TweetDay

Andrew Huang, Daniel Griffin, Nikhil Mane, and Vijay Velagapudi

Project Goals

On Twitter, individuals and outlets frequently use the acronym ICYMI to bring links to the attention of others who may not have seen them. ICYMI stands for In Case You Missed It (see figure 1.). The New York Times even has a section at the bottom of its application called “In Case You Missed It” with articles from previous days. On Twitter, it is easy to miss out on information just because you did not check in throughout the day. To a lot of people this 'missing out' and the associated experience of exclusion can be agonizing. Even if you do have time at the end of the day to check your Twitter timeline, the manner in which the tweets are presented to you makes it cognitively difficult and time consuming to identify what was happening in moments throughout the day.

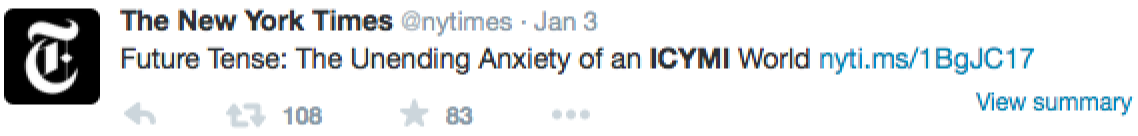

The New York Times @nytimes · Jan 3
Future Tense: The Unending Anxiety of an **ICYMI** World nyti.ms/1BgJC17

Figure 1. The New York Times tweeting an article about ICYMI: “Even the Twitter account for President Obama, one of the few people who can assume that his actions are never missed by the world, occasionally uses ICYMI...” (link to tweet; article)

We aim to create a visualization that will allow a user to see a day’s worth of information from tweets by the people they follow on Twitter.

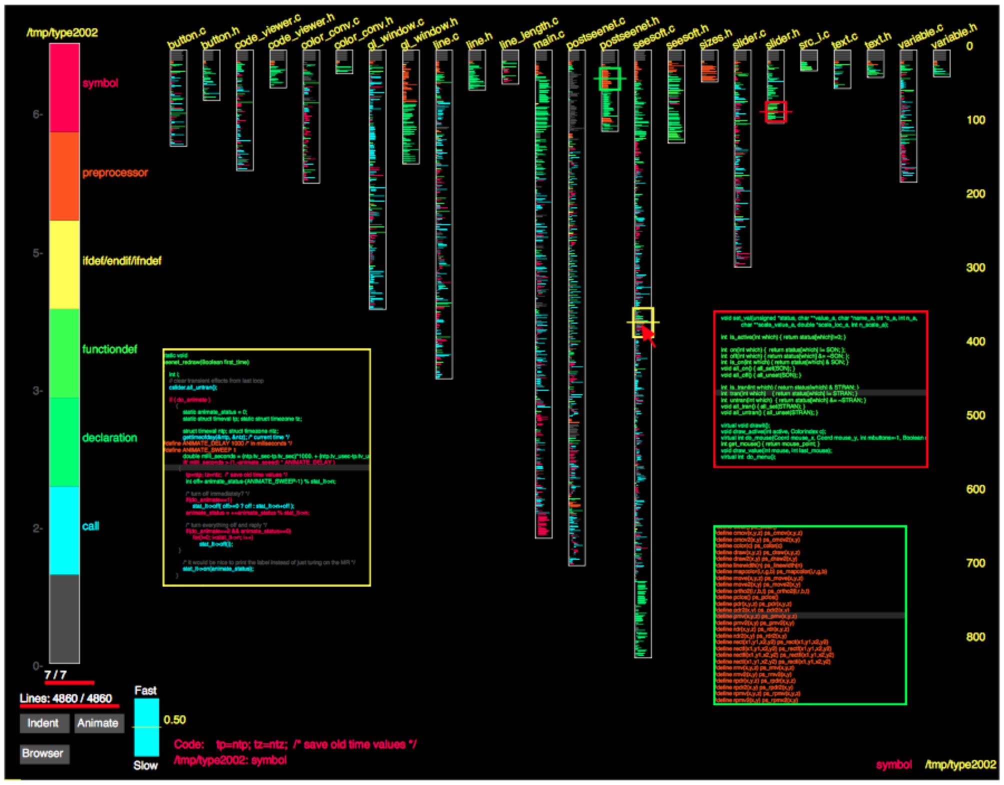
We will provide an overview of all tweets from accounts a user follows in SeeSoft fashion, explained in full below, with arcs connecting tweets to indicate replies or retweets (similar to TwitArcs). This overview will provide the context for interaction with a TreeMap-style grid displaying tweets identified as potentially notable by retweet or favorite count. We will enable interaction between the two views and an additional ability to view the contents of the tweets themselves and interact with them.

Related Work

In considering design alternatives, we looked both at the academic literature for examples in displaying the content of text and at attempts others have made in visualizing the contents of a Twitter timeline. The literature and examples are discussed below with a summarization of their strengths and weaknesses as related to our goals available in table 1.

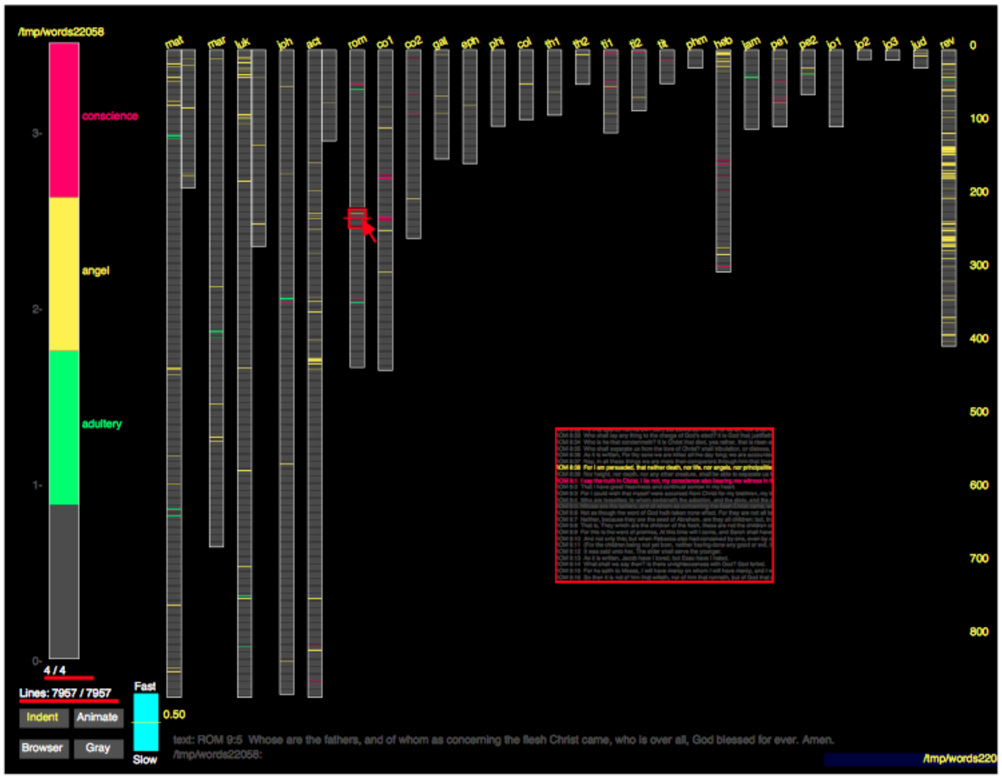
SeeSoft, SeeLog, and Information Mural

SeeSoft was proposed by Eick et al in 1992, originally as a way to visualize thousands of lines of computer code at once. Each line of code was represented as a thin row with the color of that row indicating some statistic. Selecting a portion of this overview presented the text at that location in readable form in a separate window. A later paper, Eick 1994, showed additional implementations in color, including one on natural language text from the New Testament (see figures 2 and 3).



```
Code:   tp=ntp; tz=ntz;  /* save old time values */
/tmp/type2002: symbol
```

Figure 2. A SeeSoft visualization of the SeeSoft code.



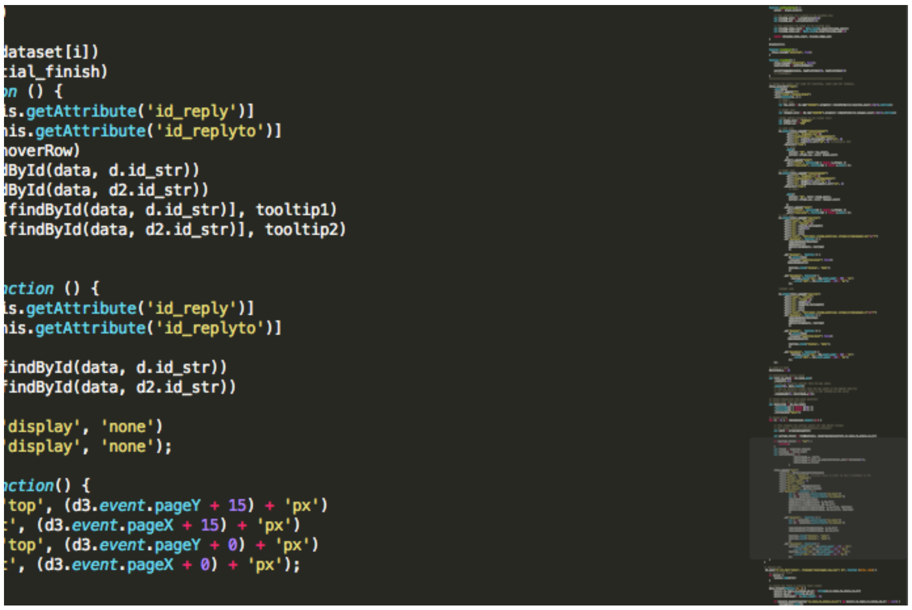
ROM 8:38 For I am persuaded, that neither death, nor life, nor angels, nor principalitie

ROM 9:1 I say the truth in Christ, I lie not, my conscience also bearing me witness in t

text: ROM 9:5 Whose are the fathers, and of whom as concerning the flesh Christ came, who is over all, God blessed for ever. Amen.
/tmp/words22058:

Figure 3. The New Testament.

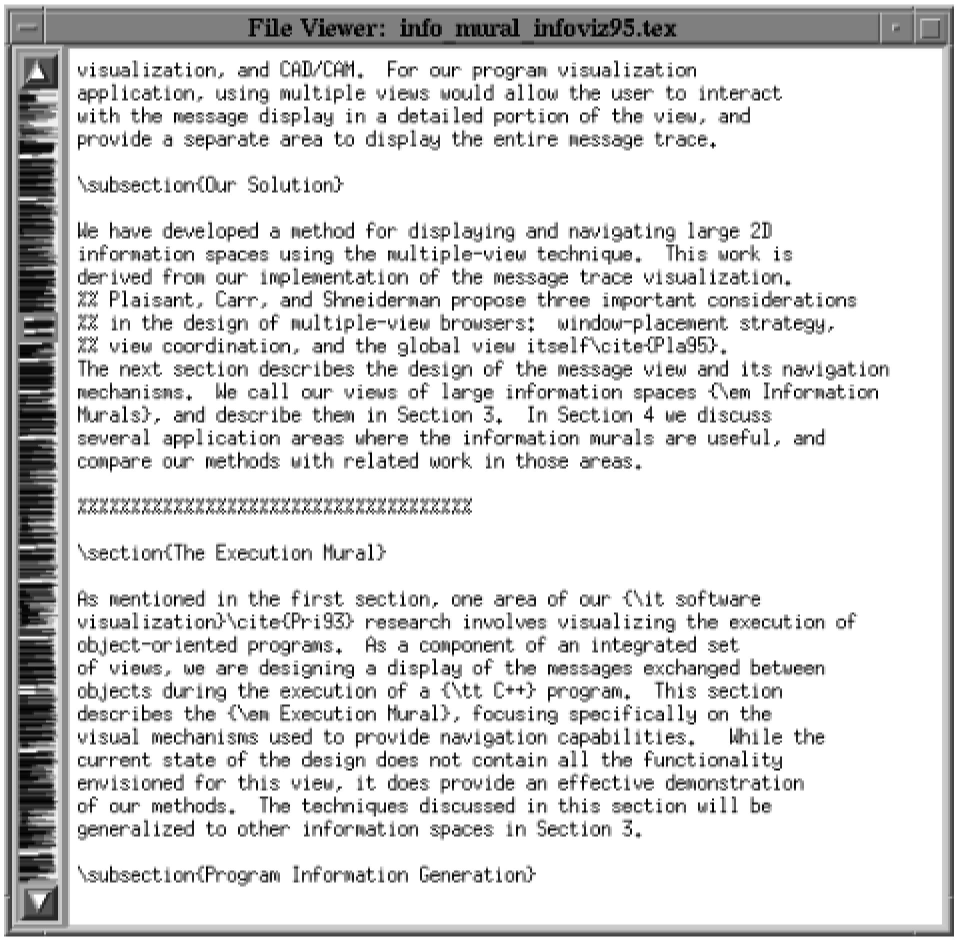
TwitArc, discussed below, is an adaption of the SeeSoft technique. Sublime Text, the text editor, has an overview window that is also similar to SeeSoft, including line highlighting based on the colors the code is represented as in the primary view (see figure 4).



```
dataset[i])
tial_finish)
on () {
is.getAttribute('id_reply')]
his.getAttribute('id_replyto')]
hoverRow)
dById(data, d.id_str))
dById(data, d2.id_str))
[findById(data, d.id_str)], tooltip1)
[findById(data, d2.id_str)], tooltip2)


ction () {
is.getAttribute('id_reply')]
his.getAttribute('id_replyto')]

findById(data, d.id_str))
findById(data, d2.id_str))

'display', 'none')
'display', 'none');

ction() {
'top', (d3.event.pageY + 15) + 'px')
', (d3.event.pageX + 15) + 'px')
'top', (d3.event.pageY + 0) + 'px')
', (d3.event.pageX + 0) + 'px');
```

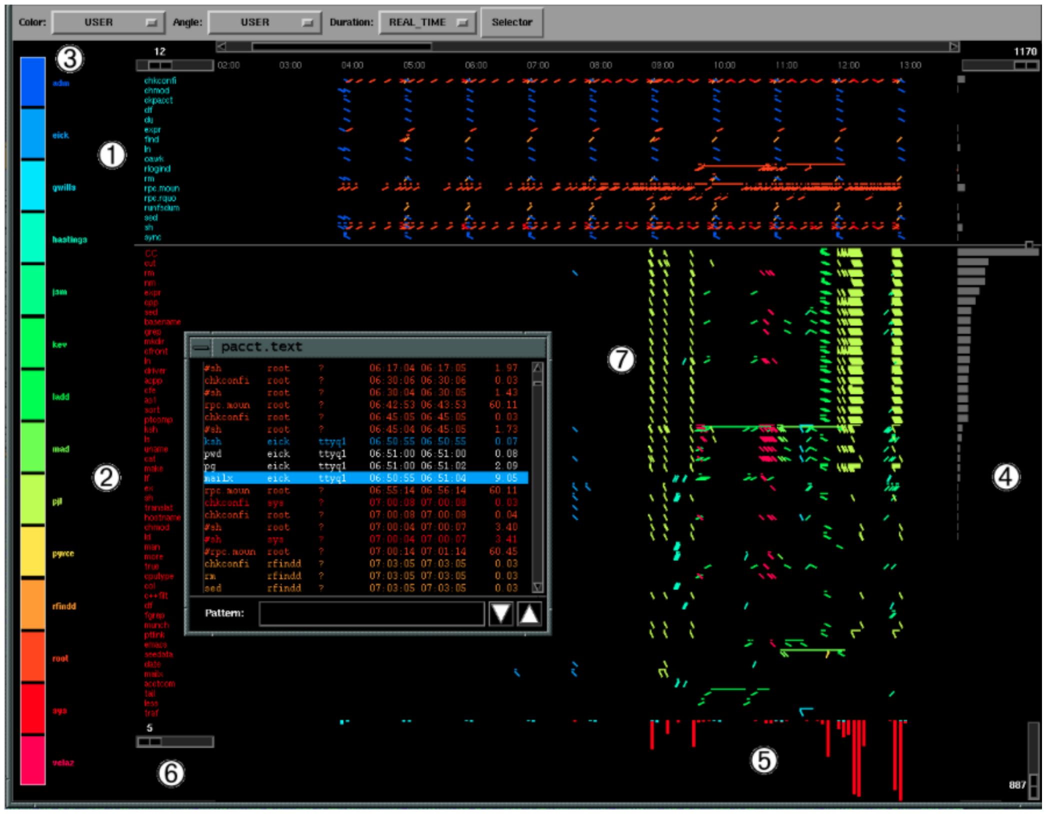
Figure 4. The Sublime text editor overview field is seen on the left. The highlighted portion is that currently in the working frame.

Following SeeSoft, SeeLog from Eick and Lucas was designed to view much larger text files, trace files. The SeeLog authors critique SeeSoft for its current implementation not displaying temporal data. As they wrote, “equal increments in text do not correspond to equal increments in time.” (Eick and Lucas 1996) Information Mural, from Jerding and Stasko, extended the SeeSoft idea to show the entirety of an information source in the visual space at one time. This was done to reduce information loss from abstraction or summarization (see figure 5). Jerding and Stasko critiqued SeeSoft for an inability to display all information at one time - because “SeeSoft requires one row of pixels for every line in the file.” (Jerding and Stasko 1998) Both of these critiques, the lack of design for temporal distribution of data and the inability to display all data in one viewing field were considerations we actively considered. We did not be using the highly abstracted marks of SeeLog nor the algorithmic scaling of Information Mural (for SeeLog’s abstracted marks, see figure 6).



```
visualization, and CAD/CAM.  For our program visualization
application, using multiple views would allow the user to interact
with the message display in a detailed portion of the view, and
provide a separate area to display the entire message trace.

\subsection{Our Solution}

We have developed a method for displaying and navigating large 2D
information spaces using the multiple-view technique.  This work is
derived from our implementation of the message trace visualization.
%% Plaisant, Carr, and Shneiderman propose three important considerations
%% in the design of multiple-view browsers:  window-placement strategy,
%% view coordination, and the global view itself\cite{Pla95}.
The next section describes the design of the message view and its navigation
mechanisms.  We call our views of large information spaces {\em Information
Murals}, and describe them in Section 3.  In Section 4 we discuss
several application areas where the information murals are useful, and
compare our methods with related work in those areas.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\section{The Execution Mural}

As mentioned in the first section, one area of our {\it software
visualization}\cite{Pri93} research involves visualizing the execution of
object-oriented programs.  As a component of an integrated set
of views, we are designing a display of the messages exchanged between
objects during the execution of a {\tt C++} program.  This section
describes the {\em Execution Mural}, focusing specifically on the
visual mechanisms used to provide navigation capabilities.   While the
current state of the design does not contain all the functionality
envisioned for this view, it does provide an effective demonstration
of our methods.  The techniques discussed in this section will be
generalized to other information spaces in Section 3.

\subsection{Program Information Generation}
```

Figure 5. This is an example from Jerding and Stasko showing the entirety of a file depicted by lines on the left of the screen.


Color: USER Angle: USER Duration: REAL_TIME Selector

```
pacct.text
#sh        root    ?      06:17:04 06:17:05   1.97
chkconfi   root    ?      06:30:06 06:30:06   0.03
#sh        root    ?      06:30:04 06:30:05   1.43
rpc.moun   root    ?      06:42:53 06:43:53  60.11
chkconfi   root    ?      06:45:05 06:45:05   0.03
#sh        root    ?      06:45:04 06:45:05   1.73
ksh        eick    ttyq1  06:50:55 06:50:55   0.07
pwd        eick    ttyq1  06:51:00 06:51:00   0.08
pg         eick    ttyq1  06:51:00 06:51:02   2.09
mailx      eick    ttyq1  06:50:55 06:51:04   9.05
rpc.moun   root    ?      06:55:14 06:56:14  60.11
chkconfi   sys     ?      07:00:08 07:00:08   0.03
chkconfi   root    ?      07:00:08 07:00:08   0.04
#sh        root    ?      07:00:04 07:00:07   3.40
#sh        sys     ?      07:00:04 07:00:07   3.41
#rpc.moun  root    ?      07:00:14 07:01:14  60.45
chkconfi   rfindd  ?      07:03:05 07:03:05   0.03
rm         rfindd  ?      07:03:05 07:03:05   0.03
sed        rfindd  ?      07:03:05 07:03:05   0.03
Pattern:
```

Figure 6. The SeeLog view has highly abstracted marks for a visualization useful only to trained expert users.

Eick, Stephen G. "Graphically displaying text." Journal of Computational and Graphical Statistics 3.2 (1994): 127-142. (link)

Eick, Stephen G., Joseph L. Steffen, and Eric E. Sumner Jr. "Seesoft-a tool for visualizing line oriented software statistics." Software Engineering, IEEE Transactions on 18.11 (1992): 957-968. (link)

Eick, Stephen G., and Paul J. Lucas. "Displaying trace files." Softw., Pract. Exper. 26.4 (1996): 399-409. (link)

Jerding, Dean F., and John T. Stasko. "The information mural: A technique for displaying and navigating large information spaces." Visualization and Computer Graphics, IEEE Transactions on 4.3 (1998): 257-271. (link)

TreeMaps
The original 1991 paper on Treemaps discusses a novel way to display hierarchical information that would be able to maximize the area used to display this data (figure 7). An updated paper discussed improvements in order to better display objects such as images. The algorithms discussed also show ways to group images and ensure that neighboring objects are kept in tact. The hierarchies discussed show innate attributes such as size of the

company in relation to the overall stock market capitalization. Another dimension of data can be applied using opposing colors, showing whether the company's stock traded up or down for that day (see figure 8).

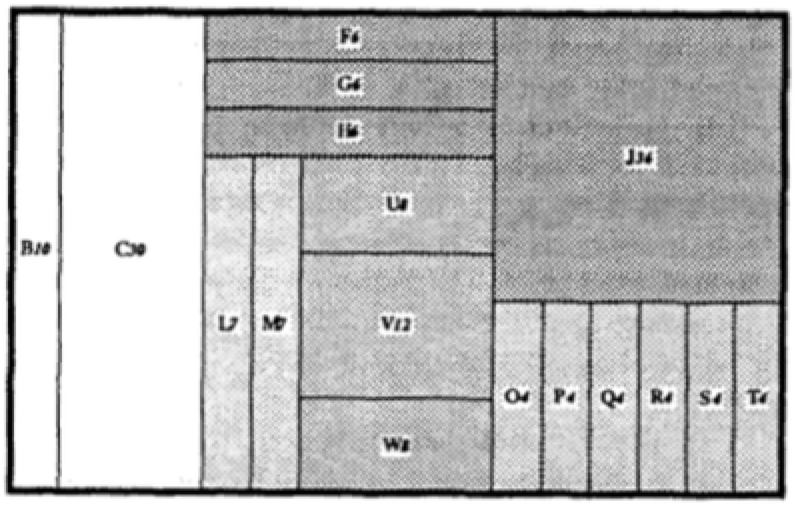


Figure 7. Original treemap.

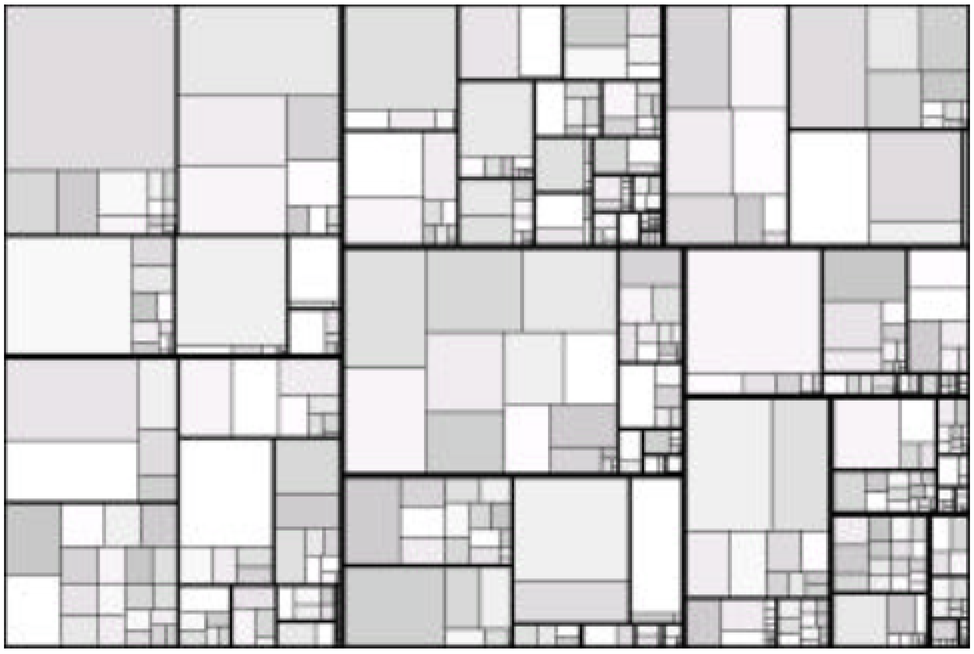


Figure 8. Nested treemap with colored grids.

Johnson, Brian, and Ben Shneiderman. "Tree-maps: A space-filling approach to the visualization of hierarchical information structures." Visualization, 1991. Visualization'91, Proceedings., IEEE Conference on. IEEE, 1991. (link)

Bederson, B.B., Shneiderman, B., and Wattenberg, M.

Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies ACM Transactions on Graphics (TOG), 21, (4), October 2002, 833-854. (link)

Social Media Visualizations in Research: Streams, Events, and Networks

Much of the published research on Twitter visualizations focuses on looking at live high-rate streams of data, event exploration and interpretation, and the networks of followers, replies, or mentions. These tools are often focused on teasing out topic models, sentiment analysis, identifying emerging trends, or influential clusters within networks. All of these are outside the problem space of our project. For instance, Diakopoulos et al. developed a workspace, Vox Civitas, for journalists to view tweets related to particular events - including an ability to view the video of an even conjointly with the tweets that appeared at that time and with sentiment analysis tools. TwitInfo, from Marcus et al., also looked at tweets surrounding events, providing visualizations of tweet frequency, geographic dispersion, and a basic pie chart of sentiment. Their main emphasis, though, was “a novel algorithm for peak detection and labeling” and not their visualization.

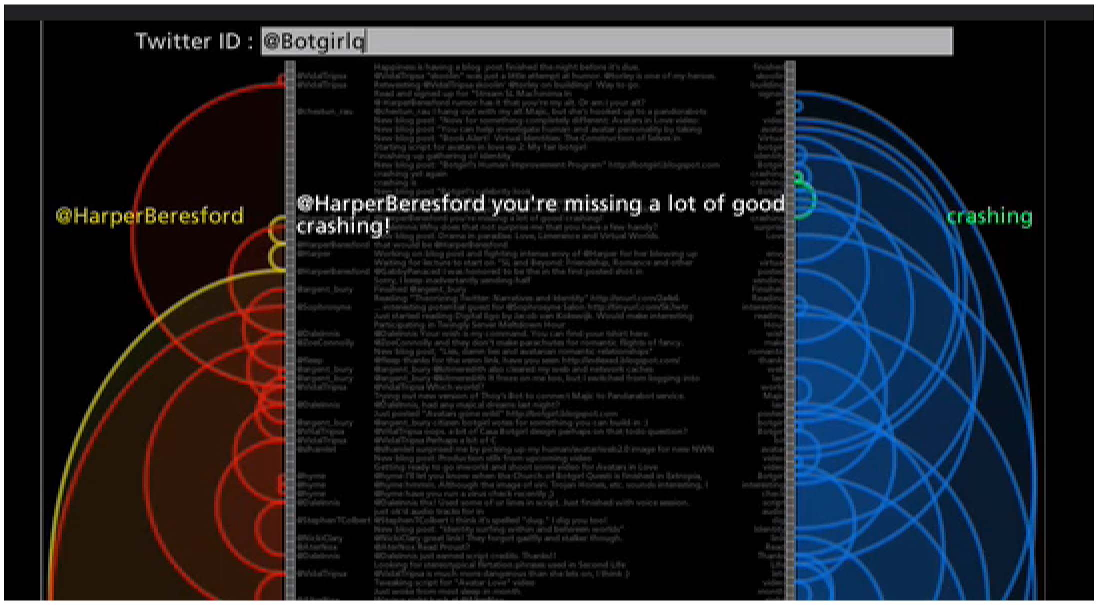
Diakopoulos, Nicholas, Mor Naaman, and Funda Kivran-Swaine. "Diamonds in the rough: Social media visual analytics for journalistic inquiry." Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on. IEEE, 2010.

Marcus, Adam, et al. "Twitinfo: aggregating and visualizing microblogs for event exploration." Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 2011.

Related Twitter Visualizations

TwitArcs

“I've combined some visuals from a side project related to linguistics with Twitter data to create TwitArcs. It takes the latest 100 tweets for a Twitter ID or term of interest and creates a list representation that has arcs connecting messages sent to the same users or that use the same primary term. You can click on the left side to load the tweets for a new user, on the right side to load the tweets for a specific term, and in the middle to visit the actual tweet.” (Figure 9; Neoformix link; Flickr link (video))

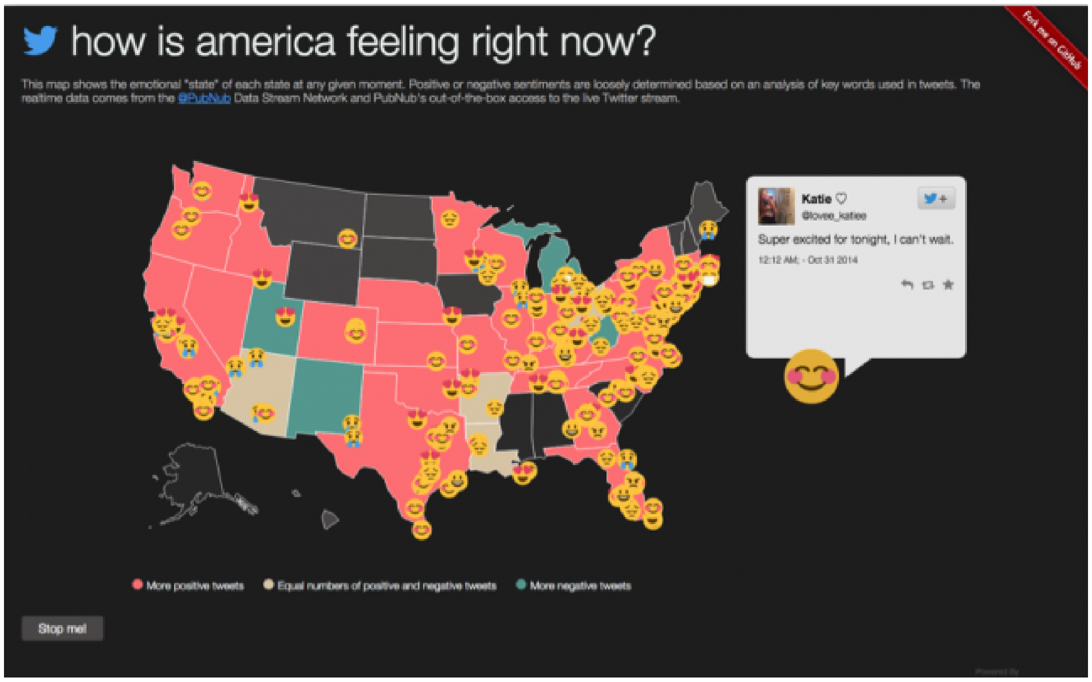


@HarperBeresford you're missing a lot of good crashing!

Figure 9. A screenshot of a video of TwitArcs in use.

Tweet Emoticon

This map shows the emotional "state" of each state at any given moment. Positive or negative sentiments are loosely determined based on an analysis of key words used in tweets. The realtime data comes from the @PubNub Data Stream Network and PubNub's out-of-the-box access to the live Twitter stream (figure 10).
(Twitter link)



how is america feeling right now?

This map shows the emotional "state" of each state at any given moment. Positive or negative sentiments are loosely determined based on an analysis of key words used in tweets. The realtime data comes from the @PubNub Data Stream Network and PubNub's out-of-the-box access to the live Twitter stream.

Katie ♡
@lovee_katiee
Super excited for tonight, I can't wait.
12:12 AM - Oct 31 2014

Figure 10.

Recap of Personal Tweets

This is a recap visualization from Twitter that allowed a user to login and view their personal year of tweets in a summarized visualization. While the visualization was attempting to do the same thing that we are attempting to do, it fell short due to the fact that it fails to provide any context to the topics that were discussed. There is very little value to knowing that you tweeted 34 times about data without having some way of seeing what discussions that sparked. This emphasizes the importance of *how* we provide context of tweets to users. This visualization takes very little advantage of interactive animations which may add some value in highlighting relevant tweets.
(FlowingData link; AdWeek link

Analogies to Exploratory Data Analysis

There are some parallels between our project and EDA: attempting to identify meaningful information from data using various exploration tools and techniques. Tukey emphasized that, in EDA, “finding the [right] question is often more important than finding the answer”. With our visualization, we acknowledged that we may not be able to consistently answer the question of what happened over a period of time in a user’s Twitter feed with predefined algorithms. Instead, our focus was on helping users discover the right question about what happened in the previous day by providing additional ways for them to interact with their Twitter timeline data.

Tukey, John W. "We need both exploratory and confirmatory." The American Statistician 34.1 (1980): 23-25. (link)

Related Work	Strengths	Weaknesses
SeeSoft	overview with details on demand, marking methods	is not designed to deal with temporal data
Table Lens	brushing/linking and EDA foundation	optimized for analysis of quantitative data
TwitArcs	essentially an implementation of SeeSoft but including more sophisticated interaction showing relationships	Current implementation is for one person's tweets. We have to optimize for a day's conversation for all followers
Information Mural	designed to display all information on the screen with no information loss, avoiding	algorithmic complexity
TreeMap	Quick view of a hierarchy of information. Also allows for interaction to highlight certain areas of the map	It may hide small pockets of information and weight things that we don't necessarily want to display.
Revisit - a Twitter wall visualization	Topic-based. Conveys temporal dynamics by emphasizing conversational threads established by retweets and @replies.	Importance of a tweet may be based on quantitative metrics. Clutter likely because of the potentially large amount of tweets.
EDA Tools	Allows users to understand the data and ask their own questions without predefining them (like with event exploration).	More difficult to do with text data than numerical
Visual Recap	Highlights topics well Showcases interesting tweets (Golden Tweet)	A bit static and does not allow much exploration context missing from topics, difficult to understand meaning of the tweets
Tweet Emotion	Good use of iconography and coloring to clearly display overview of emotions	Doesn't allow for fine levels of details based on city, county, etc. No animations to better portray a flow of information in real time

Table 1. Summarizing Strengths and Weaknesses

Description of TweetDay

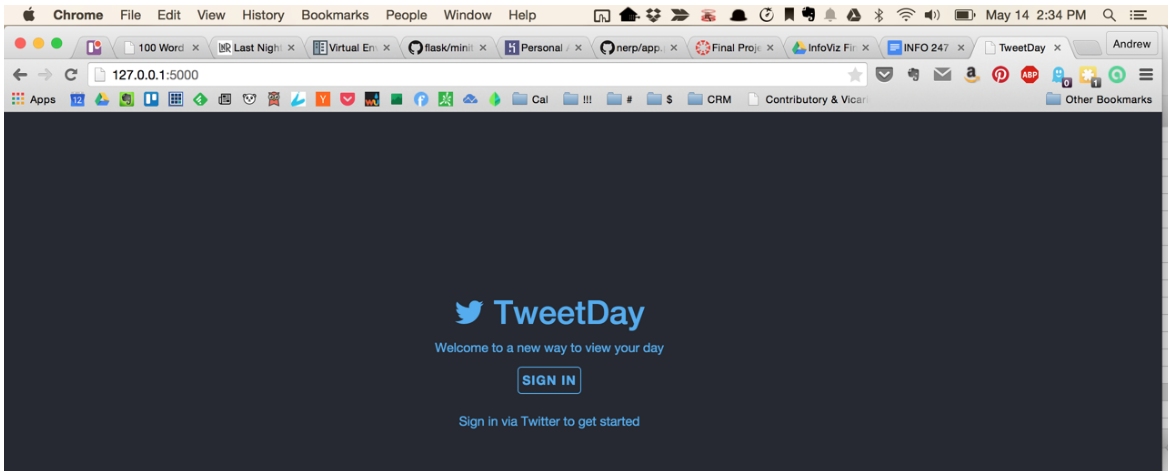

TweetDay
Welcome to a new way to view your day
SIGN IN
Sign in via Twitter to get started

Figure 11. Login screen.

Upon logging in the user is presented a screen divided into two main fields. The SeeSoft-esque view is on the left and the treemap is on the right.

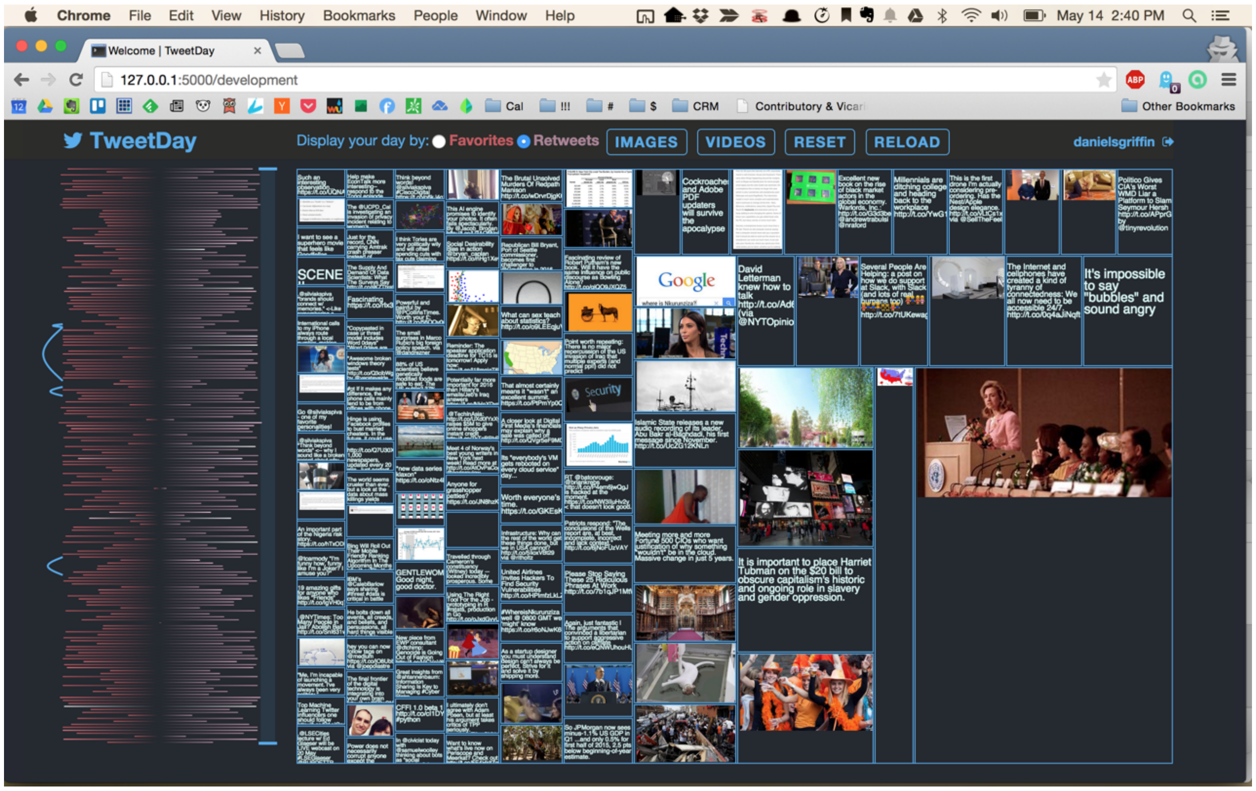
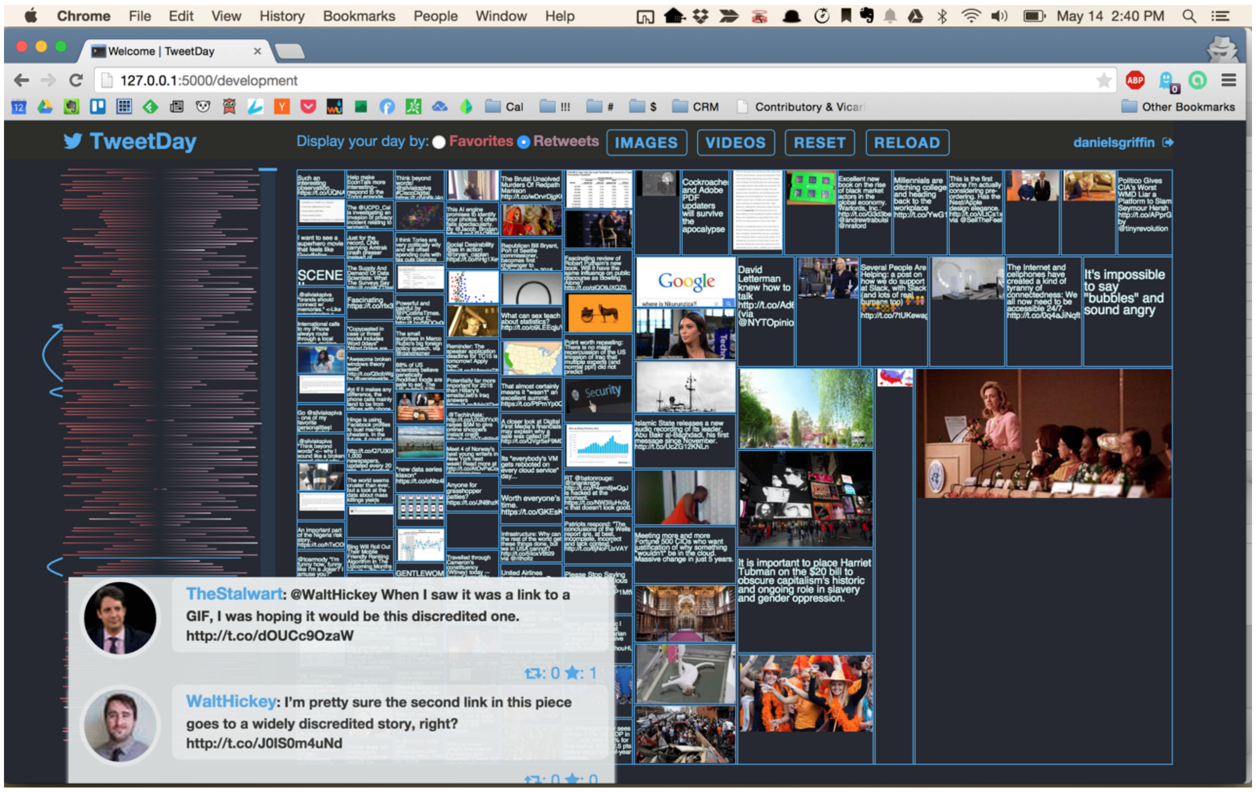


Figure 12. SeeSoft field on the left, treemap on the right.

The SeeSoft field displays a line per tweet, the length of the line proportional to the number of characters in each tweet. The line is centered on the middle, creating a mirroring effect. The opposites sides of the line are colored by the number of favorites or retweets of that tweet. The brightness of the color indicates a greater proportional number of favorites or retweets (white being the most). There is an added benefit in displaying both metrics in the same space as sometimes a tweet with have far more favorites than retweets (or the reverse) and that would be quickly signaled to the user in our design. The color disappears as approaching the center with a linear gradient. Hovering over a line brings up a tooltip showing the contents of the tweet and highlights the line (see figure 24). Scrolling through the lines with the j and k keys moves up and down from the latest line viewed and presents a tooltip showing the tweet (see figure 25).

To the left of the lines are blue curves linking together a tweet that is a reply to the tweet to which it is in reply. Hovering over a curve brings up a tooltip showing the two tweets and highlights the lines representing those tweets.



Chrome File Edit View History Bookmarks People Window Help
May 14 2:40 PM
Welcome | TweetDay
127.0.0.1:5000/development
TweetDay
Display your day by: Favorites Retweets
IMAGES
VIDEOS
RESET
RELOAD
danielsgriffin

TheStalwart: @WaltHickey When I saw it was a link to a GIF, I was hoping it would be this discredited one. http://t.co/dOUCc9OzaW

⮂: 0 ★: 1

WaltHickey: I'm pretty sure the second link in this piece goes to a widely discredited story, right? http://t.co/J0lSOm4uNd

Figure 13. Hovering over a curve displays the pair of tweets - reply and replied to.

To the right of the lines is a brush tool that allows the user to change the selection of tweets. It initially loads at the fullest extent. Changing the selection updates the tweets visible in the treemap.

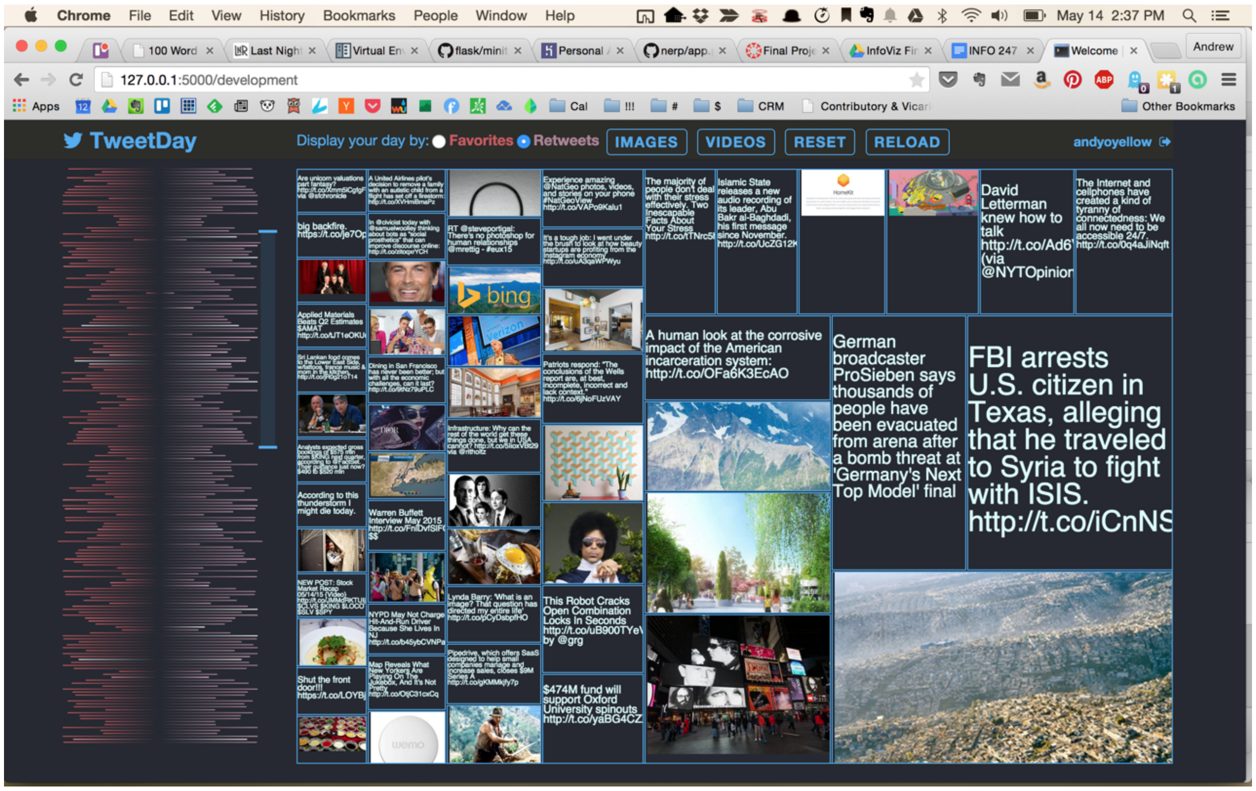
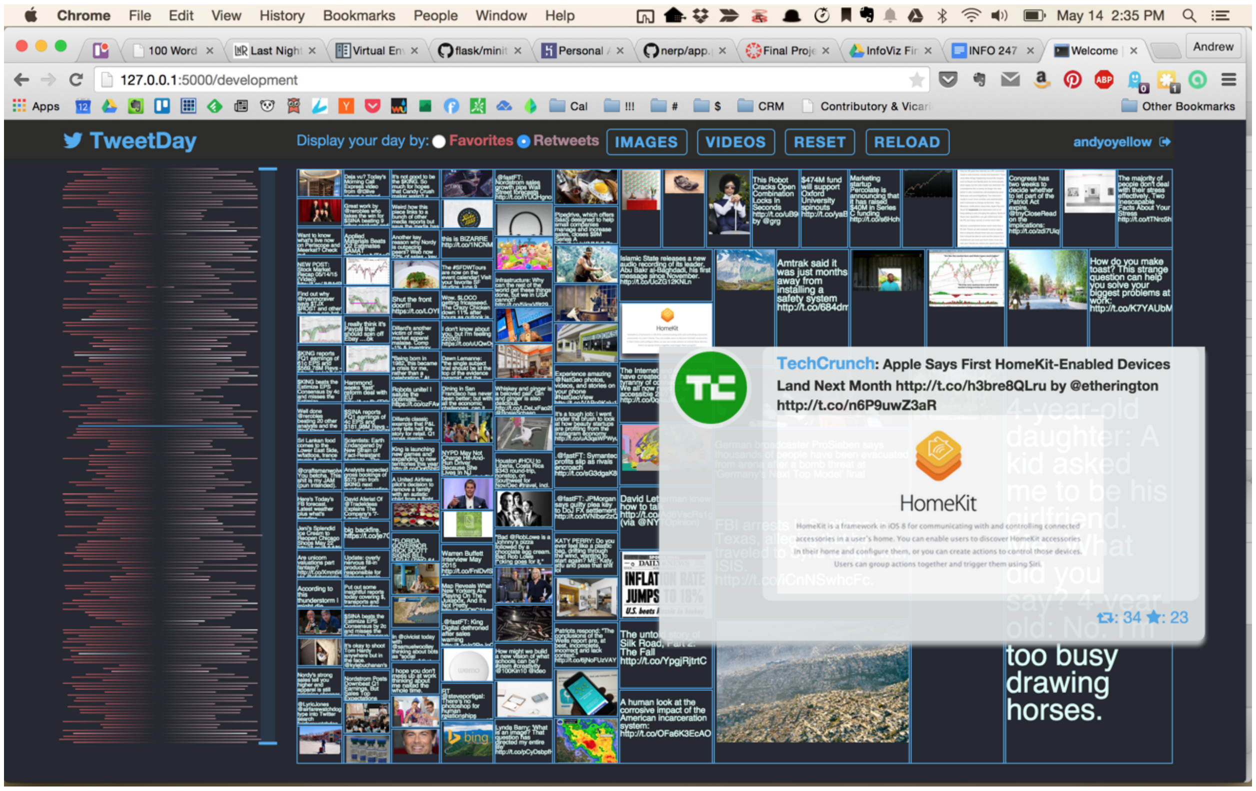


Figure 14. Note the brush tool used to reduce the tweets visible in the treemap.

The treemap field displays the tweets in boxes. The sizes of the boxes are determined by the number of favorites or retweets for that tweet (the metric can be toggled at the top). If the tweet has a media attached (a picture or video), the media is displayed in place of any text. If the media is a video it can be played from within the treemap by right-clicking on the image. Hovering over a box brings up a tooltip showing the contents of the tweet and highlights the line representing that tweet in the SeeSoft field. This highlighting grants the user the benefit of seeing the larger context within which that tweet was made. As noted above, the brush links with the treemap to update the field of available tweets represented. A user can also select to only see pictures or only see images in the treemap.



TechCrunch: Apple Says First HomeKit-Enabled Devices Land Next Month http://t.co/h3bre8QLru by @etherington http://t.co/n6P9uwZ3aR

Figure 15. The image is hovered over, bringing up the tooltip displaying the image and name of the account making the tweet, the full text of the tweet along with a larger view of the image, and the number of favorites and retweets while the location of the tweet is highlighted in blue in the left field.

Tools

We used Twitter's REST API to initially pull data from Daniel's newsfeed and used a static JSON file to build our preliminary visualization. After we had made progress on the visualization, we decided that we would get better feedback from users if we were able to let them use their own data in the visualization. Therefore, we implemented a simple Python Flask server that used the OAuth library to access Twitter's REST API to start pulling data dynamically. We initially attempted to make the application available through Amazon's EC2 servers, however, decided that Heroku offered a better solution. The application is currently hosted on Heroku and is capable of running smoothly on a very small server because most of the work is done on the client side.

Some initial design work was done in Illustrator to create visual mockups to better represent the our hybrid combination of SeeSoft, TwitArcs, and a TreeMap. Images, video screen shots, and actual tweets were copied into the architecture to a give a sense of how the actual application would look and feel like.

Our primary tool for creating the visualization was D3.js. D3 has a built-in functionality to create treemaps and svg lines for the SeeSoft implementation. Additionally, the fact that it was built with Javascript gave us the flexibility to create the type of interactions that we needed that tools like Tableau or Highcharts were capable of providing. This was particularly the case with the Bézier curves, which are a reappropriation of D3 functionality that we were able to use for the TwitArcs. Since there were four of us working on the same code at the same time, we decided to use Git/Github to manage our application. This created some overhead in terms of merging and managing different branches, however, the ultimate benefit of Git was evident in our efficiency in our ability to make changes to the application.

Design and Development Process

1. Initial Designs

Our very first designs attempted to present tweets to users on a scatter plot. This involved using most of the design space to represent only two variables, time and individual, while attempting to put too much information (number of retweets or favorites, media included, signifier of being a retweet, etc.) into the mark of the tweet itself.

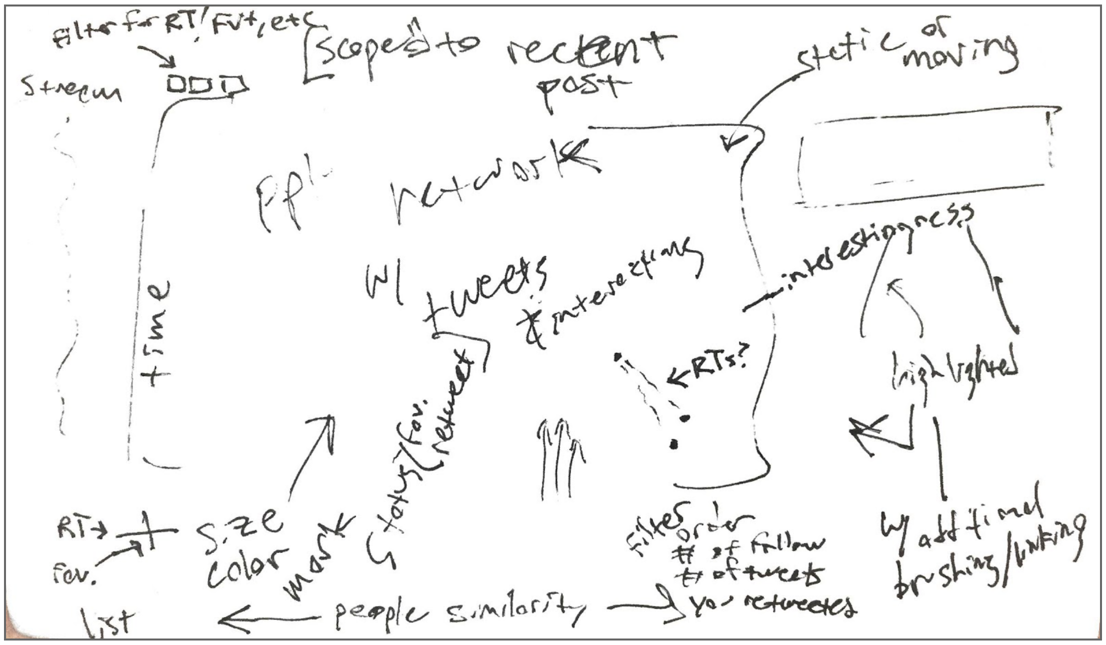
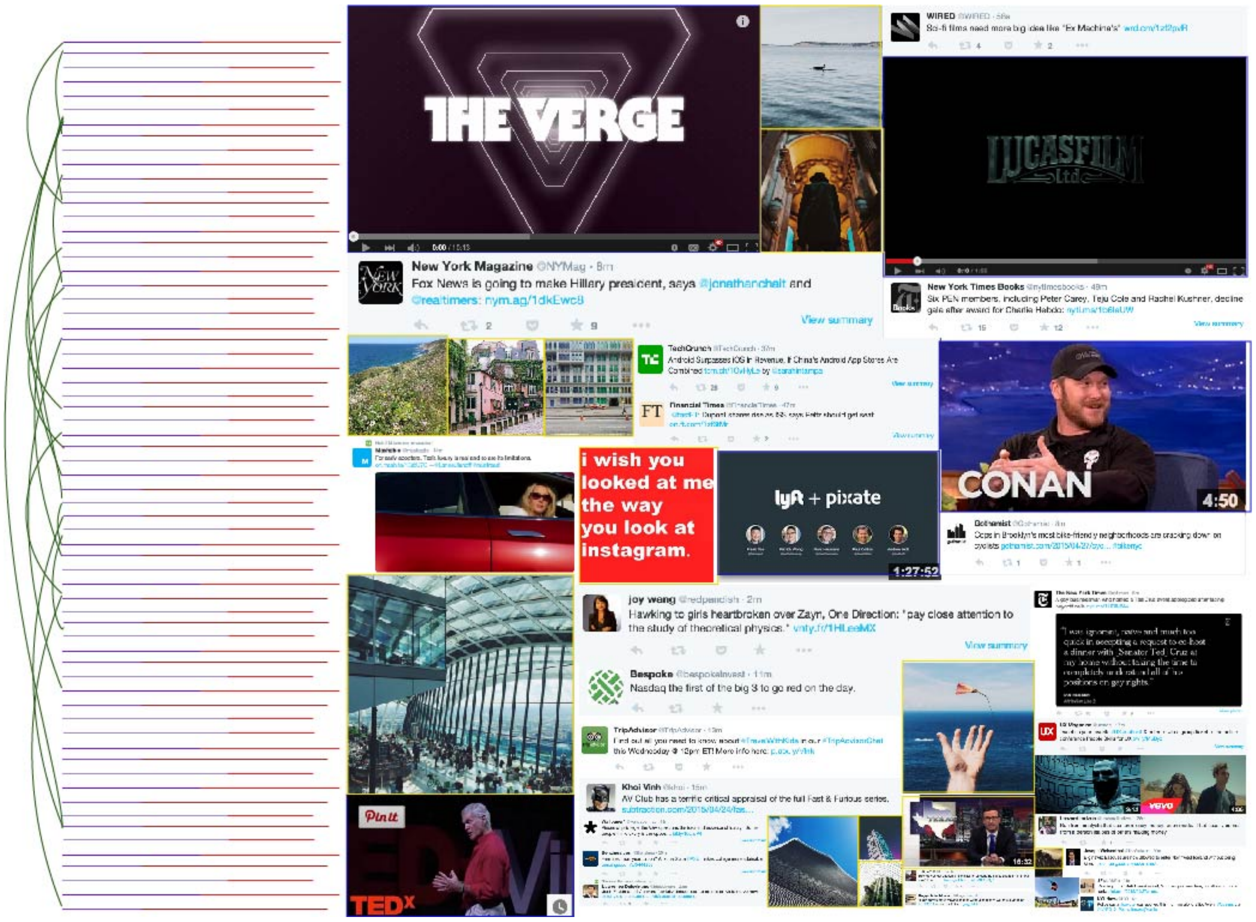


Figure 16. Our first design attempt on an index card.

Initial usability testing was quick and cheap — random twitter users in the school — were asked to interpret the design. Feedback from target users helped focus our designs on providing clear context and ready access to content. We did not find any “brick walls” — problems so serious that users cannot get past them. Anyone but project designers and engineers can be used (they tend to act as "expert reviewers" because they are too close to the project).Our next designs incorporated the benefits SeeSoft brought to visualizing text and treemap brought to using the whole design space to display content. These designs were also heavily motivated by TwitArcs.



New York Magazine @NYMag · 8m
Fox News is going to make Hillary president, says @jonathanchait and @realtimers: nym.ag/1dkEwc8

joy wang @redpandish · 2m
Hawking to girls heartbroken over Zayn, One Direction: "pay close attention to the study of theoretical physics." vnty.fr/1HLeeMX

Bespoke @bespokeinvest · 11m
Nasdaq the first of the big 3 to go red on the day.

Khoi Vinh @khoi · 15m
AV Club has a terrific critical appraisal of the full Fast & Furious series. subtraction.com/2015/04/24/fas...

Figure 17. The final design before development began. See also figures 19 and 20.

2. Static data and reviewed

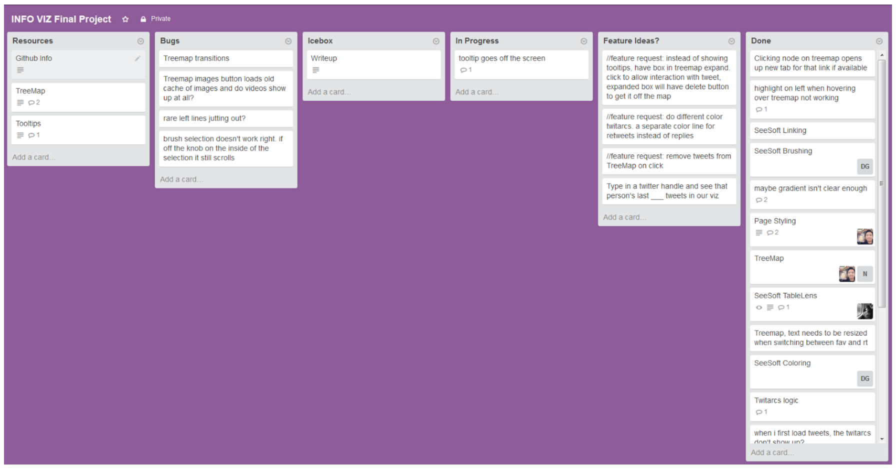
In order to validate that we could get the type of information that we needed from the Twitter API, we created a script to pull data from the Twitter API and save it as a static JSON file. We had to make some modifications to our script to account for some type changes, however, we were able to find almost all of the data that we needed for our visualization in the retrieved data.

3. Base Canvas

To have a similar starting point for our application, we decided to create some base code to put all of our visualizations in. This would allow us to work on the different aspects of the visualization separately, but, still be able to integrate the various components into the master layout file. To do this, we created the directory structure and a set of html and css files, along with the supporting JS and D3.js files. Within the HTML file, we created two separate SVG elements, one for SeeSoft and the other for the treemap. We styled these elements to have the desired width and height. We then added some small D3 code snippets that we know we would have to use, such as the code to read in the JSON and drawing some basic SVG elements on the page. This initial code base was shared with git/Github, which allowed us to work separately on different aspects as described in the following section.

4. Task Management and Process

We used Trello to coordinate tasks between us and to help us keep track of who was working on a particular feature so that we did not duplicate each other's work (see figure 18). Daniel and Vijay worked primarily on the SeeSoft and Béizer line implementations while Andy and Nikhil created the Treemap (see initial implementations in figure 21). Each feature was broken into smaller, more manageable tasks, so that none of us were overwhelmed with a given task. For example, on the SeeSoft implementation, we started initially with drawing a line for each tweet, then focused on making the line length equivalent to the length of the tweet, and then worked on creating color gradients for each of the lines. This process of breaking problems into smaller problems was extremely useful for us since none of us had any tangible experience with using Javascript at this scale. Additionally, we created separate feature branches on Git so that we could separate concerns and only merged to our master branch once we verified that the feature was working and was tested.


INFO VIZ Final Project Private

Resources
- Github Info
- TreeMap 2
- Tooltips 1
- Add a card...

Bugs
- Treemap transitions
- Treemap images button loads old cache of images and do videos show up at all?
- rare left lines jutting out?
- brush selection doesn't work right. if off the knob on the inside of the selection it still scrolls
- Add a card...

Icebox
- Writeup
- Add a card...

In Progress
- tooltip goes off the screen 1
- Add a card...

Feature Ideas?
- //feature request: instead of showing tooltips, have box in treemap expand. click to allow interaction with tweet, expanded box will have delete button to get it off the map
- //feature request: do different color twitarcs. a separate color line for retweets instead of replies
- //feature request: remove tweets from TreeMap on click
- Type in a twitter handle and see that person's last ___ tweets in our viz
- Add a card...

Done
- Clicking node on treemap opens up new tab for that link if available
- highlight on left when hovering over treemap not working 1
- SeeSoft Linking
- SeeSoft Brushing DG
- maybe gradient isn't clear enough 2
- Page Styling 2
- TreeMap N
- SeeSoft TableLens 1
- Treemap, text needs to be resized when switching between fav and rt
- SeeSoft Coloring DG
- Twitarcs logic 1
- when i first load tweets, the twitarcs don't show up?
- Add a card...

Figure 18. The Trello dashboard we used to organize our efforts.

Results

Our goal was to create a visualization that presented a bigger picture of the tweets that a user might see during a day on Twitter. We wanted to move away from the standard Twitter timeline which presents a linear chronological view. We wanted to test if this view was useful by simply asking users if they would rather use this than the Twitter application or website. One user expressed significant frustration, decrying his concern that our visualization made him want to engage with Twitter more.

Some of the quotes from our users were

> *"This is making me want to read Twitter more."*
>
> *"This is definitely something that I would want to use."*
>
> *"I don't think anything like this exists and I would like to use it when it comes out."*

Many users recognized the problem we had identified of a fear of missing out and the difficulty of ensuring conversations and events occurring throughout the day were not missed. The visualization did provide them an overview that provided a measure of a significant period of time. Multiple users while engaging with the visualization identified tweets through the coloring of the lines or that "popped out" in the treemap and remarked they would probably not have seen the tweet or noted its importance if using the standard Twitter timeline. In addition, users noted that seeing a visual "map" of the information sources they follow on Twitter was useful to them.

The goal not achieved was enabling reply and retweet interaction from within the visualization. Enabling this would largely have required more configuration and interaction with the Twitter API. Our initial experience with the API helped us realize that a significant amount of time could be spent, if we so wished, navigating the API alone. We even had to make code changes to accurately render multimedia content due to a change in the data object. We went back to the original scope of our project and in the end we focused our efforts on the visualization itself. However, we acknowledge what users said, that the lack of reply and retweet interaction would be a hurdle in their adopting this visualization.

An observation from multiple users was that locating the tweets with the most favorites or retweets in the bottom right corner of the treemap seemed a surprising choice. Changing the construction of the treemap to locate those tweets on the top left is something we would like to test. One benefit would be keeping the user's locus of attention more towards the center of the screen, closer to the SeeSoft-esque visualization and better able to benefit from the interaction between the two main fields.

There were two observations concerning the curves indicating replies. First, many pairs of replying and replied to tweets were part of larger conversations. A further step would involve identifying when that is the case and possibly showing more than just those two tweets when a user hovered over the reply curve. Second, many of the replies were from the same person initiating the conversation and continuing it -- a form of "tweet chaining" that links thoughts together to overcome Twitter's character limit (see figure 23).

We also failed to indicate in some fashion which tweets were retweets. One option considered was different colored curves. We could also have implemented a button to toggle a highlight of all retweets or to remove all retweets from the dataset being viewed. It would perhaps be interesting to users to compare the treemap with and without retweeted tweets from those not being followed.

The amount of data and interactions being presented to the user and the various purposes to which the user may have wished to put TweetDay made it clear in the end that a larger and more structured user testing phase would be beneficial.

Appendix I. Links

i. Here is the live implementation where users can sign in and see TweetDay's visualization of the tweets from those they folow: http://dry-dusk-4832.herokuapp.com
ii. Here is an implementation with stored data: http://dry-dusk-4832.herokuapp.com/demo

Appendix II. Division of Work

The work on this project was conducted with considerable collaboration among all members. Most of the work was conducted during sustained development periods in shared settings which allowed for mutual code reviews, pair programing, and minimal friction to collaboration. The following table provides a summary of the breakdown of tasks while should not be seen as indicating a lack of contribution by any one individual. The difficulty and importance of the tasks are not equivalent.

Task	Contributor
Initial Visualization Designs	ALL
Initial pull for static data	Daniel and Vijay
GitHub Guru	Vijay
SeeSoft	Vijay and Daniel
base implementation	Vijay
linear gradient	Daniel (w/ Vijay integrating)
scaling and mirroring	Andy
brush	Daniel
highlighting on hover	Vijay
keyboard scrolling (j/k)	Daniel (w/ Vijay integrating)
TreeMap	Nikhil and Andy
base implementation	Andy
integrating Twitter data with the treemap	Nikhil
photos and videos view	Nikhil
tooltip integration	Vijay
integrating SeeSoft hover highlighting	Vijay
Reply Arcs (Bézier curves)	Daniel
highlighting on hover and tooltip	Daniel and Vijay
Overall Visual Design	Andy
Tooltip Design	Andy
Flask Implementation	Vijay
Heroku Implementation	Vijay
Final Write Up	ALL

Appendix III. Additional Screenshots

Below are more screenshots of TweetDay and its design and development.

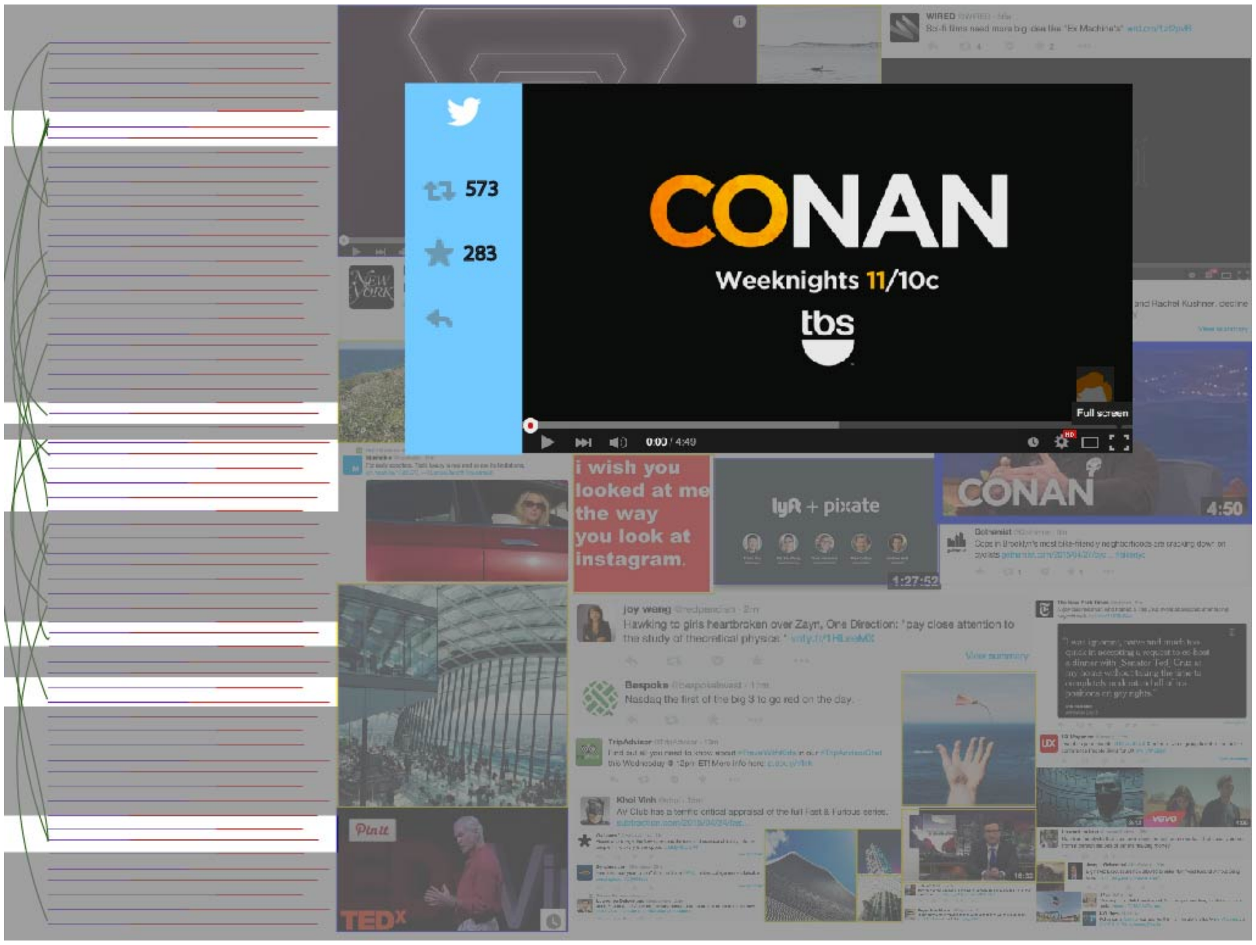
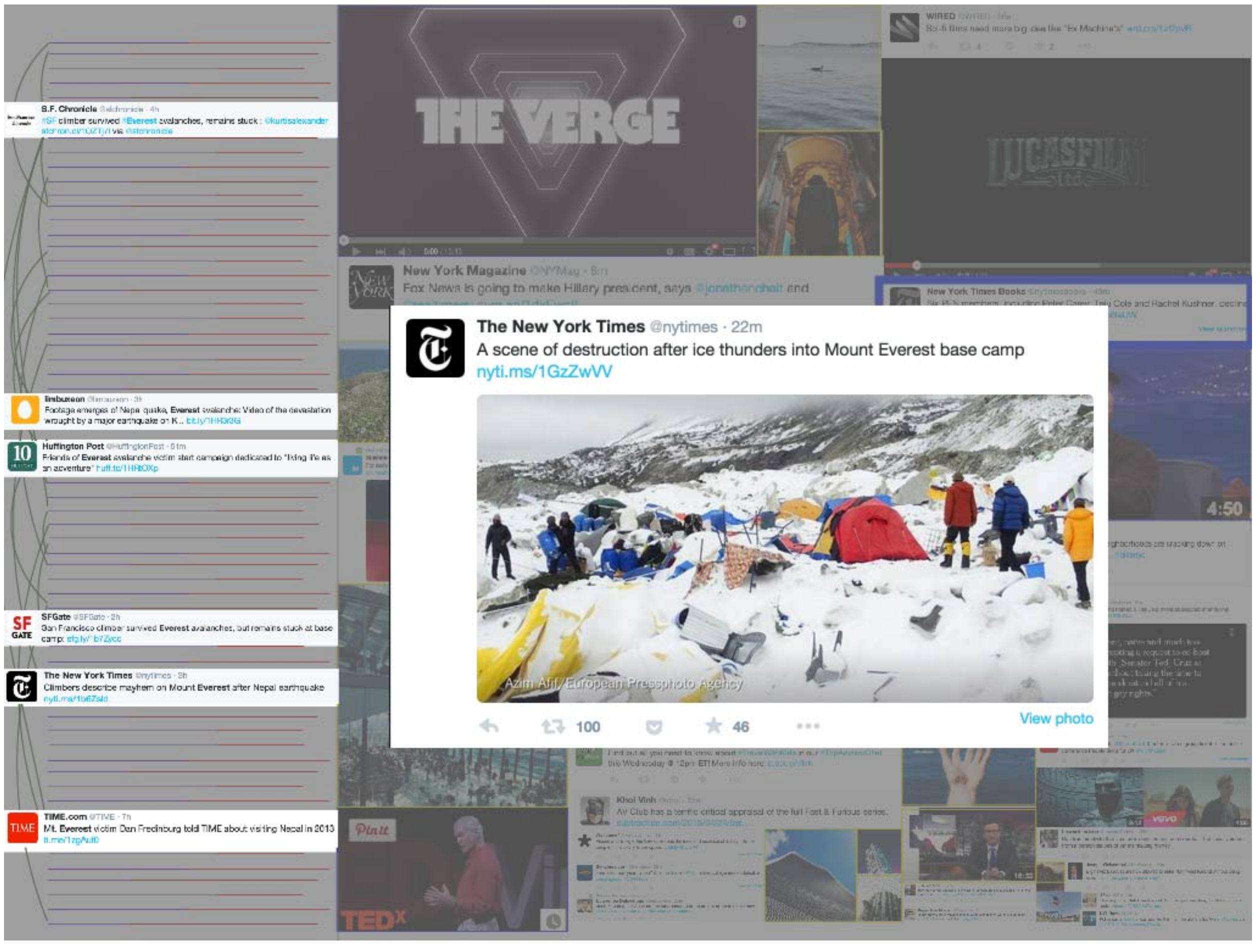


Figure 19.



S.F. Chronicle @sfchronicle · 4h
#SF climber survived #Everest avalanches, remains stuck : @kurtisalexander sfchron.cl/1OZTj/I via @sfchronicle

Imbuzeon @imbuzeon · 3h
Footage emerges of Nepal quake, Everest avalanche: Video of the devastation wrought by a major earthquake on K... bit.ly/1HR3lGG

Huffington Post @HuffingtonPost · 51m
Friends of Everest avalanche victim start campaign dedicated to "living life as an adventure" huff.to/1HRtOXp

SFGate @SFGate · 2h
San Francisco climber survived Everest avalanches, but remains stuck at base camp: sfg.ly/1b7Zycc

The New York Times @nytimes · 3h
Climbers describe mayhem on Mount Everest after Nepal earthquake nyti.ms/1b6Zsld

TIME.com @TIME · 7h
Mt. Everest victim Dan Fredinburg told TIME about visiting Nepal in 2013 ti.me/1zgAut0

The New York Times @nytimes · 22m
A scene of destruction after ice thunders into Mount Everest base camp
nyti.ms/1GzZwVV

Azim Afif/European Pressphoto Agency

100 46 View photo

New York Magazine @NYMag · 8m
Fox News is going to make Hillary president, says @jonathanchait and

Figure 20.

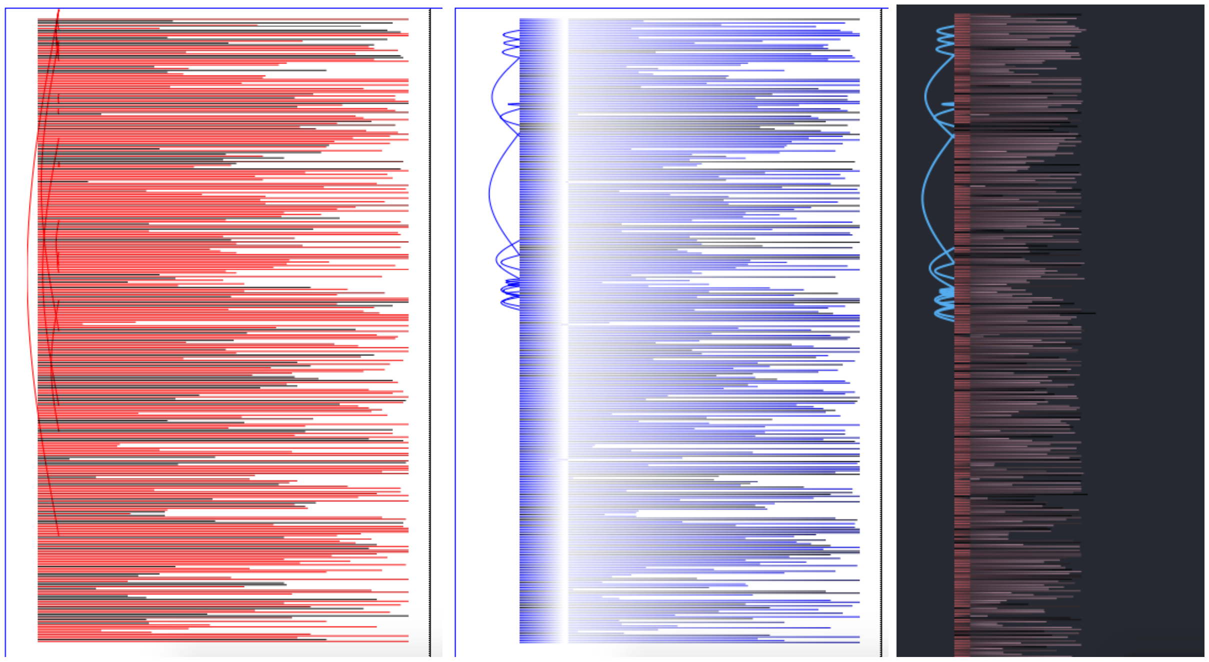


Figure 21. Initial implementations of the lines and curves.

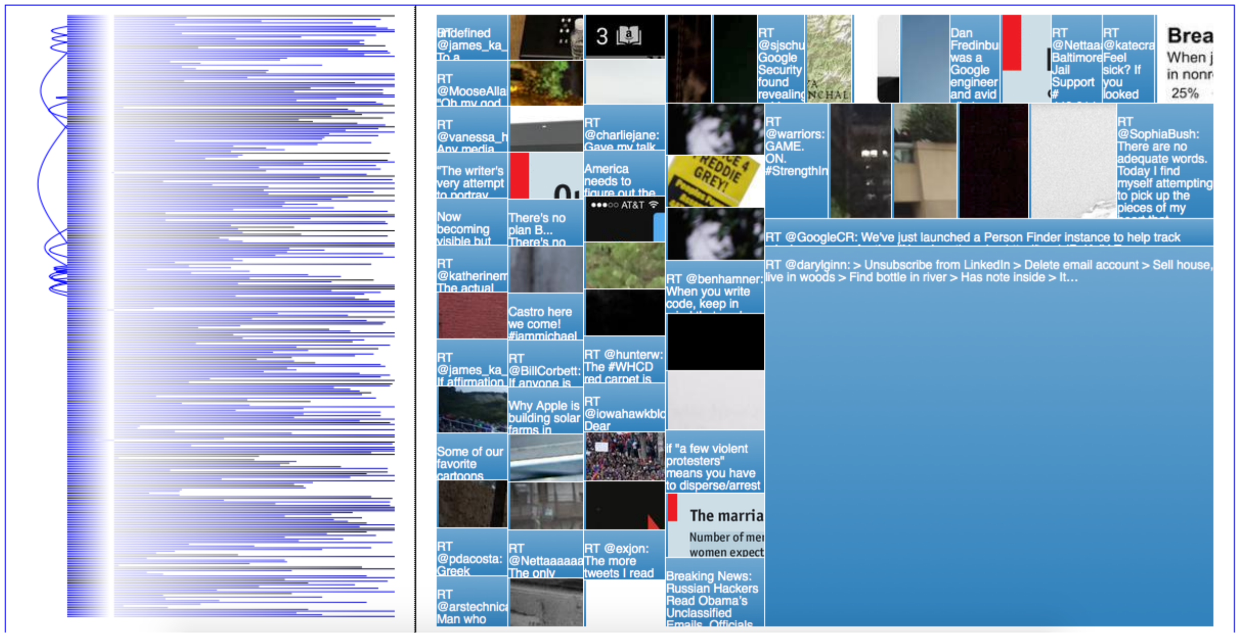
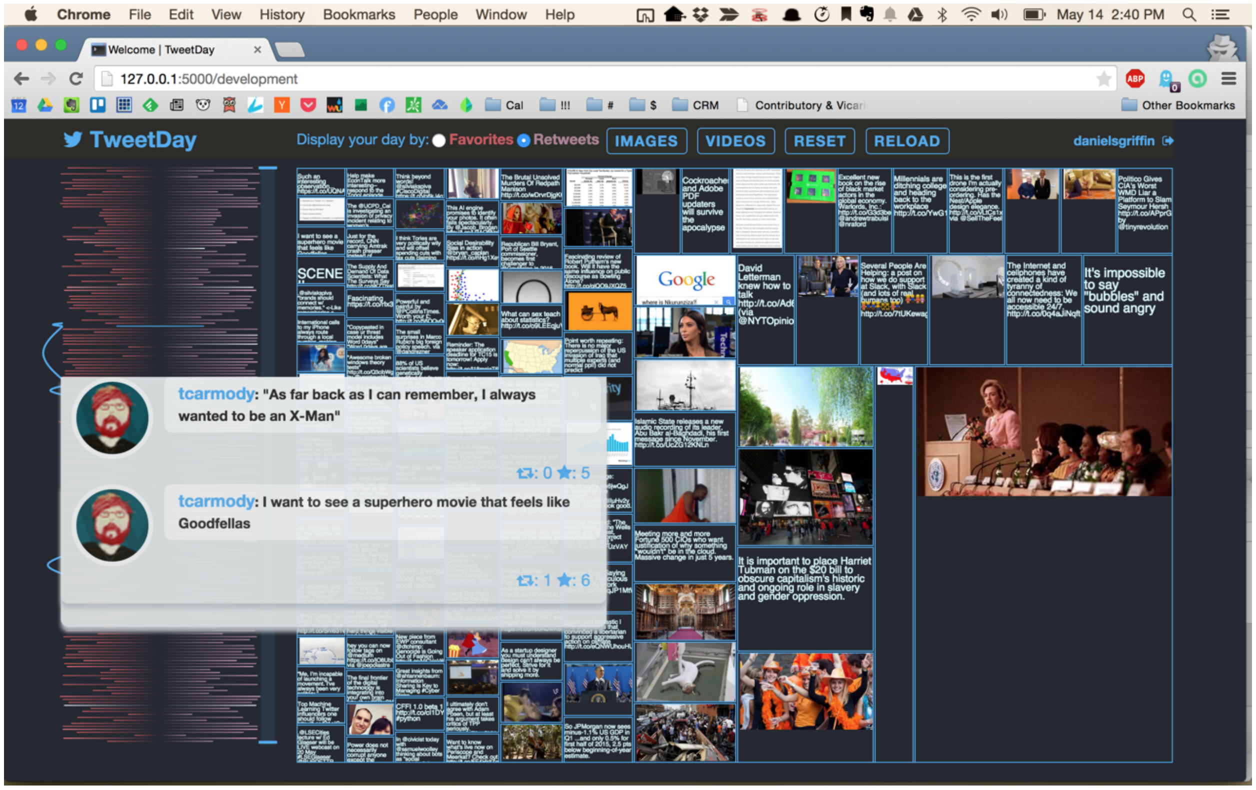


Figure 22. One of the first views of the combined fields.



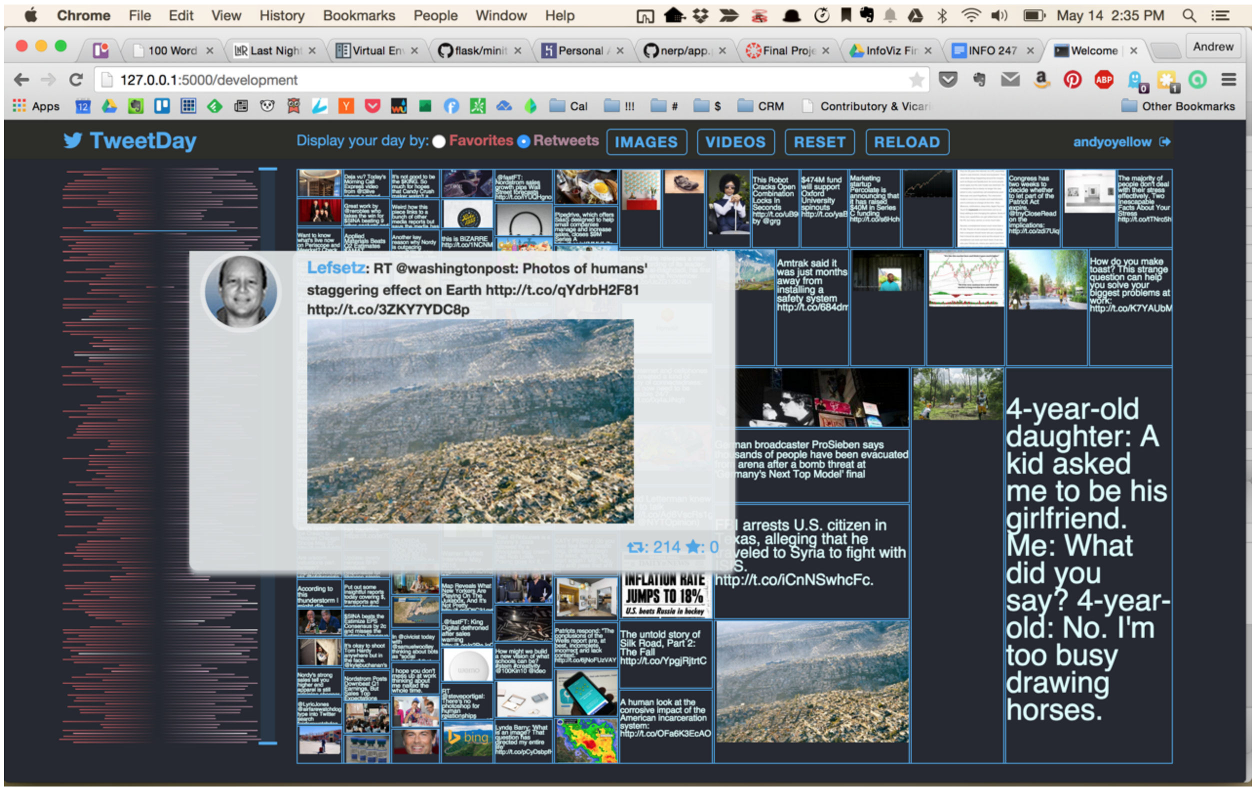
tcarmody: "As far back as I can remember, I always wanted to be an X-Man"

0 ★: 5

tcarmody: I want to see a superhero movie that feels like Goodfellas

1 ★: 6

Figure 23. When an account replies to itself it is depicted the same as other replies.

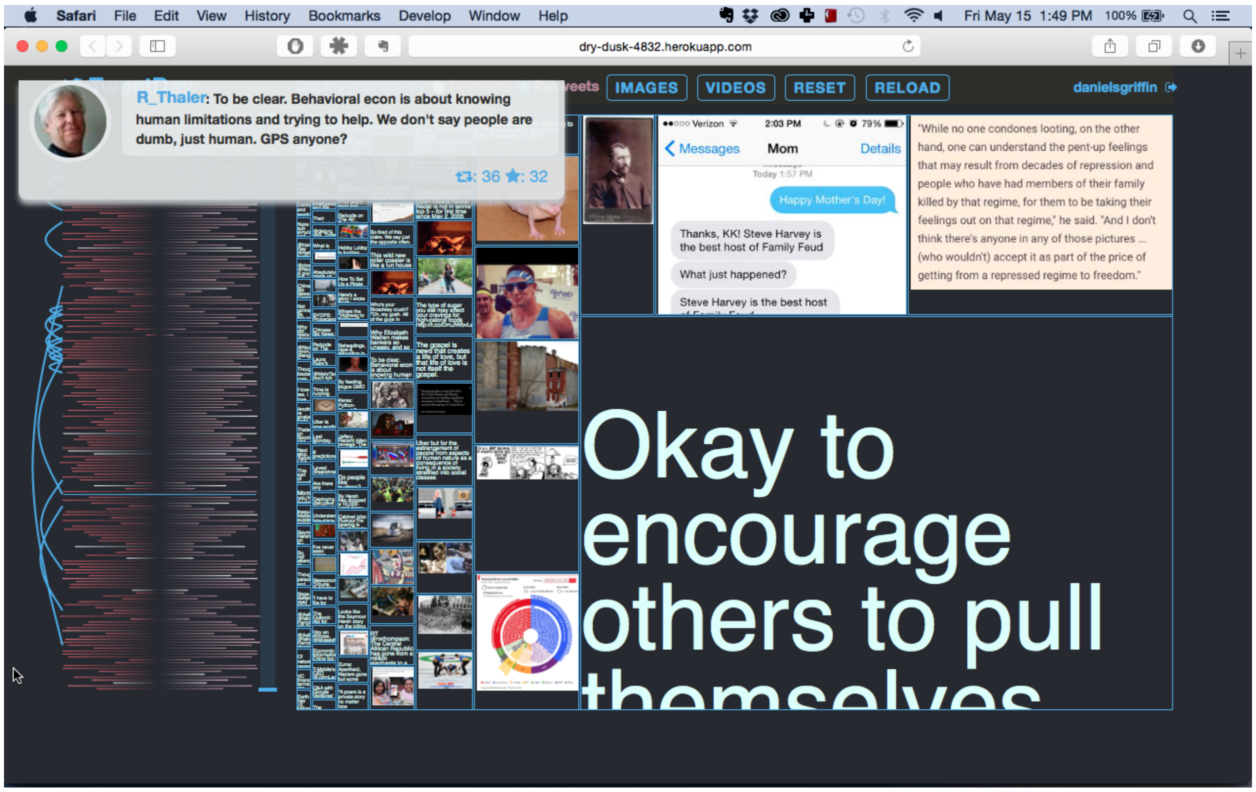


Lefsetz: RT @washingtonpost: Photos of humans' staggering effect on Earth http://t.co/qYdrbH2F81 http://t.co/3ZKY7YDC8p

214 ★: 0

4-year-old daughter: A kid asked me to be his girlfriend. Me: What did you say? 4-year-old: No. I'm too busy drawing horses.

Figure 24. Hovering over a line brings up a tooltip.



R_Thaler: To be clear. Behavioral econ is about knowing human limitations and trying to help. We don't say people are dumb, just human. GPS anyone?

: 36 ★: 32

Happy Mother's Day!

Thanks, KK! Steve Harvey is the best host of Family Feud

What just happened?

Steve Harvey is the best host

"While no one condones looting, on the other hand, one can understand the pent-up feelings that may result from decades of repression and people who have had members of their family killed by that regime, for them to be taking their feelings out on that regime," he said. "And I don't think there's anyone in any of those pictures ... (who wouldn't) accept it as part of the price of getting from a repressed regime to freedom."

Okay to encourage others to pull themselves

Figure 25. Scrolling through the lines on the left with the j and k keys presents the tooltip on the upper left.