ViralViz: Visualizing Temporal Content Flow in Social Networks

Joshua Bradley
Computer Science Dept.
University of Maryland
jgbradley1@gmail.com

Nick Fung
Computer Science Dept.
University of Maryland
nfung13@gmail.com

Isaac Julien
Computer Science Dept.
University of Maryland
ijulien4@gmail.com

Meethu Malu
Computer Science Dept.
University of Maryland
meethu@cs.umd.edu

Matthew Maurielo
Computer Science Dept.
University of Maryland
mattm401@umd.edu

ABSTRACT
With the expansion of data sharing and always-online devices, social networks have become an increasingly important aspect of society. Groups such as the Social Media Research Foundation are interested in creating new techniques and tools to analyze content flow through these networks. ViralViz is a visualization tool developed as a NodeXL extension to aid in the study of content diffusion through social networks, specifically the evolution of topics over time. ViralViz takes in GraphML files generated by NodeXL, which can import this data from a variety of social networks, including Twitter. ViralViz uses several topic analysis techniques to generate information which it then displays in a Streamgraph. Visualization controls allow the user to filter the data and manipulate the display, facilitating an interactive user experience that provides an in-depth analysis of topics over time. We chronicle the iterative development of the project including a usability study with 10 participants the results of which are distilled into application refinements. ViralViz received a number of positive remarks for vision and functionality. We record improvements that address concerns and criticisms as well as plans for future work.

Keywords
ViralViz, NodeXL, social network, Streamgraph, Twitter, viral, information visualization, diffusion of content

1. INTRODUCTION
Social scientists are often interested in studying content diffusion through social networks in order to better understand the impact social media has on a group of people. For instance, a social media manager may want to increase brand awareness of a specific product through social networks like Facebook and Twitter. The manager would be interested in discovering how awareness of similar products has spread in the past and may wish to find out who best propagates this information. He or she would also need to explore what characteristics contribute to a message or advertisement going “viral”.

Figure 1: Twitter Graph of #BigData generated using NodeXL

One popular tool for modeling social networks is NodeXL [1], an open-source interactive network visualization and analysis tool. It supports the importation of data from multiple social media platforms, including Twitter and Facebook, and allows users to analyze and visualize the data as a network (Figure 1). Users are then able to explore the graph using a number of tools that allow zooming, filtering, and grouping. NodeXL also provides the user with the capability to calculate graph metrics such as betweenness centrality, degree, and pagerank.

The NodeXL codebase is hosted and overseen by the Social Media Research Foundation (SMRF), an organization dedicated to the support of research in the area of social media. In the context of Twitter data, graphs are constructed where Twitter users are nodes and edges represent connections that are made between communicating users (retweets, mentions, etc.). NodeXL includes a number of tools including the ability to group user nodes based on interactivity or other traits. Additionally, NodeXL can analyze and extract a list of “topic” words for each subgroup based on information content.

While NodeXL is highly capable of modeling a static network or a snapshot of a dynamic network, one limitation it suffers from is the difficult problem of analyzing activity and content change over time. For example, a marketing agent may wish to visualize how topics in the graph change over several days or predict topics that might peak sometime in the future depending on current or past activity. However, NodeXL lacks tools to easily create a visualization to accomplish this task.
We report 3 main contributions in this paper:

1. We developed a tool, ViralViz, which extends the current capabilities of NodeXL, allowing analysts the ability to observe the temporal behavior of topic conversations in a social network. ViralViz allows users to import their own GraphML files generated by NodeXL. It provides a smooth transition from the static representation of dynamic data found in NodeXL to a visualization of the temporal dynamics within the data.

2. We demonstrate that a Streamgraph can be used as an effective analytics tool for temporal data, generating actionable insight.

3. Past work has centered around producing static Streamgraphs with minimal interactivity. By providing the user with a rich set of tools to filter, manipulate, and annotate topics (in the control panel) within the Streamgraph, we give the user a novel and potential beneficial interactive experience.

2. RELATED WORK

Online social networks like Twitter, Facebook, and LinkedIn have reached a point where they provide massive and rich data sets for analysis. Social media scientists, managers, and sociologists have found great use of these resources in their research. There has been some previous work that analyzes the use of Twitter hashtags, keywords, and tags over time. Tools like Conference Monitor [2] leverage backchannel conversations to provide insights on the important participants, trending topics and popular sessions. [3] explores topics from two large online communities (Twitter and Wikipedia) and establishes trends in topics over a 6 month time span. [6] analyzes diffusion of topics based on mentions in the twitter data. Previous tools like Statler [7] examine the semantics and structure of tweet messages sent during a broadcast media event, displaying segmentation, trending topics, and geo-location data. However, Statler was designed with the goal to identify specific interesting moments within the tweet stream. It does not give an adequate representation of the overall twitter stream. Twitinfo [7], a system built to visualize real time events, focuses on identifying and labeling conversation peaks during some event. While this data can be useful, it is also important to understand how and when the peaks occur, which is what ViralViz attempts to address.

The original Streamgraph paper provides a lively discussion on the difficulties faced by stacked chart over the years [9]. Starting with a case study of the design of the method, it documents the story that led to the Streamgraph creation, what geometric algorithms were used to calculate Streamgraph layers and what other design principles influenced its creation. There is also a discussion on how to order the layers in a Streamgraph, which will prove to be important when deciding on which topics should be layered on top of others.

There has been some other research that has focused on the flow of topics over time in Twitter data. [4] provides a visualization of automated topic detection and alignment over time. Though the work has significantly contributed toward visualizing topics, it places an emphasis on exploring the similarity of topics over time. ViralViz on the other hand, seeks to provide users with the power to analyze the entire data set (or a controlled subset of it), on the diffusion of content over time.

ViralViz seeks to fill a void in the NodeXL tool, specifically targeting the analysis of topic diffusion within a social network over a time period. Most tools do not provide users a way to inspect the temporal behavior of a topic along with its highs and lows. In addition, many do not provide controls to manipulate and interact with the topic stream. Contrary to [5], which asserts that Streamgraphs cannot depict complex patterns in heterogeneous data sets, ViralViz contains a rich set of tools and controls that help users explore datasets from various domains.

3. TOPICS: TOPIC MODELLING WITH MALLET AND KEYWORD EXTRACTION

To create the Streamgraph visualization, we need to define some notion of a topic. In natural language processing (NLP), this representation is typically a collection of words that describe the topic. For instance, a topic might be [“Obama,” “election,” “Romney,” “debate”]. Although no one term is chosen to represent the topic, ideally, the meaning of the topic is clear from the words that describe it.

ViralViz allows the user to choose between two approaches: Latent Dirichlet Allocation (LDA), a popular topic modeling technique, and an approach that extracts keywords based on their statistical significance. To perform LDA topic modeling, ViralViz utilizes the MAchine Learning for LanguagE Toolkit (MALLET) software package developed by Andrew McCallum of the University of Massachusetts, Amherst [8]. LDA is a hierarchical Bayesian model that describes a process in which a set of latent (unseen) topics generate words in a set of documents [1]. Each document is assumed to be a mixture of topics, and each topic generates words according to some probability distribution. MALLET provides sampling-based inference methods that estimate the parameters of this model. By binning the Twitter data by time, we can use the trained parameters to display the change in topics over time.

We also provide a statistical method for choosing keywords, similar to information theory-centric approach of the SMR Foundation. We use a likelihood ratio test to identify statistically significant collocations, groups of adjacent words, and from these collocations select k top words based on their frequencies.

The SMR Foundation’s current approach to topic analysis is to extract meaningful collocations of words (e.g. bigrams) using measurements of information content such as TF-IDF. It is our hope that utilizing a topic modeling approach will allow us to connect multiple meaningful words and phrases that belong to the same topic.
4. DESIGN OVERVIEW

ViralViz is designed to function with a web interface frontend communicating with a server back end. The frontend interface utilizes standard HyperText Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript. The user may access the frontend interface through a web browser such as Firefox, Chrome, or Internet Explorer. The server runs a Python script, which makes calls to MALLET to perform topic modeling over GraphML files. These files may either be stored on the server or specified via a uniform resource locator (URL) from the interface. The output from MALLET is stored on the server in the form of extensible markup language (XML) files. The server returns the names of these files to the interface so that the browser can obtain and process the MALLET output. The interface reads the data remotely from these XML files and displays the appropriate visualization for the user. The user may then manipulate the data using the control panel within ViralViz.

4.1 Client/Server Architecture

The ViralViz client frontend runs in a user’s browser, allowing for application to potentially be published on the Internet to reach a wider audience. It makes extensive use of CSS and JavaScript to communicate with the server, display visualizations, and allow the user to interact with the data. Through buttons, fields, and other controls, the browser can collect a set of parameters from the user to send a command to the backend topic modeling code. The client uses JavaScript to send an Asynchronous JavaScript and XML (Ajax) request to a PHP script hosted on the server. This script in turn runs a Python script with the appropriate input parameters needed to execute MALLET. The backend communicates the location and file names of the MALLET output to the server so that the information may be loaded into the visualization.

The client interprets the MALLET output files and formats this data into visualizations. The Streamgraph and other visualizations are created using the Data Driven Documents (D3) library. This JavaScript library is a powerful tool for creating, displaying, and manipulating visualizations within a web browser. The primary visualization is a Streamgraph prominently featured in the ViralViz main frame. Many of the user controls, including filters and annotations, alter the Streamgraph. Further manipulations may require additional communication with the server and can result in additional MALLET output files. The ViralViz interface has controls that allow the user to save their visualization and accompanying annotations on the client side.

4.2 Interface

The interface can be divided into three distinct sections. The control panel on the left contains fields, buttons, and menus that allow the user to interact with the data. The user can perform actions such as filtering, ordering, and data loading. The visualization panel is in the upper right portion and occupies most of the screen. This panel displays three visualizations including the Streamgraph as well as a legend and a timeline for the Streamgraph. Finally, the details panel is located at the bottom of the screen. This panel consists of two panels: a Topic tab that displays a list of topics along with some details and an Annotation tab that displays the annotations created by the user along with several details.

4.2.1 Visualization Panel

4.2.1.1 Streamgraph

The Streamgraph is a visualization that was designed to display multiple time series data [9]. This visualization was first developed by Byron and Wattenberg to visualize music preference data and gained widespread controversial popularity when the New York Times published a Streamgraph displaying box office revenue [12]. We chose to employ the Streamgraph as the primary visualization for ViralViz for several reasons. The original motivating factor behind its design is derived from Edward Tufte’s macro/micro principle - to show many individual time series while also conveying their sum. We felt it necessary to choose a visualization that would support possibly many different time series (represented as topics) due to the possibility of there being many different topics appearing in the twitter stream at any given time. Comprehension of a Streamgraph does require a change in mindset from conventional assumptions normally applied to charts, due to the fact that Streamgraphs have no global
Y axis. Therefore, data peaks will not always occur at the top of a Streamgraph nor will low data peaks occur at the bottom. In fact, high data peaks must be measured by the height of the “bulge” that is created. The Streamgraph is displaced around a central axis and designed to model changes in magnitude smoothly over time, lending itself to form a flowing, organic shape.

4.2.1.1 Streamgraph Design
Twitter activity data was organized into topics using the MALLET library, with each topic being visualized in a Streamgraph layer. The Streamgraph displays changes in the magnitude of a topic based on the activity level (number of tweets classified into that topic) over a given time interval. This data can be further analyzed by the user to identify topic trends and estimate the timing of key events that triggered changes in the activity level.

The legend element contains a list of topics color encoded the same as their corresponding topic in the Streamgraph. The legend is meant to give clarity to the Streamgraph and aid the user in quickly making sense of the visualization. To this extent, hovering over a topic in the legend will highlight the corresponding topic in the Streamgraph.

4.2.1.2 Secondary Visualizations
Also included in the visualizations panel (Figure 5), are two secondary visualizations: a bubble chart and a bar chart. The bubble chart is labeled as “Topics Sized by Volatility”. While the Streamgraph is a powerful tool for visualizing changes over time, the purpose of displaying a bubble chart in this visualization panel is to give the user information on the overall statistics of each topic. By displaying the overall volatility, the user is given information on the topics that cover the entire time span. Similar to the legend, using the mouse to point at a bubble will highlight the corresponding layer in the Streamgraph and the element in the legend. This facilitates the user’s ability to tie the visualizations together into a cohesive image. The bar chart lists the six most active Twitter handles, visually displaying their total Tweet count in the data set. The purpose is to give the user more information about the Tweets and can allow the user to operate on the data with this knowledge. For example, the user could filter out some or all of these Twitter handles using the “Exclude Users” tool in the control panel.

4.2.2 Control Panel
There are a variety of options available in the control panel shown in Figure 4 to aid exploration and manipulation of the Streamgraph. Additionally, the panel has been laid out to onboard the user experience. The steps to completing the ViralViz experience can be broken down into four main areas: (1) loading the data, (2) filtering the data by changing Mallet parameters, (3) manipulating the visual elements of the Streamgraph, and (4) exporting a report summarizing the results of the entire process.

4.2.2.1 Layer Ordering
The Streamgraph stacks the topic layers to create a seamless graphic. The ordering through which the topics are layered can aid in the user’s ability to quickly identify important trends in the data. Because trend data tends to ebb and flow over the course of a sufficiently long period of time, ordering topics by volatility can make unstable or interesting topics more readily apparent. For this paper, we define the volatility of a topic as the standard deviation of the corresponding time series associated with that topic. Thus moving more volatile topics to the top and bottom extremes of a Streamgraph will result in more stable, relatively flat topics in the middle of the visualization. This layout ordering also means that...
extreme changes in the volatile topics will be less of a distraction to the Streamgraph as a whole, having minor impact on the layers in the middle. Other orderings included in ViralViz are average, default, reverse, and inside-out. As the name suggest, the average layout ordering will place topics with a small mean value toward the center of the Streamgraph, while topics having a larger mean (i.e. more tweets on average are classified into this topic) will tend to be pushed toward the outer edges of the stream. We define the default layout order as the ordering of topics that the data is read in (i.e. the first topic read into the Streamgraph will be located at the top of the graph, the last topic will be located at the bottom of the Streamgraph). Although no intrinsic value exist for this particular layout ordering, we chose to leave it as an option in order to test the impact that more valuable layout orderings (such as volatility and average) had in generating insight. The reverse layout ordering is defined as the reverse of the default layout ordering. The inside-out layout ordering determines the order of topic layers by first sorting each layer by the index of their maximum value, then using balanced weighting.

4.2.2.2 Timeline: Zoom, Slide, and Filter

The timeline control allows a user to zoom and filter the Streamgraph in order to control the scale and timeframe of the area under examination. This simple tool allows a user to intuitively make adjustments to the degree of zoom by dragging the margins of the focus area. Shifting focus to a different timeframe simply requires the user to select the current area under focus (visually depicted as a slightly darker rectangular area) and drag. Because time sets can be very long in length, it is important for the user to be able to drill down into areas of interest and filter out times that are of no interest.

4.2.2.3 Topic Filtering

The topic filter control allows the user to selectively filter specific topics out of the Streamgraph. Such a tool can be useful when the user is interested in only a few specific topics. Streamgraphs in particular can be difficult to analyze because the layers are stacked from a centerline instead of all starting from a common baseline. Filtering topics will make it easier to compare individual topics.

4.3 DETAILS PANEL

The Details Panel is meant to give the user greater insight into the meaning behind the Streamgraph. This panel gives the full details of each topic to the user including the summary, volatility, and complete description as generated by MALLET. While the Visualization Panel will allow the user to make determinations about time specific characteristics for a topic, the Details Panel will allow the user to identify specific aspects about each topic and the subject of each topic. This information would allow the user to create a complete conclusion, connecting specific discussion topics at specific times.

4.4 ANNOTATIONS PANEL

The Annotations Panel was designed to allow users to curate the Streamgraph. The tool enables the user to mark interesting points within the stream and write small messages about them. Additionally, the initial implementation would allow us to gather feedback from users on directions to take this tool in terms of the dynamic Streamgraph.

5. TESTING

To examine our current prototype and evaluate its potential as a tool, we conducted remote and in-person usability tests with ten users.

5.1 METHOD

Testing was performed in person or using remote desktop technology which allowed participants the ability to operate the ViralViz tool hosted on a personal computer. This was necessary to allow the development team more control over the test environment and avoid breaking any security policies that can occur with third party servers limiting access to application and system calls. We conducted usability testing of ViralViz with both domain experts and domain novices recruited through a combination of random selection and direct solicitation by way of the team’s accessible social networks. The goal was to obtain feedback on the tool’s ability to facilitate the analysis of diffusion of content in large social networks over time from a variety of datasets. The demographic details and prior experience in social network analysis of each test subject was recorded at the onset of testing. Of particular interest was the user’s previous experience using tools like NodeXL or Gephi. Users with no previous experience in analyzing temporal content in large networks were additionally asked for their impressions of such a task and the type of content they would like to see. This survey took approximately 5 - 10 minutes for each participant.

Once the pre-activity survey was completed, users were asked to access the ViralViz interface. Remote testers were given detailed instructions to log into one of the researcher’s machines to access the tool. As part of this activity, we asked the users to explore the interface until they were comfortable. Once the users had indicated that they were comfortable, they were instructed to perform 3 different tasks. We employed the Think-Aloud protocol, which encourages a user to vocalize their thoughts and talk through their reasoning. While performing these tasks, the researchers asked the subject questions designed to promote activity and elicit user feedback. Users were asked if they could identify any source of frustration or if they noticed any features that were lacking. In total, this activity lasted 20-30 minutes.

The first task assigned to the users was to identify interesting topics by loading a particular data set using ViralViz. The users were asked to utilize the tooltip in drawing insights about the data set. The users were also asked to identify key people in the data set. As a second task, the users were instructed to load another dataset and compare the different topics within this dataset. The users were advised to use the filtering tools provided in the control panel to aid in this analysis. The third task was designed to be more open-ended, asking the users to use all the features of this tool on a third data set. The users were asked to give feedback on each of the features and draw their own conclusions from the data. To aid the users, we drew their attention to the annotation tool in the control panel. The users were able to use the tool to mark the
Streamgraph to identify interesting points within the data set. A point of interest for this task was to examine what kinds of annotations the users attributed to different points in the Streamgraph.

After completing the three tasks, the users were then asked to fill out a satisfaction questionnaire, which focused on their usage of ViralViz. To understand the usefulness of features and the participant’s satisfaction with the results in ViralViz, we used Likert scales to capture subjective responses. The questionnaire also asked the users open-ended questions about future use, comparison with current practices, and innovativeness. The satisfaction questionnaire took 5 - 7 minutes per participant. All sessions were recorded using screen capture technology for post-study analysis.

5.2 PARTICIPANTS
We performed usability testing with 10 participants (25% female) ranging from domain experts in the area of social media research to domain novices (i.e. computer-science graduate students with some experience in information visualization). We found that our participants used NodeXL (50%), UCINet (10%), and Excel (10%); some users did not identify particular toolsets but had past experience in information visualization.

5.3 RESULTS
Likert questions implemented on a 9-point difficulty scale ordered very easy (1) to very difficult (9) or agreement scales ordered strongly disagree (1) to strongly agree (9); 5 was neutral in both. We report averages (M=X) and standard deviation (SD=X). The 10 participants are identified as ‘P1’ through ‘P10’.

Task Difficulty
The tasks, as described in the method section, were designed to progressively reinforce the actions necessary for participants to get a sense of what the final application experience would be like. As anticipated, participants reported that the average difficulty of performing the loading and exploration task was 7.7 (SD=1.25) or relatively easy. While completing the second task, participants reported that adding mallet filtering to the task increased the difficulty to 6.5 (SD=1.63), but still the task was perceived as being easy. By the third task, participants were asked to first manipulate and annotate the stream, and then export the results in addition to the previous steps and this is rated slightly less difficult at 6.7 (SD=1.63), which remains an easy task. From these results, we can conclude that our participants were able to easily complete the intended tasks as described to them.

Task Satisfaction
Having assessed the difficulty of the tasks, we then asked the participants to subjectively rate how satisfied they were with the results of the tasks. Task one allowed the users to engage with the application for the first time with very little understanding of what to expect, thus users rated their satisfaction with completing the task as being a 7.5 (SD=1.43) or satisfied. Similarly, satisfaction with the results of tasks two and three were 6.3 (SD=1.76) and 6.9 (SD=1.52) respectively.

Individual Features
Most participants commented that they were surprised about the number of features implemented by ViralViz. We chose to ask our users about a subset of these features that seemed most representative in the experience. Participants rated the data granularity parameter as not being very useful (M=5.3, SD=2.05). Participants rated the number of topics parameter as being fairly useful (M=8.2, SD=1.13). Participants rated the reordering mechanism as being somewhat useful (M=6.9, SD=1.52). Participants rated the annotations feature as being useful (M=7.3, SD=1.49). Based on these rating and our observations, it is apparent that the participants were successful in using the included tools and controls to manipulate the data and draw some preliminary conclusions. However, some participants needed prompting to make use of the manipulatable timeline feature. Participants requested clearer directions and labels.

Overall Experience
The participants rated the experience of using ViralViz positively (M=6.8, SD=1.13). P2 stated that ViralViz had the potential to “supplement, not replace what I do currently. I do a lot of tiresome looking through datasets. This would cut that down.” and P4 stated that “some of the text analysis tools (e.g., topic identification) were more useful than what I currently use.” Participants expressed positive impressions on the overall design and functionality of the ViralViz tool. They had little difficulty navigating through the collapsible menu interface and utilizing the data loading and display controls. Participants were able to interpret the basic structure of the Streamgraph: activity level over time. However, some users did not immediately grasp that the displayed visualization represented Twitter data divided into topics. Additionally, participants expressed some confusion in understanding the additional bubble and bar chart displays. Finally, all the participants would likely use ViralViz in the future (M=7, SD=1.15).

Drawing Insights
All of our users were able to draw their own insights using the control panel and the Streamgraph visualization it produced. For
example, P1 stated that ViralViz “quickly led me to investigate the topics that had unique or extreme temporal qualities.” and P5 was specifically able “see (a) when the topic (e.g., polio) is booming, and (b) what sub topic is contributing to that boom.” However, some users were more critical. P1 expressed that he was unable to draw conclusions from the topic descriptions because they are just lists of words. He suggested using stop words to avoid including common words such as “the” and “rt” in the topic descriptions. Similarly, many participants expressed a desire to include the actual Tweet data or some subset of highly representative Tweets in order to better express the representation of each topic. P1 also expressed concern that the layers of the Streamgraph are not grounded on a common axis. This can make it difficult to compare peak values for two given topics, as the widths must be compared, not the peaks.

5.4 REFINEMENTS
After receiving feedback from the usability study, a number of improvements were made on the interface and functionality of the ViralViz tool. To improve the readability of the interface, we included additional labels, tooltips, and reworded titles of existing panels. To make controls more understandable, explanatory labels were included in close proximity to or in place of technical terms that may have been confusing. The terminology was explained or simplified and focused on the result of the operation rather than the technical name.

To emphasize the connection between the different visualizations, we added in support for brushing and linking, writing additional JavaScript code to coordinate the Streamgraph, bubble chart, and legend. This feature allowed the user to highlight a topic by hovering over a Streamgraph layer, a bubble chart circle, or an element in the legend. Performing a mouse over action highlights the topic in all three elements. In addition, functionality was added to click on a highly active Twitter user in the bar chart in order to bring his or her Twitter page up in a new browser tab.

To increase readability of the visualization panel, the color scheme was altered to make each layer more visually distinct. The original color scheme was limited to shades of green in order to maintain a consistent aesthetic appeal. However, user feedback indicated that the ability to differentiate between the topic layers was more important.

6. DISCUSSION
6.1 FEATURES
Our results for individual features indicate that the satisfaction of task two was the lowest of the three tasks. The difficulty of task two was also rated as the highest. Task two required that the users compare two topics within the polio.xml dataset. This dataset only encompasses a small time frame, which may have been a leading factor in the users’ inability to identify trends over time. In addition, lack of domain knowledge among the users may have contributed to the task difficulty and the inability to generate interesting conclusions. One user expressed a desire to change datasets to HPV.xml, as he was more familiar with biological topics and would be better able to analyze the data.

Most users expressed their confusion in regards to the ‘Data Granularity’ feature. Users were uncertain of the effect of this option and how it would be used. This conclusion is reflected in the survey data, in which 30% of the users rated this feature from “neutral” to “least useful”. Our results also indicate that the ‘Number of Topics’ parameter was well received by the users. A majority of the users valued this feature more than others. It allowed users to read the Streamgraph more accurately and identify interesting characteristics about the data.

Users expressed their concern about the tooltips, indicating that it was “annoying” to mouse over and read the long descriptions. This was especially true for layout ordering, where all five layouts are explained in one tooltip. However, this was less common among the domain experts. These users were more familiar with the concepts and were not troubled by this issue.

While manipulating the Streamgraph, the available controls allowed users to exclude specific Twitter users and their Tweets from the topic discovery function. Users P5 and P7 confused this tool to instead include the selected Twitter users in the Streamgraph, excluding the rest. These testers later realized their mistake and indicated that inclusion, rather than exclusion, was more intuitive and would add more value to the tool.

It took little time for participants to figure out how annotations could be created or removed from the Streamgraph, although prompts from the examiners were sometimes required. Once users began exploring the use of this tool, they found great value in the functionality and indicated that they could envision social media scientists wanting to come back and go over their annotations. None of the users imagined that they could export their work and when the researchers prompted them to use that feature under the Workspace tab. They thought that it was a useful and “ neat” functionality to save their results for future work or collaboration.

6.2 VISUALIZATIONS
Almost all users were fascinated to see the Streamgraph transition from one layout ordering to another. While some users commented on how each of these layouts can be useful, some were neutral towards them and said they were not too sure how it would be useful.

One user felt that a Treemap would be useful to understand the dataset better. Another user stated that a different visualization could be incorporated to work with the Streamgraph in order to express a more complete view and aid in analysis of the data. Some users expressed the need for more interactivity between the visualizations. An example was given in which clicking on the topic bubbles would trigger something in the main Streamgraph. In our current system, we have brushing and linking that allows users to choose data in one visualization and see it highlighted in the other visualizations.

Almost all users spoke about how they could see different topics peaking at different time intervals, which made them think about events that could have triggered such peaks. One user said he would like to visualize events and news along with the Streamgraph to make more sense out of this data.

6.3 STREAMGRAPH COLOR SCHEME
In the system we used for testing, our Streamgraph used different shades of green to represent the different topics (streams). Users complained about not being able to distinguish the shades in the Streamgraphs as the difference between layers was too subtle. The text in the stream legend was also difficult to read because of the dark green used with a dark font color. We realized the problem...
and iterated on the colors and our current system uses different color scheme that also provides better readability.

6.4 CONTENT
An important piece of feedback received from many of the domain experts was the relevance of the data being displayed by the Streamgraph. Although the trends in activity are interesting, it is difficult to draw concrete, usable conclusions. It was suggested that ViralViz include additional, detail-oriented data such as influential Tweets that have many retweets. Similarly, while the most active users are useful, the more relevant statistic would be the most influential users. These users could be those that were retweeted or mentioned often. Other feedback suggested that the number of user followers could be used as a measure of influence. While we agreed with this assessment, we were unable to incorporate this information within the project timeline.

User testing also showed that it was not obvious that the timeline could be manipulated as a sliding window. In order to emphasize that the timeline can be manipulated to perform zoom and filter actions in the time dimension the timeline needs to be updated to include clear visual affordances of these actions.

6.5 INNOVATION
Our satisfaction questionnaire asked if users found any innovative features in ViralViz compared to the current tools they were using for which P5 said:

"Topic to stream graph connection is useful Selection and annotation features allow users to add value as subject matter experts to the data the export to pdf is useful: generating a consumable product for non-analysts is a common user task"

Participant P7 commented about the usefulness of layout ordering:

“One thing that looked different to me was the ordering of the graph. I thought it was interesting and new to see the option of ordering the graph based on things liked the average, or flipped inside out. I’m not sure how the flipped one would be used, but I did think it was interesting. I could probably, however, think up some ways to use the average sorting tool.”

P10 sums up his entire experience of using ViralViz as follows:

“Some network analysis tools are very powerful, but using their interface is like pulling teeth. Although I didn’t instantly understand 100% of what was going on in ViralViz, the “big picture” experience was relatively straightforward. topic identification and the visualizations were new.”

7. CONCLUSION
ViralViz has shown promise in fulfilling its intended task of aiding in the analysis of diffusion of content within a social network. We were able to create a system that can take in social media data over a time period from GraphML formatted file, process it using a topic modeling method, and display the results in a Streamgraph. Although limited in size, our usability study produced promising feedback and we were able to address many of the shortcomings found in this study. However, there is still room for improvement that could increase the proficiency of the tool. The topic summaries and descriptions could be more relevant and more closely resemble rational, grammatically correct phrases. To address this issue, we would look to further tune output from MALLET or explore other topic discovery algorithms. In addition, the secondary visuals could be more closely related to the Streamgraph and possibly provide more useful information. As an example, the bubble chart could change depending on the filtering actions of the user or size the bubbles on different values. Nevertheless, ViralViz is effective in its current format and merits further effort.

8. FUTURE WORK
While ViralViz has undergone a number of improvements and has received conceptual praise from our usability testers, we know that the tool can be further improved. The project as a whole could benefit from additional usability testing and possibly a long-term case study. Along similar lines, ViralViz could be tested utilizing data from other social media sites besides Twitter such as Facebook, YouTube, or Steam. In addition, we would like to explore the possibility of making better use of the secondary visualizations. Additional tools could be added to the Control Panel that would allow for control and exploration of these visualizations to deepen the depth ViralViz. The functionality of the annotations tool needs to be iterated on to increase the visibility and functionality of this tool. Finally, more information could be added to the Details Panel including the ability to annotated time frames and create better topic details.

ViralViz has proven to be effective in its current capabilities, however it can serve as the starting point for a more comprehensive tool. ViralViz models content diffusion by a measure of activity level. Fundamentally, this is a measurement of communications per time period. While this is a good approximation model of content diffusion, more advanced metrics can be applied. As an example, an interesting aspect of content diffusion is the activity of influential members. Further incarnations of ViralViz could include the identification of key members of the social network or key communication messages. Key members or messages could be identified through metrics such as betweenness centrality, number of followers, or number of mentions. Incorporation of such analytics would make ViralViz even more effective.

9. CREDITS

<table>
<thead>
<tr>
<th>Team Member</th>
<th>Primary Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joshua Bradley</td>
<td>Streamgraph Design, Streamgraph Programming</td>
</tr>
<tr>
<td>Nick Fung</td>
<td>Frontend/Backend Interface, Secondary Visualizations</td>
</tr>
<tr>
<td>Isaac Julien</td>
<td>Backend (Mallet and Statistical Method for Topics)</td>
</tr>
<tr>
<td>Meethu Malu</td>
<td>Frontend Design, Study Design</td>
</tr>
<tr>
<td>Matthew Mauriello</td>
<td>Frontend Design, Frontend Programming, Study Design</td>
</tr>
</tbody>
</table>

10. ACKNOWLEDGEMENTS
Our thanks to Marc Smith mentoring this project and for helping us recruit social media scientists for our study, as well as the participants in the user study. Additionally, we would like to
thank Ben Shneiderman and the entire CMSC 734 class for helping us develop this study.

11. REFERENCES


