**ViralViz: Visualizing Temporal Content Flow in Social Networks**

**Josh 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  
julien4@gmail.com

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

**Matthew Mauriello**  
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 SMR Foundation are constantly on the lookout for new tools to analyze these networks, particularly their behaviors over time. ViralViz is a visualization tool meant to aid in the study of the diffusion of content through social networks. It operates on GraphML formatted data, which can be gathered from Twitter through NodeXL. ViralViz uses topic-modeling tool MALLET to create topics from the data and display them as layers in a Streamgraph. ViralViz also provides tools to the user to allow for manipulation of the display and filtering of the data to aid in analyzing the social network. We chronicle the iterative development of the project including usability studies and refinements. Although receiving positive remarks for vision and functionality, a number of improvements are recorded here for future work.

**Keywords**  
ViralViz, NodeXL, Social Network, Streamgraph, Twitter, Viral, Information Visualization, Diffusion of content

1. INTRODUCTION

Social scientists are often tasked with studying the diffusion of content through social networks. For instance, a social media manager may want to increase brand awareness of a specific product through social networks such as Facebook and Twitter. This manager would be interested in how awareness for similar products has spread in the past and may study which people best propagate this information and what factors contribute to something going “viral”.

One tool that provides the capability to model social networks is NodeXL[1], an open-source network analysis and visualization software package for Microsoft Excel. NodeXL is designed for users with little programming experience and represents the network graph in a collection of Excel spreadsheets. Social Media Importer is an extension for NodeXL that provides the ability to import data from outside sources including Facebook and Twitter.

The Social Media Research (SMR) Foundation uses NodeXL to create and explore a temporal representation of the social network. The graphs display connections between Twitter users based on who is Tweeting and who is receiving Tweets. NodeXL includes a number of tools including the ability to partition the graph into subgroups of users that tend to intercommunicate. Additionally, NodeXL can analyze and extract a list of “topic” words for each subgroup, based on information content.

While NodeXL is a highly capable program, one limitation it suffers from is the difficulty in analyzing activity and content change over time. A marketing agent may wish to visualize how topics in the graph change over several days or predict topics that might peak sometime in future depending on what the key people are tweeting. However, NodeXL lacks tools to easily create a visualization to accomplish this task.

We report 2 main contributions in this paper: (1) We created a tool that was capable of analyzing Twitter data collected through NodeXL and exported into a GraphML data file which allowed users to create and explore a temporal representation of the Twitter data that helped them analyze trends and diffusion of content through a social network. The different layers of the Stream Graph represented different topics stacked over each other. (2) We provided a rich set of tools to filter, reorder layout, add and remove topics and annotate topics in the stream thus providing greater interactivity with the Stream Graph.

2. RELATED WORK

Online social networks like Twitter, Facebook, LinkedIn have reached a point where they provide massive and rich data sets for analysis. Social media scientists, social media managers, sociologists have found great use of this resource for their research. There has been a lot of work on analyzing Twitter hashtags, keywords, and tags over time. Tools like Conference Monitor [2] leverages 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. Other tools like Statler [7] examines short messages from Twitter thus providing segmentation, trending topics, level of interest and Tweet geo-locations. Statler only provides specific interesting moments within the Tweet.
stream without providing a complete flow of the stream. Twitinfo[7] a system built to visualize real time events focuses on identifying and labeling peaks during the event. While this data can be useful, its also important to understand how and when the peaks occurred which we try to achieve in ViralViz.

There hasn’t been much research surrounding flow of topics over time on Twitter. TopicFlow: Visualizing Topic Alignment of Twitter Data over Time [4] provides visualization of automatic topic detection and alignment over time. Though the work has significantly contributed towards helping visualize topics, it lacks the ability to visualize the continuous movement of how topics flow in time.

However, none of these tools provide users a way to look at how a topic progresses over time along with the highs and lows of that topic. There aren’t any tools that let you manipulate and interact with the flow of topics too. Contrary to [5] which discusses about how Stream Graphs cannot depict complex patterns in heterogeneous data sets, ViralViz with its rich set of left panel tools, helps users in exploring datasets from varied domains.

3. DESIGN AND IMPLEMENTATION

ViralViz is designed to function with a web interface frontend communicating with a server back end. The interface is written PHP utilizing cascading style sheets (CSS) and JavaScript. The user can access the frontend interface through a web browser such as Chrome or Internet Explorer. The server runs a Python script that calls the Machine Learning for Language Toolkit (MALLET) to perform topic modeling on 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 appropriate visualizations for the user. The user may then manipulate the data using the ViralViz tools.

3.1 Topic Modeling: MALLET

Topic modeling is a key component in creating the Streamgraphs. Topic modeling is a statistical model used to derive topics that fit a collection of documents. Topics can be an abstract collection of words or a grammatically correct description, although such accurate descriptions tend to be complex and unreliable. On the most basic level, topics can be a collection of frequently used words commonly used across a subset of the documents. More complex and accurate algorithms will employ statistical modeling or other advanced techniques.

We perform topic modeling on the Twitter data using the MALLET software package developed by Andrew McCallum of the University of Massachusetts, Amherst [8]. MALLET was used to bin the individual Tweets in the GraphML file into different topics. Each topic can then be displayed as a layer in the Streamgraph.

MALLET allows for several different models to be used for the classification process. Latent Dirichlet Allocation (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 [10]. 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. ViralViz also includes the ability to use Information Content as a modeling technique. The SMR Foundation’s current approach to topic analysis is to extract meaningful collocations of words (for example, bigrams) using measurements of information content such as TF-IDF. We hope that a topic modeling approach will allow us to connect multiple meaningful words and phrases that belong to the same topic, since a topic is a distribution over all seen words. One challenge will be then representing each topic for the user. Our first approach will be to simply choose the k highest probability words for each topic.

3.2 Client/Server Architecture

The ViralViz client front end runs in the user’s browser, allowing for the potential to publish the application on the Internet to reach a wide 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 intractable, the browser can collect a set of parameters from the user to send a command to the MALLET topic modeler. The client uses a 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 input parameters to perform a MALLET operation. The location and file names for the MALLET output are communicated back to the server for visualizing.

The client can interpret the MALLET output files and can format 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.

3.3 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.

![Figure 1: ViralViz with Streamgraph, Hot Topics, Top Tweeters, Stream Legend and a Table at the bottom](image)

### 3.3.1 Visualization Panel

#### 3.3.1.1. Streamgraph:
The Streamgraph is a visualization that is primarily intended to showing multi-layered data over a time period. This visualization was developed by Byron to visualize musical preference data and was utilized in the New York Times to visualize movie sales data [9]. We chose to employ the Streamgraph as the primary visualization for ViralViz because it naturally lent itself to modeling the changes in Twitter trends over a period of time.

Twitter activity data was organized into topics, with each topic being visualized in a Streamgraph layer. The Streamgraph structure visualizes the changes in the topics over a time period. This can be analyzed by the user to identify topic trends and estimate the timing of key events that triggered changes in this data.

The legend element contains a list of the topics colored the same as their corresponding topic in the Streamgraph. The legend is meant to give clarity to the Streamgraph and help the user to quickly make sense of the visualization. To this extent, placing the mouse pointer over a topic in the legend will highlight the corresponding topic in the Streamgraph.

#### 3.3.1.2. Secondary Visualization:

Also included in the visualizations panel 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 covers 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.

### 3.3.2 Control Panel

#### 3.3.2.1 Layout Ordering

The Streamgraph stacks the data 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 time period, ordering the topics by volatility can make more unstable topics more readily apparent. Volatility is a measure of standard deviation in a topic. Thus by moving more volatile topics to the top and bottom extremes of a Streamgraph will result in more stable, relatively flat topics toward the middle of the visualization. This layout will make the extreme changes in the volatile topics more visible. Other orderings included in ViralViz are average, reverse, and inside-out.

#### 3.3.2.2 Timeline: Zoom, Slide, and Filter:

The timeline control allows user to zoom and filter the Streamgraph in order to control the scale and timeframe of the area being examined. The bar is based on the sliding window tool that can be found in other visualizations such as Google Finance. This simple tool provides adjustments to zoom, timeline range in an easy to understand format. Because time sets can be very long in length, it is important for the user to be able to zoom in to areas of interest and filter out times that are not being currently examines.

#### 3.3.2.3 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 “ground” value.

### 3.4 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. METHOD
We conducted remote usability testing to evaluate if ViralViz was helpful for social media managers or social scientists in tracking diffusion of content in large social networks over time. The procedure started with participants giving their consent to participate in the study. At the beginning of the session, we asked the participants to fill a short online survey which collected demographics details along with their previous experience analyzing large social networks using different tools like NodeXL, Gephi and any other if they use. Table [1] shows the current tools used by our users. For users with no previous experience in analyzing temporal content in large networks, they were asked about the idea and the type of content they would like to see. This survey took about 5 - 10 minutes for each participant.

Once the pre-activity survey was completed, the users were asked to remotely log into one of the researcher’s machine to access ViralViz. As part of this activity, we asked the users to play around with the tool until they were comfortable. Once they were ready to move on, we asked them to perform 3 tasks. We used the Think-Aloud protocol and asked the user to talk out loud about what they were seeing and doing on the screen. While performing these tasks, the researchers prompted questions like were there things they couldn’t find while using the tool, who were the key people involved in the current topic, and many others. This activity lasted 20-30 minutes.

The first task asked the users to find any number of interesting topics by loading a particular data set using ViralViz. The users were asked if the tool helped them draw any insights. They were also asked to identify the key people associated with the topic and also the tweets in each topic. Once they were done with the first activity, we asked the users to load another dataset and then compare topics within this dataset using the filtering options provided on the left side. We observed what features the users used to filter and compare their data. The third task was more about making the users use all the features of this tool and getting feedback on each one of them. So we asked them to annotate one of the Streamgraphs to see what kinds of annotations users come up with and if the feature is useful at all.

After completing the three tasks, the users were then asked to fill out a satisfaction questionnaire that focused on their usage of ViralViz. To understand the usefulness of features in ViralViz, we used a 9-point Likert Scale to capture the responses. The questionnaire also asked the users open-ended questions about future use, comparison with current practices and innovativeness about ViralViz. The satisfaction questionnaire took 5 - 7 minutes per participant.

4.1.1 Participants:
We performed usability testing with two separate subjects who are domain experts in the area of social media research. The testing was performed remotely using remote desktop technology to allow the participants the ability to operate the ViralViz tool hosted on a personal computer. This allowed the development team more control over the test environment and avoided the permissions problem that can occur with third party servers. These web servers often limit access to application and Python calls for security reasons.

**In all we interviewed 7 users, out of which *x* were females and the rest were males. talk about the occupation of these participants - social scientists, professors, sociologists. talk about their previous experience analyzing diffusion of content over large datasets**

Table 1. This table shows the distribution of different tools used by the participants

<table>
<thead>
<tr>
<th>Tool Name</th>
<th>Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>End</td>
</tr>
<tr>
<td>Figures</td>
<td>Good</td>
</tr>
</tbody>
</table>

5. RESULTS
Both participants expressed positive impressions on the overall design and functionality of the ViralViz tool. They had no difficulty in navigating through the collapsed menu interface and utilizing the data loading and display controls. Both participants were able to interpret the basic structure of the Streamgraph: magnitude data over time. However, one user did not immediately grasp that the displayed visualization represented Twitter data divided into topics. Both participants expressed their confusion in understanding the additional bubble and bar chart displays.

The participants were successful in using the included tools and controls to manipulate the data and draw some preliminary conclusions. However, both participants needed some prompting to make use of the manipulatable timeline feature. Both participants requested clearer directions and labels. One participant had a number of suggestions to ease the burden of feature discovery on the user.

One participant 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, the other participant expressed his desire to include the actually Tweet data or some subset of highly representative Tweets in order to better express the representation of each topic. This same participant also expressed his 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.

5.1.1 Refinements:
After receiving feedback from the two domain expert participants of the usability study, we worked to improve the ViralViz tool to address its shortcomings. To improve the readability of the interface, we included additional labels and reworded titles of existing panels. For controls labeled with technical terms, we simplified the terminology and focused on the result of the

---

Figure 2: [Graph representing the usefulness of each feature in ViralViz **need to insert real data**]
operation rather than the technical name. In order to emphasize that the timeline can be manipulated to perform zoom and filter actions in the time dimension.

To emphasize the connection between the different visualizations, we created additional JavaScript code to coordinate the Streamgraph, bubble chart, and legend. This feature allows the user to highlight a topic by using the mouse to point at 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.

6. DISCUSSION

/**The results from the usability testing showed that the overall design of ViralViz is a promising and useful tool. Both of the usability test participants reinforced that the Streamgraph could be used to identify interesting events and could be used to measure information diffusion across the network. However the tool as a whole still needs refinement. The test participants needed clearer instructions and better design of control and display elements in order to more easily understand the visualizations. In addition, the topic modeling results need to be cleaned and refined. As they are currently obtained and displayed, the descriptions do not provide much useful information as to the Tweets incorporated into a given topic. Additional information such as common phrases or the Tweet data itself could clarify the issue if it were to also be included/**

7. LIMITATIONS AND FUTURE WORK

[male/female ratio for usability testing currently it's a very complicated setup which involves users remote logging into our machines which might've resulted in some user frustration limited number of users our dataset was limited and didn’t allow users to select something of their choice (also future work)]

While ViralViz has undergone a number of improvements and has received conceptual praise from our usability testers, we feel that the tool can be further improved. We feel that the project as a whole could benefit from additional usability testing and possible 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. Finally, more information could be added to the Details Panel including the ability to annotated time frames and create better topic details.

8. CONCLUSION

ViralViz has shown promise in fulfilling its intended task. We were able to create a system that can take in social media data over a time period in GraphML format, 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 can be more relevant and more closely resemble rational, grammatically correct phrases. In addition, the secondary visuals can be more closely related to the Streamgraph and possibly provide more useful information. Nevertheless, ViralViz is effective in its current format and merits further effort.

9. ACKNOWLEDGMENTS

Our thanks to Marc Smith for helping us recruit social media scientists for our study and Ben Shneiderman for helping us develop this study.

10. REFERENCES

[1] NodeXL (http://nodexl.codeplex.com/)