Twitter Trendfinder: Visualization and Analysis of Twitter Trends

Peixin Gao†, Milad Gholami∗, Chris Imbriano∗, Mahfuza Sharmin‡, Jin Sun∗
University of Maryland, College Park, USA
gaopei@umd.edu†, mgholami@cs.umd.edu∗, imbriano@umd.edu∗,
msharmin@umiacs.umd.edu‡, jinsun@cs.umd.edu∗

Abstract—With the rapid growth of social networks, people started to realize the influence of the social networks like Twitter in terms of information diffusion over the networks. Numerous researches on social network analysis have been conducted exploring the information diffusion models and influence patterns. Previous works on finding influence patterns in social networks mainly focused on building statistical models using tons of training data. However, many details and key characters of the information diffusion process may be eliminated via the statistical approach of analysis. In this project, we investigate and analyze the influence pattern behavior directly through visualization. We build a tool to show tweet-retweet interaction graph over time on Twitter data with purpose to find most influential user for a given topic. Usability test has been conducted and helped improving our design.

Keywords—Social network analysis, information visualization, information trend, influence pattern

I. INTRODUCTION

The Internet has spawned different types of information sharing systems, including the Web. Recently, online social networks have gained significant popularity and are now considered as the most popular sites. Unlike the Web, which is largely organized around content, online social networks are organized around users. Participating users join a network, publish their content, and create links to any other users with whom they associate. The resulting social network provides a basis for maintaining social relationships and spreading information.

Twitter is one of the most famous and successful social networks, offering online social networking and microblogging service that enables users to send and read “tweets”, which are text messages limited to 140 characters. Social networks, especially one like Twitter, not only connect people together, but also play as media to disseminate information like news and messages. A lot of research has been conducted on investigating the role of Twitter as a social media, analyzing the difference of Twitter with traditional media, developing the information diffusion models within Twitter. The research is mainly about three aspects: (i) which pieces of information or topics are popular and diffuse the most, (ii) why and how information is diffusing, and will be diffused in the future, (iii) which members of the network play important roles in the diffusion process.

There are three important definitions related to the information diffusion models. The first one is social influence, a phenomenon that the social behavior of user is induced by his/her connections and vice versa. Such influence appears explicitly when someone retweets someone else’s “tweet”, for example. The second definition is herd behavior, a social behavior when a sequence of individuals make an identical action, not necessarily ignoring their private information signals. And the last one is called information cascade, which is a behavior of information adoption by people in a social network resulting from the fact that people ignore their own information signals and make decisions from inferences based on earlier peoples actions.

Information diffusion process in Twitter can be summarized as a piece of information carried out by messages spread along the edges of the network. Pal et al. developed a non-graph based, topic-sensitive method, where a set of nodal and topical features are defined for characterizing the network members. Using probabilistic clustering over this feature space, they rank nodes with a within-cluster ranking procedure to identify the most influential and authoritative people for a given topic. Weng et al. also developed a topic-sensitive version of the Page Rank algorithm dedicated to Twitter, which is called TwitterRank. H Kwak et al conducted a quantitative study on the entire Twittersphere and information
diffusion on it, where three ranking systems are used to identify influential users. They found a non-power-law follower distribution, a short effective diameter, and low reciprocity, which all mark a deviation from known characteristics of human social networks. However, these methods only exploit the topology of the Twitter network, yet ignore other important properties, such as nodes features and the domains of the topics spread in the network. No individual can be a universal influencer, and influential members of the network tend to be influential only in one or some specific domains of knowledge.

We argue that in order to understand the information dissemination process in Twitter, detail information like types of trends and features of accounts spreading the trends should be taken into consideration. In this project, we build up a visualization tool to visualize the tweet-retweet pattern of different trends in Twitter. The tool can be used to discover the user(s) who are responsible for initiating the trend and/or for propagating the trend, it can also be used to verify our idea that the detail information of both accounts and trends' topics matter in the information diffusion process. We also try to find a better way to determine the correlation between hot trends and correspondingly influential accounts via the visualization tool.

II. MOTIVATION

As we mentioned above, most of the current research on the topic of information distribution and influence diffusion is mainly about global analysis which may fail in discovering a lot of detail information. People will not be able to find the evolution patterns of a specific trend from macroscope analysis. Identifying influential patterns in Twitter will be more meaningful if it is explored on each topic separately. Because the set of people who are considered as important with respect to one sector might not the same as people who are influential in another sector.

With the visualization tool developed in our project, the microscopic information of details can be seen and more insights about influence flow pattern in Twitter can be found. To the best of our knowledge, this kind of visualizing attempt has not been taken in previous work.

The visualization tool can be useful in various situations, for example, social scientists or other media people might be interested to detect who are the “big guy” in a particular sector. With similar information, the advertisement agencies are able to conduct advertising of high accuracy and increase profit. Also social scientists can ask whether trend-initiating people can be location specific, sex-specific etc. Security people can use the tool to watch for any revolution, any kind of anomaly detection in advance by watching them. Another contribution of this project is categorizing the trend by the importance of user vs. that of topic. The broadly retweeted trends may be due to accounts of popularity, they may also caused by the importance of information carried by the tweets. The difference between these two types could be identified from our visualization tool.

III. RELATED WORKS

A significant amount of study has been done in the area of Twitter data analysis and their visualization. These works focused on different aspects and we discuss them in the following sections.

A. Social Network Analysis: Users and Tweets

Aarts et al [1] analyzed characteristics of users, messages communicated among them, and networking behaviors or activities. Some of their interesting findings are: high buzz is associated with large number of retweets, retweets are correlated with follower counts etc. Ota et al [12] proposed a graph based method to find users who propagate tweets of matching interest. And they visualized that graph (overlap graph) to check the frequency and content of retweets for validation purpose. Analysis of group behavior and growth patterns based on twitter usage was conducted by Krishnamurthy et al [7].

The characteristics of tweets and their flowing patterns and disseminations is also a concern for study. Suh et al [16] applied machine learning (PCA and GLM) to analyze the features of tweets. In another study temporal changes of retweets was conducted [17]. Another large scale study on deleted tweets was done to examine several aggregate property like the connection of those tweets with users, comments, geotagging information etc [2]. Wong et al [18] conducted an interesting study analyzing how twitter users can differ from larger online population with respect to movie ratings and trying to to answer whether or not Twitter data is a good predictor for box office (movie) success.
B. Influential Accounts within Social Networks

Hao et al [5] proposed a method to discover influential users by modeling user relations as a graph and using community detection algorithms to classify users with regard to their influence. Cappelletti et al[3] ranks users on twitter based on their information amplification potential. The potential is defined with two factors: one indicates the tendency of a user to be retweeted or mentioned, second is proportional to the size of the audience, i.e. the number of followers. The ranking algorithm is similar to that of PageRank.

Li et al [10] addressed this problem from a game theory approach. They defined standards of influential users by set of parameters such as message quality, message generation rate, following tactics and following rate. This paper reveals the strategy to become an influential user. A game theory model is used to simulate the evolution of the network and the spread of the influence. Having also built a graph to represent user interactions in social network, Han et al [4] takes a different perspective in modeling the information flow and identifies influential users using random walks. The intuition is that influential mobile users may be visited more frequently by random walks initialized from different small group of users.

C. Trend Detection and Monitoring

Trends are topics that involve a significant amount of users discussions. One interesting question is when does one topic become a trend, and how? Mathioudakis et al [11] proposed a method to detect trend over twitter stream in real time. Trend is found by detecting and grouping bursty keywords (“keywords that suddenly appear in tweets at an unusually high rate” [11]). This is closely related to our task which is to retrieve the moment when a topic becomes a trend over time.

D. Visualization Tools

Conference Monitor is a tool which provides useful visualization of hashtags over time [15]. Conference Monitor uses backchannel conversation to identify popular sessions, important participants, and trending topics. Similarly, Statler developed by Yahoo labs examines data corresponding to broadcast events [13]. TweetInfo visualized the summary of twitter events highlighting peaks of high tweet activity over time [14]. Nokia Internet Pulse offers visualization of current discussion over time around particular topic using tag clouds [6]. SparkClouds also uses tag clouds to convey trends among multiple tag clouds [8]. TopicFlow [9] is another tool that visualizes topic detection and topic alignment between sets of tweets over time. Social Media Importer [20] is an extension for NodeXL [19], which is a software package for Microsoft Excel. SMI can be used to import data from social networks like Facebook and Twitter. NodeXL is a powerful tool for visualizing and data analysis. So, tweets can be imported from Twitter and different kinds of graph can be generated for analysis. However, NodeXL can only generate static graph and have limited dynamic interactions. The data representation, visualization and the corresponding analysis that we are offering in our tool (described in subsequent sections) can not be achieved using NodeXL. In addition to that, in NodeXL the nodes are users and it shows how topic flows. However, our goal is to watch how the user influence an action, so better representation for visualization would be actions as nodes and showing flow from user to user.

IV. DATA PROCESSING AND ARCHITECTURE

Twitter TrendFinder is a web based interactive tool to visualize tweets of one particular topic. We collected original tweets data using Twitter Stream API via a Ruby wrapper. Twitter Stream API offers way to sampling from the public data flowing through Twitter. A ruby script is created to send request through API to track 100 tweets about topic “#ObamaCare” for a short time period. Tweets are stored in JSON file and build in a small collection. One limitation of the API is that it can only sample a limited number of recent tweets. Most tweets we collected appear to be either singletons, i.e. do not retweet or reply any other tweets, or have only one-to-one connection among others. Furthermore, Stream API has limited access to historical data. The tweets we collected are mostly created near the time tracking was done. All these aspects limit our ability to explore more interesting structure from the data using designed visualization. Ideally we would expect to have tweets with relative long time span and more complex network structure (e.g., tweet has multiple retweet/reply). To achieve this, one solution
is to keep tracking the same topic for a much longer time period, for example three months.

We wrote a Ruby script to create a synthetic dataset that would exhibit some interesting features that we did not see in our small collections. The small datasets did not have tweet "chains" longer than two; that is, there weren’t any instances of a retweet being retweeted or of a reply being replied to. Further, the density of tweets was very high for the period of time we were collecting and quite sparse for all other times. Our script then creates a random number of reply chains, retweet trees and ignored tweets for a period of up to 7 days in the past. The script gives each tree an increasing probability of termination at every subsequent layer, so that they didn’t get excessively long.

While not a perfect representation of real user data, the synthetic data set allowed us to test our interface with a larger number of nodes and more interesting node relationships. These features exposed problems with occlusion and data density, most of which were subsequently addressed in the development of our tool.

Whether obtained via Twitter API or produced synthetically, the data is then ingested into a PostgreSQL database. PostgreSQL is an enterprise class open source object-relational database management system. In conjunction with the database, we developed a Ruby on Rails engine to serve the data out of our database and communicate with front end. The Rails engine had a variety of URL endpoints which would respond with data in JSON format.

1) /graph.json: Full dataset in standard node/edges format
2) /tree.json: Full dataset in alternate tree format
3) /tweets/:id.json: Details for tweet with :id

The front end runs in the web browser like Chrome, Mozilla Firefox etc. We choose web interface as our front end to reach comparatively wider set of audience. The engine serves the tweet data in json format. The engine can provide data in two formats. One format is graph representation of tweets, another format is individual tweets. In graph representation, each tweet is represented as node containing basic information (tweet id, created time of tweet etc) regarding the tweet.

Individual tweet format gives full information of each tweet including user details. The reason for keeping two different format is to provide the minimal amount of data to serve each component. For initial graph drawing we do not need all the information. If the graph contains every single details that we included in our analysis, the data size will be huge. This can make the initial data loading slower. To make the data load faster we keep the graph size minimal. So graph.json should be the minimal amount of data in order to produce the “overview” visualization. We don’t need the specific tweet details for every node in the network in this view. One a node is click though, then we want to see the details but just for that specific node, not all of them. Hence as the user wants to inspect more, information of single or subset of tweets can be provided on demand through second representation. Both of the data representation is sent as json format to the front end and we have URLs that serve various amounts of data. A sample tweet data is shown below.

"tweets": [{
  "id":1,
  "text":"RT@LadyVeteran23:ObamaCare",
  "created_at":"2013-11-02T19:58:50",
  "twitter_id":"396728123380609024",
  "retweeted_id":"396723459423608833",
  "in_reply_to_status_str":null,
  "created_at_numeric":1383422330,
  "url":"localhost/tweets/1.json"
}]

Here “id” is the identifier we assigned for each tweet. “text” contains the content of the tweet. “created_at” shows the time this tweet was created. “retweeted_id” is the id of the tweet being retweeted by this tweet. This field is critical since we need it to trace back to the very tweet which started the current trend. These four fields are what we need to build our primary visualization.

After importing the json file to our D3 visualization engine, we are able to draw the graph such that each node is an individual tweet and each link represents the relation between tweets (retweeted/reply).

V. OVERVIEW OF DESIGN

Our front end uses CSS and JavaScript to communicate via Ruby on Rails engine, displays visu-
alizations and allows the user to interact with the data. The visualization is created using Data Driven Documents (D3).

Twitter Trendfinder visualization follows the Visual Information Seeking Mantra: overview first, zoom and filter, then details-on-demand. First by default we show the overview picture of tweets occurrence over time. Then the brushing enables user to zoom and focus in a specific time period for closer investigation. Finally, by clicking on the tweet node, detailed information is presented in the information panel. The interface is illustrated in Fig-1.

A. Graph View

Tweets are represented as nodes in graph with three types including independent tweet, reply and retweet. Edges are drawn between nodes for following two cases:

1) if tweet 2 is retweeted from tweet 1, we add an edge from node 1 to node 2.
2) if tweet 2 is a reply of tweet 1, we add an edge from node 1 to node 2.

The graph is drawn using force layout in D3 (Region 1 in Fig-1). The x-coordinate of the tweet is set by the time it was created. The y-coordinate is automatically calculated by algorithm used in force layout.

In the graph view we have added interactive functionalities. For example, user can select a specific node. On selection of a node (representative of a tweet/reply/retweet) user can see the detailed information of that tweet. Also, to see the trend more clearly from the graph, we highlight the whole subgraph of which the node is part of (see 2 of Fig-3). The highlight of subgraph is important because this shows how much reply/retweet has been made from the selected tweet. Thus a clear impression is made about whether the corresponding user is a potential influential person or not.

In situations that a lot of tweets are created in a short time period, the nodes density is high and nodes get occluded. For better arrangement of the nodes we implemented force layout for the visualization. Even with the force layout the graph can be difficult to inspect with large number of tweets. That's why we have added brushing in both x and y-axis (Region 2 and 3 in Fig-1). Context+Focus with Brushing helps to see the dense graph closely by selectively zoom in area of interest. As we move the brushing towards left (right) the nodes/tweets in the left (right) becomes invisible.

B. Information Pane

The Information Pane serves the purpose of details on demand. It contains three components: Tweet Details, User Details and Top Tweets (Region 4, 5, 6 in Fig-1 respectively). “Tweet Details” shows information about tweet, e.g. tweet text, tweet creation time, tweet type (reply/retweet). “User Details” serves the goal to find who plays “powerful” role for spreading messages. This component displays how many friend and followers the user have, where the tweet has been made etc. “Top Tweets” lists top 10 tweets which has highest number of immediate retweets/reply. Clicking on one of the item in top tweets list will highlight the corresponding tweet node in the graph.

C. Control Panel

We include a control panel to be able to filter tweets (Region 7 in Fig-1). There are two types of filtering in our implementation. We can show/hide tweets which are singletons (no one retweeted and no one replied) and/or tweets which are either retweeted or replied by other tweets. This function is provided by the option “Tweets with Edges”. The other filter option is to show only reply tweets or retweet tweets. We also have a search box to accept user name as input and show corresponding tweets.

VI. Evaluation and Results

The purpose of our evaluation is to gauge the usability and ease of navigation of the Twitter Trendfinder web site for users. To evaluate our tool we conducted usability test. The usability test consisted of a session of 5 minutes instruction, a set of pre-designed tasks to be done by test subjects, and feedback from the users. Our target users are social scientists, social media related IT companies, and curious citizens. Because of time limit, we selected 7 graduate students as our test subjects. Among them 50% were male and 50% were female, also 60% were from CS department and 40% were from statistics department. The experience of earlier Twitter usage of the test subjects was also 50-50, i.e. half of them used Twitter and half of them did
not use Twitter earlier. For some test subjects we conducted remote usability tests and for others the test was taken online.

We started the test with a brief introduction and explanation of the tool. Then we allowed the participants to explore the site for a couple of minutes to prepare themselves ready for the test. We asked them to finish seven tasks.

The tasks designed for the usability test are as follows:
1) Given the a certain date D, find the tweets about the trend A and the amount of the tweets.
2) On what date the trend A started to blow up.
3) What’s the peak (maximum amount of tweets per day) about topic A.
4) Who is the person that started the trend A.
5) Given a certain tweet, find its children, and tell if they are retweets or replies.
6) Find all the singletons on a certain date.
7) Zoom in to a certain date.

Here, tasks 1-3 are designed to see whether the users can extract necessary information from the visualization of the tool. Tasks 4-5 focus on the usability of the information pane and tasks 6-7 takes care of the usefulness of the control panel. After finishing the tasks we check the correctness of the answers, which is one of the important measures of the usability of our design. We found that 3 out of 35 answers were incorrect. Their incorrectness stemmed from the fact that some nodes are hidden behind other nodes, also nodes can not be moved. Node movement can reduce the errors, but one careful design consideration here is the fact that x-axis in the plot is time and any node/tweet movement should be restricted to only along y-axis. This might be irritating for the user. The quick solution we come up with is to color code the nodes and assign transparent colors. The use of transparent color made it easier to understand whether there is any other node along z-axis. In the color coding, green nodes represents retweets, red nodes are reply and orange ones are independent tweets (singletons).

During our usability test, we also measured the time that the participants took to complete all the tasks. The average completion time was around 12 minutes. After finishing all the tasks, the participants were asked to fill in a feedback form of twelve questions. Some of them are Likert Scale style questions, others are specific feedback about our tool. The follow-up questions are listed in the following.

1) Do you like the interface design?
2) Is the interface intuitive?
3) Can you find the information you need from the interface easily?
4) Is the interface more informative than literature?
5) Do you like the color scheme used in this website?

These are 10-point Likert Scale questionnaire to capture users' responses. The corresponding feedback is presented in figure 2. The bar chart shows how much the user rated our design, interface, color scheme etc. In a 1 to 10-point scale, for all the questions below the average likeness is around 6 to 8 (blue bars). Red bars in the figure indicates the low variance of the fact that most users liked the different components of the tool and their feelings about the tool do not vary too much.

The other questions for feedback are listed below.
1) What features of the Twitter TrendFinder web site were vague or confusing to you, if any?
2) What is your impression about navigating the site? Does it seem easy or difficult? What makes it that way?
3) What else should be included on the Twitter TrendFinder web site?
4) What did you like best about the site?
5) What did you like the least?
6) Do you think some people would have problems using the Twitter TrendFinder web site? What kinds of people? What kinds of problems?
7) Would you like to make any other comments about Twitter TrendFinder?

From the second part of feedback we received enthusiastic responses. Most users liked the tool and they gave lots of suggestions and comments. All users found the tool very easy to use, fast, informative, and useful. Some of them expressed their extreme interest to see their own tweets in our tool. None of them faced any problem with color but the users who do not use Twitter took longer time to grasp the idea of our tool. Those users also commented that people who never used Twitter might not find the tool very intuitive.

A. Redesign and Future Work

Based on the results of usability test, we modified our visualization tool in many aspects and made it more useful and attractive. For example, in earlier design we implemented range slider for zooming. One participant did not notice that the slider can be used to zoom. We decided to replace the slider with “context and focus with brushing” 3. The new design shows two components: graph of tweets at the top and volume of tweets at the bottom. Both of them are arranged according to horizontal time scale. User can select a piece of time from volume panel and focus that part in the graph view panel. A vertical brush component is also added for zooming in vertical direction. Additionally, the time axis is now shown in finer scale as another user suggested. We also used color coding to differentiate different kinds of tweets without making any mouseclick.

Due to time limit we could not incorporate everything that our test participants suggested. One of the tasks in usability test required to inspect
specific tweet at specific time. The feedback from user with respect to that task was to make the background of the visualization panel like a “transparent graph paper with large squares”. This will make the inspection easier. In latest implementation, the transparency of the node and the volume panel showing the number of nodes at a particular time, solves the problem of node acclussion. However, allowing mobility of node along y-axis can be worth to pursue in future. Before that another round of usability test should also be considered. Two users opined that this is a useful tool and the designers should consider integrating the tool with Twitter so that data can be fed real time and see how the trend looks like. We also agree that such integration is going to be a worthwhile future work.

CREDITS

All team members were engaged in the discussions and meetings regarding problem formulation and proposing idea for solution. Following are the specific contributions of the team members (Alphabetical order by last name).


Milad Golami: Data processing, Implementation of user interaction with graph.

Chris Imbriano: Implementation of ruby on rails engine, Data processing, Graph visualization, Color coding.

Mahfuza Sharmin: Layout and Control Panel design, Report writing, Running Usability Test.

Jin Sun: Graph visualization with force layout, Implementation of context+focus with brushing, Data processing.

ACKNOWLEDGMENT

The authors would like to thank Dr. Ben Shneiderman, Professor, University of Maryland, Mark Smith and Dr. Jimmy Lin, Associate Professor, University of Maryland for their mentorship and guidance.

REFERENCES


