

Visual Analysis of Temporal Trends in Social Networks Using Edge Color Coding and Metric Timelines

Udayan Khurana*, Viet-An Nguyen*, Hsueh-Chien Cheng*[†], Jae-wook Ahn*[†], Xi (Stephen) Chen*, Ben Shneiderman*[†]

* Department of Computer Science, University of Maryland, College Park

[†] Human-Computer Interaction Lab, University of Maryland, College Park

Email: {udayan,vietan,cheng,jahn,xichen,ben}@cs.umd.edu

Abstract—We present *NetEvViz*, a visualization tool for analysis and exploration of a dynamic social network. There are plenty of visual social network analysis tools but few provide features for visualization of dynamically changing networks featuring the addition or deletion of nodes or edges. Our tool extends the codebase of the NodeXL template for Microsoft Excel, a popular network visualization tool. The key features of this work are (1) The ability of the user to specify and edit temporal annotations to the network components in an Excel sheet, (2) See the dynamics of the network with multiple graph metrics plotted over the time span of the graph, called the *Timeline*, and (3) Temporal exploration of the network layout using an edge coloring scheme and a dynamic *Timeslider*. The objectives of the new features presented in this paper are to let the data analysts, computer scientists and others to observe the dynamics or evolution in a network interactively. We presented *NetEvViz* to five users of NodeXL and received positive responses.

I. INTRODUCTION

The visualization of network structures has gained much attention recently. The analysis of large scale complex networks formed by social media has become an interesting topic with great potential in terms of research and business. Through a combination of computers and the cognitive capability of the human visual system, visualization has been considered to be one of the most powerful approaches for network analysis.

Dynamic networks are extensions of conventional networks with temporal attributes attached to the vertices and edges. Since the structure of a dynamic network evolves over time, interesting questions related to temporal attributes arise. However, the analysis of a dynamic network can be challenging without proper handling of the temporal attribute. When a static network is simply shown at a certain time, the relations between the previous and the next timestamp disappear. Even if the network are shown side-by-side for comparison, the differences can still be difficult to be found.

Of the many tools developed to perform network visualization, NodeXL is a highly recognized tool based on Microsoft Excel which supports flexible manipulation of the visualization. However, there exists only limited ways to analyze dynamic network in the latest version of NodeXL. In order to let users of NodeXL visually analyze the dynamic nature of networks, more advanced features are required.

We present *NetEvViz*, a prototype system to visualize the changes in dynamic networks. The prototype is an extension of NodeXL. Two temporal attributes, *Start Time* and *End Time*, represent the time each node or edge entered or left the network, respectively. An example of applying temporal attributes can be found in a social network, where each vertex corresponds to a specific user and each edge corresponds to the relation between a pair of users. The start and end times of a vertex are the time points when the user joins or leaves the social network by creating or deleting the profile, respectively. The start and end times of an edge can be used to represent the time when a be-friend-with relation is established or terminated.

Our prototype aids the visualization of the change in a temporal network by highlighting the state of the graph around two timestamps. Given the two timestamps, which are specified by the user, the vertices and edges are categorized into groups with respect to their temporal attributes. Each group is assigned a different color, which enables the user to recognize subtle patterns otherwise hard to observe in a homogenous graph. In order to select the points of interest, the user is presented with the summary plots of certain metrics over the timespan of the graph.

II. RELATED WORK

A. Dynamic Network Visualization

Visualization is an important and useful approach to analyze network data. Network visualization techniques have been proposed in various domains including social networks [21], [12], biological networks [20], and computer networks [7]. The problem of representing temporal knowledge and temporal reasoning has long been studied [5], in order to discover relations and patterns [6] and to learn from the past to predict, plan, and build the future [4]. The temporal dimension is crucial for social network analysis and visualization too. With the recent popularity and availability of temporal or longitudinal network data, there has been an increasing interest in developing visualization techniques for dynamic networks. Ahn et al. [3] surveyed the tasks of temporal network analysis using temporal network visualizations. In general, these visu-

alization techniques use two popular approaches to represent the network data.

The first approach uses the traditional node-link representation to visualize the network. To add the temporal information, the authors of [9] proposed a 3D layered visualization design in which each newly introduced part of the network is shown in a layer of its own and the composition of consecutive layers represents the corresponding state of the network. [17], [8] studied two types of visualizations of temporal social networks: (1) static flip books (a combination of fixed node layout and dynamic social relationships), where the node positions remain constant but the edges accumulate over time [10], and (2) dynamic movies where nodes move as a function of changes in relations [11]. The authors of [2] proposed five principles for implementation of temporal visualizations for social networks and presented two network prototypes NodeXL and TempoVis.

The second approach represents a network by an adjacent matrix. TimeMatrix [22] proposed a matrix-based visualization for temporal network data in which each cell in the matrix contains a bar chart glyph to show how a certain attribute of the corresponding nodes or edges change over time.

Finally, a hybrid approach which combines both node-link-based and matrix-based representations have also been used to visualize static networks such as MatrixExplorer [13], MatLink [14] and NodeTrix [15]. However, to the best of our knowledge there is no hybrid approach based tool built for temporal network visualization.

B. Topological Network Statistics

Besides showing the node-link-based layout of the network, our visualization attempts to incorporate a set of network topology metrics [21], [19] to show quantitatively how the network changes over time. This can be done by plotting a set of network topological measurements at different time points. Network topology has been extensively studied on a wide range of networks including social, biological and computer networks. A variety of topological metrics have been proposed to measure important different types of network properties.

Broadly, the network topology measurements can be divided into three groups from a lower to a higher abstraction level: local, group (or community) and global metrics. Note that this categorization of metrics is not mutually exclusive. Many metrics at lower levels can be aggregated (e.g., sum, average etc) over multiple nodes to achieve values or distributions at higher levels. *Local metrics* measure the topological properties of individual nodes in the network. This group of metrics includes degree, betweenness, closeness, eigenvector and PageRank centralities. In addition, clustering coefficient and assortativity coefficients are a measurement of how nodes locally connect to each other. *Group metrics* measure the topological properties of a group or community of nodes in the network which include modularity [18] and group evolution metric [20]. *Global metrics* quantify different properties of the whole network such as number of nodes, number of edges, density, etc.

Besides measurements for static networks at different points in time, a number of metrics have been proposed to explicitly quantify the change of network over time. [16] introduced a novel dynamic centrality metric which measures how well connected a node is over time. [20] proposed two metrics to quantify how groups of nodes in the network evolve over time.

C. NodeXL

NodeXL is a free and open source tool for Network Analysis and Visualization [12], [1]. The tool provides a rich set of functionality for visual analysis of static networks. In the section on design and implementation, we describe in more detail the features of NodeXL and why it is a suitable candidate for extension towards dynamic network visualization.

III. KEY DESIGNS AND IMPLEMENTATIONS

A. Introduction to NodeXL

As mentioned before, NodeXL provides many capabilities for Social Network Analysis and Visualization within Excel. Visualizations can be done using different algorithms like Fruchterman-Reingold, Harel-Koren etc. Using select and drag, the user can choose to manually change the layout that was computed algorithmically. Also, the network can be annotated with the names or images of the nodes or edges and the edges can be filtered based upon a numerical attribute associated with that edge. Different nodes can be colored according to a cluster specification or individually as well. The tool also contains algorithms for calculating network metrics like degree centrality, betweenness centrality, clustering coefficient, eigenvector centrality etc.

The ease of data storage and manipulation with Microsoft Excel, combined with the rich network visualization feature set of NodeXL make the choice of extending NodeXL instead of writing software from scratch a natural one. As explained later, we were also able to utilize Excel's API for plotting the *Timeline*. Finally, since NodeXL is open-source software, our future goals of distributing these changes to a large audience of interest are better pursued through this choice.

In order to understand the functionalities provided by NetEvViz, we require the reader to have a working knowledge of NodeXL. For the readers unfamiliar with it, we provide a very brief account of NodeXL in the following sentences and strongly recommend going through the 40-page tutorial that can be found at [1] and optionally refer to [12] as well. In order to use NodeXL in Microsoft Excel 2007 or 2011, the user needs to install the required plug-in and bring up a NodeXL workbook template as shown below. Two worksheets, *Vertices* and *Edges*, need to be populated in order to specify the the input network. *Vertices* and *Edges* can either be populated entirely manually, imported in several formats like Pajek, or imported directly from a Twitter account. Upon data import, the user can calculate various metrics, compute clusters or lay out the graph on the pane shown in the right side of the excel plug-in. The NodeXL ribbon provides various functionalities that can be used to refine the visualization and aid in the network analysis process.

B. Implementation with NodeXL

In order to visualize the temporal aspects of a network, we were required to make a list of changes and add new features to the NodeXL codebase. To start with, the start and end time of a node are essential inputs to the tool for mapping the temporal dimension of data. We added *Start Time* and *End Time* columns to the Vertices and Edges worksheets, which specify the time at which a node or an edge joined and the time after which it was no longer a member of the network, respectively. In many networks, a node joins the network and stays forever, whereas in others it can leave the network or break a relationship at some later point in time. Hence, we consider it mandatory to enter the Start Time for temporal analysis, but the absence of an end time is understood to be the current time by default. Note that the absence of a timestamp in any of the rows does not affect any existing functionality and hence backwards compatibility is guaranteed. This means that all of NodeXL's features can still be performed with NetEvViz in the same way. Currently, we have put these two columns as the first two of the user-defined columns and we proposed that this be made a part of the mainstream NodeXL template. For this version of our tool, we have designed two main features - *Timeline*, to view the dynamics of the network using a change in graph metrics, and a *Timeslider* for Temporal Comparison. Accordingly, we created a tab in the NodeXL ribbon. These can be seen in Figure 1. The details of the features follow in the next two subsections.

	A	B	H	I	J	M	N
1			Label			Other Columns	
2	Vertex 1	Vertex 2	Label	Label Text	Label Font	Start Time	End Time
3	martinbutts	diesel_lady				10/14/2010 11:01	11/4/2010 20:58
4	rickr2010	diesel_lady				10/14/2010 11:02	10/14/2010 13:09
5	lima57					10/14/2010 11:04	10/14/2010 16:05
6	mmgoc					10/14/2010 11:34	10/14/2010 23:03
7	cellia	spaceyg				10/14/2010 12:58	
8	thedavidbran	dt				10/14/2010 13:03	
9	bethhawkspr	spaceyg				10/14/2010 13:09	10/15/2010 0:04
10	ngordon	ajcgeorgia				10/14/2010 16:05	10/15/2010 13:57
11	tahdeetz	votedeal				10/14/2010 23:03	10/15/2010 14:43
12	timothy_wats	jaseliberty				10/14/2010 23:32	10/15/2010 17:50

Fig. 1. The Edges worksheet of NetEvViz for Twitter dataset with Start and End Times

C. Network Timeline

In order to provide a high level overview of the dynamics of a network, we present the user with a *Timeline* of different graph metrics. The essential idea is to let the user view the change in metrics over time and identify points of interest with respect to those metrics. Graph metrics are different from node metrics in the sense that they present a property of the whole graph instead of components of the graph individually. Examples of such metrics are overall node count, overall edge count, graph density, average betweenness centrality, and average clustering coefficient. We believe that for a user analyzing a new dataset, getting an overview at the highest level is the first

important insight. In this project, we implemented the three most basic metrics for temporal graphs, namely, temporal node count, temporal edge count and temporal edge-node density. The user can compute these metrics and plots by clicking the *Timeline* button on the *Temporal* tab of the NodeXL Ribbon. The user can also select the number of sampling points from the drop down box named *Steps*. Three options are provided - 5, 10 or 15, the first one being the default value. According to the number of steps selected, the tool calculates the values of metrics at the different timepoints. The selection of actual timepoints is a separate problem in itself. One way of going about it is selecting the points of maximum change reflected in the graph. However, time being a continuous quantity, it is a rather ill-defined problem to take such an approach. Another option could be to calculate and plot these metrics at all times where there was a change. However, there are obvious performance issues with this approach. In the absence of an obvious method to figure out the timepoints of importance, we adopted the selection of timepoints at uniform intervals of time. Starting from the start time of the first edge and ending in the last time of the edge, the *Timeline* consists of 5, 10 or 15 equally spaced timepoints. Figure 2 shows the *Timeline* metric values and plots for a dataset consisting of Twitter replies and mentions described in the section on Evaluation.

D. Timeslider

NodeXL provides a set of dynamic filtering sliders corresponding to a column, specifying a certain feature of the nodes or edges in the network. These double-headed sliders allow the users to interactively filter out nodes and/or edges from the network by selecting the ranges of different features. In NetEvViz, we introduce *Timeslider*, which is implemented based on the built-in dynamic filter sliders in NodeXL. Similar to the above-mentioned sliders, the *Timeslider* is also a double-headed trackbar which allows users to move its two handles to select a time period. A major difference is that the *Timeslider*'s values are drawn from the contents of both the *Start Time* and the *End Time* columns. Thus, the range of the *Timeslider* would be from the earliest time stamp in the *Start Time* column to the latest time stamp in the *End Time* column. We denote these two time points t_{min} and t_{max} , respectively.

By selecting two time points of interest, the user can visualize the differences of the network at the two specified time points. The two selected time points are denoted by t_{lower} and t_{upper} , respectively. Figure 3 shows how the *Lower* and *Upper Selected Time* divide the whole time range into three periods. Each edge of the network can be categorized into one of the six disjoint sets based on the relationship between their start and end times with the lower and upper selected times.

Let t_s and t_e denote the start and end time of an edge. The six categories of edges are as follows:

- 1) $t_{min} \leq t_s < t_e < t_{lower}$: Set of edges that were added and deleted **before** the lower selected time t_{lower} .

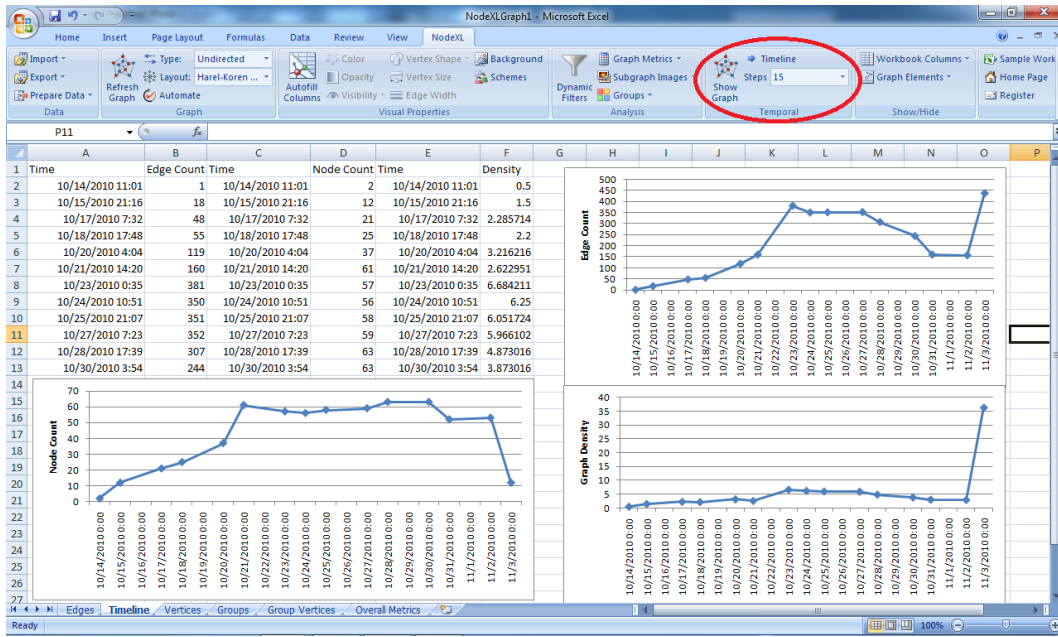


Fig. 2. A Timeline of fifteen steps for the Twitter network, showing plots for edge count, node count and edge:node density. The *Temporal* tab in the NodeXL ribbon is circled in red.

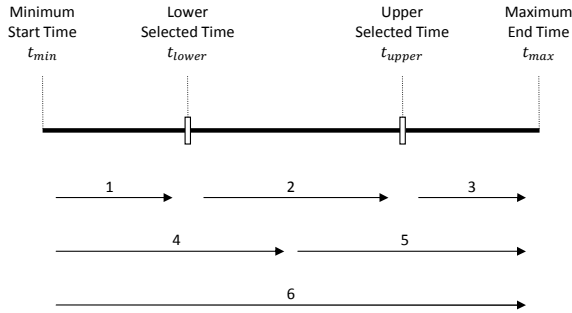


Fig. 3. The *Timeslider* with two selected time points dividing edges into six categories

- 2) $t_{lower} < t_s < t_e < t_{upper}$: Set of edges that were added and deleted **between** (but not inclusive) the lower selected time t_{min} and the upper selected time t_{max} .
- 3) $t_{upper} < t_s < t_e \leq t_{max}$: Set of edges that were added and deleted **after** the upper selected time t_{max} .
- 4) $t_{min} \leq t_s \leq t_{lower} < t_e < t_{upper}$: Set of edges added before the lower selected time t_{min} and deleted between (but not inclusive) the two selected time points.
- 5) $t_{lower} < t_s \leq t_{upper} < t_e \leq t_{max}$: Set of edges that were added between the two selected time points and deleted after the upper selected time t_{upper} .
- 6) $t_{min} \leq t_s \leq t_{lower} < t_{upper} \leq t_e \leq t_{max}$: Set of edges that were added before the lower selected time t_{lower} and deleted after the upper selected time t_{upper} .

In order to help the user distinguish different sets of edges described above, we add four buttons to let the user choose the colors to represent four different types of edges (Figure 4). In principle, since there are six categories of edges, one can use

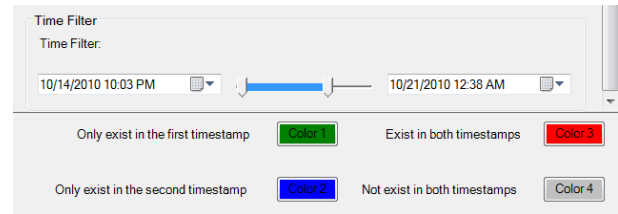
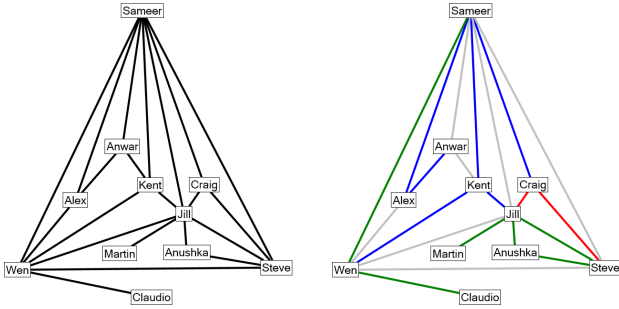


Fig. 4. The *Timeslider* with double headers to select two time points. Four buttons select colors for different edge categories

six colors. However, since edges in categories 1, 2 and 3 do not exist in the network at either of the two selected time points, we decided to use the same color to represent edges in these categories. By default, this group of edges is colored in gray and denoted by “Does not exist in either timestamps” as shown in Figure 4. The other three edge categories are represented by three different colors to be chosen by the user. The default color for category 4, “Only exists in the first timestamp”, is green, for category 5, “Only exists in the second timestamp”, is blue and for category 6, “Exist in both timestamps”, is red.

To illustrate, we use an example of a network consisting of 11 nodes (people) and 21 edges (friendship). Each edge in the network consists of start and end time ranging from 14-Oct-2010 12:00 AM to 25-Oct-2010 12:00 AM. Figure 5a shows the visualization of the network using NodeXL. Figure 5b shows the visualization of the network with the same layout when the *Timeslider* is used and the two time points selected are 14-Oct-2010 10:03 PM and 21-Oct-2010 12:38 AM, respectively. In this new visualization, we can clearly see edges in different categories with respect to the two selected time points. The set of edges which only appear at the first time



(a) Original NodeXL without the temporal color coding ability (b) Visualization in NodeXL using NetEvViz's Timeslider

Fig. 5. A temporal network drawn without and with a time based edge coloring scheme

point is coded in green containing 6 following edges (Wen, Sameer), (Claudio, Wen), (Martin, Jill), (Jill, Steve), (Jill, Anushka) and (Steve, Anushka). There are only two edges which appear in both time points, (Craig, Jill) and (Craig, Steve), and they are coded in red. The blue edges are the ones which only appear at the second time point. This set contains 6 edges (Alex, Anwar), (Alex, Sameer), (Wen, Kent), (Craig, Sameer), (Sameer, Kent) and (Jill, Kent). The remaining edges are coded in gray, which indicates the set that does not exist at both time points.

IV. USABILITY SCENARIOS

In this section, we provide a few examples to demonstrate *NetEvViz* for dynamic social network exploration.

A. Dataset

The dataset used in our evaluation was collected from Twitter using the NodeXL's Twitter Search importer [12] focusing on the toss-up gubernatorial races in the 2010 United States midterm elections in Georgia between Nathan Deal and Roy Barnes. The importer collected all Twitter users who included a candidate's full name in their Tweet (e.g., "Nathan Deal") or alternate variations of their names (e.g., "NathanDeal"). We considered each Twitter user a node and a *reply-to* or *mention* relationship as an edge in our network. The choice of including only *reply-to*'s and *mention*'s but not the *following* relationships was due to the fact that only *reply-to* and *mention* relationships have the exact time stamps. The data were collected on three days: a) October 12, three weeks before the election, b) Nov 2, on the election day and c) Nov 4, post election day. In total, there are 1188 edges and 1015 nodes in the network.

Since our visualization is designed specifically for networks which have edge additions and deletions, the piece of information for the edges' *End Time* is a crucial component of the input data. However, in the original dataset there is no notion of "deletion" of a *reply-to* or *mention* relationship. To make the dataset more interesting to look at, we assumed that an edge's *End Time* is a random date after the edge's *Start Time* and the lifespans of all edges are in the range of 3 to 10 days.

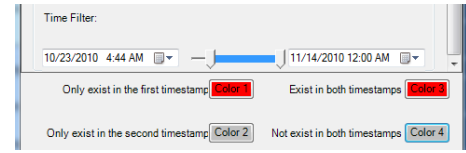


Fig. 6. Configuration of the *Timeslider* for October 23 in Scenario 1

A node's timestamp is that of the first edge that the node has in our dataset.

B. Case Study

Given the *Timeline* presented in Figure 2, the color coding scheme presented in this paper can be used in various ways to visualize temporal aspects of the network. The specific choice of hue or shade a category of edges and the combination of choices for different categories is important in generating a meaningful layout. We illustrate few of the case for the given Twitter dataset below.

1) *Scenario 1*: Find out the the areas with high conversation activity around October 23, 2010 and compare it to the state of the network few days before and after the peak.

We chose the first slider on the Time filter as our date of interest, and the second slider at any arbitrary position. The edges that belong to the categories of first timestamp only and both were chosen to be coded in red. This included all edges that were active at the first timestamp. The other two categories were coded to be in Grey. The idea is to show the valid edges at the chosen timepoint to be highlighted in red against the background of all the dormant edges in red. Figure 6 shows the time sliders for October 23. Figures 7a, 7b and 7c show the layouts in this color coding at October 21, October 23 and October 28, respectively. Figure 7b shows strong activity in the single largest and the second largest cluster and moderate activity in other smaller clusters. Whereas, in Figure 7a and Figure 7c, the activity is only moderate activity in smaller clusters and mild or no activity in the big cluster.

2) *Scenario 2*: Visually compare the activity on October 26 and November 6 for the given network.

In order to show a direct comparison between the networks at two different timepoints, we chose to locate the two sliders at the given times, as shown in Figure 8a. The color of the edges valid only at the first timestamp are chosen to be in Red, whereas the ones valid only at the second timestamp are in Green. Note that colors are chosen to provide a contrast between the edges at two timepoints. The edges that belong to both are neutral and represented in Black, and those belonging to neither are made invisible by selecting a white color in a white background. Figure 8b shows the difference in the concentration of activity at the two given dates using the color scheme mentioned above.

3) *Scenario 3*: What are the conversations that were alive for 5 days (approx.) or more?

In order to find out conversations that lasted longer than a certain duration, the color coding scheme shown in Figure 9a is useful. The two sliders are adjusted so to be apart by 5 days. Only the edges belonging to both timestamps are colored, the

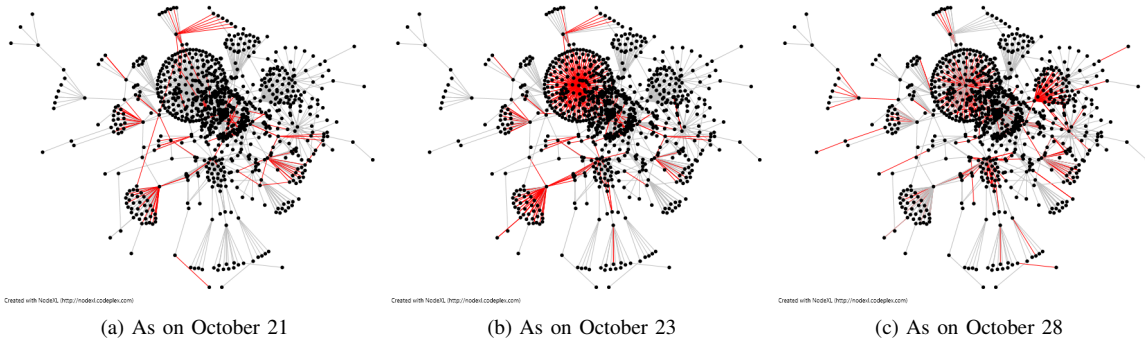
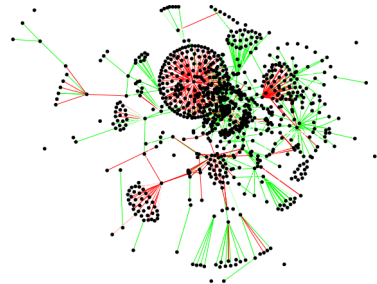


Fig. 7. Network activity on different days, depicted by edges colored in Red



(a) Configuration of the *Timeslider*



(b) Layout depicting activity on October 26 in Red and November 6 in Green

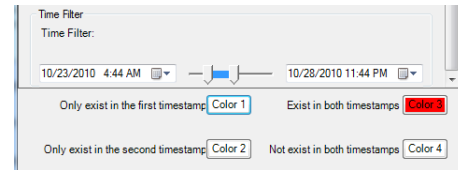
Fig. 8. Comparative activities on two different days using different colors in a single layout

rest being made invisible. After making this selection, we grab the bar between two slider points and drag it slowly along the time axis. This will keep the difference between the two sliders constant (duration) but vary their positions (start and end times). Along the way, one can observe the conversations in color that outlived the current duration indicated by the two sliders. One such instance is shown in figure Figure 9b, corresponding to the slider positions in figure Figure 9a.

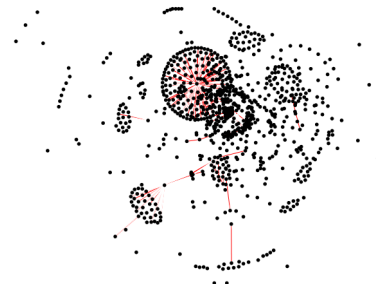
Other scenarios may require a different combination of colors on the *Timeslider*. The scenarios mentioned in this paper are aimed at presenting the basic guidelines on how to make those choices.

V. EVALUATION

In order to assess the effectiveness of our new features to NodeXL, we designed and performed a user study. We first decided on a list of tasks related to temporal network analysis together with a questionnaire for the user to evaluate and make suggestions after performing the tasks. A pilot study was then conducted on one subject to evaluate our designed tasks. Based on the results and feedback from the pilot study, we modified



(a) Configuration of the *Timeslider*



(b) Network layout

Fig. 9. Conversations that lasted longer than five days

the tasks and questionnaire and performed the actual usability test on five subjects.

A. Usability Study

1) *Subjects*: We required all the participants to have experience with both Microsoft Excel and NodeXL. We aimed to design NetEvVis as a tool for expert network analysts. With this requirement, we recruited five graduate students (denoted by P1 to P5) who took a graduate-level Information Visualization course at the University of Maryland. They were trained to use NodeXL for visual network analysis in the class. Among these participants, P1 and P3 were also familiar with Twitter data.

2) *Procedure*: We used the Twitter dataset described in the previous sections for the usability testing. Subjects were tested independently, according to the following procedure:

Introduction & Entry Questionnaire: Before conducting the study, all participants were told in general what they were expected to do. They were then instructed to read and sign the Study Consent Agreement. Then they completed a questionnaire asking about their experience on NodeXL and temporal network analysis.

TABLE I
USER STUDY TASKS

($T_1=2010-10-20$ 4:04~ $2010-10-21$ 14:20, $T_2=2010-10-21$ 14:20~ $2010-10-23$ 0:35, $t_1=2010-10-18$ 20:00, $t_2=2010-10-25$ 21:00)

Task 1: Visualize network statistics over time
1. How many edges were added? (between time T_1 and T_2)
2. How many nodes were added? (between T_1 and T_2)
3. How does the network density change between time t_1 and t_2 ?
Task 2: Visualize the differences of network at different points in time
1. Move the <i>Timeslider</i> and compare the network at time t_1 and t_2
2. Change the edge colors that appear at time t_1 but not at time t_2 to Red
3. Change the edge colors that appear at time t_2 but not at time t_1 to Blue
4. Change the edge colors that appear at both t_1 and t_2 to Green
5. How many edges appear in the network at time t_1 but not at time t_2 ?
6. How many edges appear in the network at time t_2 but not at time t_1 ?
7. How many edges appear in the network at both t_1 and t_2 ?
Task 3: Visualize temporal network with interactive spreadsheet
1. Add some new edges with time stamps to the Edge Sheet
2. List all the edges that appear in the network at time t_1 but not time t_2
3. List all the edges that appear in the network at time t_2 but not time t_1
4. List all the edges that appear in the network at both t_1 and t_2

TABLE II
POST STUDY QUESTIONNAIRE

<ol style="list-style-type: none"> 1. How do you rate your overall performance on the tasks given? 2. I am comfortable using the <i>Timeline</i>. 3. I find the <i>Timeline</i> plots showing the three network statistics (number of edges, number of nodes and density) changing over time useful. 4. I would suggest adding the following network statistics. 5. I am comfortable using the <i>Timeslider</i>. 6. I find changing the color for different types of edges useful. 7. I am comfortable with detecting different types of edges given two selected time points. 8. I am comfortable with changing the color of different types of edges. 9. I find the ability to edit the spreadsheet with visualization is useful. 10. I find the <i>Timeslider</i> handles move smoothly and correctly. 11. Please provide the names of the tools that support temporal network comparisons. 12. My overall impression of the NetEvViz is: 13. I think NetEvViz will be an useful extension for NodeXL. 14. Please leave any comments to the NetEvViz team.
--

Training: All participants were then given a 10-minute tutorial about NodeXL and NetEvViz including: (1) Introduction to the new features of NetEvViz including visualizing network metrics over time using the *Timeline*, comparing network edges at two different time points using the *Timeslider* and editing network data directly from the worksheet; and (2) Two sample tasks: clicking the *Timeline* button to generate the graph metrics and sliding the *Timeslider* randomly to visualize the changes in the graph.

Testing: The participants were asked to perform the three tasks shown in Table I. They were asked to think aloud of what they were doing and report any problems encountered. At the same time, the observer made notes about their actions.

Post Questionnaire: Finally, the participants were asked to fill out a post study questionnaire described in Table II. We used a 9-point scale for measuring each variable. They were also asked to leave comments about their experience.

B. Results and Analysis

1) **Task Completion Analysis:** The main session of the study was to perform three tasks (each contains three to seven

subtasks as shown in Table I). The average time spent on each task was 6, 15, and 10 minutes for Task 1, 2, and 3 respectively. It took all the participants from one to three minutes to complete each subtask. They spent a large portion of the time reading the tasks, understanding the meaning of each metric generated by the *Timeline* and selecting specific edges in the graph visualization.

P1 finished all the tasks successfully. However, P1 had some trouble understanding the function of the *Timeslider*. P1 thought that the *Timeslider* defined a time range, not two specific time points. P1 expressed an opinion: “I think you should use two single sliders instead of a range slider.” P2 performed the tasks quickly. P2 thought that the concept of using a *Timeslider* for temporal network analysis was novel. P2, however, encountered a problem with selecting the specific time in the *Timeslider*. P3 did not finish task 3 because P3 was not able to select edges and nodes in the graph visualization. The selection function, which does not support selecting edges, is implemented in NodeXL, so we assume this is an existing problem in NodeXL. P4 did a careful study and provided many useful suggestions, especially in color selection and coding. P4 mentioned that the labels should be attached to the color button. P4 also commented on the color coding of the visualization. P4 said “It is very difficult to detect the edges that are activated during specific time period. So it would be much better to filter out gray edges.” We agree with her suggestion and expect to make this change in the next release of NetEvViz. P5 also liked the idea of using colors to compare the temporal network at different time points. P5 also thought that it was interesting to use the *Timeline* to generate and compare network metrics over time. However, P5 thought it would be more convenient to add the selection of date and time in the Excel worksheet so that when editing the times of edges, users did not need to manually enter the full text.

2) **Questionnaire Analysis:** After the test, participants were asked to fill out the post-study usability questionnaire shown in Table II. The questions cover the overall performance, the usability of the *Timeline* and *Timeslider* and the effectiveness of the color coding. The average score (Figure 10) is 6.38 out of 9 ($SD=2.23$). Among the questions, users have highest evaluation for Q6: “I find the feature of changing the color for different types of edges useful” (8.4 points); Q8: “I am comfortable with changing the color of different types of edges” (8.2), and Q13: “I think NetEvViz will be an useful extension for NodeXL” (7.4). The lowest scored questions are Q10: “I find the *Timeslider* functions smoothly and correctly as I move the handlers” (4.6); Q1: “How to you rate your overall performance on the tasks given” (5.2) and Q5: “I am comfortable with using the *Timeslider*” (5.4).

From this questionnaire and the comments of participants, we can conclude that users in general had a positive experience in using NetEvViz. They found it useful to use NetEvViz to visualize and compare temporal networks at two different time points. They were especially in favor of the feature of color selection and comparison. However, they also reported that the

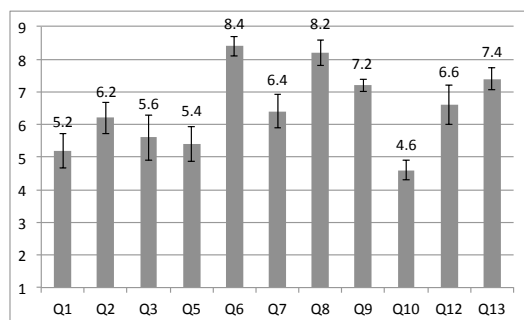


Fig. 10. Summary of subjective user feedbacks

occlusion of edges made it hard to detect the target edges, and that it was difficult to select the exact time in the *Timeslider*.

VI. CONCLUSION

In this work, a prototype is built upon NodeXL to explore the potential of using visualization to help the analysis of dynamic networks. The prototype contains two new features, the *Timeline* and the *Timeslider*. The *Timeline* gives the user a summary of the network over time, which allows the user to obtain an overview of the changes in the network. *Timeslider* allows for precise comparison of the differences in the network, which are highlighted with colors, given two timestamps which are specified by the user.

A user study was conducted to confirm the usefulness of our prototype. Five participants with prior NodeXL experience were recruited. The study follows the procedure of training, pre-study questionnaire, performing predefined tasks and post study usability questionnaire. Summarized from the feedback of the user study, the users had a positive experience and found it is useful to use our NetEvViz features to visualize and analyze temporal network data with Excel. Users acknowledged the novelty in the idea of using *Timeslider* to select the time point and then compare the network instances in the same layout. The users also liked the *Timeline* and the ability to dynamically edit edges and nodes in the spreadsheet despite some inconvenience in the user interface and color coding, which were duly noted as software bugs.

NetEvViz is a first step to understanding how NodeXL can be used to analyze dynamic networks. Future work includes:

- More sophisticated metrics (e.g. clustering coefficient) can be added to the *Timeline* beside the three metrics we implemented. The challenge lies in the computation overhead, which may have to be addressed specifically.
- The *Timeslider* can be extended to color the nodes as well as edges. Also, it would be interesting to look at visual properties beside color (e.g. shape or opacity).
- Support the analysis of egocentric network by allowing the users to specify the vertex they are interested in.

VII. ACKNOWLEDGMENTS

We appreciate the support of Microsoft External Research and Mr. Tony Capone for the development of NodeXL and helping us extend the codebase, and National Science

Foundation grant IIS0968521, plus National Science Foundation/Computing Research Association Computing Innovation Fellow, Postdoctoral Research Grant for Jae-wook Ahn. We are also grateful to Dr. Derek Hansen for sharing the dataset.

REFERENCES

- [1] "Nodexl: Network overview, discovery and exploration for excel," <http://nodexl.codeplex.com>.
- [2] J. Ahn, M. Taieb-Maimon, A. Sopan, C. Plaisant, and B. Shneiderman, "Temporal Visualization of Social Network Dynamics: Prototypes for Nation of Neighbors," *Social Computing, Behavioral-Cultural Modeling and Prediction*, pp. 309–316, 2011.
- [3] J. Ahn, C. Plaisant, and B. Shneiderman, "A task taxonomy of network evolution analysis," Human-Computer Interaction Lab, University of Maryland, Tech. Rep. HCIL-2011-09, 2011.
- [4] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski, "Visualizing time-oriented data—a systematic view," *Computers & Graphics*, vol. 31, no. 3, pp. 401–409, 2007.
- [5] J. F. Allen, "Maintaining knowledge about temporal intervals," *Commun. ACM*, vol. 26, pp. 832–843, November 1983.
- [6] A. Aris, B. Shneiderman, C. Plaisant, G. Shmueli, and W. Jank, "Representing unevenly-spaced time series data for visualization and interactive exploration," *Human-Computer Interaction-INTERACT 2005*, pp. 835–846, 2005.
- [7] R. Becker, S. Eick, and A. Wilks, "Visualizing network data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 1, pp. 16–28, March 1995.
- [8] S. Bender-deMoll and D. McFarland, "The art and science of dynamic network visualization," *Journal of Social Structure*, vol. 7, no. 2, 2006.
- [9] U. Brandes and S. Corman, "Visual unrolling of network evolution and the analysis of dynamic discourse," *Information Visualization*, vol. 2, no. 1, pp. 40–50, March 2003.
- [10] K. Durant, A. McCray, and C. Safran, "Modeling the temporal evolution of an online cancer forum," in *1st ACM International Health Informatics Symposium (IHI 2010)*, November 2010, pp. 356–365.
- [11] P. Gloor, J. Krauss, S. Nann, K. Fischbach, and D. Schoder, "Web science 2.0: Identifying trends through semantic social network analysis," in *Computational Science and Engineering, 2009. CSE'09. International Conference on*, vol. 4. IEEE, 2009, pp. 215–222.
- [12] D. L. Hansen, B. Shneiderman, and M. A. Smith, *Analyzing Social Media Networks with NodeXL: Insights from a Connected World*. Morgan Kaufmann, 2010.
- [13] N. Henry and J.-D. Fekete, "Matrixexplorer: a dual-representation system to explore social networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 677–684, 2006.
- [14] —, "Matlink: Enhanced matrix visualization for analyzing social networks," in *Human-Computer Interaction – INTERACT 2007*, ser. Lecture Notes in Computer Science, C. Baranauskas, P. Palanque, J. Abascal, and S. Barbosa, Eds. Springer Berlin / Heidelberg, 2007, vol. 4663, pp. 288–302.
- [15] N. Henry, J.-D. Fekete, and M. J. McGuffin, "Nodetrix: a hybrid visualization of social networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 1302–1309, 2007.
- [16] K. Lerman, R. Ghosh, and J. H. Kang, "Centrality metric for dynamic networks," in *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, ser. MLG '10. New York, NY, USA: ACM, 2010, pp. 70–77.
- [17] J. Moody, D. McFarland, and S. Bender-deMoll, "Dynamic network visualization," *American Journal of Sociology*, vol. 110, no. 4, pp. 1206–41, 2005.
- [18] M. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.
- [19] —, *Networks: an introduction*. Oxford Univ Press, 2010.
- [20] G. Palla, A. Barabási, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, 2007.
- [21] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*. Cambridge university press, 1995.
- [22] J. S. Yi, N. Elmqvist, and S. Lee, "TimeMatrix: Analyzing Temporal Social Networks Using Interactive Matrix-Based Visualizations," *International Journal of Human-Computer Interaction*, vol. 26, no. 11, pp. 1031–1051, 2010.