# A Visualization Interface for Twitter Timeline Activity

Wesley Waldner and Julita Vassileva
Department of Computer Science
University of Saskatchewan
Saskatoon, SK, Canada
{w.waldner, julita.vassileva}@usask.ca

## ABSTRACT

Social media streams are a useful source of current, targeted information, but such a stream can be overwhelming if there are too many sources contributing to it. In order to combat this information overload problem, rather than by filtering the stream, users may be able to more efficiently consume the most impactful content by way of a visualization that emphasizes more recent, popular, relevant, and interesting updates. Such a visualization system should provide means for user control over stream consumption while not excluding any information sources in the stream, allowing users to broaden their source networking without becoming overwhelmed. This paper presents a visualization for the Twitter home timeline that allows users to quickly identify which updates are most likely to be interesting, which updates they have and have not read, and which have been posted most recently. A small-scale pilot study suggests that improvements to the prototype are required before carrying out a larger-scale experiment. The effects of recommendation presentation on subjective measures of recommender accuracy will be studied as future work using this application as a framework.

## Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems – *human information processing*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *information filtering;* H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *user-centered design.*

## Keywords

Recommender systems; Social media; Social visualization

## 1. INTRODUCTION

In public social networks, where status updates can be viewed by any and all users of the system, a social activity stream is a useful tool that can help avoid information overload by collecting in a single location all updates from only those users in one's own social network. Social network users will typically connect with other users they are interested in, and, ideally, their activity stream will therefore consist of updates on topics that match their interest as well. However, it is impossible for all updates to be interesting or relevant to the user. Thus recommender systems can be introduced into social networks to serve two primary purposes. The first is to recommend additional sources of information to the activity stream, which involves adding nodes to one's social network. As the network grows, however, at some point throughput

can become so great that it is impractical to consume every new piece of information flowing through the stream. In addition, the quantity of uninteresting content also increases with the interesting content. At this stage users have the option either to reduce the size of their network, resulting in a stream that is easier to handle, or to risk missing some particularly relevant or interesting updates.

The second common use of recommender systems in social activity streams is to try to avoid this problem by filtering the stream to show only the most relevant updates to the user. The ideal filtering recommender would reduce the stream throughput to a manageable amount, and would consistently predict with perfect accuracy the updates that the user would most like to consume. While it is unreasonable to expect perfection, such filtering mechanisms are intuitively useful in dealing with the information overload problem.

The stream filtering approach, however, has some potentially undesirable side effects [3]. Even if the recommender models a user's interests perfectly, she can become trapped inside a "filter bubble," engineered to match her interests at a particular point in time, but making it difficult to discover potentially new areas of interest. More realistically, the stream is also not being filtered perfectly. In either case, it can be difficult for the user to escape the filter bubble to receive serendipitous updates or expand her interests, especially since most filtering mechanisms do not provide much if any control to the user. When consuming filtered streams, users will also have a skewed perception of activity within their network. As preferences and interests may change over time, so too might the behaviour of other members in the network. If updates from these nodes are being filtered out of the stream, this may have unintended consequences on the user who might be interested in these activities but may never know of them because the nodes lie outside of her filter bubble.

Stream filtering, despite its shortcomings, is a commonly-used strategy for dealing with information overload in social activity streams. However, it is possible to emphasize certain updates without filtering others from the stream completely. In systems that show the entire stream by default without filtering, such as Twitter, each update is normally given equal visual prominence regardless of its popularity, relevance, or interest to the user. Therefore, the passive viewer cannot have any awareness of the popularity or social impact of posts just by consuming the basic stream. As a result, users will need to read each update to determine its relevance, at which point their time already will have been spent. Furthermore, if a user has not visited his stream in a while, he will be unable to catch up on the most important updates from that time period without consuming the entire stream.

A stream visualization that simultaneously depicts all updates from within a specific time range and differentiates between the most popular and impactful ones is a potentially useful alternative to stream filtering, as it allows users to explore more or less deeply depending on the amount of time they have available. By using a multi-dimensional nonlinear visualization that recommends and

emphasizes the most important and interesting status updates for a particular user at a particular time, users will have increased awareness of the most impactful updates in their networks, will be able to consume time-relevant updates more effectively and efficiently without needing to filter their social streams, and will have increased trust in the system compared to a system without emphasis that filters out the least interesting updates.

## 2. BACKGROUND

### 2.1 Social Activity Stream Recommendation

There are a number of differences to consider when recommending for social activity streams versus traditional product recommendations. For one, there is usually a much larger amount of non-redundant data. For example, users may find thousands of social updates relevant at any given time. However, if a system is trying to recommend a new camera, the user is likely to buy only one and then not need any more help. Also, social updates may only be relevant for a very short period of time and may be targeted to a specific audience with special knowledge.

Though precision may be more important than recall in recommendations involving items that require a large commitment of time or resources [1, 7], recall intuitively seems to be more important when evaluating social activity stream recommenders. A small number of uninteresting updates appearing throughout the stream will not cost the user much time, perhaps as little as a few seconds, meaning that a lower level of precision may not cause much harm. Incorrect product recommendations, on the other hand, can have a greater negative effect. For example, if a user purchases an item that turns out not to be a good fit she may not be able to return the item to retrieve the money she spent. Conversely, it is undesirable to miss out on very important updates in a social activity stream, meaning that a lower level of recall may cause a great amount of relative harm. Ultimately, user satisfaction is the most important factor. Social activity streams are similar to subscription services in this way: there are no individual purchases to consider, and they interact with the system many times within a short span. What matters most is that people continue to use the system and have a good overall experience.

### 2.2 Visualization

Social visualization is an important aspect of recommender presentation that goes beyond the context in which items are presented and considers the structure that the presented data takes. When used in conjunction with a recommender system, social visualization can help the user understand how the recommender system is working [6]. There are many examples of systems that allow users to visualize their social networks[1]. These tools often simply map the connections between nodes without taking into account the activity of those nodes. However, previous studies have applied visualizations to the realm of social network activity and social activity streams. Some relevant examples are described in Section 6.

## 3. TWITTER STREAM VISUALIZATION

### 3.1 Main Idea

The main goal of this paper and future related work is to show that a multi-dimensional nonlinear visualization that emphasizes

---

[1] e.g.: http://keylines.com (general); http://mentionmapp.com, http://tweepsmap.com (Twitter); http://socilab.com (LinkedIn); https://immersion.media.mit.edu (email); http://www.touchgraph.com/facebook, https://friend-wheel.com (Facebook);

recommended content in a users' social activity streams will increase user awareness of impactful updates, increase user trust in the recommender system compared to one that employs filtering, and enable users to more effectively and efficiently consume the most relevant and interesting updates in their streams. To this end, we have developed an application that displays data collected from users' social activity streams in Twitter. The visualization represents updates as circles on a two-dimensional display, with different properties mapped to different visual dimensions (see Table 1 for a listing and Subsection 3.2 for full details). Recency and interest level, two important factors in supporting user awareness of the most relevant social network activity, receive the greatest focus and most prominent visual coding. However, to avoid misleading inferences about activity levels, no updates are filtered out of the system in this visualization, regardless of how irrelevant or uninteresting they may seem. In an effort to provide a more usable product, these updates are de-emphasized so as to be easier to ignore if the user so chooses. A content-based recommender learns from user behaviour and predicts the user's level of interest in every new update that appears in the stream. The visualization design supports chronological consumption of stream content, while highlighting the most relevant content to the targeted user and simultaneously depicting rises and falls in activity levels across the user's network.

### 3.2 Visual Design

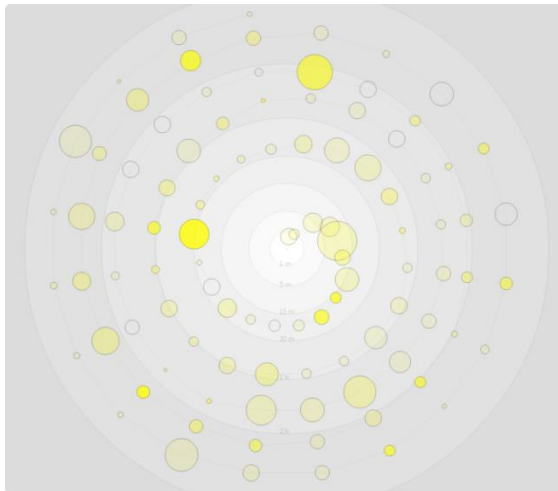#### 3.2.1 Two-dimensional Timeline Visualization

The backdrop for the stream visualization comprises a number of concentric circles about a central point. This point can be thought of as the immediate present. Each background circle, in increasing distance from this central point, represents an older point in time in the past. The distance between circles remains close to constant, but the time represented increases at greater distances from the centre to allow more room at the present where there is less angular spread and where users are more likely to focus their attention in order to read the latest updates. Thus the amount of time since an update was posted is coded in the visualization as distance from the centre. Because of the importance of size in the perception of visual prominence [2], Tweet relevance is coded with circle radius. With this combination of visual mappings, Tweets that are more recent and more relevant to the user will occupy more space close to the central region of the visualization. Appropriate default minimum and maximum values are in place to prevent unreadable results, and users are able to personalize the appearance so that it works best for the throughput level of their stream. More details on personalization options are discussed in Subsection 3.5.4 on the client implementation. The rest of the visual mappings are shown in Table 1.

**Table 1. Mappings between variables and visual dimensions**

| Variable | Visual Dimension |
|---|---|
| Recency | Distance from origin |
| Recommendation strength | Size |
| Popularity | Colour opacity |
| Unread/read | Shape (circle/horizontal line) |

Colour opacity was chosen for Tweet popularity, which is calculated as a normalized sum of the number of retweets and number of favorites. There is some concern that very popular Tweets, even when small due to a weak recommendation value, could dominate visually. However, popularity reflects social impact, which is an important factor for users to understand in order to be socially aware, so popular Tweets should be prominent.

**Figure 1: Two-dimensional stream visualization**

Showing hundreds of complete Tweets onscreen at one time would of course cause overcrowding and would overwhelm the user; this is why circles are being used as placeholders. The actual content of the Tweets is hidden until the user's cursor hovers over one of the circles. On hover, a small card-like element will appear next to the cursor that displays the Tweet's content, including thumbnails of any embedded images, and the Tweet author's user name and avatar. Additionally, there is a linear stream panel that can be docked along the right side of the window. When the user interacts with a Tweet in either view, the corresponding Tweet in the other view (including the circle representation in the visualization) will be highlighted to help draw a connection between the two stream presentations. This may be helpful for a user who is reading the linear stream and wants to see the impact of a particular Tweet in relation to others around it. It also makes it easier to switch back and forth between views at any given time.

### 3.2.2 Linear Textual Timeline Visualization



**Figure 2: Screenshot of the textual linear stream showing all three recommendation tiers**

As mentioned, a textual timeline was also presented to complement the two-dimensional visualized timeline. Here, rather than using a continuous scale, recommendation scores are mapped to three discrete tiers. Tweets in the highest tier are larger in area as well as font size, have a stronger yellow colour, and are aligned further to the left. Tweets in the lowest tier are the smallest, have no colour, and are aligned further to the right, while the middle tier Tweets are in between the two extreme tiers in all qualities. The elements used to represent Tweets in this timeline view are

exactly identical to the cards that pop up in the two-dimensional visualization, both visually and functionally. Tweets are displayed from top to bottom in chronological order, from newest to oldest.

## 3.3 Feature Design

Since this design builds upon the existing infrastructure of a major social network, features are already available to users for communication. However, these existing features have some limitations. A recommender needs a way to infer the utility of a particular item for a particular user. In Twitter, a user's appreciation for a Tweet can be explicitly indicated by a "retweet" or "favorite" action. One potential downside to these built-in actions is that they are completely public: any Twitter user can see which Tweets you have retweeted or favorite, which, depending on the situation, can be an incentive or deterrent to performing those actions. For the purposes of training a recommender, it would be preferable to have private ways of indicating interest for those situations in which a user might not want to publicize her opinion. Twitter also does not provide a way to indicate disinterest in a Tweet. To address these shortcomings, this application provides "like" and "dislike" functions, which are used exclusively to train the recommender. These two actions are denoted by familiar "thumb-up" and "thumb-down" icons.

Another feature, implemented to complement the recommender, is a manual user influence scale, which is shown in Figure 3 as "Relative Volume (User)". Users can manually adjust a recommendation multiplication factor that is effective across all Tweets by a particular member of their follow network by using a slider that can scale their influence up or down. For example, if the minimum influence level is chosen for User A, then all Tweets from this user will be shown as if they were given the minimum possible recommendation value from the recommender system. Similarly, if the maximum level were chosen, all Tweets from this user would be shown as the maximum recommendation level. The scale is quasi-continuous, and the chosen value is used as a multiplier as a final step after the initial value is passed from the recommender system running on the server.

A filtering feature was also added in order to test how trust in is affected when users, rather than the system, have full control over filtering. Users can move two sliders, one labelled "Min" and the other labelled "Max", to select a range of recommendation scores to allow through the filter. Setting the minimum value higher will exclude Tweets with low scores, while setting the maximum value lower will exclude Tweets with high scores.

## 3.4 Implementation Details

### 3.4.1 Overview

The software implementation of this application consists of three basic components: a client, server, and database. The server connects directly to the Twitter API and to the database and sends only the necessary updates to the client, which consists of the graphical user interface and visualization. A full-JavaScript software stack was used to develop the application.

### 3.4.2 Recommender

The recommender system implemented is similar to the one described by Wang et al. [9] to identify the most interesting updates from the Twitter user's home timeline. Users are given the ability through the graphical user interface to rate individual Tweets as interesting or uninteresting by clicking the "like" and "dislike" icons. These ratings are sent to the server and stored so that the recommender can be trained in the future as the user continues to give new ratings. As with any recommender system, more data is better: getting users to contribute ratings is one of the most im-

portant problems in social computing, but in this system users are encouraged to rate more and more Tweets as they see highly-rated Tweets that they are not interested in. These high ratings will appear to the user to be out of place, and with a single click they can be corrected. As new ratings are provided, the recommender will be re-trained and the interface updated; this quick feedback provides additional incentive to the users to continue training.

The recommender uses a naïve Bayes classifier trained using features from the rated Tweets stored in the database to predict whether unrated Tweets are interesting to the authenticated user. Then all unrated Tweets are classified as interesting or uninteresting. Using the Bayesian probability model, the posterior probabilities of the Tweet belonging to each of the two classes is calculated. The overall recommendation score from 0 to 1 is then determined by calculating the probability of the Tweet being interesting *given* the assumption, used for simplicity, that it is either interesting or uninteresting. Then, where $T$ is the Tweet being classified, $I$ is the set of interesting Tweets, and $U$ is the set of uninteresting Tweets, we have:

$$score = P\big(T \in I \big| T \in (U \cup I)\big)$$

Using the conditional probability formula for dependent events, we get:

$$score = \frac{P([T \in I] \cap [T \in (U \cup I)])}{P[T \in (U \cup I)]}$$

Since $T$ can only be an element of $I$ if it is also an element of $U \cup I$, the numerator can be simplified. The denominator can also be expressed as a simple sum because the sets $U$ and $I$ are mutually exclusive by definition. So we have:

$$score = \frac{P(T \in I)}{P(T \in I) + P(T \in U)}$$

In other words, the total recommender score is the ratio of the posterior probability that the Tweet is interesting to the sum of the posterior probability that the Tweet is interesting and the posterior probability that the Tweet is uninteresting. This will result in an average score (close to 0.5) when a Tweet fits equally well into either category and a more extreme score (closer to 0 or 1) when the Tweet fits into one of the two classes exceptionally well.

The following features are included in the classification procedure:

- Content author
- Content retweeter (if applicable)
- All hashtags
- All user mentions
- Tweet type(s): photo, link, retweet, reply, quote, manual retweet, and/or comment
- Number of retweets
- Number of favorites
- Length of text
- Number of numeric digits

The features are all used in an attempt to classify different types of Tweets. For example, a user may be partial to relatively long Tweets containing many numbers and no links that have been retweeted many times. The naïve Bayes classifier treats each feature as independent, however, so interactions between these features will not be accurately represented. A recommender that will take these interaction effects into account is left for future work. It would be interesting to try to classify Tweets based on topic to improve the recommender. Sriram [5] presents some promising work that uses text mining to classify different types of Tweets, while Wang et al. [9] used text mining to improve recommendations with similar machine learning techniques to those used here.

### 3.4.3 Client
HTML5 canvas was considered for rendering the visualization, but elements and event handlers would be easier to manage if each component was a node in the DOM tree. Instead, Scalable Vector Graphics (SVG) technology was used to allow for creation of vector images, which can scale to arbitrary sizes without losing detail. SVG elements are defined using XML and can be used in HTML5 markup just like regular DOM elements. Because all of the graphics are scalable, we added a feature that allows the user to zoom in and out to the position of the mouse cursor by scrolling the mouse wheel. This is perhaps the greatest benefit of using SVG instead of HTML5 canvas. Panning in the visualization is also allowed by clicking on an open area and dragging the cursor in any direction.
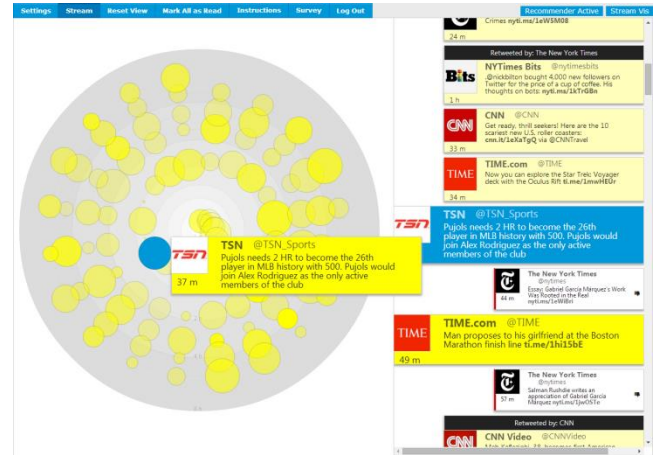


**Figure 3: The client application running in a web browser**

## 4. PILOT STUDY

### 4.1 Goals
Before carrying out a large-scale quantitative study using this visualization tool, a smaller pilot study was necessary to identify pain points, streamline the experimental process, and determine the best way to collect the necessary relevant data. The pilot experiment tested the usability of the system and the appropriateness of the variable coding arrangement. Feedback was gained from the users on the following qualities of the system:

- Usefulness of the visual-emphasis approach to presenting recommendations
- Usefulness of user-controlled filtering feature
- Usability in general
- Sources of particular difficulty

### 4.2 Procedure
Two Twitter users were recruited via Facebook and were required to complete, in order, all of the tasks listed in this subsection.

### 4.2.1 Explore and Rate Tweets
Users were required to rate Tweets to train the recommender. To do this, they were instructed to read through either the textual or visualization timeline in chronological order, rating especially interesting and uninteresting Tweets along the way. Thirty ratings were sufficient to produce what users deemed to be accurate recommendations in a previous small-scale study using only the textual timeline with the three tiers of recommendation strength [8], so the recommender was activated after thirty ratings. At this

point users were to make any necessary adjustments to the default settings now that the size of the Tweets had changed to reflect recommendation scores.

### 4.2.2 Timeline Reading

Users were instructed to traverse their timelines chronologically, reading *only* the emphasized Tweets, first using the textual timeline, and then using the two-dimensional visualization.

### 4.2.3 User Volume

In order to evaluate the usefulness of the User Volume feature, users were instructed to identify some users they wanted to see more or less of in their timeline and then to use the User Volume slider to make that user's updates more or less visually prominent.

### 4.2.4 Filtered Timeline Reading

Finally, users adjusted the Filter settings to test the recommender and visualization's joint effectiveness in another way. First they increased the minimum filter amount to show only the most highly-recommended Tweets, and then they reset and decreased the maximum filter amount to show only the least highly-recommended Tweets.

### 4.2.5 Survey

A link to a questionnaire appeared after the recommender became active. Users completed this survey as the final step in the study.

## 4.3 Survey Responses

The survey consisted of a 20-part questionnaire. The questions were broken down into the following categories:

1. Twitter usage
2. Recommendation presentation
3. Recommender performance
4. Design feedback

The results for categories 2–4 are outlined in the following sections. Responses for categories 2 and 3 were on a six-point Likert scale.

### 4.3.1 Recommendation Presentation

Users were asked the following set of three questions for both the textual stream and the visualized stream:

1. How easy was it to read only the most emphasized Tweets in your timeline?
2. How easy was it to ignore the de-emphasized Tweets in your timeline?
3. How easy was it to read through all Tweets in the timeline together in chronological order while the recommender was active?

Responses to these questions are shown in Tables 2 and 3.

**Table 2. Recommendation presentation responses for the textual stream**

|  | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Emphasized** | - | - | - | - | 1 | 1 |
| **De-emphasized** | - | - | - | - | 1 | 1 |
| **Combined** | - | - | - | 1 | 1 | - |

**Table 3. Recommendation presentation responses for the visualized stream**

|  | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Emphasized** | 1 | - | - | 1 | - | - |
| **De-emphasized** | - | - | - | 1 | 1 | - |
| **Combined** | 1 | - | 1 | - | - | - |

Generally the response to the textual recommendation presentation was very positive, while response to the two-dimensional stream visualization was mixed. Both users found it at least as difficult to read the entire stream chronologically in both cases as it was to read only the emphasized Tweets or ignore the de-emphasized Tweets. This can be considered a positive result because it suggests that recommendation emphasis may be a viable alternative to filtering for stream consumption.

### 4.3.2 Recommender Performance

With regard to recommender performance, the following questions were asked:

1. How accurate was the recommender in emphasizing interesting Tweets?
2. How accurate was the recommender in de-emphasizing uninteresting Tweets?
3. How strongly do you agree with the following statement? "As you increased the minimum Filter value, the application showed a generally more interesting timeline."
4. How strongly do you agree with the following statement? "As you decreased the maximum Filter value, the application showed a generally less interesting timeline."

Responses to these questions are shown in Table 4.

**Table 4. Recommender performance responses**

|  | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Interesting** | - | - | - | - | 2 | - |
| **Uninteresting** | - | - | - | 1 | 1 | - |
| **More Interesting** | - | - | - | - | 2 | - |
| **Less Interesting** | - | - | - | - | 2 | - |

Subjective evaluations of recommendation accuracy do not necessarily tell the whole story, but it is a very important component, especially in social activity stream recommendation. It is possible that an unbiased test of the recommender using pre-determined ratings in training and test sets and cross-validation would tell a different story and that users are more forgiving of recommendations that are slightly off or just better than the alternative. Users may especially be forgiving in this setting because reading an uninteresting Tweet causes little harm. The naïve Bayes classifier used here should infer preferences of users quite well if they follow others tweeting about only a narrow range of topics, but to get a more reliable indication of recommender performance using this subjective method of testing, a larger sample size is needed. On the other hand, the results are promising given the small amount of effort required to train the recommender.

### 4.3.3 Design Feedback

With regard to the user interface and feature design, the following questions were asked:

1. How useful was the "User Volume" feature?
2. Which timeline presentation style would you most prefer for regular use?
3. What did you like most about the user interface?
4. What did you like least about the user interface?
5. Which application feature did you like most?
6. Which application feature did you like least?
7. Do you have any other comments or suggestions?

The first question had responses on a four-point Likert scale, while the second question asked the users to choose between the textual and visualized versions of the timeline. The others were all text fields that allowed for open-ended responses.

Tweet interest can sometimes fluctuate greatly even within the set of Tweets from a given user. Because of this fact, it was unclear how helpful the "User Volume" feature would be, which allows users to manually adjust the influence of Tweet authorship on recommendation scores. However, both users reported that the feature was useful.

When asked which timeline presentation style they would most prefer for regular use, participants were given the choice between showing everything equally, showing everything with varying levels of emphasis, and filtering out the most uninteresting Tweets. Neither participant said that they would prefer everything to be shown equally, while each of the other two options was selected once. Without more participants these responses are not very useful, but it does suggest an appetite for users to have some processing done on the content in their stream, as not all updates are created equal.

The open-ended responses revealed some useful suggestions for future improvement. Users found the two-dimensional visualization relatively difficult to use and understand, suggesting that the presentation and interface could be more intuitive. The greatest source of trouble was lag due to frequent re-calculations in the application's script, which used the AngularJS framework. Significant performance enhancements may be possible, but has proven difficult without removing one of the visualizations from the page. Simplifying the two-dimensional visualization would reduce the need for so much processing to constantly be done. Creating a custom JavaScript framework optimized for this particular application would allow for maximum flexibility, but would require much more development time and would add much complexity.

In designing the visualization, we attempted to mitigate the performance problems by allowing users to limit the number of Tweets displayed on the page at one time, and in testing this seemed to work well. It is unclear whether the users missed reading about this feature in the instructions or if it did not have the same positive effect in their environments. It may also not be as practical in higher-throughput streams to limit the number of Tweets shown too much.

A larger sample size is desirable before writing off the two-dimensional visualization as a tool for stream consumption, but it would likely benefit from some design changes. It is possible that the visualization is better served as a complementary view to provide social activity awareness and a general view to support a primary linear textual stream. Some possible reasons users preferred the textual stream are that it supports a more passive browsing style, shows more information at one time, is more familiar, and contains larger targets for mouse interaction. More information will be gathered about the weaknesses of the existing system, and more usability testing will be done to improve it before carrying out a large-scale user study.

## 4.4 Limitations

The greatest drawback to this pilot study was the limited sample size. Of course a pilot study using even a small number of participants is more helpful than none at all, since it forces the designer to consider implications of releasing a system to the public further in advance. While many of the comments were very helpful, it is impossible to make any firm conclusions about the results gathered from the Likert-scale questions because of the small sample.

In general, the questions asked in the questionnaire were subjective and may have been positively biased, though some of the answers on the extreme negative end of the scale suggest this was not an issue for all participants. A quantitative study comparing the two presentation styles to measure interaction data, user preference, and subjective assessments of recommendation accuracy would be much more likely to avoid such biases and give more useful results.

## 5. PROPOSED EXPERIMENT

### 5.1 Goals

The main goal of the proposed large-scale experiment is to investigate the effects of recommendation presentation methods on users' subjective evaluations of the underlying recommender mechanism. In other words, we want to determine if the different ways of presenting social activity stream recommendations to users will affect how accurate they perceive the recommender to be. To measure this, metrics of trust, transparency, persuasiveness, effectiveness, efficiency, and satisfaction will be collected.

### 5.2 Design

In order to eliminate as many potential biases as possible, as well as to study interaction effects between different factors, a $2^2$ factorial experiment design will be used. Participants will randomly be assigned to one of two groups, one of which will use the visualized stream, while the other half uses the linear textual stream. Meanwhile, half of each of those groups will be divided by presentation methods of visual emphasis with user-controlled filtering or automated filtering where hidden updates are recoverable but not shown in the main timeline. The participants will have no knowledge of the existence of the other groups.

**Table 5. Treatment combinations for the proposed experiment**

| Textual Stream & Emphasis | Visualized Stream & Emphasis |
|---|---|
| Textual Stream & Filtering | Visualized Stream & Filtering |

In contrast to the brief pilot study conducted and described in this paper, the proposed experiment will take place over a period of two weeks, with participants using the system several times throughout that period. Several questionnaire responses will be required so as to measure the evolution of participant opinion over time. The questions will be similar to those used in the pilot study, but will focus more on recommender performance and trust and less on aspects of usability. User interaction data may also be collected and analyzed. We would like to recruit 100 participants so that an adequate sample size is reached for each factor group.

### 5.3 Expected Results

We expect that participants who use the systems with visual emphasis will rate the equivalent recommender system as being more accurate than will those using the systems with automatic filtering. Besides higher raw subjective scores for recommender accuracy, we expect to observe the following three results:

- Filtering will cause decreased trust
- Emphasis will cause increased transparency
- Emphasis will cause increased persuasiveness.

Trust, transparency, and persuasiveness, as they relate to recommendations, have been defined by Tintarev and Masthoff [6].

It is unknown whether the interface (textual vs. visualized) will have any effect, but any such effects will be observed. Participants may perceive more trustworthiness in the text stream case because less information is being "hidden" until the user interacts with the interface, but the visualization shows additional information that the text stream does not. For example, the visualization codes popularity and shows more data on the screen at one time. These factors may not be factors at all, or they may cancel each other out. Whatever the result, it will serve to guide future development of such systems for consumption of social activity streams.

## 6. RELATED WORK

As mentioned, the typical approach to the primary problem of information overload in social streams is to use some form of stream filtering. Naturally, there has been plenty of work done in this area, and several examples of stream filtering can even be found in the major social networking sites. Facebook's news feed, for example, reorganizes updates using an unknown algorithm of which post date is only one of multiple factors. It also is able to filter out particular updates, and this filter can be trained by user feedback. This method of reorganizing information, however, can mislead the users with respect to social activity since updates are not presented in chronological order.

The issue of recommendation in Twitter timelines in particular from a filtering approach has been tackled by Sriram [5]. In addition to a naïve Bayes classifier, C4.5 decision tree and sequential minimal optimization algorithms were used to classify Tweets into categories such as "news," "opinion," "deals," and "events." Support was also added for user-defined classes, which could be a useful addition to this project. Adding user-defined classes beyond just "interesting" and "uninteresting," but using the tiered model and visual emphasis instead of stream filtering could be a possible direction for future enhancement. Sriram attained a very high level of categorization accuracy using a more complex feature set that may be worth emulating in future work as well.

Wang et al. [9] also studied recommendations of updates across both Twitter and Facebook, focusing only on recommendation effectiveness without suggesting filtering as a solution to the information overload problem. They studied the value of textual and non-textual features in accurately predicting whether an update will be liked, disliked, or neutral. Machine learning algorithms such as decision trees, support vector machines, Bayesian networks, and radial basis functions were compared for performance. This paper was a helpful starting point for generating recommendations from basic features of social activity stream updates.

Some of the drawbacks of information filtering in social streams have been addressed by Nagulendra and Vassileva [3]. The "Filter Bubble" visualization in social networking site Madmica, shown in Figure 5, allows users to view which updates have been hidden, and it also gives control to show or hide posts on certain topics from certain users. However, it remains difficult to get a sense of where posts belong in the context of the social stream without restoring them to a visible status. This is likely not as important for Madmica as it is in Twitter, where updates may quickly become less relevant as they age, but it is one reason this
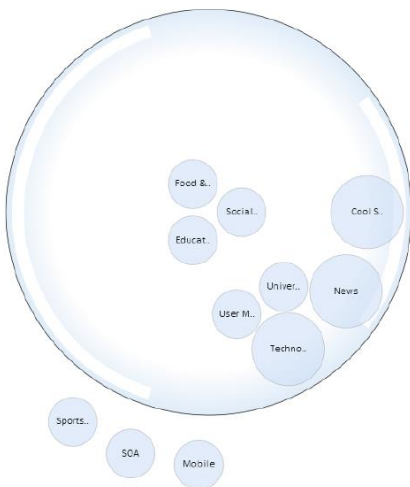
project explores a complete view with emphasis rather than a filtered stream.

Webster and Vassileva's work in the Comtella-D online discussion forum [11] was the original inspiration for the strategy of recommendation presentation using emphasis rather than filtering. In their system, recommendations are made collaboratively by and for other members of the community. The most recommended posts are shown in a brighter colour and with larger text in order to be visually attractive and more noticeable. The chosen colours in that case fit with an "energy" metaphor, with the more recommended posts displaying more life while the least recommended posts have a dull and lifeless appearance. However, a horizontal offset was not employed in this system, and the method of collaborative recommendations used within this closed community is not replicable in the vast open world of Twitter.

**Figure 5: Visual emphasis of collaborative recommendations in Comtella-D**

Rings[2] [4] is a visualization system for Facebook friend networks that codes recency, quantity of recent posts, and average social impact of those posts. The system successfully increased user awareness of lurkers and the most active recent contributors in one's own network but did not focus on which individual posts were most impactful, choosing rather to focus on the users and their relative activity levels within the friend network. The information that the visualization provided was interesting for users and was not easily discoverable through Facebook's own default interface, but it was not necessarily useful for popular Facebook functions such as everyday social stream consumption. The visual design was the main inspiration for the visualization described in
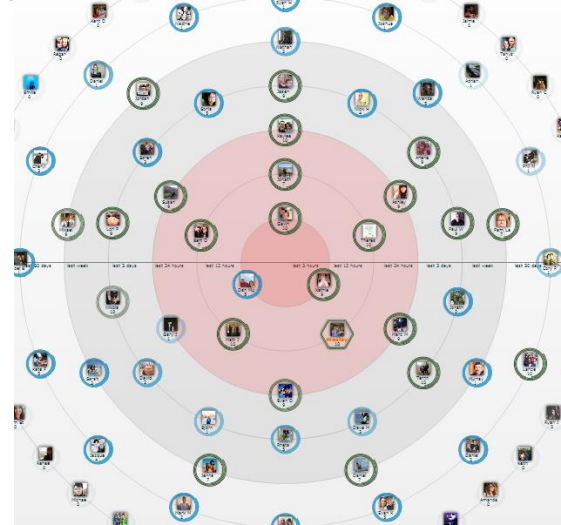
**Figure 4: Filter bubble visualization in Madmica**

**Figure 6: Screenshot of the Rings Facebook visualization**

---

[2] http://rings.usask.ca

this paper. This new design also attempts to address some of the shortcomings of Rings by facilitating Twitter's typical use cases.

KeepUP [10] visualizes a user's network of influence in an RSS recommender system that allows for user interaction. While it does primarily model the network rather than the posts, it also tracks topics that each user has commonly liked or disliked. The transparency provided and affordance of user control over others' influence on recommendations allows users to shape their own filter bubble. The User Volume feature provided in the visualization system described in this paper was adapted from the idea that users can choose which members of their network should have the most influence on their recommendations.
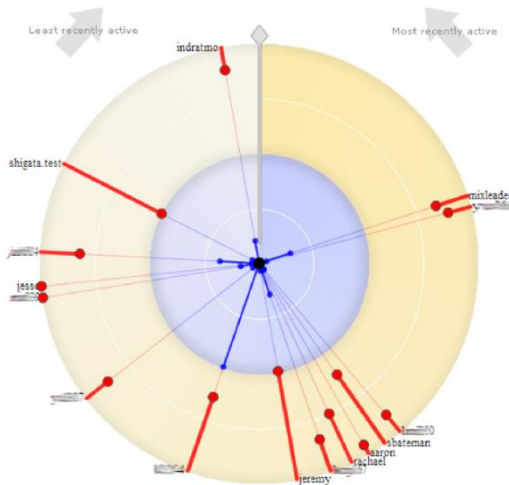


**Figure 7: Visualization of neighbour influence in KeepUP**

## 7. SUMMARY
This paper expands on work done in the area of social visualization and recommender systems by developing an application that can be used to study the effects of recommendation presentations on subjective measures of recommender performance. It is understood from the related work that visual emphasis can be a useful way to draw users' attention to more interesting or relevant content in social activity streams and that giving users control over stream filtering can increase their trust in these systems. The ultimate result that all of this is working toward is improved social activity streams wherein users spend more of their time reading the content best suited to them personally and are more aware of the full extent of activity in their social networks. Gaining a better understanding of the user and of how design decisions affect user opinions of the systems recommending and presenting that content is an important next step toward achieving those goals.

## 8. FUTURE WORK
Besides carrying out the experiment outlined in this paper, this application can be extended in a number of different ways for future research with the goal of understanding how best to increase user awareness and present recommendations of time-relevant updates in social activity streams. Potential future work that would extend or expand upon the research described here includes:

- Determining the optimal number of ratings required to strike the right balance between recommender effectiveness and user satisfaction.

- Improving Tweet classification in the recommender system, including accounting for interaction effects of classification features.
- Incorporating text mining to enhance classification and recommendation based on topics.
- Incorporating more user control by allowing users to specify why they liked or disliked a particular Tweet, including the ability to identify combinations of contributing factors.

The ultimate goal with this future work is to enhance the user experience through effective recommendations and presentations. Explanations and control are facets of recommender systems research that can lead to greater user acceptance, satisfaction, and trust in these systems. Applications of these facets to this unfiltered social activity stream recommender concept will be explored in greater detail in the future.

## 9. REFERENCES
[1] Gunawardana, A. & Shani, G. 2009. A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. *Journal of Machine Learning Research* 10, 2935–2962.
[2] Healey, C.G. 1992. Visualization of Multivariate Data Using Preattentive Processing. M.Sc. thesis, The University of British Columbia.
[3] Nagulendra, S. & Vassileva, J. 2013. Providing Awareness, Understanding and Control of Personalized Stream Filtering in a P2P Social Network. In *Collaboration and Technology: 19th International Conference, CRIWG 2013, Wellington, New Zealand, Oct. 30–Nov. 1, 2013, Proceedings*, 61–76.
[4] Shi, S. 2011. Keeping up with the Information Glut by Visualizing Patterns of Posting by Friends on Facebook. M.Sc. thesis, University of Saskatchewan.
[5] Sriram, B. 2010. Short Text Classification in Twitter to Improve Information Filtering. M.Sc. thesis, The Ohio State University.
[6] Tintarev, N. & Masthoff, J. 2007. A Survey of Explanations in Recommender Systems. In *ICDEW '07 Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, 801–810. DOI= http://dx.doi.org/10.1109/ICDEW.2007.4401070.
[7] Tyler, S.K. & Zhang, Y. 2008. Open Domain Recommendation: Social Networks and Collaborative Filtering. In *ADMA '08 Proceedings of the 4th International Conference on Advanced Data Mining and Applications*, 330–341. DOI= http://dx.doi.org/10.1007/978-3-540-88192-6_31.
[8] Waldner, W. & Vassileva, J. 2014. Emphasize, Don't Filter! Displaying Recommendations in Twitter Timeline*s*. In *RecSys'14 Proceedings of the 8th ACM Conference on Recommender Systems*, Oct. 6–10, 2014. DOI= http://dx.doi.org/10.1145/2645710.2645762.
[9] Wang, Y., Zhang, J., & Vassileva, J. 2010. Towards Effective Recommendation of Social Data across Social Networking Sites. In *AIMSA'10 Proceedings of the 14th International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, 61–70.
[10] Webster, A.S. & Vassileva, J. 2007. The KeepUP Recommender System. In *RecSys'07 Proceedings of the 2007 ACM Conference on Recommender Systems*, 173–176. DOI= http://dx.doi.org/10.1145/1297231.129726
[11] Webster, A.S. & Vassileva, J. 2006. Visualizing personal relations in online communities. In *AH'06 Proceedings of the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 223–233. DOI=http://dx.doi.org/10.1007/11768012_24.