Brianna Gainley, Nick St. Pierre, Alan Estrada

brdsi:
A Twitter Newsfeed Aggregate

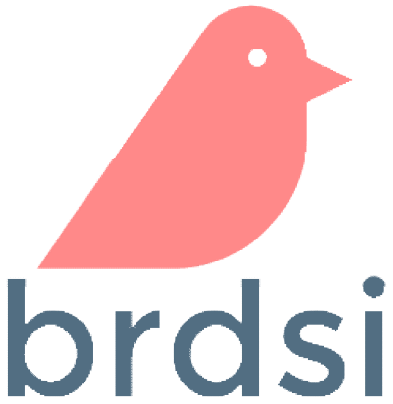

Table of Contents

Project Goal

Twitter offers a wealth of data generated by people all over the world, but does not offer a user-friendly means to sort and access this information. Our group will create a web app that acts as a “popular” news site by allowing users to sort trending tweets by geographic location or personal interest with an in-depth filter system. These tweets will populate a home page to give the user access to the most relevant and current trends happening around the globe.

Application Features

Our web app brdsi will make full use of Twitter’s APIs and available libraries to aggregate information on the most current trends and user tweets from around the world. These tweets will be filtered into a home dashboard and presented to the user visually. Trend and tweet updates will happen in near-real time so that the user constantly has access to the most current information, thereby giving them what is essentially a newsfeed of topics that are most popular at the time. Users will be able to sort this information with our extensive filter system, allowing them to view trends by geography, shared interest with friends, or personal preference of topic. We would also like to include the feature to sort for only tweets that include a link to an outside news article or other external media, for people who would like to learn more on the topics of their interest.

Figure 1

Layout and Options

When users first visit the site, they will be presented with a generic population of trending tweets from across the globe. These tweets will be randomly pulled from Twitter's Streaming API and displayed in groups to represent the top ten or so trends. On this page, a navigation bar will display some of the more prominent options, such as sort by location, friend list, or tags. The streaming tweets will be the highlight of the page, giving focus to the trend the user is interested in.

Figure 1 shows a potential GUI that the user can engage with. The login fields to the left will enable the user to login to their pre-existing Twitter account so that our app may access their friend list. Users will not be able to post tweets from our site, but instead may filter trends based on tweets their friends have posted, so their feed

displays tweets on relevant trends that only people they know have created. The "friend trends" option in the top nav bar would be greyed out if the user chose not to login to their account, as is shown here.

Additionally, users may sort trends by tags that they create using our search bar. The difference between this search feature and Twitter's generic search is that users will be limited to sorting solely on trends. To ease this restraint and perhaps to suggest what they may want to search for, an auto-fill drop down may be added that populates with only trending terms as the user types.

The furthest option in the nav bar is one that we are considering adding, time pending at the end. We would like to incorporate a separate filter that can scan geographic locations for their "moods". For instance, the filter would scan for words like "happy, good, excited" etc. and give the region a positive mood rating. The filter would conversely also search for words like "unhappy, frustrated, miserable" etc. and label the region negatively. We would like to document these trends over time for regions and dynamically generate graphs per monthly or weekly review of these ratings to visually display a region's mood or attitude and how it changes through the year. This would be an interesting way to document how a populace is feeling as good or bad events occur in their region each year.

Software Component Requirements

Twitter Integration

Our web app will make heavy use of the Twitter API. We need access to tweets as they are created, so we will specifically be using the Streaming API, which returns tweets just being created and is easily searched on and filtered. The Streaming API

has built-in geolocation parameters as well, which will be immensely useful to our filtering system. The Streaming API requires authentication to access Twitter's servers, so we will also be using oAuth, which is Twitter's preferred method of authenticating users to its servers.

In addition, we will be using a library to ease the integration of Twitter's features into brdsi. Twitter has many supported libraries, but we decided to use PHP to build the site for two reasons: first, most of the team is comfortable and familiar with PHP from prior coursework, and second, of the many supported PHP libraries, there are a few that integrate particularly well with the Streaming API and are well documented with examples. We are looking into using 140dev's Twitter Framework for the above-cited reasons.

Visual GUI Creation

In keeping with creating a site that interfaces well with Twitter, we would like to use Twitter Bootstrap as a basis for creating brdsi's GUI. Bootstrap provides a comprehensive framework that comes with many highly customizable and adaptable UI components, such as login forms, search bars, tags, and nav bars, all of which will be included on our site. For any additional components that we may need, we plan to use jQuery UI, which is equally adaptable. If time permits and we are able to create our regional mood meter, we will likely also look into using HTML5 canvas to create our dynamic trend graphs to display our findings.

Intended User Base

Our intended user base would mostly be made up of people that are already familiar with and enjoy Twitter as a social media feed. The flow of the site and the use of

trends to fill a newsfeed would make the most sense to people accustomed to Twitter's service. Having a familiarity with the service would also enable some extra features for Twitter users, such as the ability to integrate their friends list.

Other possible users include people that want to get easily digestible snippets of news and people that like to hear the population's take on current events rather than traditional media's. Each tweet is at most 140 characters, so each news item is short and brief. Most tweets will only be the author's opinion on the subject, so the bias of traditional media is left out as well, leaving only the opinions of ordinary people.

Potential Issues

Software

Based on our initial tests, we don't think that Twitter integration to our web app will be much of a challenge. Twitter has a very well documented API that makes it incredibly easy for developers to hit the ground running in fusing the developer's system with their own. One challenge we may come across is the integration of the oAuth authentication method, which none of us have familiarity with. However, the documentation for incorporating this is also very well done, which will ease the integration process.

When we first started to list the features we wanted for our app, one of the first we decided on was to have a nearly-live feed of new tweets and data. A problem that we may run into with this is Twitter's hard cap on requests per hour. With the Streaming API, we are only allowed to make requests to Twitter's servers 100 times per hour. This at first seemed like it wouldn't be enough for a live feed, but this actually means we can make a request for new information once every 35 seconds. This should be more than sufficient, as anything faster would make it difficult for the

user to process what is already on their screen. We may even consider making fewer requests so that the site updates once every one or two minutes instead.

One final issue that we discovered was the ability to see all of our tokens and requests in plain text if the site was hosted on the school's servers. If someone were to accidentally view our site from cs.uml.edu instead of weblab, the PHP code is visible which is undesirable. To mitigate this, we will likely host the site on one of our team member's personal sites, on a subdomain. That way, there is no issue of trying to configure weblab servers to hide sensitive information. This will also allow our app to have a permanent home even after the scope of this course.

GUI

We don't foresee having many, if any issues at all with the GUI development portion of this app, due to the ease of configuring the Bootstrap framework. The biggest problem we have encountered so far is finding a good way to format the data from the tweets in such a way that is familiar to Twitter users, but not a direct copy. We want to put our own mark on the site so it doesn't look like a Twitter clone, so our design must be careful to emulate but not completely imitate Twitter's design decisions.

Timeline

	Bri	Nick	Alan
Weeks of 2/10-2/17:Create brdsi base GUI and publish/configure subdomain on bricreates.com	X		
Weeks of 2/10-2/17:Integrate Twitter Streaming API and display trending tweets.		X	X

By Alpha Release: Functioning web site with streaming trends (no tagging, logins, or geolocation yet)	X	X	X
Month of March: Expand UI to include tags, and searching. Work on filter for links and media.	X		
Month of March: Work on filter for geolocation and displaying trends based on either location selected or location detected from device.		X	
Month of March: Work on filter for user-defined searches on trends and auto-complete of search-box for trends only.			X
For Beta Release: Expanded website that allows geo-tagging, and user-defined searching with auto-complete	X	X	X
Month of April: Integrate user login to the UI and design friend page to display friend-specific tweets.	X		

Month of April: Enable oAuth so that users can query Twitter's servers to match friends' tweets against trends		X	
Month of April: Integrate previously created filters for use with user's friend list to sort on specific friend's tweets			X
Month of April (Time Permitting): Add additional page to display graph of moods for a region and use HTML5 Canvas to graph	X		
Month of April (Time Permitting): Create a tracking system for regional "moods" by creating a keyword filter and backend logging to use for graphical display		X	X
Final Release: Fully fleshed out site with geo-tagging, user-searches, media filters, and logins	X	X	X

Acceptability Criteria

Release Requirements

At our release, we will have a fully functioning web app that integrates completely with Twitter's Streaming API. The site will be able to pull in, format, and display newly created tweets that match up with geographic trends. Users will be able to filter these tweets based on the location they are interested in, the search terms they use, or by tweets that include external links and media only. Users will also be able to link to their actual Twitter profile so that they can see only trending tweets that their friends have created.

Extra Features (Time Permitting)

We would additionally like to include a "mood meter" that tracks how regions are feeling based on certain positive or negative keywords that we search users' tweets for. These moods will be tracked over time and graphed using HTML5 canvas to give a visual display of how a region has been "feeling" over the course of a year.

Future Considerations

If this project goes well and we would like to extend it, mobile versions for Android and iOS would be a welcome addition to the app so that users could get their newsfeeds delivered directly to their phone in an easily digestible format. The short easily readable length of tweets makes sense to translate to phones so that people on the go can get their news-bites on the go too.