Development of a Web Application to Explore Twitter Organizations and Collaborations
Scott Longwell, Class of 2012

Websites such as Facebook, MySpace, and Twitter have become a central aspect of modern life, changing the way people interact and communicate. As representations of social networks, they also provide excellent platforms for sociological research. Many of these websites offer an application programming interface (API), which allows a developer to write code that can access and modify the site’s data in a relatively simple fashion. Twitter, a “microblogging” website where user posts (or “tweets”) are limited to 140 characters, provides such functionality. Twitter’s APIs come in two main flavors: a REST API and a Streaming API. The REST API can return a variety of information depending on the request. Profile information about a specified user, the users a given user is following and is followed by, and recent tweets on the public timeline are all accessible through the REST API. On the other hand, the Streaming API continuously forwards all tweets meeting criteria specified in the request, along with information about the posting user. The criteria can be keywords, specific users, and/or tweet location. Both APIs are needed to thoroughly investigate the wealth of social network data Twitter provides.

My work for the summer was to program a web application utilizing the Twitter APIs as an avenue for sociological research. In essence, there are three sequential elements of the project: data collection, processing and storage, and analysis and visualization. For the first phase, I wrote code in PHP (a web friendly scripting language) to make requests to both of the Twitter APIs based on keyword parameters from a basic web form. In the request scripts, I also implemented an authorization scheme called OAuth, which Twitter requires for certain API calls. These same scripts then place the raw data returned from the APIs into a database called MySQL. Raw collected data is then processed before it is stored in its final state. The goal of this phase is to extract additional information from the raw fields which Twitter’s API’s return. For example, a tweet’s text can be parsed to determine if it contains @mentions (tweets are directed to other users through the @username convention), is a “retweet” (a tweet that user A posts and user B reads and posts again, signified through the “RT @username” convention), or contains hyperlinks (through parsing for “http://”). In order to be useful, some raw data is also cleaned up. I wrote another script to sort through user-specified locations, which can be well formatted (“Akron, Ohio, USA”), colloquial (“The Big Apple”), or junk (“The jungle baby!”). The script uses two Yahoo! API services to take well-formatted and colloquial type locations and fit them into a standardized set of data fields (latitude, longitude, city, state, and country).

The finalized data is then retrieved from MySQL through a PHP script in order to be analyzed in a variety of ways. Through Google Map’s JavaScript API, I created a webpage that displays pushpins representing the locations of gathered tweets (data returned on tweets occasionally contains coordinates based on IP address or a mobile device’s GPS). Upon mouse over, these pushpins display additional information about the tweet, including the text, username, date, and source. Additionally, I wrote a PHP script to format and export files for the network visualization program Pajek, which is capable of rendering social graphs in three dimensions (the nodes of such a graph are users, who are connected through @mentions to other users). In addition to general statistics, these visualizations are the net product of the web application and are what will allow researchers to test hypotheses about the social networks Twitter represents.

Though the web application in its current state has proven to be useful, a number of challenges remain for future work to address. I am currently reworking the MySQL database and relevant scripts to incorporate a buffer for the Streaming API that will hold data until it is processed into a new, restructured database. This update will allow the processing stage of the project to be automated and ease analysis of the data. In addition, a script to identify and mark “spam” tweets based on established criteria would be a very useful feature for the application. The current user interface for the web application, a simple set of web forms, could also be embellished and extended to comprehensively cover all aspects from data collection to analysis. Realization of these goals should provide a fully functional tool for probing and analyzing Twitter in a sociological context.

Faculty Mentor: Prof. Dhiraj Murthy
Funded by the Gibbons Summer Research Program