LogRank: Summarizing Social Activity Logs

Abhijith Kashyap
Dept. of Computer Science and Engineering,
University of California, Riverside.
akash001@ucr.edu

Vagelis Hristidis
Dept. of Computer Science and Engineering,
University of California, Riverside.
vagelis@cs.ucr.edu

ABSTRACT

Online Social Networks (OSNs) allow users to create and share content (e.g., posts, status updates, comments) in real-time. These *activities* are collected in an *activity log*, (e.g. Facebook Wall, Google+ Stream, etc.) on the user's social network profile. With time, the activity logs of users, which record the sequences of social activities, become too long and consequently hard to view and navigate. To alleviate this *cluttering*, it is useful to select a small subset of the social activities within the specified time-period as representative, i.e., as summary, for this time-period.

In this paper, we study the novel problem of social *activity log summarization*. We propose LogRank, a novel and principled algorithm to select activities that satisfy three desirable criteria: First, activities must be *important* for the user. Second, they must be *diverse* in terms of topic, e.g., cover several of the major topics in the activity log. Third, they should be *time-dispersed*, that is, be spread across the specified time range of the activity log. LogRank operates on an appropriately augmented social interaction graph and employs random-walk techniques to holistically balance all three criteria. We evaluate LogRank and its variants on a real dataset from the Google+ social network and show that they outperform baseline approaches.

1. INTRODUCTION

Online Social Networks (OSNs) are continuously updating their services to facilitate and promote user interactions. The principal example of a feature that facilitates such communications is the Facebook *Wall*. This feature, available in some form in most OSNs, records and displays users' *activities* (e.g., posts, status updates, comments). The Wall, which we refer as *activity log* in this paper, serves as a consolidation point for all activities in a social network relevant to the owner of the Wall. While initially limited to short text posts, they now include web links, photos, videos, and personal updates such as life events (graduation, job changes etc.) interests and moods. A recent survey reported that in 2011, over 75,000 Wall posts (wall posts, status updates, links etc.) were added to Facebook every minute and about 41 posts are added to the wall of an *average* Facebook user each month[1].

While OSNs have focused on finding new ways to enable users to generate and share content, little work has studied the difficulties faced by users in consuming this ever-increasing content stream. The accumulation of activities and posts on a user's activity log makes it difficult for users to follow and keep up with the activities of *friends*. As an example, consider the Google+ posts of Barack Obama (managed by his re-election campaign staff). Figure 1 shows a small selection of posts from his activity log (*Stream* in Google+ lingo) between December 2011 and January 2012. The posts vary from insignificant (e.g. (#2) birthday wishes to a Hollywood legend), to important policy decision announcements (#7 and #8). A *follower* reading through the posts finds it difficult to keep track of the large and ever-growing content. Instead of displaying the entire content stream, it is better to display a small selection of *representative* posts as a summary of the activity log. The problem becomes trickier if in addition to posts, there are also other types of activities such as personal status updates. To partially address this problem, Facebook recently rolled out the *Timeline* feature, which arranges activities along a temporal axis and highlights important activities from predetermined time-periods. However, this feature depends on the user to select important activities and by default selects activities that have the most feedback or reaction (e.g. comments, likes etc.)

The choice of a representative subset of activities to display from a user's activity log is a subjective matter. Intuitively, it is better to select and display the most *important* posts. However, judging the importance of posts is also inherently subjective and depends on a variety of factors. The *type* of the post is an important indicator of its importance. For example, the post about an important life event such as marriage is more important than, say, a mood post about a particular day's traffic. The importance of a post can also be judged by the reaction to a post by members (friends and followers) in the social network. In a social network, friends can react to a post by commenting on it, sharing the post or by indicating a (positive) feedback (e.g. Facebook *Like* or Google+ '*+1*') on it. The importance of friends providing the reaction is also a factor. For example, a comment from a friend who interacts regularly is more important than a *Like* from a friend who interacts rarely.

In addition to the importance, it is also necessary to consider the variety or *diversity* amongst the representative activities. For example, activities #7 and #8 shown in Figure 1 can be considered important due to the sheer amount of reaction (sum total of shares, comments and *+1*s) on these activities. However, the posts discuss the same topic of 'payroll tax cut' and including only one in the summary is sufficient. Furthermore, it is desirable to choose activities that are *time-dispersed*, that is, be spread across the specified time range of the activity log. For example, for a 6-month range, it is generally not desirable to have all representative activities be from the same month, assuming that other months also have some important activities. In the example in Figure 1, the posts relating to his opponent Mitt Romney (#5 & #6) are on the same day, and including both of them in the representative summary would hide other some other important activity in a size-constrained representative summary.

In this paper, we study the novel problem of *summarizing an activity log, by selecting a set of representative activities that are important, diverse and time-dispersed*. The proposed approach, LogRank, addresses all three criteria in a principled way. The importance of the post is computed by combining the factors identified above: *(a)* the type of the activity *(b)* the type and

[1] Facebook Activity Statistics:
http://www.digitalbuzzblog.com/facebook-statistics-stats-facts-2011/

Fifteenth International Workshop on the Web and Databases (WebDB 2012), May 20, 2012 - Scottsdale, AZ, USA

Figure 1. Excerpt of Barack Obama's Google+ Stream

frequency of reactions, and *(c)* the importance of the friends providing the reaction. To achieve this, we propose a rich graph-based model called *Social Interaction Graph (SIG)* that captures users and their associations, activities and reactions to activities and also models the time distances between activities. LogRank is a random walk-based algorithm that operates on SIG and selects a set of activities by balancing importance, diversity and time-dispersity. The paper makes the following contributions:

1. We propose SIG, a rich graph-based model that captures the various activities of a user, the reactions to these activities, and the time-dispersity of the activities (Section 2).
2. We propose LogRank, a principled authority-flow based algorithm to compute a representative summary of the user's activities by selecting activities that are simultaneously important, diverse and time-dispersed (Section 3).
3. We present experimental results and a preliminary user study of applying our techniques on a real OSN, to summarize the Google+ activity logs of 2012 US presidential candidates. We compare against baseline approaches (Section 4).

We discuss related work in Section 5 and conclude in Section 6.

2. FRAMEWORK AND DATA-MODEL

The entities in an OSN and their interactions are captured by a composite model, which we term the *social interaction model* and is illustrated in Figure 2. The figure shows the entity *types* in an OSN namely users, their activities and reactions to these activities. Further, it shows the most common relationships that exist between activities:

1. **friend(user1,user2):** Users establish friendships with other users. This typically symmetric relationship is an implicit indication of trust and allows a user to view, follow and participate in activities of friends. Another means of establishing interpersonal relationships is by subscribing to content of another user. This *follower* relationship is more restrictive and is typically asymmetric.
2. **activity(user,activityType,date-time,data):** The activities of a user, which are posted on her activity log, can be personal status updates, text posts, location check-ins, etc.
3. **reaction(activity,user,reactionType)**: A *user (friend or follower)* can react to the activities of another user. This reaction can be of various *types* such as *comment*, *share* (*retweet*), *Like* (*+1*), etc. The importance of an activity can be judged by the *type* and *amount* of reaction to it. Further, reactions by trusted users (e.g., close friends who frequently interact with each other) have higher impact.

In addition to the explicit relationships described above, the social interaction model can capture several implicit relationships. For this paper, we capture the implicit relationship between activities:

4. **distance(activity1,activity2):** This relationship represents the *content similarity* and *time difference* between activities in the social interaction graph.

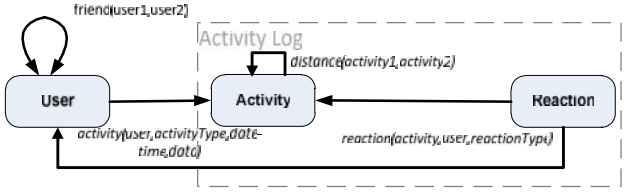

Figure 2. Social Interaction Model

We note that our model does not capture all the interactions in a social network. For example, typically a reaction (comment) can also have an associated reaction (*Like* or a comment reply) and these interactions can be leveraged to further refine our model. However, to keep the model simple, we choose to model interactions that significantly affect the relative importance of activities and defer capturing other interactions to future work.

Social Interaction Graph (SIG): From the social interaction model (which can be viewed as the schema), we create a *SIG* (which can be viewed as the instance), which consists of instantiated entities and edges between these entities corresponding to interaction relationships. The algorithms presented in Section 3 estimate the score of each activity (activities are nodes in SIG) using authority flow ranking methods [1-3], which require assigning *authority transfer weights* to each edge of SIG, given its heterogeneous nature. The authority transfer weight of an edge denotes how much of the score (authority) of a node should be transferred to its neighbor. The SIG graph $G(V,E)$ is a labeled directed graph where each node $v \in V$ has an associated $type(v)$, which is one of the types in the social interaction model, i.e., *user*, *activity* or *reaction*. An example SIG is illustrated in Figure 3.

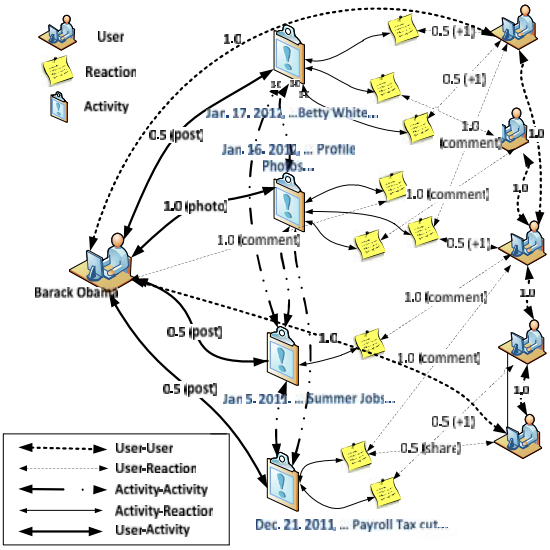

Figure 3. Social Interaction Graph (SIG)

For each relationship between two entities u, v in the social interaction model, we create two edges $e^f(u \to v)$ and $e^b(v \to u)$ in SIG. For example, if there is an activity added by a given user, *Activity*(*Barack Obama*, *post*, 12.21.2011,' ... *Payroll Tax cut*'), then we create two edges, one in each direction (Figure 3) between nodes representing the user *Barack Obama* and the corresponding activity in the SIG graph. The rationale for creating two edges is that authority flow in each direction can be potentially different, as discussed in Section 3. Each edge has an associated authority transfer weight $w(e) \in [0,1]$. These edge weights capture the strength of the association between two entity instances. Next, we formally define the weights w on edges of G.

User-User Edges: *Friend*ship is a Boolean relationship between user nodes and we set its weight in SIG as follows:

$$w(u_1 \to u_2) = w(u_1 \to u_2) = \begin{cases} 1 & if\ Friend(u_1, u_2) \\ 0 & otherwise \end{cases}$$

User-Activity Edges: For each activity a of a user u, we create two weighted edges:

$$w(a \to u) = w(u \to a) = W_A(a.activityType)$$

where $W_A(activityType) \in [0,1]$ is a function that assigns relative weights based on activity type (e.g., *post*, *statusChange*). As we mentioned in Section 1, $activityType$ is a factor in deciding the importance of an activity and the weight function assigns weights based on type. In experiments, which are based on Google+ data, we set $W_A(photo) = 1$ and $W_A(post) = 0.5$, since we believe that posts with *photo* carry double the importance of text *posts*. We explore more principled ways of setting weights in future work.

Activity-Reaction Edges: As in the case of User-Activity edges, the edges between an activity and a reaction are weighted based on the $reactionType$ (e.g., *comment*, *+1*, *Like etc*). For each reaction r to an activity a we create two edges:

$$w(a \to r) = w(r \to a) = W_R(r.reactionType$$

where $W_R(r.reactionType) \in [0,1]$. In experiments, we assign $W_R(comment) = 1$ and $W_R(+1) = 0.5$, because we believe that *comments* carry double the importance of *+1*.

Reaction-User Edges: For each reaction r in the social network model, we add edges between r and the user who performed the reaction:

$$w(r \to u) = w(u \to r) = 1$$

Activity-Activity Edges: These edges are added to capture the *content similarity* and *time difference* between activities a_1 and a_2. Specifically, the distance edges have weights:

$$w_s(a_1 \to a_2) = w_s(a_2 \to a_1) = \frac{distance(a_1, a_2)}{\max_{a_i \to a_j} distance(aa_i, a_j)}$$

In this paper, we compute the content similarity between two activities based on Information Retrieval text similarity measures. In particular we use the Lucene Similarity Scoring formula[4] and we normalize it by dividing by the maximum similarity score amongst the activities. Regarding time difference, we create edges with weights:

$$w_t(a_1 \to a_2) = w_t(a_2 \to a_1) = 1/(df + 1)$$

where *df* is the time difference (in days) between a_2, a_1.

We combine the two factors by a linear combination:

$$w(a_1 \to a_2) = w(a_2 \to a_1) = \delta \times w_s + (1 - \delta) \times w_t \qquad (1)$$

where $\delta \in [0,1]$ balances the importance of topic diversity and time dispersity, and can be set by the domain expert or through trial and error or user feedback. We set $\delta = 0.5$ in our experiments and defer selecting an optimal value for δ, which may be application- and user-dependent to future work. Further, for performance reasons, we only add activity-activity edges if $w(a_1 \to a_2)$ is above a threshold (we use 0.3).

3. ALGORITHMS

In this section, we present algorithms to compute a representative summary of the user's activity log. As we mentioned in Section 1, these representative activities should not only highlight the important activities of the user but also be topic-diverse and time-dispersed. Authority-flow based algorithms ([1, 3]) have proved useful in computing the global importance of nodes in data graphs like the Web graph or blog graphs [5]. However, these algorithms compute solely the relative relevance or importance of nodes. To incorporate diversity and time-dispersity, we adapt the GRASSHOPPER algorithm [6] which ranks nodes on a homogenous graph with emphasis on diversity.

Background (PageRank): Let $G(V,E)$ be a graph with set of nodes $V = \{v_1, \ldots, v_n\}$ and set of edges E. The PageRank $r(v_i)$ of a node v_i is a measure of its global *importance* in G, where the importance is based on the recursive notion that important nodes are linked to by other important nodes. Starting from a random node v_i of V, a random surfer either follows an out-link of v_i

with probability c or jumps to a random node with probability $(1-c)$. Let $\boldsymbol{r} = [r(v_1), \ldots, r(v_i), \ldots, r(v_n)]^T$ be the rank vector. The global PageRank can be computed by the following equation:

$$\boldsymbol{r} = c\boldsymbol{A}\boldsymbol{r} + (1-c)\boldsymbol{e}/|V| \qquad (2)$$

where, $\boldsymbol{e}$ is the uniform vector $[1, \ldots 1]^T$ and $\boldsymbol{A}$ is a $n \times n$ matrix of transition probabilities with $A_{ij} = 1/O(v_i)$, if there is an edge $e: v_i \rightarrow v_j \in G$, and 0 otherwise and $O(v_i)$ is the out-degree of node v_i. PageRank assigns a global score to nodes in G and does not consider any preferences, say for a given user or query. To account for preferences, several personalized versions of PageRank have been proposed [1, 7, 8]. By selecting a set of nodes $S \subseteq V$ as the *base set* to which the surfer jumps randomly, the PageRank score (a.k.a. authority) associated with nodes in S and the ones close to them is increased. In particular, instead of using the uniform vector $\boldsymbol{e}$, a *base set vector* $\boldsymbol{s} = [s_0, \ldots, s_n]^T$ with $s_i = 1$ if $v_i \in \boldsymbol{s}$ (and 0 otherwise) can be used. The PageRank equation (Equation 2) is rewritten as:

$$\boldsymbol{r} = c\boldsymbol{A}\boldsymbol{r} + (1-c)\boldsymbol{s}/|S| \qquad (3)$$

Note that Equations 2 and 3 do not account for diversity in ranking nodes. For example, all nodes of a highly connected cluster would receive high score.

Background (GRASSHOPPER): The GRASSHOPPER algorithm[6] addresses the problem of computing a diverse ranking of nodes in information-networks. This algorithm generates a diverse ranking of nodes as follows. It first executes PageRank (Equation 2 or 3), outputs the node with highest rank, and then makes this node an absorbing state for the random walk. Since random walk-based algorithms compute the rank of a node by combining the relative importance of all neighboring nodes, converting a node to an *absorbing node* will reduce the score of all its neighbors and hence avoid ranking highly two nodes that are tightly linked to each other (i.e., similar to each other), which in turns achieves diversity. Then, the node with the highest score (probability) in the stationary distribution of a random walk with absorbing nodes is selected as the next item. However, random walk on a network with absorbing states is ill-defined since any walk on a connected graph will eventually be absorbed. Instead, in [6] the authors propose to compute the expected number of visits to each node before absorption as a measure of a node's importance. Intuitively, any node that is connected to absorbing nodes in a random walk will be absorbed much sooner and therefore would have fewer expected number of visits. Equation 3 can be rewritten in matrix form as $\boldsymbol{r} = \tilde{\boldsymbol{A}}\boldsymbol{r}$ where:

$$\tilde{\boldsymbol{A}} = c\boldsymbol{A} + (1-c)\boldsymbol{1}\boldsymbol{s}^T/|S| \qquad (4)$$

Let R be the set of nodes ranked so far. A node $v \in R$ with index i in $\tilde{\boldsymbol{A}}$ is converted into an absorbing state by setting $\tilde{A}_{ii} = 1$ and $\tilde{A}_{ij} = 0$, $\forall j \neq i$. The transition matrix in Equation 4 can be rearranged by putting all absorbing states first:

$$\tilde{A}' = \begin{pmatrix} \boldsymbol{I_h} & \boldsymbol{0} \\ \boldsymbol{P} & \boldsymbol{Q} \end{pmatrix} \qquad (5)$$

where I_h is a $h \times h$ unit matrix where h is the number of absorbing states. The expected number of visits in an absorbing random walk is computed as [9]:

$$\boldsymbol{z}^T = \boldsymbol{1}^T(\boldsymbol{I} - \boldsymbol{Q})^{-1}/(n - |R|) \qquad (6)$$

The node with the highest number of expected visits in $\boldsymbol{z}$ is chosen and converted into an absorbing node and the algorithm repeats by recomputing Equations 5 and 6.

LogRank: The algorithms presented so far, operate on a homogenous unweighted graph (e.g. hyperlinked Web documents in PageRank or sentences for text summarization task in GRASSHOPPER). In these networks, the $n \times n$ authority flow matrix A is created by normalizing weights across rows, that is, by the inverse of the out-degree, i.e. $A_{ij} = w_{ij}/\sum_{k=1}^{n} w_{ik}$, where w_{ij} is 1 if edge $i \rightarrow j$ exists and 0 otherwise. In contrast, SIG consists of heterogeneous entities and relationships between them. In such a scenario it is critical to carefully distribute authority based on the semantics of the nodes and edges, and not just on the number of connecting paths. For example, consider a social interaction graph in which a user is connected to 100 other users in a friendship relation and she comments on 2 activities of a user whose activity log is being summarized. In this case, the transition probability from the user to an activity or user node would be $1/102$ and therefore most of the authority will be transferred to user nodes. However, intuitively both activities should receive also significant authority from the user node.

To fairly distribute the authority to its neighbors based on type, we normalize the row weights based on node types as follows:

$$A_{ij} = \alpha(i,j) \times w_{ij} \Big/ \sum_{k \in type(j)} w_{ik} \qquad (7)$$

where $type(j)$ is the type of node with index j in A, and weight w_{ij} is computed according to the formulas in Section 2.1. Normalizing weights based on types results in the total transition probability out of a node to be greater than 1. Therefore, an *authority flow bound* $\alpha(i,j)$, which depends on the type of edge $i \rightarrow j$, is introduced so that they sum to at-most 1. In this work, we set $\alpha(i,j)$ to $1/3$ for all node pairs, given that we have at most 3 types of edges incident on a node (Figure 2). As we mention in Section 6, in the future we will study more elaborate ways of computing $\alpha(i,j)$ to bias the effect of various relationship types.

Another key difference of LogRank is the way that the *base set* is defined. We are interested in computing a representative set for activities $\{a_{t1}, .., a_{tk}\}$ in the activity log of a user u. Therefore, we set the base set S to $\{u\}$, which means that all random walks start from *u*. This biases LogRank towards activities important for user *u*. For instance, a post by a user u_1 who is closely connected to *u* will receive higher LogRank than a post from another user u_2. Note that LogRank assigns a score to all nodes in SIG; however, it outputs only activity nodes that belong to the activity log of the user of interest by ignoring other nodes selected during evaluation. However, the choice of the *base set* ensures quick convergence of the algorithm as the random-walk is biased towards nodes close to the *baseset* i.e. activities and they are likely to be picked during initial iterations.

4. EXPERIMENTAL EVALUATION

In this section we present the initial results of an experimental evaluation and a user study of LogRank on a real dataset collected from public profiles on the Google+ OSN.

4.1 Experimental Setup

Dataset: We chose Google+ due to its non-restrictive data usage policies. We focused on 2012 US Presidential candidates, since these profiles are public and are highly active. We seeded our crawler with the profile IDs of Barack Obama and three Republican candidates (Mitt Romney, Newt Gingrich and Rick Santorum). The crawler downloaded all the activities on these profiles and reactions to these activities, and other public activities of users who contributed these reactions. The data spanned over a period of 5 months between October 2011 and February 2012.

Summarization algorithms: We considered three algorithms:

ReactionAmt: As mentioned in Section 1, most social networks rank activities based on their type and *amount* of reaction on it. We treat all activity types as equally important and score an activity a solely by its reaction as:

$$score(a) = \#comments(a) + 0.5 \cdot \#plusOnes(a) + \#shares(a)$$

LogRank-NoTime: This LogRank variant was established to study the effectiveness of time-dispersity of LogRank. We set $\delta = 1$ in Equation 1, that is, we ignore time-dispersity.

LogRank: As described in Section 3.

Measures: In Section 1, we argued that a good summary of the activity log should contain activities that are not only important, but also content-diverse and time-dispersed. However, there is no well-established metric to measure the diversity of a ranking algorithm [10]. In [6], the authors propose a measure that indirectly estimates diversity by measuring the coverage of auxiliary nodes (they measure the coverage of movies to estimate the diversity of rankings of actors), that is, nodes that are not output by the ranking algorithm. In our case, the auxiliary nodes are users and reactions, since LogRank only outputs activity nodes. We only measure the coverage of users, since reactions are intuitively less important to cover. The user coverage is the number of unique users who either performed or reacted to one of the result activities. Higher coverage is an indication of high diversity. Another measure proposed in [11] measures the diversity in information networks by measuring the *density* $d(G_S)$ of the results subgraph $G_S(V_S, E_S)$ which is constructed with nodes in the diversified resultset and all edges between them. The density $d(G_S)$ is defined as the ratio of number of edges in G_S to the number of edges in a complete directed graph with $|V_S|$ nodes. The key intuition here is that the number of interconnections would be low in a diverse resultset resulting in a low density score. In our problem, G_S contains nodes for activities in summary S and all (distance) edges between them in SIG. We use the following density measure:

$$d(G_S) = \sum_{i,j \in S} distance_{norm}(i,j) / \left(|V_S| \cdot (|V_S| - 1)\right) \qquad (8)$$

where $distance_{norm}(i,j)$ is the normalized $distance$ between two activities in S. Lower density implies a more diverse summary.

Methodology: We partitioned the 5-month activity log of seed profiles (four Presidential candidates) into 3 overlapping ranges, where each range contains activities over a 3 month period with overlap of 2 months i.e. we have a total of 12 activity logs. Each range contained between 44 and 98 activities with 62 activities on average. For each range, we created a SIG and then summarized it with summaries consisting of 2 to 10 activities.

4.2 Results

Figure 4 shows the average coverage of users by summaries of various sizes constructed using baseline methods and LogRank. ReactionAmt performs better when summary sizes are small (≤ 3) since it selects activities by amount of reactions (comments, *+1*, etc.) and activities selected first tend to have a large number of user reactions. However, many users tend to cluster around a small set of *similar* activities (such as multiple posts criticizing opponents) and ReactionAmt only covers these users and does not include users who react to diverse activities and therefore coverage decreases as the summary size increases. The coverage achieved by LogRank is better on average (by 14%) as compared to LogRank-NoTime. The improvement is marginal for small sized summaries since factors other than time (reactions, content similarity) dominate selection. However, with increase in summary size, activities with fewer reactions are considered and time weighted distance edges play an increasing role in driving selection of activities that are spread across time and therefore include even more users.

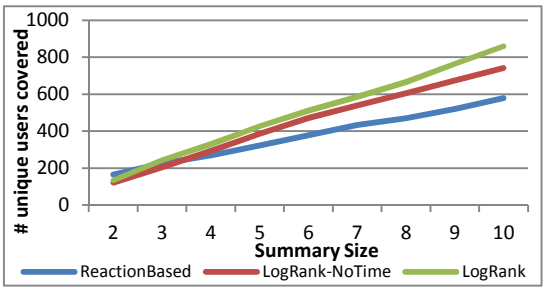

Figure 4. User Coverage

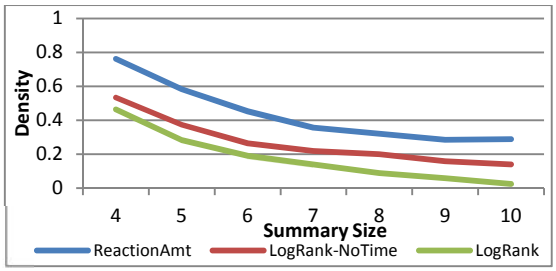

Figure 5. Density (lower is better)

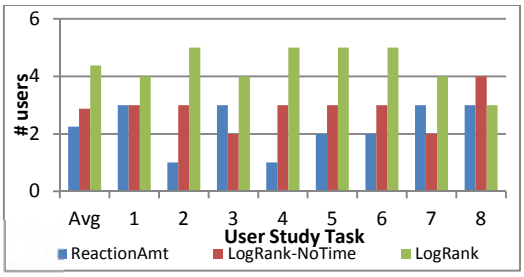

Figure 6. User Study

Figure 5 shows the average density, computed using Equation 8, of the subgraph constructed using only activities in the summary and their interconnections. As seen in the figure, ReactionAmt has much higher density (38% on average) on average as compared to other two methods. Activities selected by this method are often similar (based on content and time-dispersion) to other activities in the summary and therefore the graph constructed has many interconnections between them. The density scores for LogRank are less on average, compared to LogRank-NoTime. This is because the activities are spread across time in addition to being diverse in content and reactions. The decrease in density is higher (better) on average as the summary size increases demonstrating the effectiveness of our model in incorporating time-dispersion. However, the decrease is not highly significant (~10%) due to impact of other factors (reactions etc.) in summary computation.

4.3 User Study

The experiments in Section 4.2 demonstrated the effectiveness of LogRank in constructing representative summaries using the indirect diversity measures of coverage and density. Given the difficulty of evaluating the effectiveness of a diversified ranking method, we performed a preliminary user evaluation to judge the effectiveness of our approach. We constructed 8 sample activity-log windows from the Google+ Stream of the presidential candidates described in Section 4.1, where each window contains

20 activities. Next, we constructed representative summaries, with 5 activities each, using the three algorithms described in Section 4.1. We presented these summaries (un-labeled and in random placement order) alongside the activity-log to the users and asked them to pick the most representative summaries (one or more). For this preliminary evaluation, we asked 10 graduate students at our university to perform this task.

Figure 6 shows the number of times a summary was picked by users as the best representative summary for each task. Users picked LogRank as the best summary in a majority of tasks (7 out of 8). Furthermore, LogRank-NoTime outperformed ReactionAmt by a significant margin (21%). This is a significant improvement given that the tasks involved summarizing just 20 activities. For a larger activity log, we anticipate a larger improvement as diversity and time-dispersity play a larger role.

5. RELATED WORK

Authority-Based Ranking: Authority-flow based methods are widely popular in ranking nodes in information-networks such as hyperlinked web-documents [3], personalized web-search [7, 8] text-summarization [2], ranking in structured databases [1] and many others. However, these works focus solely on computing relevance and do not take into account diversity in ranking.

Diversified Ranking: The importance of diversity has been widely recognized in various scenarios such as diversifying search results [6, 12, 13], summarization [14] among many others. Recently, several works have proposed ranking diversification on information-networks. In , the authors propose an adaptation of widely accepted Maximal Marginal Relevance (MMR) measure[13] to graph structured data whereas DivRank [11] uses reinforced random walk model [15] to improve diversity. In our work, we use the GRASSHOPPER framework [6] and adapt it to work with heterogeneous social network data.

Graph Summarization: Graph summarization [14, 16-18] has been widely studied in various contexts including data-mining, compression and social network analysis. Most of these methods are based on grouping several nodes or sub-graphs into a *super-node* that summarizes these nodes. Instead, our method is based on selecting a subset of nodes to form representative summary of the entire graphs. Such sample-based summarization methods have been proposed for text summarization [16, 19] and image search[20]**.** However, these methods use ad-hoc measures and clustering techniques to summarize data whereas we focus on a more principled approach allowed by authority-flow techniques.

6. CONCLUSIONS AND FUTURE WORK

We proposed LogRank, a method to automatically construct a representative summary of the *activity log* of a user's social network profile. LogRank constructs a summary that contains important activities that are also topic-diverse and time-dispersed. We empirically demonstrated the effectiveness of LogRank with experiments and a small-scale user study. The preliminary version of LogRank can be extended in various interesting ways. One immediate avenue is calibrating various weights and parameters (W_A, W_R, δ *etc.*) used in LogRank. We plan to explore ways of inferring these weights based on the distribution of nodes and edges in SIG. Yet another direction is incorporating other relationships into the LogRank framework, such as reaction-reaction mentioned in Section 2 and user-group interactions. On the technical side, we will study how to improve the time performance of LogRank. One direction could be bringing LogRank computation closer to more efficient PageRank by allowing a small escape probability from *absorbing states* as suggested in [6].

7. ACKNOWLEDGEMENTS

This project was supported in part by National Science Foundation grants IIS-1216032 and IIS-1216007.

8. REFERENCES

[1] Balmin, A., Hristidis, V. and Papakonstantinou, Y. *Objectrank: authority-based keyword search in databases*. In *Proceedings of VLDB*, 2004.

[2] Erkan, G. and Radev, D. R. *LexRank: graph-based lexical centrality as salience in text summarization*. J. Artif. Int. Res., 22, 1 2004.

[3] Page, L., Brin, S., Motwani, R. and Winograd, T. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report, Stanford InfoLab, 1999.

[4] *Lucene*: http://lucene.apache.org/

[5] Hu, M., Sun, A. and Lim, E.-P. *Comments-oriented blog summarization by sentence extraction*. In Proceedings of CIKM, 2007.

[6] X. Zhu, A. B. Goldberg, J Van Gael, D. Andrzejewski, Q. *Improving Diversity in Ranking using Absorbing Random Walks*. In Proceedings of NAACL-HLT, 2007.

[7] Haveliwala, T. Topic-Sensitive PageRank. In Proceedings of WWW, 2002.

[8] Jeh. G, Widom. J., *Scaling personalized web search.* In Proceedings of WWW, 2002.

[9] Peter G. Doyle and Snell, J. L. *Random Walks and Electrical Networks*. 1984.

[10] Radlinski, F., Bennett, P. N., Carterette, B. and Joachims, T. *Redundancy, diversity and interdependent document relevance*. SIGIR Forum, 43, 2 2009, 46-52.

[11] Mei, Q., Guo, J. and Radev, D. *DivRank: the interplay of prestige and diversity in information networks*. In Proceedings of SIGKDD,2010

[12] Agrawal, R., Gollapudi, S., Halverson, A. and Ieong, S. *Diversifying search results*. In Proceedings of WSDM, 2009.

[13] Carbonell, J. and Goldstein, J. *The use of MMR, diversity-based reranking for reordering documents and producing summaries*. In Proceedings of SIGIR, 1998.

[14] Candan, K. and Li, W.-S. *Discovering Web Document Associations for Web Site Summarization*. In Proceedings of DaWaK, 2001.

[15] Pemantle, R. Vertex-reinforced random walk. *Probability Theory and Related Fields*, 92, 1 1992, 117-136.

[16] Mihalcea, R. *Graph-based ranking algorithms for sentence extraction, applied to text summarization*. In Proceedings of ACL, 2004.

[17] Navlakha, S., Rastogi, R. and Shrivastava, N. *Graph summarization with bounded error*. In Proceedings of SIGMOD, 2008.

[18] Tian, Y., Hankins, R. A. and Patel, J. M. *Efficient aggregation for graph summarization*. In Proceedings of SIGMOD,2008.

[19] Vanderwende, L. and Banko, M. *Event-centric Summary Generation*. In Document Understanding Conference 2004.

[20] Kennedy, L. S. and Naaman, M. *Generating diverse and representative image search results for landmarks.* In Proceedings of WWW, 2008.