User-Rank: Generic Query Optimization for Participatory Social Applications

Danny Hughes, Caren Crowley, Wilfried Daniels, Rafael Bachiller and Wouter Joosen
iMinds-DistriNet, KU Leuven, Leuven, B-3001, Belgium.
{firstname.lastname}@cs.kuleuven.be

Abstract

Online social networks have become a key communication medium for millions of Internet users. Building on this success a new class of social applications have emerged that use online social networks as a platform to access an established community of users. While some social applications act as a forum for users to interact, an emerging class of participatory social applications creates value by using data received from users to deliver a service. The success of participatory social applications is dependent on user responsiveness to application queries. In this paper we propose 'User-Rank', an application-independent approach to optimizing the querying of social network users. The design of User-Rank is based upon a 30-day experiment, which involved sending 3,055 messages to 70 users to determine the factors influencing response time and message loss. User-Rank achieves significant performance improvements relative to random querying and is lightweight with minimal memory and computational requirements.

1. Introduction

Online Social Networks (OSNs) have emerged as a key Internet communication medium. At the time of writing, Facebook has more than 1 billion monthly active users [1], while Twitter has more than 200 million [2]. As the scale of these networks continues to increase, so too does their potential for supporting novel applications.

Social applications create value by providing a forum for users to interact with one another, or by using data provided by users to deliver a service. Social applications that act as a forum to connect users are the most common, for instance 'Farmville', which runs on the Facebook OSN reports more than 50 million monthly active users [4]. Social applications that exploit data provided by users are also becoming increasingly popular. Such applications use crowd-sourced resources to provide a service such as smart parking [5], or weather monitoring [6]. User input may include smart phone sensor readings, photos or text content. This application model is particularly attractive as it provides a means to create large-scale applications with minimal infrastructure costs.

The success or failure of social applications that rely on user input is dependent on user responsiveness to requests. A high rate of message loss and long response times may preclude the development of time sensitive social applications. For instance, the weather radar application developed by Demirbas et al. [6] achieved a response rate of only 15%, when using Twitter to gather weather reports, while Nazir et al. [7] report an average response time of 16.52hrs for OSN users engaged in a social application running on Facebook. Lindsay et al. [3] reflect these concerns, cautioning that social networks *"should be further examined and researched before being adopted and used for emergencies and disasters"*. This clearly motivates the need for a better understanding of user responsiveness on OSNs and also mechanisms to optimize the querying of users.

In this paper, we consider the important problem of understanding and managing user response times and message loss in communications between social applications and their users. Our approach is based on an original 30-day study of 70 OSN users located in 11 countries. The purpose of the experiment was to identify statistically significant factors impacting message loss and message response times. Based on the results of this study we devised the "User-Rank" algorithm, which optimizes the querying of OSN users. We evaluate User-Rank on trace data from our experiment and find that it reduces the number of users that are required to answer a query by an average of 39.44% in comparison to random querying.

The contributions of this paper are threefold. Firstly, we identify the significant determinants of user responsiveness on OSNs. Secondly, we propose the generic User-Rank algorithm, which optimizes the querying of OSN users. Thirdly we demonstrate through our evaluation of User-Rank that consideration of user behavior can significantly improve user performance in social applications.

The remainder of this paper is structured as follows: We begin in Section 2 by reviewing related work. Section 3 describes our experiment. Section 4 introduces the User Rank algorithm. Section 5 evaluates the algorithm. Section 6 concludes. Finally, section 7 discusses the limitations of this work and directions for future research.

2. RELATED WORK

Boyd et al. [9] define Online Social Networks (OSNs) as *"web-based services that allow individuals to (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and those made by others within the system"*.

A key feature that has contributed to the popularity of online social networks is the ability of such networks to act as application hosting platforms for third party social applications. Social applications create value for their users by providing a forum for users to interact. The first social applications to emerge were games. *Fighters Club* [8] launched in 2007 is a gaming social application that allows users to pick virtual fights with Facebook friends, lasting from 15 to 48 hours. *Farmville* [20] launched in 2009 and grew to support over 50 million monthly active users. A number of non-game applications have also been successful. For example, *Hugged* is a simple application, launched in 2008, that allows users to send virtual 'hugs' to friends [8]. Now, a new class of social applications is emerging that creates value by using data provided by users. We refer to this class of applications as *participatory social applications*.

2.1 Participatory Social Applications

Participatory social applications are distinct from other social applications in terms of how they create value, which is often through the sensing of physical or social phenomena. Participatory sensing refers to user data that is a combination of both passive *sensed data* and more active *content uploading*. Participatory sensing as we use the term is distinct from user-generated content as defined by the OECD which stipulates that user-generated content *"must be published on a publicly available website or on a social network accessible to a selected group of people; it needs to show a certain amount of creative effort and needs to be created outside of professional routines or practices"* [11]. In contrast, the data generated by participatory sensing is often not publicly available, is not necessarily creative

(e.g. sensor readings) and often has a commercial purpose.

An early example of a participatory social application is the *Twitter Weather Radar* provided by Demirbas et al. [6]. Their application examines the feasibility of crowd-sourcing the problem of monitoring and predicting weather conditions. An application was developed that uses a dedicated twitter account to send 'tweets' (i.e. Twitter messages) to its 'followers' (i.e. users who subscribe to messages from the account). The tweets requested either current or next-day weather conditions in various locations. Followers of the application responded by manually sending a tweet with the requested weather conditions. The weather monitoring application achieved an average accuracy of 85% for current weather conditions and 46% for next-day weather predictions. In terms of user participation, Demirbas et al. [6] reported only a 15% response rate and slow response times, with 50% of responses taking longer than 30 minutes.

Miluzzo et al. [11] proposed the CenceMe application, which uses standard mobile phone sensors to infer the current activity of the user. To achieve this, a range of software classifiers were applied to data from the phone's microphone, camera, accelerometer, Bluetooth radio and GPS receiver to identify activities including: sitting, standing, running and walking. The users of CenceMe were invited to share their activity context using online social networks such as MySpace and Facebook. In terms of user participation, Miluzzo et al. [11] find that CenceMe users were willing to share their status via online social networks. Rosi et al. [12] highlight the potential of online social networks to detect social phenomena such as crowds and to support location aware recommendation systems. Noulas et al. [13] follow this approach, using GPS location data from mobile phones together with recommendations from the FourSquare and Gowalla (closed in 2012) social networks to suggest new venues. Participatory social applications, such as those discussed in this section are time sensitive and therefore have more stringent requirements in terms of user responsiveness relative to more traditional social applications, which create value by acting as a forum for interaction [7].

2.2 Studies of OSN User Behavior

The success or failure of participatory social applications that rely on user data input is dependent on user responsiveness to requests. In terms of user behavior in social applications, research has tended to focus on social and emotional issues [14]. For instance, the Forrester survey found that entertainment driven users generate more content relative to career driven

users [16]. While increasing the entertainment element of applications may increase user responsiveness, other more fundamental issues relating to how message requests are communicated to users have received relatively little attention.

While social applications may provide significant new value for their users, they also introduce new concerns over privacy and acceptability. Efstratiou et al. [17] investigate user perceptions of privacy and acceptability for participatory sensing applications using surveys and interviews. This study was conducted in the context of a participatory sensing application deployed at a UK research institute, which monitored: user location, identified co-location of users and detected conversations. This information was then shared using online social networks. Efstratiou et al. [17] find that, while many users had privacy concerns, they were willing to accept participatory sensing applications, where they are seen to provide sufficient value. In addition, Efstratiou et al [17] observe that control over how sensed data is released to the online social network is critical to minimizing user privacy concerns. As such direct user input in the generation of social application data is both necessary and desirable.

While issues of privacy and trust govern users decision to participate in social applications other factors may influence user responsiveness (i.e. the likelihood that a message will be lost or have a long response time). The timeliness of users responses is especially critical in terms of the feasibility of participatory social applications that rely on time sensitive data. Nazir et al. [7] developed and launched three social gaming applications on Facebook and used data collected from these applications to examine user response times (i.e. the time period that a user must wait for a response from another user). Message requests were sent in an unpredictable fashion from one user to another with users often being located in different countries. The authors report an average response time of 16.52hrs with the longest response taking as much as 567hrs. If such high response times were inherent to OSNs, this would preclude their use in many time-sensitive applications.

2.3 Social Network Theory
In examining the factors that affect the flow of information from users to participatory social applications we draw on and contribute to existing social network theory. A social network is broadly defined as a set of actors connected by a set of ties. The actors are nodes in a graph and the ties between them are edges. The actors may be, but are not limited to, persons, terms, organizations or concepts [21]. Ties

between actors may represent the flow of resources, interactions, similarities (location) or social relations [22]. Ties may be directed, such as 'gives advice to' or undirected, such as 'similar location' [22].

In this paper we consider a 2-mode network wherein both 'users' and 'social applications' are modeled as actors within the network. Messages sent between the application and users and between the users and the application, are the ties of interest within the network. A significant body of research has examined attributes of actors as affecting the flow of information within the network [23,24,25]. However relatively little work has examined attributes of the ties as affecting information flow, a notable exception being research differentiating between strong and weak ties. The strength of weak ties theory argues that weak ties offer more novel information relative to strong ties. Granovetter (1978) distinguishes between strong and weak ties based on the amount of time, intensity and exchange of reciprocal services. This distinction between strong and weak ties has also been applied to online environments [27].

In this paper we focus on how the attributes of ties (messages) between social applications and their users influence the flow of information within the network. In so doing we examine whether it is possible to optimize the querying of users by selecting the users with based on their observed level of message responsiveness which is here used as a proxy for tie strength. In Section 3 we describe an experiment that measures user responsiveness in social applications and based upon this, in Section 4 we then introduce the User-Rank algorithm that optimizes the targeting of queries to OSN users based upon the results of this study.

3. EXPERIMENT
In order to examine factors affecting message loss and message responsiveness we designed an original experiment that ran for 30 days from December 15th 2012 to January 14th 2013. In total, 70 users were recruited for the study and a total of 3,055 *request messages* were sent to users during the experiment.

3.1 Data Collection
We created a short message to recruit participants, which fit within the 140-character constraint of Twitter. This message contained a link to a web-based participant registration system. By clicking the link users were able to enter their social network usernames, their name and email address. This recruitment message was shared by four of the authors

on this paper on their chosen OSN (i.e. Facebook and/or Twitter). Registered participants were also encouraged to share this message with their contacts by sharing a link to the registration page using their chosen OSN(s) thus allowing the recruitment message to spread over the social network graph. The recruitment process continued throughout the experiment. Our participants responded from Belgium (57%), the UK (18%), Australia (9%), the US (5%), Spain (3%), Poland (2%), India (2%), Ireland (1%), Sri Lanka (1%), Portugal (1%) and Andorra (1%). Of these participants 1% used only Twitter, 23% used only Facebook and 76% used both social networks.

Custom client applications were created for each online social network, with a dedicated account used for Facebook and Twitter communication. In the case of Twitter, the client sent a directed tweet (Twitter message) to participants, while in the case of Facebook; the client sent a private chat message to participants. The distribution of messages to users via the online social network clients was scheduled using standard Linux CRON jobs. The time of day when messages were sent was selected randomly by the experimental scripts but with controlled daily frequency for each participant. Each message contained the following text: *"OSN Connectivity - click this link to help by checking-in: [link]"*. Wherein *[link]* is a HTTP link to the check-in web page that logs the response time, device description and the IP address of the replying node to the experimental database. The check-in web page also presents users with feedback on how many times they have checked in and a link allowing them to drop out of the experiment or share the recruitment message with their neighbors on the social graph.

3.2 Results of Data Analysis
In assessing user responsiveness we consider two key factors: (a) message loss and (b) response time. During our study we sent a total of 3,055 messages. Of these, 1,538 (50.34%) were responded to, while 1,517 (49.66%) were lost. Based upon the time taken for a user to click a web link embedded in the message, we are able to measure response time. Messages not responded to after 12 hours are treated as lost messages.

3.2.1 Message Loss
In order to examine factors affecting message loss we performed a logistic regression analysis of our data. The Time of day at which message is sent (morning, afternoon, evening and night), the OSN used to deliver the message (Facebook or Twitter) and the Rate of messaging are included as explanatory variables. To

examine the effect of Rate of messaging on Message Loss, we split users into two groups. Starting on day 19, half of the group (even user ID numbers) was kept on a constant rate of 1 message per day. The other users were put on an increasing rate of 2, 3, 4 and finally 5 messages per day by the end of experiment.

Variables with a significance level of $p < 0.05$ were judged to have a statistically significant impact on the explanatory variable. The results of our logistic regression analysis are reported in Table 1. In logistic regression with multiple categorical variables, one category is used as a reference and compared to the other categories. We find significant relationships between Time and Rate on Message Loss, however we do not find a statistically significant relationship between OSN and Message Loss.

Table 1: Message Loss						
	B	S.E.	Wald	Df	Sig	Exp(B)
Time *			11.931	3	.008	
Time(1)*	-.360	.105	11.873	1	.001	.698
Time(2)	-.176	.106	2.727	1	.099	.839
Time(3)	.198	.107	3.420	1	.064	.820
Rate*			68.282	4	.000	
Rate(1)*	-.788	.119	43.724	1	.000	.455
Rate(2)*	-.606	.150	16.412	1	.000	.545
Rate(3)	-.181	.140	1.669	1	.196	.835
Rate(4)	-.172	.141	1.501	1	.221	.842
OSN	-.071	.083	.732	1	.392	.931
Constant	.676	.115	34.599	1	.000	1.967
Overall Percentage Correct: 58.2 (50.3)[1]						
Nagalkerke R Squared: .044[2]						
No. of observations: 3,055						

The local time of the user (Time) is found to be statistically significant overall (p=0.008). *Time(4)-night* is used as the reference category and compared to the other time periods. *Time(1)–morning* is statistically significant as such we can say that message loss is significantly different during the morning than at night. *Thus, in order to minimize message loss send messages during the morning period (06.00-11.59).*

We find that the number of messages sent to the user per day (Rate) has statistically significant impact overall (p<0.001). The reference category is *Rate(5)* which indicates 5 messages sent per 24hrs and is compared to the other Rate categories. *Rate(1)* and

---

[1] Overall Percentage Correct refers to the overall percentage of messages correctly classified as lost or not-lost when our predictors (Time, Rate, OSN) are included in the model. This increases from 50.3 (no predictors) to 58.2.

[2] Nagelkerke R Squared is 0.044, which indicates that our model predicts 4% of the variance in message loss. We acknowledge that this value is low, however, we believe that this is acceptable for an exploratory study.

Rate(2) are statistically significant as such we can say that message loss is significantly different when the rate of messaging is 1 or 2 per day compared to when the rate of messaging is 5 messages per day. Thus, *in order to minimize message loss keep the rate of messaging at or below 2 messages per 24hrs.*

3.2.2 Message Response Time

We examine the relationship between Response Time, measured in seconds and the explanatory variables Time of day, Online Social Network and Device (cellular or non-cellular) used to respond to messages using linear regression analysis. The results of the regression analysis are reported in Table 2. We again use $p<0.05$ as our cut-off significance value.

We find a statistically significant relationship between the Time of day at which the message is sent and message response time. If we examine the Beta coefficient (B) we see that a one unit increase in Time of day (e.g. moving from morning to afternoon, or afternoon to evening) leads, on average to a 2,971 second increase in response time. Comparing the standardized Beta coefficients (Std. B), it can be seen that Time of Day has the greatest impact on response time. Thus, *in order to minimize message response time, send messages earlier in the day.*

Table 2: Message Response Time					
	B	S.E.	Std. B	t	sig
(Constant)	3720.93	717.20		5.188	.000
Time of day*	2971.67	236.65	.304	12.557	.000
OSN*	-1327.75	531.39	-.061	-2.499	.013
Device*	-1629.33	551.85	-.072	-2.952	.003
R-Squared: .101[3]					
No. of Observations: 1,538					

The relationships between online social network (OSN) used and message response time was also found to be significant. The negative Beta coefficient indicates that messages delivered via Twitter are, on average, responded to 1,3227 seconds faster than messages delivered via Facebook. Thus, *in order to minimize message response time, send messages via Twitter rather than via Facebook.*

We also find a significant relationship between the Device used and message Response Time. The negative Beta coefficient implies that messages responded to via a cellular devices exhibit a response time, on average, 1,629 seconds faster than messages

---

[3] The R-square of the model is .101, which indicates that the model explains 10% of the variance in message loss. We acknowledge that this value is low, however, we believe that it is acceptable for exploratory research.

responded to on non-cellular device. Thus, *in order to minimize message response time, send messages to users who access online social networks using a cellular device.*

3.2.3 Summary of Experiment Findings

In summary, the findings of our study have a number of implications for the designers of participatory social applications:

(a) *The time at which users are messaged has a significant impact on response times and the number of messages that are lost.* Our results indicate that social application developers can minimize loss and response times by sending queries in the morning where possible. It should also be noted that online social networks are geographically distributed and thus it is the local time of the user that should be considered, rather than the current time at the sender.

(b) *The rate at which users are messaged has a significant impact on message loss.* As the rate of messaging increased above 2 messages per day we observe a significant increase in message loss.

(c) *The online social network via which users are messaged has a significant impact on response time.* We find that messages sent via Twitter have significantly lower response times relative to Facebook users. As such, Twitter is a more effective platform for the delivery of time sensitive messages than Facebook.

(d) *The device used to respond to messages has a significant impact on response time.* Individuals who responded to messages via cellular devices demonstrated significantly lower response times. The efficiency and effectiveness of social applications is likely to increase as smart-phones become ubiquitous.

In the following Section, we describe how we have used the findings of this study to create User-Rank a generic approach to optimizing the querying of social network users.

4. OPTIMIZED QUERYING OF USERS

User-Rank aims to maximize the efficiency of user querying for participatory social applications. This is achieved by selecting users based on their last observed message response characteristics (i.e. device used, OSN used and current local time) and the number of messages sent from the application to the user during a 24hr period (i.e. rate). Based on these characteristics, each user is then assigned a ranking that estimates their potential for answering a query. A

high ranking indicates that the user is likely to be very responsive to data requests (strong tie). Application developers may use this ranking to optimize querying by messaging the highest ranked users first and then continuing down the list until the required number of responses has been achieved. User-Rank thus allows time-sensitive queries to be successfully executed using fewer users.

4.1 User Model and Data Gathering

Scalability is critical for participatory applications that are required to support and gather data from millions of users. It is therefore critical to minimize the memory requirements of modeling each user. Specifically, User-Rank stores a user ID variable, and one variable for each of the four determinants of user responsiveness (OSN type, local time, rate, and device) as identified by the study described in Section 3. In the case of local time, the 'time-difference' is stored based upon the time zone from which the user was last observed to respond. The time-difference is then used to calculate the users current local time whenever a query is ready to be sent. A description of each variable follows:

- *ID*: the unique text username of this social network user (i.e Twitter handle or Facebook login).
- *OSN:* a numeric identifier that represents the social network that the user is using.
- *rate*: a numeric value that represents the number of messages that have been sent to the user during the current day according to the sender's clock.
- *time-difference:* a numeric value that represents the offset in units of 30 minutes between the local time at the sender and the local time at the user.
- *device:* is a Boolean value that represents whether the user is on a cellular device.

Our approach to gathering user data requires only a standard web server and an OSN client that is capable of messaging the users. Each request that is sent to an OSN user embeds a unique web link that associates each response to its corresponding query. The user must click the link in order to respond. When the user follows the embedded link, the web server resolves the IP address of the user to a location, which is used to set the *time-difference* variable. The browser agent string, a standard feature of the HTTP protocol [19], is then used to set the *device* variable. The *OSN* and *ID* fields are identified based upon the embedded hyperlink, which associates a response to a specific user query. The *rate* variable, which is measured in messages-per-day is simply incremented every time a user is messaged, until it is reset to zero at midnight.

Two features of the data gathering process are noteworthy: (a.) our approach gathers no more data than a standard web-site tracker and (b.) all of the data required to model a user can be encoded in just 54 bytes of memory: 50 bytes for OSN username and 1 byte each to store OSN, rate, time offset and device type. Naturally, the inclusion of more meta-data on users would allow for more specific filtering; however this comes at the cost of more invasive monitoring and greater memory requirements.

4.2 Weighting of the Rank Variables

The weight of the ranking variables are set based upon the observed effect of these variables on user response times and message loss in a set of 'training data' (i.e. data that is used to set the weightings of the algorithm). In Section 5 we create four instances of the algorithm each using one weeks worth of training data and then evaluate the performance of each instance on the three other weeks of data on which it was not trained.

For every *categorical variable*: "time", "OSN" and "device", the weight for each category-value is set to the average observed decrease in response time (measured in seconds) observed for users exhibiting that category, compared to the average response time observed across all training data. For example: if user response times are on average 100 seconds quicker than average where the users local time is 'morning', the weight for "time = morning" is set to 100. For the continuous variable "rate", weights are set as the average decrease in response time for every 1 unit increase in the rate of messages. For example, if response times increase on average by 100 seconds for every additional message per day, the weight for "rate = 5" would be set to 5 * -100 = -500. The total *Rank* for each user is thus the sum of weights for all observed values. The ranking is then converted to *Normalized-Rank* on a 0 to 100 scale using the following formula:

$$Normalized\text{-}Rank = 100/Maximum\text{-}Rank * Rank$$

Where *Maximum-Rank* is the maximum ranking assigned to a user in the training data. The use of a normalized scale allows for the use of fixed-size variables within the User-Rank algorithm.

4.3 The User-Rank Algorithm

The User-Rank algorithm takes an unordered, set of OSN users as input (`population`) and returns these OSN users ranked according to their potential for answering queries (`ranked_population`). The User-Rank algorithm is simple and operates as follows:

1. For each user in the filtered set, a rank is calculated based upon the weightings described in Section 4.1.
2. The filtered and ranked set of users is sorted by rank using the well-known Quick-Sort algorithm [18] and returned.

This algorithm is shown in pseudo-code in Listing 1. Numbered comments indicate which block of code corresponds to each stage of the algorithm as described above. Weighting variable names are summarized in Table 3.

Listing 1: The User Rank Algorithm

```
User-Rank(population){

rankedPopulation = new set;

for all users in population {

  u = next user from population;

  // 1. Set user ranking and add to set
  rank = 0;
  time = local_time + u.time-difference;
  if (time = Morning) rank = rank+WTm;
  if (time = Afternoon) rank = rank+WTa;
  if (time = Evening) rank = rank+WTe;
  if (time = Night) rank = rank+WTn;
  if (u.device = Cell) rank = rank+WDc;
  if (u.device = Non-Cell) rank = rank+WDn;
  if (u.osn = Twitter) rank = rank+WOt;
  if (u.osn = Facebook) rank = rank+WOf;
  rank = rank+(Wr*rate);
  u.rank = rank;
  add u to rankedPopulation;
  }
}

// 2. Sort set by rank and return
Quick-Sort(ranked_population by rank);
return ranked_population;
}
```

Table 3: Weighting Variable Descriptions

Variable	Description
WTm, WTa, WTe, WTn	Weight for Time = *m*orning, *a*fternoon, *e*vening or *n*ight.
WDc, WDn	Weight for Device = *c*ellular or *n*on-cellular.
WOt, WOf	Weight for OSN = *t*witter or *f*acebook.
Wr	Weighting for rate = *r*.

5. EVALUATION

In this section, we evaluate the performance of User-Rank using the trace data we gathered during the empirical study described in Section 3.

5.1 Evaluation Environment

User-Rank was implemented in Java 1.6 and all tests were executed on Mac OS X Version 10.8.4 running on a 1.3GHz Intel Core i5 with 4GB of RAM. To evaluate User-Rank, we used our trace data from the 30-day experiment described in Section 3. We split this data set into four samples of 1 week each. These data sets contained on average 750 queries, allowing us to simulate a user population of the same size. The data from each week was then used to train a unique instance of the User-Rank algorithm (UR-WK1 to UR-WK4) by setting the variable weightings as described in Section 4.2. The four algorithm instances were then tested on the other three weeks of data on which they had not been trained. It is important to note that all instances of the algorithm were tested on *different* data sets than those they were trained on, thus ensuring that the performance results are representative. For each test case we recorded the number of users that must be messaged in order to receive a certain number of responses (from 10 to 50) within 60 minutes. The performance of each instance of the User-Rank algorithm is then compared against random querying. We also measured the time that the algorithm takes to generate the ranked data set in each case.

5.2 Evaluation Results

Figures 1 to 4 show the number of users that must be queried in order to receive 10 to 50 responses within 60 minutes using the week 1 to 4 data sets respectively. The performance of random querying (RANDOM) is compared to User-Rank instances trained on data from each of the other three weeks (labeled UR-WK1 to UR-WK4).
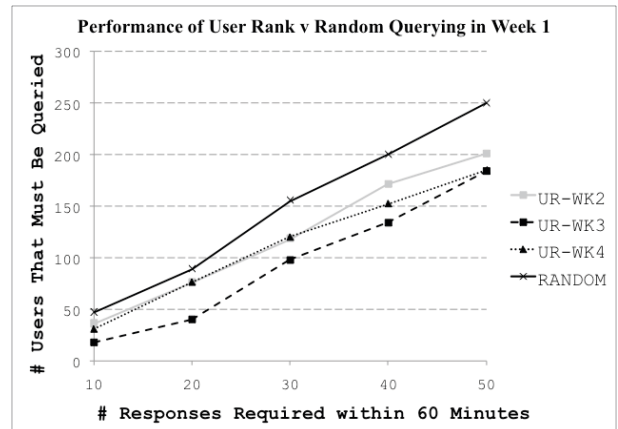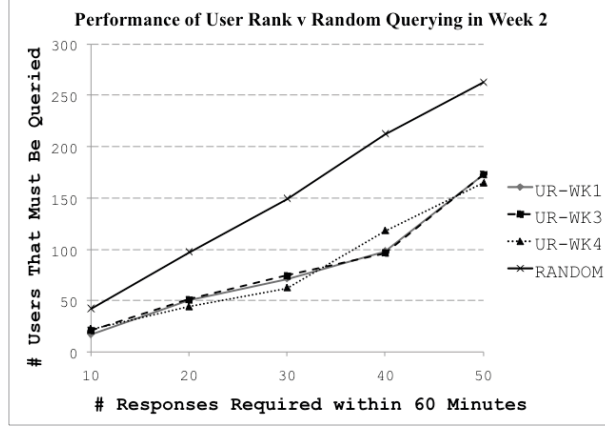

Figure 1 – Performance of User Rank in Week 1

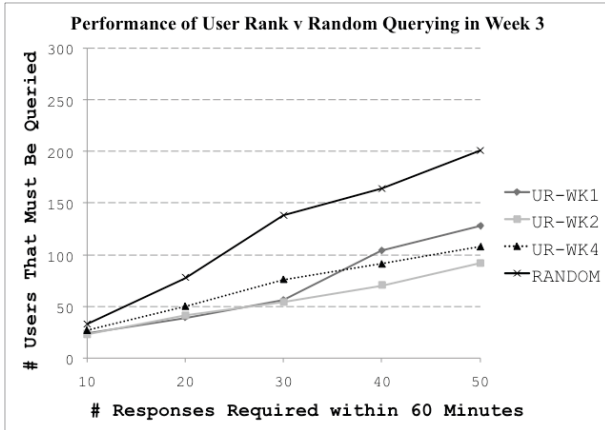**Figure 2 – Performance of User Rank in Week 2**

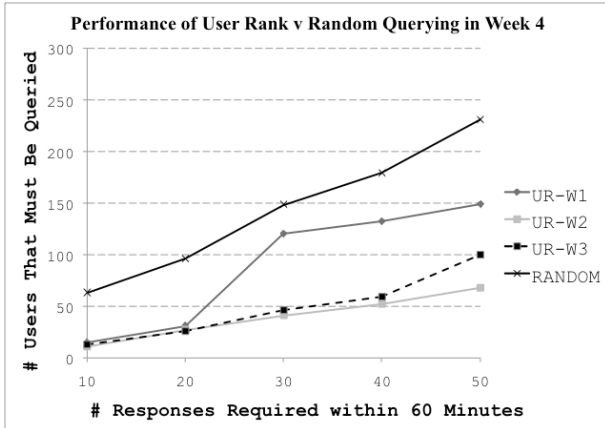**Figure 3 – Performance of User Rank in Week 3**

Figure 4 – Performance of User Rank in Week 4

Querying Efficiency: As can be seen from Figures 1 to 4, User-Rank outperforms random querying for all instances on all data sets. In the worst case, running UR-WK2 on the Week 1 data set, User-Rank reduced the number of users that must be messaged by 19%. In the best case, running UR-WK2 on the Week 4 data set, User-Rank reduced the number of users that must be messaged by 72%. The average saving across all test cases was 39.44%.

Algorithm Performance: Table 4 summarizes the execution time of the algorithm for all test cases. For each test the algorithm was executed 10 times and the results were averaged. As can be seen from Table 4, User-Rank is fast, taking a maximum of 2ms to rank the target data set.

Table 4: Test-case Execution Time

Test #	Execution Time (ms)
Week 1	1.4
Week 2	1.8
Week 3	1.9
Week 4	2.0

Efficient Use of Population: an interesting feature of User-Rank is that it makes good use of the full user population. On average, the ranking of a user varies by 55 points on the 100-point normalized ranking scale during the course of the experiment. Furthermore, as the rate at which a user is messaged increases, their rank decreases, ensuring that no user is flooded with messages. Over 78% of our experimental participants were ranked in both the top and bottom quartile at some point in our 30-day experiment.

6. DISCUSSION AND CONCLUSIONS

This paper has considered the problem of user responsiveness in participatory social applications. To better understand this problem we performed a 30-day study of 70 users on the Twitter and Facebook social networks, wherein we sent 3,055 request messages to participants and logged the response time and loss rates of users. This study allowed us to identify the statistically significant determinants of message loss (i.e. local time, rate of querying) and response time (i.e. local time, OSN and device). Based upon the findings of our study, we propose User-Rank, which optimizes the querying of OSN users by considering the effects of these determinants on user responsiveness.

At a theoretical level our results demonstrate that the notion of strong and weak ties can be usefully applied to online social networks. In Granovetter's [26] formulation tie strength is a property of the emotional relationship between two actors. However in online social networks such a relationship may be difficult to measure. In this paper we have demonstrated a more general approach, wherein tie strength is distinguished based on their contribution to the outcome of interest, in this case user responsiveness to messages sent. The User Rank algorithm operates by analyzing the last tie

observed between a certain user and the application and applying weight to the attributes of the tie proportional to their observed impact on message responsiveness. We then use the estimated current tie strength to optimize which users are selected for interaction with the application. We expect these findings to be generally applicable across social networks, particularly those networks that have a central coordinating actor(s).

At a practical level our preliminary evaluation of User-Rank shows that it is capable of reducing the number of users that must be messaged in order to successfully fulfill a query, where success is defined as receiving a certain number of response messages by a specified deadline. This tackles the critical problems of high message loss and slow user response times as identified in prior work by Demirbas et al. [6] and Nazier et al. [7]. User-Rank significantly outperforms random querying of users in all cases, reducing the number of users that must be messaged by an average of 39.44%. Furthermore, the user rank algorithm has minimal computational and memory requirements. The full 3055 response data set can be stored in under 165KB of memory and the worst-case execution time of User-Rank is 2ms. This implies that User-Rank may be applied directly to Internet-scale data sets.

7. Limitations and Future Work

Limitations of our data set: The work presented in this paper is based on a relatively small-scale data set of ,3055 queries sent to 70 users over 30 days. While our experimental participants were diverse, responding from 11 countries, it is likely that the population exhibits different demographics to a 'typical' population of OSN users. Extending our experiments to greater scale in terms of number of users and time-scale is a priority for our future work. This will also provide the opportunity to test User-Rank on new data 'in the wild'.

Software architecture: It should be noted that while User-Rank provides an important building block that is required to implement a complete querying system for OSN users, important elements are still missing. Here, our priorities are two-fold: (a) to develop a supporting middleware platform that is capable of scaling to support many simultaneous queries and (b) to create a suite of user-facing tools that allow for the formulation of accurate and efficient queries.

Modeling the user: Our findings imply that significant optimizations can be made to user querying without the need for detailed user information or complex algorithms. However, User-Rank is a first step and it likely that further research in this area will contribute more complex algorithms that are capable of more accurately predicting the performance of users in answering queries. Specifically, in our future work we will strive to build more complete models of user responsiveness, in particular incorporating attributes of users into the model. In particular, future work will focus on users motivation for participating in social applications.

8. Acknowledgements

9. References

[1] Facebook user numbers: http://newsroom.fb.com/Key-Facts

[2] Twitter user numbers: http://mashable.com/2012/12/18/twitter-200-million-active-users/

[3] B.R. Lindsay, "Social Media and Disasters: Current Uses, Future Options and Policy Considerations", CRS Report for Congress, www.crs.gov, 2011, Washington, pp1-10

[4] "Farmville 2 nears 50 million monthly-active users", available online at: http://www.gamesindustry.biz/articles/2012-10-17-farmville-2-nears-50-million-monthly-users

[5] X. Chen, E. Santos-Neto, M. Ripeanu, "Crowd-based Smart Parking: A Case Study for Mobile Crowdsourcing", 5th International Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications(MOBILWARE 2012), Berlin, Germany.

[6] M. Demirbas, M. A. Bayir, C. G. Akcora, Y.S. Yilmaz and H. Ferhatosmanoglu, "Crowd-sourced sensing and collaboration using twitter, "World of Wireless Mobile and Multimedia Networks (WoWMoM), IEEE International Symposium on a, 2010, pp.1-9

[7] A. Nazir, S. Razir and C-N. Chuah, "Unveiling Facebook: A Measurement Study of Social Network Based Applications", IMC '08, ACM, New York, 2008, pp43-56

[8] V. Lenders, E. Koukoumidis, P. Zhang and M. Martonsi, "Location-based Trust for Mobile User-generated Content: Applications, Challenges and Implementations", HotMobile'08, Napa Valley CA, 2008, pp.60-64

[9] Boyd, d. m., & Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, *13*(1), article 11.

[10] OECD, "Participative web and user-created content: Web 2.0 wikis, and social networking", Organisation for Economic Co-operation and Development, Paris, 2009.

[11] E. Miluzzo, N.D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S.B. Eisenman, Z. Zheng and A. Campbell, "Sensing Meets Mobile Social Networks: The Design Implementation and Evaluation of the CenceMe Applicatio (SenSys '08), ACM, North Carolina, 2008, pp.337-350

[12] A. Rosi, M. Mamei, F. Zambonelli, S. Dobson, G. Stevenson, and J. Ye, "Social sensors and pervasive services: Approaches and perspectives", 2nd IEEE Workshop on Pervasive Collaboration and Social Networking (PerCol'11), IEEE, Seattle, pp. 525-530.

[13] Noulas, S. Scellato, C. Mascolo and M. Pontil, "An Empirical Study of Geographic User Activity Patterns in Foursquare", Proceedings of the 5th International AAAI Conference on Weblogs and Social Media, 2011, pp.570-573

[14] R. Chen and Y.Sakamoto, "Perspective Matters: Sharing Crisis Information in Social Media", 46th Hawaii International Conference on Systems Sciences, IEEE, 2013, pp.2033-2041

[15] S. Deterding, M. Sicart, L. Nacke, K. O'Hara and D. Dixon, "Gamification. Using Game-design Elements in Non-Gaming Contexts, Proceedings of the 2011 annual conference extended abstracts on Human Factors in Computing Systems, Vancouver Canada, 2011, pp.2425-2428.

[16] C. Li, "Mapping Participation in Activities Forms the Foundation of a Social Strategy", Social Technograohics Trennds Report, Forrester Reserch Inc., www.forrester.com/go?docid=42057, 2007

[17] C. Efstratiou, I. Leontiadis, M. Picone, K. Rachuri, C. Mascolo and J. Crowcroft, "Sense and sensibility in a pervasive world", Proceedings of the 10th International Conference on Pervasive Computing (Pervasive '12), 2012, Springer-Verlag, Berlin, pp.406-42

[18] Sedgewick, R. (1978). "Implementing Quicksort programs". Comm. ACM 21 (10): 847–857.

[19] R. Fielding, J. Gettys, M. Jeffrey, H. Nielsen, L. Mainster, P. Leach and B. Lee (June 1999). "RFC 2616: Hypertext Transfer Protocol – HTTP/1.1"

[20] Farmville, available online at: http://www.farmville.com

[21] S. Borgatti and P. Foster, (2003), "The Network Paradigm in Organizational Research: A Review and Typology", Journal of Management 29(6) pp.991-1013

[22] S. Borgatti, A. Mehra, D. Brass and J. Labianca, (2009) "Network Analysis in the Social Sciences, Science, 13 February 2009, pp.829-895

[23] S. Borgatti and R. Cross, (2003) "A Relational View of Information Seeking and Learning in Social Network", Management Science 49(4) pp.432-445

[24] A. Mehra, M. Kilduff, D. Brass, (1998) "At the Margins, A Distinctiveness Approach to Social Identity and Social Networks of Underrepresented Groups", Academy of Management Journal 41(4) pp.441-452

[25] D. Brass, (1981) "Structural Relationships, Job Characteristics and Worker Satisfaction and Performance", Administrative Science Quarterly, 26(3), pp. 441-452

[26] M. Granovetter, (1978), "The Strength of Weak Ties", American Journal of Sociology, 78(6) pp.1360-1380

[27] E. Gilbert and K. Karahalios, (2009), "Predicting Tie Strength with Social Media", CHI 2009- Online Relationships, pp.119-135