state imposes an unfair bias toward states with a large relative non-tweeting population. The same goes for states with a large relative tweeting population. However, this imposes a problem of its own. Namely, users only get the relative frequency of states to other states, not the overall frequency of the keywords in question, to each other or to other keywords that are not currently being queried. To counterbalance this, there are graphs at the bottom of the visualization that relay raw counts of keywords. As far as the level of resolution goes, state level is not preferable. However, displaying on a county-level would require much more processing on the Java server and, therefore, more wait time for the user. Something that could be implemented to alleviate this problem is the ability for the user to zoom into part of the map and only query for counties inside of that region. Likely, it would be necessary, at least initially, to limit this to state level instead of an arbitrary division between states, again for processing concerns. This design decision reinforced Brooks's Lesson 3. [Brooks, Fred. *The Design of Design*. Boston: Pearson, 2010. 173. Print.] The developers had to examine whether they were designing the right thing, since there were several constraints. By collaborating with the project mentor, the developers were able to develop this program to suit the needs of the medical researcher and are satisfied with the performance and current capabilities of Tweet Mapper.

6. [authors: Adams, Cox, Nash] User Manual

Warnings

Use of this software comes with inherent risks, including, but not limited to, risks that affect a user's physical, mental, and emotional well-being. The user agrees to accept these risks and hold the software creators harmless for any negative effects the user experiences by using this software.

Note that this software can potentially display many different graphics with traits that include, but are not limited to, bright colors, rapid motions, and crude text. Users that could be negatively affected by these graphics, including, but not limited to, users that have a high propensity of *suffering from epileptic seizures, should use this program with caution and consult with a licensed physician prior to using this program.*

Note that this software requires the use of a physical computing device, including, but not limited to, a computer with keyboard and mouse. Users should follow the instructions and warnings of the device manufacturer(s) in order to safely use the devices with minimal risk of suffering from physical injury, including both short-term and long-term injuries.

Terms and Conditions
Tweet Mapper 1.0
Copyright (c) 2013 Michael Adams, Shawn Cox, Drew Nash

In order to use Tweet Mapper, the user must agree to abide by the Terms and Conditions presented when loading the software program.

COPYRIGHT NOTICES:

Tweet Mapper 1.0 relies on other software programs developed by third-party developers. Below are the appropriate copyright notices.

nvD3.js:
Copyright (c) 2013, Michael Bostock
All rights reserved.

jquery.js:
Copyright 2010, John Resig
 * Dual licensed under the MIT or GPL Version 2 licenses.
 * http://jquery.org/license

moment-min.js:
moment.js
version : 2.4.0
authors : Tim Wood, Iskren Chernev, Moment.js contributors
license : MIT
momentjs.com

pikaday.js:
Pikaday
Copyright © 2013 David Bushell | BSD & MIT license | https://github.com/dbushell/Pikaday

Set-up
Prior to configuring the Tweet Mapper visualization tool, the appropriate xAMP tool must be

properly installed on the machine that is intended to host the project. For Windows, this is WAMP, which is available from http://www.wampserver.com/en/. For Mac OS X, MAMP can be downloaded at http://www.mamp.info/en/index.html. On Linux, LAMP can be installed from the command line using the appropriate package installation tool for your distribution of Linux (*apt-get, tasksel,* etc.). Please consult the provided documentation from the provider for help installing WAMP and MAMP. For help installing LAMP, please refer to the help forums of your Linux distribution for help from experienced users. The installation and use of these applications is not supported by the developers of Tweet Mapper.

To set up the Tweet Mapper program, download the *TweetMapper.zip* file to your computer. This file must be obtained directly from the developers, with permission. It is recommended that this be stored on the Desktop for easy access. See Figure 5.1.1 for a visualization of this step.

Figure 5.1.1

Unzip the file by double-clicking on it. Windows, Macintosh, and Linux should support the decompression of a ZIP file.

Once *TweetMapper.zip* has been unzipped, a folder titled *TweetMapper* should appear. Open this folder by double-clicking on it. This should work for Windows, Macintosh, and Linux. See Figure 5.1.2 for a visualization of this step.

Figure 5.1.2

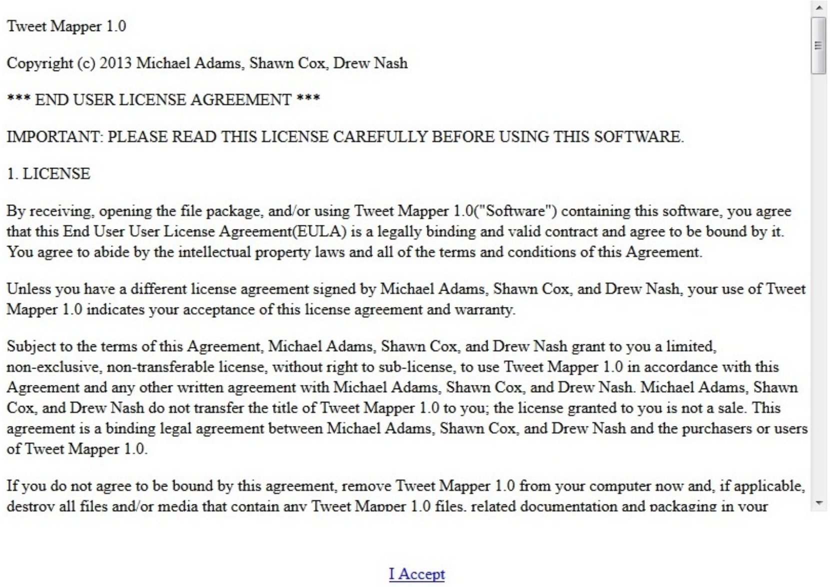
Once the folder is open, ensure that the *index.html* file is present. If so, proceed to the *Use* section to start using Tweet Mapper!

Use

Navigate to the location on your computer where you saved the Tweet Mapper files. If you are unsure of where this location is, consult the *Set-up* instructions, and consider loading another copy of Tweet Mapper on your computer.

Open the *index.html* file by double-clicking on it. A window will open.

In order to use the Tweet Mapper program, you must accept the Terms and Conditions. Please read them carefully, and click *I Accept* at the bottom if you agree to abide by these Terms and Conditions. See Figure 5.2.1 for a visualization of this step.



Tweet Mapper 1.0

Copyright (c) 2013 Michael Adams, Shawn Cox, Drew Nash

*** END USER LICENSE AGREEMENT ***

IMPORTANT: PLEASE READ THIS LICENSE CAREFULLY BEFORE USING THIS SOFTWARE.

1. LICENSE

By receiving, opening the file package, and/or using Tweet Mapper 1.0("Software") containing this software, you agree that this End User User License Agreement(EULA) is a legally binding and valid contract and agree to be bound by it. You agree to abide by the intellectual property laws and all of the terms and conditions of this Agreement.

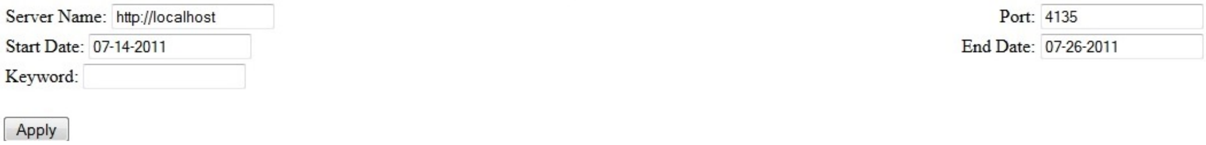
Unless you have a different license agreement signed by Michael Adams, Shawn Cox, and Drew Nash, your use of Tweet Mapper 1.0 indicates your acceptance of this license agreement and warranty.

Subject to the terms of this Agreement, Michael Adams, Shawn Cox, and Drew Nash grant to you a limited, non-exclusive, non-transferable license, without right to sub-license, to use Tweet Mapper 1.0 in accordance with this Agreement and any other written agreement with Michael Adams, Shawn Cox, and Drew Nash. Michael Adams, Shawn Cox, and Drew Nash do not transfer the title of Tweet Mapper 1.0 to you; the license granted to you is not a sale. This agreement is a binding legal agreement between Michael Adams, Shawn Cox, and Drew Nash and the purchasers or users of Tweet Mapper 1.0.

If you do not agree to be bound by this agreement, remove Tweet Mapper 1.0 from your computer now and, if applicable, destroy all files and/or media that contain any Tweet Mapper 1.0 files, related documentation and packaging in your

Figure 5.2.1

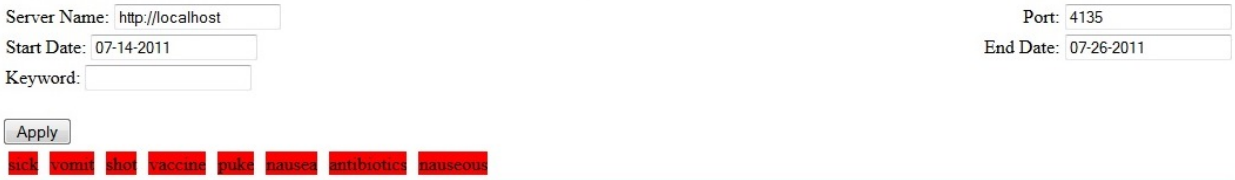
After accepting the Terms and Conditions, you can enter a *Server Name*, a *Port*, a *Start Date* and *End Date* for the tweets that you want to visualize. The start and end dates should be in the range of tweets that are housed in your database. The server name and port will be unique to the user. For assistance gathering this information, consult the *Code Organization* and *Set-up* instructions. See Figure 5.2.2 for a visualization of this step.



Server Name: http://localhost
Port: 4135
Start Date: 07-14-2011
End Date: 07-26-2011
Keyword:

Figure 5.2.2

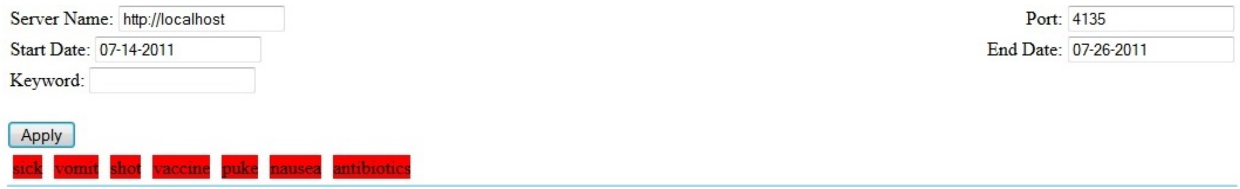
After setting these initial parameters, you may now choose the keywords that you want to visualize. In other words, if you choose “flu” as a keyword, tweets in the database with the word “flu” will be visualized on the map and in the graphs generated by Tweet Mapper. To add a keyword, type the word in the *Keyword* text box and click *Apply*. See Figure 5.2.3 for a visualization of this step.



Server Name: http://localhost
Port: 4135
Start Date: 07-14-2011
End Date: 07-26-2011
Keyword:

Figure 5.2.3

The keywords that have been added will appear inside of red boxes. To remove a keyword, simply click anywhere in its red box and click *Apply.* See Figure 5.2.4 for a visualization of this step.



Server Name: http://localhost

Port: 4135

Start Date: 07-14-2011

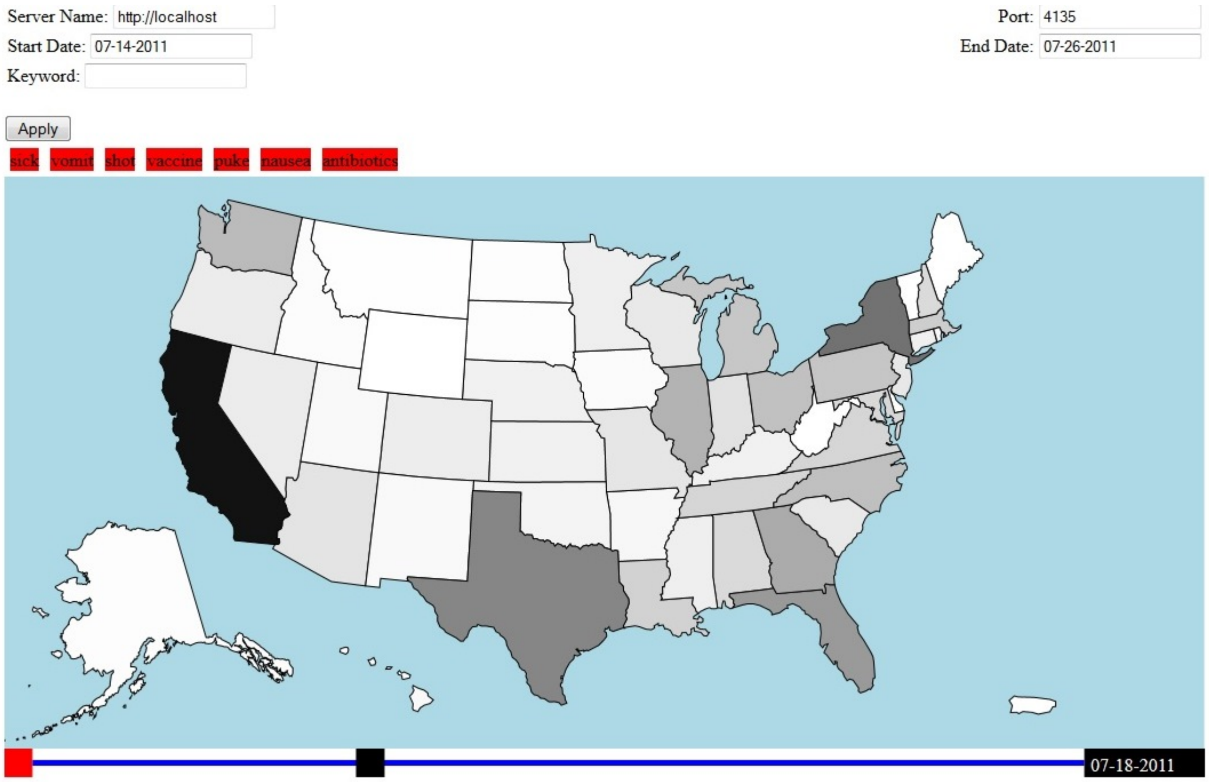
End Date: 07-26-2011

Keyword:

sick vomit shot vaccine puke nausea antibiotics

Figure 5.2.4

To see the tweets visualized on a map of the United States, the user can either drag the black control box to the right to move forward in time, or to the left to move back in time. To have the program automatically iterate through time, the user can click on the red box on the left side, which is the *Play* button. The box is red when the map is paused and green when the map is moving through time. Notice that the date that is being depicted on the map at a given time is listed on the right side of the time scroll bar at the bottom of the map. See Figure 5.2.5 for a visualization of this step.



Server Name: http://localhost

Port: 4135

Start Date: 07-14-2011

End Date: 07-26-2011

Keyword:

sick vomit shot vaccine puke nausea antibiotics

Figure 5.2.5

To see graphs on time versus the number of tweets, scroll down to the bottom of the page. The graph on top is a line graph, while the graph on the bottom is a bar graph. They both depict the same data, just in a different form. See Figure 5.2.6 for a visualization of this step.

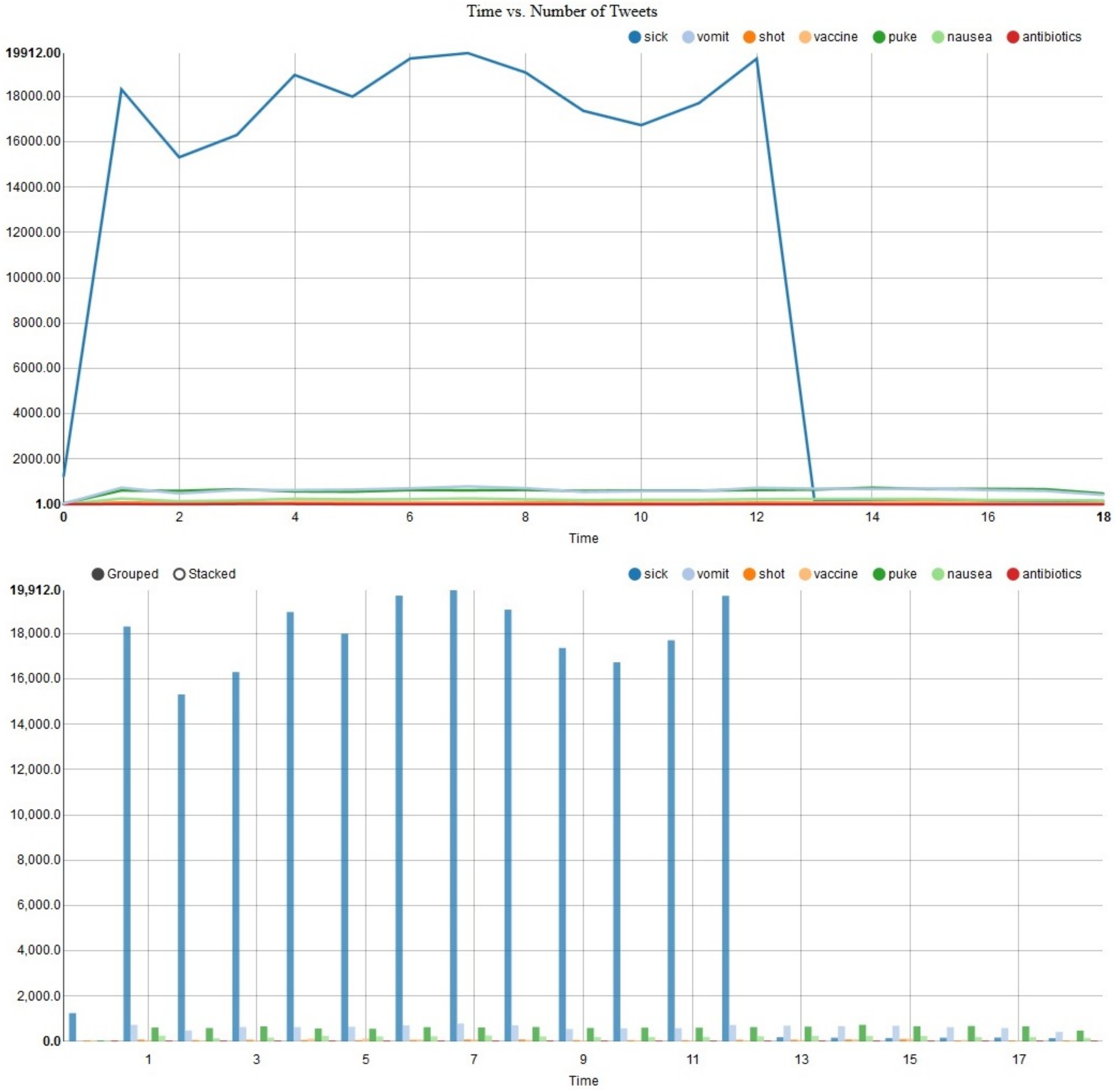


Figure 5.2.6

To remove a keyword from a graph, just click on it. The bar graph can be changed to stacked form by clicking on S*tacked*. It can be changed back to a grouped form by clicking on *Grouped.* See Figure 5.2.7 for a visualization of this step.

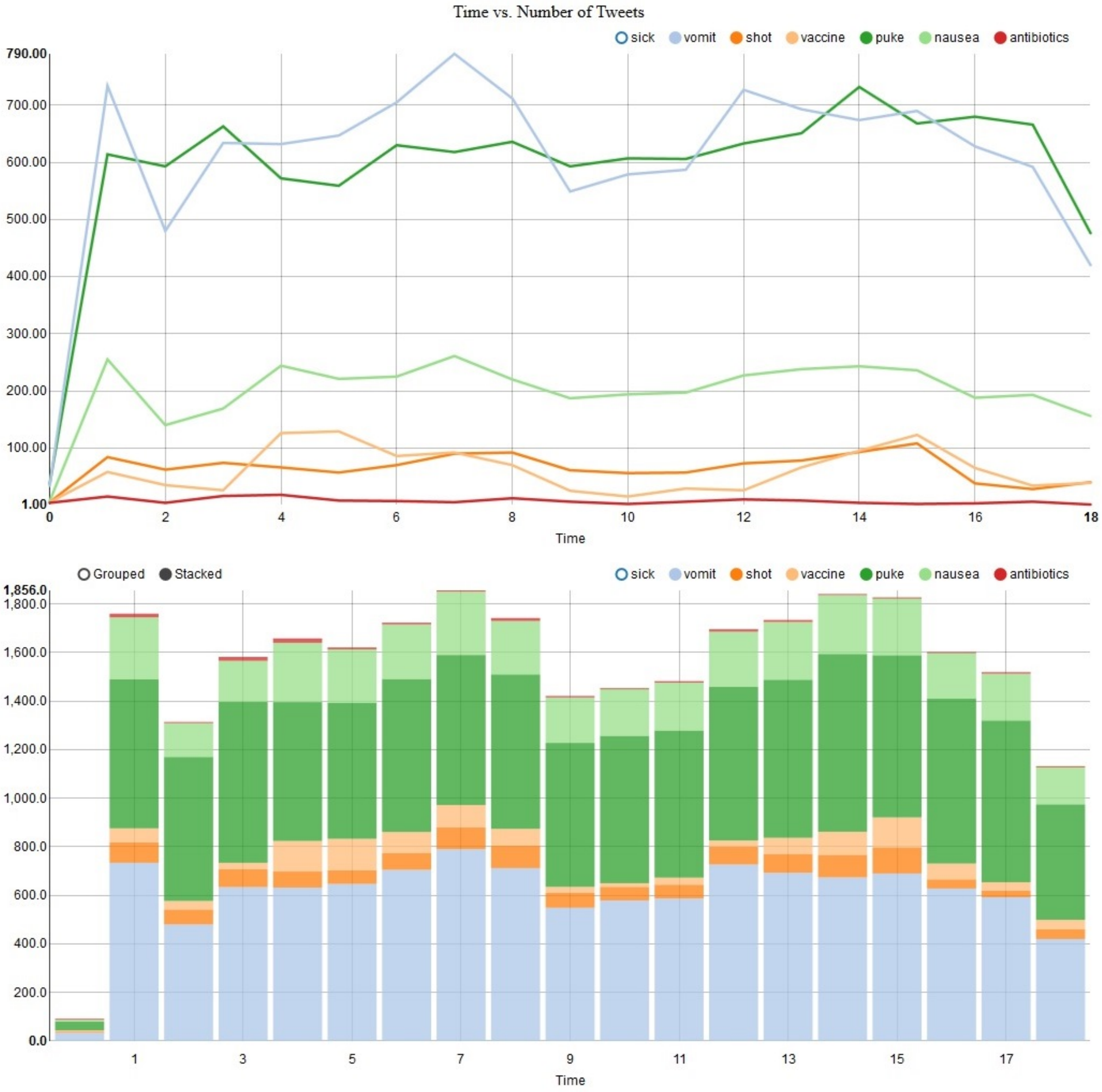


Figure 5.2.7

To close the Tweet Mapper program, simply close your web browser normally.

A demonstration of Tweet Mapper in action is available on YouTube.
https://www.youtube.com/watch?v=LyZl9pEzAkg

Code organization.
The code for Tweet Mapper is broken into two three distinct sections: the Java code for generating the database necessary for the visualization, the Java/Apache server responsible for querying the SQL database, and the JavaScript code that handles the visualization in the dynamically generated web page.

The first section of code is provided without warranty and is only applicable in certain situations. The codes require a pre-generated XML file of data from Twitter in a predetermined format and will, in parallel, read the individual tweets into memory to generate and execute the corresponding SQL statements. These codes are maintained by the developers and are available upon request. Due to the specific nature of the SQL generation, these codes will, in most cases, be of little use. Instead, the user would need to supply their own SQL database for query. If the user is not able to generate a SQL database, then that user should consider hiring a professional database consulting firm to do this.

The second section of the code forms the Java/Apache server that handles both the SQL queries and the HTTP requests of the visualization, each of which has its own compartmentalized code. The SQL queries are handled through the Java Database Connector (JDBC) library. The visualization relies on the Apache Tomcat Web Server infrastructure to accept the communications from the JavaScript code that handles the visualizations. Both the JDBC and the Apache Tomcat libraries and dependencies are compiled into the Executable Jar File and do not require external libraries for compilation or runtime invocation.

The final section of the code is the JavaScript visualization itself. The execution of the relevant codes require the use of an AMP server as noted in the *Set-up* section. The distribution, installation, and support of these AMP servers is in no way supported by the Tweet Mapper developers and must be used at the user’s own risk. By hosting the *TweetMapper* visualization directory provided in the given web service, the HTML page can be generated and used as noted in the *Use* section above. The webpage can then be accessed by using Mozilla Firefox (or many other common web browsers) from any location, provided that the computer has network access to the AMP server.

Since Tweet Mapper was developed for an academic researcher, it is not publicly available for download. However, the developers are willing to work with programmers that may want to experiment with Tweet Mapper and/or contribute to the project. For more information about downloading Tweet Mapper, email the developers.

Works Cited

Bensinger, Ken. “Boston Bombing: Social Media Spirals out of Control.” *LA Times*, Apr. 2013. Web. 3 Dec. 2013.

Brooks, Fred. *The Design of Design*. Boston: Pearson, 2010. 173. Print.

Curtis, Sophia. “Twitter IPO: 14 fun facts.” *The Telegraph*, Nov 2013. Web. 3 Dec. 2013.

Ginsberg, Jeremy. “Detecting Influenza Epidemics Using Search Engine Query Data.” *Nature* 457 (2008): 1012-1014. Print.

Rutenberg, Jim. “Data You Can Believe In.” *New York Times*, 20 Jun. 2013. Web. 3 Dec. 2013.

Stein, Sam. “NIH Losing $1.7 Billion, 700 Research Grants Due to Sequestration.” *Huffington Post*, 4 June 2013. Web. 3 Dec. 2013.