Preserving Trend History in Real-Time Twitter Visualizations

David Pitt
Stanford University
dpitt1@stanford.edu

Stephanie Guo
Stanford University
stguo@stanford.edu

Ollie Khakwani
Stanford University
olliek1@stanford.edu

Hugh Cunningham
Stanford University
hughec@stanford.edu

ABSTRACT
Twitter presents an enormous dataset with over half a billion active users who generate a billion tweets every three days. The dataset is still far from being fully leveraged as a tool to explore trends, especially highlighting them in real time. We explore an approach to visualizing Twitter data in real time while compressing the timeline to prevent any loss of earlier data and generate a view of a longer-trend pattern, while maintaining the ability to view data at a micro-level. We aim to provide the basis for further exploration of real-time Twitter data mining and analysis for journalistic and other purposes.

Author Keywords
Real Time, Social Media Visualization, Timeline Compression

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
Currently, there are no systems that allow users to explore the changing popularity of a topic on Twitter in real time in an intelligible way. Our work aims to improve on previous visualizations that attempt to display trends in real time. These related works fail to aggregate the information that is retrieved, fail to show data in real time, and fail to show the larger trend, or any combination of the above. Our visualization endeavors to overcome all the limitations listed above.

With around 340 million tweets published a day and 500 million active users, Twitter has become the hive mind of the Internet, providing an insight into what the world is thinking about at any given moment. Creating ways to visualize this vast quantity of information in an intelligible and user-controlled way could provide significant insight into the way that trends grow, spread, and decay, and could also potentially lead to the discovery of ways to detect the formation of a trend in its early stages, and be used to study the network effects involved in the spread of information. While our research goal is more limited, focusing on creating an interface that updates in real time while aggregating all the data that is collected, ultimately we hope our work can help to solve one of the primary barriers to using the rich data provided by social media sites like Twitter for the analytic purposes just mentioned.

PREVIOUS WORK

Previous work on visualizing Twitter has almost always focused on visualizing the smaller trends surrounding a particular event; Shamma from Yahoo! Research in particular has explored ways to surface emerging trends in real time to build richer media experiences around live televised events. His Statler system (available in reduced form at http://shamurai.com/bin/statler/) mined data from President Obama’s inauguration in 2009, performing analysis of the online conversations and matching them up against real world events to give “an ambient awareness of what conversations are unfolding online.” His system used similar analytics to the ones we explored with Tweet Stream, emphasizing the use of frequency of tweets as a metric to detect the occurrence of importance of events, although Statler also developed algorithms to create scales of “Importance” and “Chattiness” at given moments. Shamma et al. also extracted the most frequent words that occurred in inauguration-related tweets to keep provide users with a simulated real-time update of the top-20 terms,
somewhat similar to the data that Twitter Stream Graph (see the Methods section) provides, in addition to providing a changing map with the tweets’ geo-data. They also provide an overview of the ‘relevance’ of each moment over the course of the speech.

Vox Civitas is a Twitter analytics system that derives a similar set of metrics to Statler based on the social media activity surrounding large-scale broadcast events (in this case, the State of the Union Address), but is designed specifically as an exploratory tool for journalists to extract news stories. Like Statler, the goal is to allow the user to visually interpret the data at an aggregate level, although as a journalistic tool, it places more emphasis on the ability to interpret data on the individual tweet-level as well. Vox Civitas provides a wide-range of filters on keywords, which tweets were direct quotes from the address, and performs sentiment analysis. However, unlike the MTV VMA visualization, none of the data was provided in real time - the tweets were collected just after the event and then analyzed, presumably because the level of analysis performed meant that it could not be done in real time in a browser. One of the future research goals for Vox Civitas was to create a more generalizable system that could be applied to other types of broadcast events; we hope that our work will help to extend this a step further and eventually create a real-time analysis system that does not need to be tied to a broadcast event.

Brasoveanu, Hubmann-Haidvogel, and Scharl have also done research into visual interfaces to display aggregations of social media data in real time, with a particular emphasis on effectively showing the emergence and decay of topics. They focus on three representations: stacked and color-coded sentiment bars, threaded arcs to visualize interactions...
between tweeters on top of the number of tweets, and falling bar graphs to visualize the most important topics.

**METHODS**

Tweet Stream is a tool to visualize Twitter trends in real time while preserving the history of the trend from the start of a user session. We measure a Twitter trend by a search term’s popularity. We tally the number of tweets that contain a keyword over intervals of 5 seconds starting from the time the user sends the search query. Tweet Stream focuses on a global, high-level trend instead of a low-level glimpse of raw tweets. The first part of this section will explain the limitations of previous Twitter visualization tools and the techniques Tweet Stream uses to overcome those limitations. The second part of this section will explain the technical aspects of the application.

We preface the discussion of previous visualization with the key asset of Tweet Stream that previous visualizations lack. Tweet Stream preserves the history of a Twitter trend on a single graph of constant size for any amount of data; the older information is not pushed off the edge of the screen as new real-time data is added. Tweet Stream achieves this ‘memory’ of data by compressing the graph horizontally as new data appears. The leftmost point on the x-axis timeline is always the time at which the users initiated the Tweet Stream session. The rightmost point on the x-axis timeline is always the current time. As the latest frequency count is added to the rightmost side of the screen, the x-axis re-scales to accommodate the new information. This technique allows the user to see a trend over ten seconds as easily as a trend over ten minutes, and additionally the user can see the smooth transformation from the ten second graph to the ten minute graph.

The MTV VMA and Statler visualizations mentioned above have the advantages of displaying tweet counts in real time, showing the trend over time, and enriching the visualization with lower level data analyses such as the frequency of related keywords within the trending tweets. Both visualizations preserve the history of the trend without a loss of data; however the preservation of history is not the result of a clever graphing technique but the result of knowing the event duration ahead of time. The architects of the MTV VMA and Statler visualizations know that they will be recording tweets for a set amount of time and they can set the x-axis of their trend graph accordingly. If the visualizations were to work for an indeterminate amount of time, their design would fail to preserve the history of the trend because newer tweets would eventually push older tweets off the left end of the timeline. The above is true of Vox Civitas as well. Vox Civitas is particularly limited because the tweet data was collected after the event, so the visualization is not in genuine real time.

Monitter is a real time Twitter visualization tool that works for any search term over any amount of time, as Tweet Stream does. Monitter allows the user to specify one to three search terms and it will push the text of all tweets containing that search term to the page. Monitter has significant limitations. It fails to provide any high level graphic; the only information displayed is the raw text of the tweets and the user who published it. For many topics, particularly at high-traffic times, Tweets move down the screen so quickly that the user can hardly read them. Monitter has no mechanism to preserve the history of tweets. The incessant cascade of text pushes previous tweets off the bottom on the screen almost immediately after they appear.

Tweet Stream is a flexible tool that will track any search term and can run for any amount of time. Tweet Stream counts the number of tweets containing the user's search term over an interval of five seconds and plots the count on a line graph. Tweet Stream uses the following techniques to remember trend history, preserve low-level information, and display the graph to the user in an intelligible way.

**Compression and Re-Scaling** - We eliminate the problem of loss of old information by condensing the graph as new tweets come in; no data will ever disappear behind the y-axis line. The smooth “slow-in, slow-out” animation makes it clear to the user that they are viewing the same dataset as before with the incoming data tacked on the end.

**Tweets per Second** - We made the number of tweets per second on the chosen topic our primary data point. By reducing the complexity at first glance to a single number, the mountain of data becomes simple enough for the user to easily spot changes and trends, even as the graph continually updates to include the data arriving in real time. The number of tweets per second also serves as a useful proxy for the amount of attention a subject is getting on Twitter, and served as the backbone of previous visualizations like the MTV VMA visualization, Statlet, and Vox Civitas.

**Using a Sample of Tweets** - Twitter will only make 1% of worldwide tweets available through its API and so we take a random sample of all tweets and use them to represent the entire population. Our sampling does not filter any of the tweets by keyword until the user enters one, giving our visualization the flexibility to display data on any topic, rather than being tied to a specific hashtag like Vox Civitas or the MTV Video Music Awards visualizations.

**Tooltip** - The tooltip allows us to preserve the display of the larger trend while adding the ability to view the lowest-level information, i.e. the actual tweet. At the same time,
bringing up the detailed data only when prompted avoids the problem that Monitter had of obscuring the larger trend by quickly pushing out the previously retrieved data. By more explicitly placing the tweet on the graph showing the entire timeline, our use of the tooltip also places the tweet more obviously in context of the larger trend.

On the server side, we use the Ruby Event Machine to connect to a public endpoint of the Twitter Streaming API at 'https://stream.twitter.com/1.1/statuses/filter.json?locations =-122.75,36.8,-121.75,37.8'. This is a POST request, supplying the filtering conditions to the streaming server. Our server consumes this stream and places tweets into an array as they come in. Before placing a tweet in the array, we also note the time that we received it (the reason for this will become clear later). Once the array reaches some maximum length (currently set to 1000), it starts deleting the tweet at the beginning of the array and shifting everything down by one to make room for the new tweet at the end of the array.

We use the Ruby gems Thin and Sinatra to run the server app and deploy on Heroku. You can find it running at http://safe-coast-8169.herokuapp.com/tweets. That page prints out all of the tweets that are stored in the array, so as you refresh the page, more are added/subtracted. We have another route on the server at http://safe-coast-8169.herokuapp.com/search.json/:query where ':query' is the term to search for in the tweets. When a request is made to this URL, we note the time at which that request was received. We then go through the tweets of the array and pick out the ones that contain our search term, placing them in a new array. During this filtering step we also require that the time that we received the tweet was no more than X seconds before the time we received the search request, where X is the interval at which we make requests to the server and update our visualization. Right now, X=10. We do this so that the server will not duplicate the results that it sends to a browser and result in a single tweet being counted multiple times. After filtering, the server responds with a JSON representation of the new array.

We also have a route at http://safe-coast-8169.herokuapp.com/server.json that returns the entire array of tweets that the server has stored as JSON.

RESULTS

Our implementation of Tweet Stream accomplished the goals we had set. The application plots a comprehensible, rescaling line graph that updates in real time and shows a trend of a search term’s popularity over time.

Our implementation reliably updates in real time, compressing the graph and adding a new data point every 5 seconds without any lag time. The initial load time of 5 seconds is less than half that of one of the existing Twitter visualizations, Tweet Stream Graph, and is a necessary lag to count the number of tweets needed to generate the first data point.

One of the biggest successes for Tweet Stream is that users we tested with instinctively understood the compression and re-scaling of the graph, and that the data was being updated in real time. Although some users were confused by the first re-scaling given that it caused the graph to compress to half its size, on subsequent re-scalings they quickly grasped what was happening. On their first use, users also appeared to engage with Tweet Stream, spending an average of around 3 minutes examining the visualization.

However, although most users agreed that they liked being able to choose the term they searched for, they also expressed a desire to be able to compare the rate of tweets between two or more search terms and believed this would be the primary appeal of being able to see the frequency of tweets in real time.

Another weakness is that many search terms did not return any results because of the relatively small size of the sample we are working with - this is a limitation of Twitter’s API which only gives us access to 1% of the total
tweets (roughly 30 per second on average), but it weakens the impact of allowing a user to define the search term themselves.

Users also remarked that for what appeared to be an exploratory tool, there was a dearth of interactivity in the application; several commented that they enjoyed seeing the full text of the tweet with the tooltip, but felt like having a single tweet available for each 5 second segment did not provide enough interaction to make them want to engage with the tool over a longer period of time.

**DISCUSSION**

Other users suggested having user control of the granularity of the data - as the graph compresses over time, the number of tweets appears increasingly volatile because the gap between each data point stays fixed at 5 seconds. Creating the option to alter the frequency at which data is retrieved so the graph can aggregate the number of tweets over 10, or 15 second periods for example so that the graph appears smoother could address this. It would also make the data points that the tooltip can use easier for users to click on by reducing their overcrowding as the visualization continues to run.

Journalists who specialize in social media coverage believed the tool could be extended in a number of ways to provide value to journalists covering events like the election - they were especially interested in filtering methods that could strip out "fake tweets" and social media activity pushed by either party to provide a more accurate real-time picture of public reactions to events, enabling them to call whether debates were going one way or another, or the likely future impact of political gaffes.

Because of the limited size of the dataset we had available, restricted to only 1% of the total tweets, Tweet Stream could have more potential to display the full text of each tweet, which an interface that leveraged every tweet would not necessarily be able to do. One concern that was raised during user evaluation of Tweet Stream was that having the tooltip display a randomly selected tweet did not seem as informative as displaying the most retweeted tweet, or the tweet from the person with the most followers would have been. One idea to further develop the project would be to leverage the y-axis as more than a position encoding of the rate of tweeting by lining up each tweet made over the period, starting with the 'least significant' tweet (from the user with the least followers, or the least retweeted tweet) at the bottom, going up to the most important.

**FUTURE WORK**

As our visualization continues to run and the pattern expands, we lose the fine-grained details of the data being processed, most notably the ability to view individual tweets. Our system deliberately took the approach of condensing a larger sample of data to just a few statistics that were the most informative because current examples of real-time Twitter visualizations went the opposite route and provided highly detailed information about a small subset, often anchored to a specific event. Working on creating interfaces that allow the user to view broader trends while retaining the ability to view the full text of the Tweet and the wealth of detailed metadata associated with each tweet is a clear avenue for further research, and an area that has not yet been fully explored by our research.

Further research into the visualization of broad trends and finer detail in huge, rapidly updating datasets is likely to prove a fruitful area of exploration. The interfaces produced from such research could transfer to visualizations in other domains that deal with a large number of entities that each have many sub-fields attached to them, for example, in biosciences. As well as this, previous work in the area has focused on the possibility to using massive social media datasets for journalistic purposes, and expanding the functionality of our visualization to include more of the analytics that Vox Civitas and Statler provided could have a profound impact on the relevance and reliability of "citizen journalism."

As referenced in the discussion section, Tweet Stream attracted attention from journalists whose work includes finding social media trends and gauging the response to political events. Developing smarter methods to filter types of tweet to reach the ‘real’ content, meaning genuine user reactions rather than retweets of political organizers or straightforward ‘cheerleading’ encouraged by grassroots organizers, is one area of concern. In addition, filtration methods that could leverage geo-location data, perform sentiment analysis, and other more detailed analysis while maintaining real-time updates, are of special interest to political journalists.

One area for exploration in visualizing social media data is in creating intuitive interfaces that give the user more control of the time period they explore, including the ability to pause, zoom in and out, and scroll back and forth through time, so that specific ranges of time could be viewed in more detail, including perhaps the full text of the tweets.

Alternatively, more work could be done in leveraging the full set of tweets that can be harvested in real time. In its current state, our visualization throws away the tweets that do not contain the search term, similar to previous examples that extracted only tweets containing keywords that related to a particular news event they wanted to cover. Using the full set of tweets that can be extracted via Twitter’s API would likely require research into way to rapidly classify the tweets, for example by sentiment analysis, which given the large and increasing volume of tweets shared on Twitter
would necessitate further research into algorithms that could perform this classification rapidly enough to function in a browser in real time.

Finally, plotting two or more trend lines on a single graph is a needed extension if the user is to make comparisons between two or more trending tweets. An anticipated difficulty with this extension is how to properly baseline two trend lines such that they may be easily compared. In some cases it may make sense to baseline both trend lines at the same point in the y-axis and measure relative rises and falls in the number of tweets. In other cases basing each trend line at their absolute starting number makes most sense. Another problem will be scaling trend lines of different magnitudes of popularity such that they can be easily compared. If one trend is vastly more popular than the other, the popular trend line will dwarf the smaller one and make comparison impossible. Plotting the lines on a log scale, or measuring the derivative of the trend line, are possible solutions to this problem; or perhaps an entirely new algorithm will be necessary.

CONCLUSION

Tweet Stream confronts a problem in Twitter real-time visualization and real time data visualization in general: the difficulty of preserving the history of a trend while still updating information in real time. Tweet Stream achieves this through rescaling a line graph along the x-axis such that the graph compresses horizontally as new data comes in. Previous Twitter visualization tools pioneer some of the techniques we use to visualize data comprehensively in real time. The MTV Video Music Awards visualization, Vox Civitas, and Statler combine real time tweet count updates with a trend line of tweet counts over a time range. However these visualizations are tailored to a specific event of known duration and are not real time capable for an analysis of indeterminate time. Monitter is a flexible tool that shows real time tweet data for any search term and any duration of time, however Monitter does not show any high level trends and fails to preserve previous data as new data comes in. Tweet Stream’s rescaling graph solves the limitations listed above. Tweet Stream's successes include reliable real time updating without delay and intuitive understanding of the graph’s content by users. Tweet Stream suffers from the scarcity of tweet data that the Twitter API provides; some search queries consistently fetch fewer than five tweets per interval or none at all. The application fails to capture a robust amount of low-level data; only one randomly selected tweet is displayed in the tooltip for a point plotted on the graph. Tweet Stream in its current version does not offer users sufficient interactivity. We observed that users enter a search term only to passively watch the screen fill with data. Users requested the following interactive features:

1) The ability to change the granularity of the graph by combining points on the x-axis,
2) The ability to manually filter out sources of information to exclude misinformation and disinformation, especially from political organizations for the purposes of political journalism; and
3) The ability to see more sample tweets and have them ranked along the x-axis by reputability.

An important direction for future work is graphing two or more trend lines simultaneously, and figuring out how to make comparison between trend lines comprehensible through effective base lining, scaling, or other transformations.

REFERENCES
