Reorder User's Tweets

KEYI SHEN, Tencent
JIANMIN WU, Yahoo! Beijing R&D Center
YA ZHANG, Shanghai Jiao Tong University
YIPING HAN, Yahoo! Beijing R&D Center
XIAOKANG YANG, LI SONG, and XIAO GU, Shanghai Jiao Tong University

Twitter displays the tweets a user received in a reversed chronological order, which is not always the best choice. As Twitter is full of messages of very different qualities, many informative or relevant tweets might be flooded or displayed at the bottom while some nonsense buzzes might be ranked higher. In this work, we present a supervised learning method for personalized tweets reordering based on user interests. User activities on Twitter, in terms of tweeting, retweeting, and replying, are leveraged to obtain the training data for reordering models. Through exploring a rich set of social and personalized features, we model the relevance of tweets by minimizing the pairwise loss of relevant and irrelevant tweets. The tweets are then reordered according to the predicted relevance scores. Experimental results with real twitter user activities demonstrated the effectiveness of our method. The new method achieved above 30% accuracy gain compared with the default ordering in twitter based on time.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Personalization, reorder, retweet, twitter

ACM Reference Format:
Shen, K., Wu, J., Zhang, Y., Han, Y., Yang, X., Song, L., and Gu, X. 2013. Recorder user's tweets. ACM Trans. Intell. Syst. Technol. 4, 1, Article 6 (January 2013), 17 pages.
DOI = 10.1145/2414425.2414431 http://doi.acm.org/10.1145/2414425.2414431

6

1. INTRODUCTION

Twitter, as a widely used social networking and microblogging service, enables users to instantly share their latest status and thoughts in the form of short messages of no more than 140 characters known as tweets. A tweet may be an original message, a forwarded message called retweet, or a reply to a message which is only visible to the two users involved and their mutual friends. A user may choose to "follow" other users on Twitter and become their friends/fans, that is, subscribe to their messages, so that the subscribed messages are displayed on his/her own Twitter page.

This work was partially supported by National Basic Research Program of China (2010CB731401 and 2010CB731406), Shanghai Science and Technology Rising Star Program (11QA1403500), Shanghai Talent Development Fund (2010002), and STCSM (12DZ2272600 and 2011XJRHBT0078).
Authors' addresses: K. Shen (corresponding author), Tencent; email: shenkeyi@gmail.com; J. Wu, Yahoo! Beijing R&D Center, China; Y. Zhang, Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, China; Y. Han, Yahoo! Beijing R&D Center, China; X. Yang, L. Song, and X. Gu, Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, China.

DOI 10.1145/2414425.2414431 http://doi.acm.org/10.1145/2414425.2414431

Twitter messages are believed to contain a rich source of instantly updated information about the world. Several studies have successfully used Twitter messages to predict the stock market [Bollen et al. 2010], election results [Tumasjan et al. 2010], swine flu pandemics [Ritterman et al. 2009], and box-office revenues for movies [Asur and Huberman 2010].

With the increasing popularity of Twitter, a large volume of messages are produced. As of June 2011, Twitter receives more than 200 million tweets per day. As a matter of fact, many users on Twitter are flooded with a large volume of messages of different quality. People tend not to have enough time to read all their friends' tweets carefully all the time. On the other hand, a study by Pear Analytics has put tweets into six categories: pointless babble (41%), conversational (38%), pass-along value (9%), self-promotion (6%), spam (4%), and news (4%) [Kelly 2009]. A significant portion of tweets are in fact meaningless. It is hence desirable to order the tweets in a meaningful way so as to relieve the information overloading problem and facilitate user browsing.

So far Twitter allows users to follow an individual or a list of individuals. But a user is often only interested in part of the messages his/her followees publish. For example, a student in computer science may follow a computer expert due to the interest in his comments on IT technology instead of his minutiae of daily life. On the other hand, a friend of the computer expert may want to follow him to stay updated about his personal life. As a matter of fact, tweets of different topics from the same person may have different relevance to different readers. Hence, it is desirable to order the tweets according to the the interests of a user.

Twitter currently orders tweets by the reversed chronological order. The problem of ordering users' tweets becomes nontrivial if we want to consider the differences in tweet quality and user interests. First of all, tweets are restricted to 140 characters in length and usually in informal language. These characteristics make it hard to perform semantic analysis such as topic detection on the tweets. Secondly, personalization of the ordering involves the understanding of users' preference and interests, which makes the problem even more difficult. Thirdly, the evaluation of the effectiveness of the personalized reordering is also a challenging problem. Generally speaking, we do not have the ground truth of each individual user's real information need, though we may resort to editorial help for model evaluation. But the editorial labels are time consuming and expensive to obtain. Moreover, editors are often not able to accurately judge the personalized information need of individuals, making the editorial-based evaluation method not applicable in this scenario.

Here we adopt machine-learning-based approaches to reorder tweets according to their quality and user interests so that tweets with low quality will rank low, and tweets that are attractive to the user are ranked high in the user's homepage.

To identify the topics in the tweet corpus and users' interests over these topics, we first aggregate the tweets generated or consumed by each user. Then a topic model is built by treating the aggregated tweets associated with each user as a single document. The word co-occurrence in the aggregated tweets is more reliable for topic model than using a single tweet as document. Moreover, the word distribution of each topic is well tailored with regards to the languages used in tweets. We may then infer topics of each tweet with the learned topic model.

We leverage users' feedbacks such as retweeting and replying activities for training and evaluation. The underlying assumption is that a user's interests in different topics are implicitly reflected in those observable activities. For example, a user is more likely to be interested in the tweets that he publishes than those that he does not interact with. These behavior-based data are used to both train our machine learning models and evaluate the prediction of our machine learning model. We further explore a set of features derived from the content of the tweets, users' behaviors, and users' social

graph for their prediction power of personalized relevance. We show that with the help of tweets reordering, users are able to find more informative messages, which is especially useful when users have lots of newly received tweets to read.

The major contributions of the article are as follows. First, we use the observable activities of users on Twitter such as retweeting and replying as the ground truth to train and evaluate our machine learning model, which empowers us to build a personalized ranking model. Second, we explore a wide variety of features for reordering, including many personalized features. Our experimental results demonstrate the effectiveness of personalized features in reordering, that is, the personalized reordering algorithm outperforms the default baseline and the same machine learning method that excludes the personalized features.

The rest of the article is organized as follows. We provide related work in literature in Section 2. In Section 3, we introduce details of the proposed social and personalized features. The training process utilizing users' activities on Twitter for relative relevance labeling and the machine learning ranking model for tweet reordering are presented in Section 4. In Section 5, we present the experimental results on Twitter data, which illustrate the effectiveness of our method. We conclude the article with Section 6.

2. RELATED WORK

Messages on Twitter are mostly time sensitive.

Many commercial search engines offer organic search and ranking for tweets. Y. Duan et al. have used a learning to rank method [Joachims 2002] to rank the tweets with a set of 20 queries for testing [Duan et al. 2010]. R. Nagmoti et al. have proposed and empirically compared several strategies to rerank the top-k tweets returned by an existing tweets search engine [Nagmoti et al. 2010].

Analyzing of Twitter users has also drawn much attention from both the industrial and research community. *Tweetfind.com* and *Twitority.com* rank the tweets according to the authority of the corresponding users on Twitter. J. Weng et al. focus on the problem of identifying influential Twitter users over different topics [Weng et al. 2010]. H. Kwak et al. rank the users on Twitter by accounting for the number of followers, PageRank [Page et al. 1998] score from Twitter users' friendship graph, and the number of tweets being retweeted [Kwak et al. 2010].

Recommendation or reorganization of tweets is another interesting topic on Twitter research. J. Chen et al. present a way to recommend URLs in tweets [Chen et al. 2010]. Their recommendation system takes the content, topic interest, and social voting into consideration. The Web sites *Paper.li* and *Twitter Times* turn the user's tweets stream into an online newspaper. *Paper.li* offers tags to help users generate their newspapers by help of different categories and topics. But this personalized newspaper only updates once every 24 hours. *Twitter Times* displays materials based on how many of one's friends and friends' friends have retweeted this tweet. C. Castillo et al. studied the information quality on Twitter [Castillo et al. 2011]. They classified news on Twitter as credible or not credible by a set of features extracted from tweets.

In Suh et al. [2010], B. Suh et al. studied the information diffusion on Twitter by examining a set of features that might affect the probability of a tweet being retweeted. Z. Yang et al. proposed a factor graph model to predict retweet behaviors [Yang et al. 2010].

All of the afore mentioned studies primarily focus on the tweet search ranking or recommendation, and they involve tweets from all Twitter users. However, our tweets reordering work only considers the tweets from the user's own feeds. Since the tweets that the user u received are generated by u's followees, we can utilize the personalized features between the information publisher and the receivers. But in the case of ranking or recommendation, the candidate tweets may come from unknown

Table I. The Comparison among Tweets Ranking, Recommendation and Reordering

	With query?	Tweet candidates	Personalized features
Ranking	Yes	All tweets	no/seldom
Recommendation	No	All tweets	some
Reordering	No	Followees' tweets	more

authors, and thus no historical information about the user's preference is available to this author. Table I gives a comparison among the tweets ranking, recommendation, and our reordering method. Due to the controlled source of the candidate tweets, our Twitter reordering method can utilize more personalized and social features to make the reordering more relevant.

3. FEATURE EXTRACTION FOR REORDERING TWEETS

To achieve personalized reordering of tweets, we explore a set of features to capture a user' interest level regarding the tweets he/she receives. For a user u, a tweet m posted by the user a, and the time t when u read the tweet m, the extracted features include the freshness and quality of the tweet m, the authority of the author a, as well as personalized social features such as interest match between the user u and the author a. We then use one of the state-of-the-art machine learning methods to build reordering models based on the extracted features. In the rest of this section, we describe these personalized reordering features in detail.

3.1. Freshness of the Tweets

The most notable trait of Twitter is its capability to capture recency-sensitive information. Hence, we expect freshness to be an important factor in tweet reordering. We define two features based on freshness of a tweet as follows.

—$\phi_1 = FT(t, u, m)$, *time freshness*: the difference between the time t the user u saw this tweet m and the time this tweet was posted.
—$\phi_2 = FR(t, u, m)$, *rank freshness*: the rank of the tweet m in the Twitter's timeline at the time t when the user u visits Twitter.

$FR(t, u, m)$ and $FT(t, u, m)$ measure the freshness of tweets from slightly different points of view, in terms of absolute time and relative rank, respectively. Two adjacent tweets in the timeline, which have minor difference in terms of $FR(t, u, m)$, could have large difference in terms of $FT(t, u, m)$, in the case that no new tweet is received by the user during a long period of time. On the other hand, two tweets that were posted at almost the same time could have a large difference on $FR(t, u, m)$ if a large number of tweets are posted by the user u's followees at the same time.

3.2. Influence of Authors

A user's interest to a tweet might be consciously or unconsciously influenced by the corresponding author's authority. Here we attempt to use the following set of features to reflect the influence of an author a, including one's popularity and active level on Twitter. Some of the features such as active level by their own may not be highly correlated to an author's influence. We still include them in this study because they are likely to contribute to an author's influence.

—$\phi_3 = FoC(a)$, *follower count*: the number of followers for the author a. This feature may be perceived as an indication of the popularity of the author.
Note that celebrities usually have lots of followers on Twitter. Even though the tweets from a celebrity are not always of high quality, the followers are interested in the tweets in general because they are interested in knowing almost everything about

the celebrity. These people with large number of followers are generally influential ones in the Twitter community. According to Kwak et al. [2010], the result of ranking the Twitter users according to their $FoC(a)$ is similar to that of the PageRank [Page et al. 1998] ranking based on the following/follower graph of Twitter users.

—$\phi_4 = FeC(a)$, *followee count*: the number of users that the author a follows. The followee count is a counterpart of the follower count. If a has lots of followers but few followees, a might pay more attention on sharing information; if user a has lots of followees, a might focus on what's happening among a's followees.

—$\phi_5 = LC(a)$, *list count*: the number of lists the author a belongs to. The study in Duan et al. [2010] suggests that the number of times a is listed by other users is an effective representation for a's authority.

—$\phi_6 = SC(a)$, *status count*: the number of tweets the author a posts each day. We expect that the influence of a is not proportional to $SC(a)$. But this value characterizes the author of the tweet.

—$\phi_7 = D(a)$, *days*: the number of days since a's account was created. According to the result in Suh et al. [2010], tweets from "senior" Twitter users (whose accounts were created before 1 year ago) and recent users (whose accounts were created less than a month ago) are more likely retweeted.

—$\phi_8 = VA(a)$, *verified account*: a binary number to indicate whether the author a is verified by Twitter officially or not. Most of the verified Twitter accounts belong to celebrities and well-known people. Hence, this feature is another indicator for popularity or authority.

3.3. Quality of Tweets

The quality of the tweet is expected to be one of the key factors that determine the rank of tweet. We define the following features to measure the quality of tweets.

—$\phi_9 = L(m)$, *length*: the length of the tweet m normalized by the maximum number of characters allowed. Considering many of the tweets are just some buzzes with very short length, longer tweets usually are more formal and informative.

—$\phi_{10} = CU(m)$, *containing URL*: a binary feature to indicate whether the tweet m contains URLs. With the length limitation for tweets, it is quite common for tweets to contain a short URL which points to a Web page with more details. Therefore, tweets with short URLs could provide more information.

—$\phi_{11} = HC(m)$, *hashtag count*: the number of hashtags that appear in the tweet m. Hashtag is a special word in tweet with a leading character "#", which is designed to indicate topics being covered in this tweet. Tweets with appropriate hashtags usually can be classified with less efforts.

—$\phi_{12} = RtC'(t, m)$, *retweet count*: the number of retweets that rooted from the tweet m by the time t when the user reads it. This feature represents the social voting for the tweet m from all other users. The more people share this tweet, the more likely that it is of high quality. The retweet count might be manipulated by the spammers [Duan et al. 2010] for general tweets search. However, in our case, the number is a reliable criterion because all the tweets we consider are posted by the user u's followees.

—$\phi_{13} = RtC''(t, u, m)$, *retweet count by followee*: the number of times that the tweet m being retweeted by the user u's followees by the time t when u saw it. A user tends to share similar interest to his social contacts. So retweet count by followees can be perceived as a type of recommendation by one's social contacts.

3.4. Interests of the Users

Match between the topic of a tweet and a user's interest is an important factor for ordering the newly received tweets. We infer the interests of a user by the topic model

based on the tweets the user generated and consumed in the past. Topics of the received tweets are inferred by the same topic model. To be more concrete, for each user, we represents this user as a "document" with words from the tweets this user generated and consumed. Then we train the PLSA [Hofmann 1999] topic model with the corpus of all Twitter users in our dataset.

Given the number of topics K, the training of a PLSA model is to maximize the likelihood of the corpus with the factorized conditional distribution: $P(w|u) = \sum_z P(w|z)P(z|u)$. The resulting model consists of the distribution of user interests over topics $P(z|u)$ and the word distribution $P(w|z)$ of topic z, here $z \in Z \triangleq \{1, \cdots, K\}$ represents the topic. Topic distribution $P(z|m)$ of a tweet m can then be inferred by the same fold-in scheme as in Hofmann [1999]. Given a tweet m, we can use vector $\vec{T}(m) = c_1(m)\vec{z}_1 + c_2(m)\vec{z}_2 + \cdots + c_K(m)\vec{z}_K = (c_1(m), c_2(m), \cdots, c_K(m))$ to represent the tweet m's topic distribution over K topics (i.e., topic $\vec{z}_1$ to $\vec{z}_K$), where $c_i(m)$ is the probability that tweet m is related to topic $\vec{z}_i$. And similarly a user u's interest over different topics can be denoted as $\vec{T}(u) = (c_1(u), c_2(u), \cdots, c_K(u))$. Here $\vec{T}(u)$ can be calculated by the tweets u generated and consumed and the topic distribution $\vec{T}(m)$ of those tweets.

With the topic models, we propose the following two interests matching features.

—$\phi_{14} = IMT(u, m)$, *match between tweet and user interests*: for the tweet m and the user u, the interests match is measured by the inner product of the vectors $\vec{T}(u)$ and $\vec{T}(m)$, that is, $\vec{T}(u) \bullet \vec{T}(m)$.

—$\phi_{15} = IMA(u, a)$, *interest match between the author and the user*: for the author a and the user u, the interests match is measured by the inner product of the vectors $\vec{T}(u)$ and $\vec{T}(a)$, that is, $\vec{T}(u) \bullet \vec{T}(a)$.

These two topic-model-based features enable us to encode the topic-wise personalized information in the reordering model without additional editorial efforts required. We will analyze the detailed benefits from these features in the experiment part.

3.5. Personalized Social Features

Social characteristics should be an important factor when talking about reordering tweets. Here we consider several personalized social features which represent the social relationship between the user u and the author a based on historical behaviors. The resulting model based on these features is thus personalized for each individual Twitter user.

—$\phi_{16} = RtC(u, a)$, *retweet count by user*: the number of a's tweets being retweeted by u in the past. This is an important indicator for u's interests on a's tweets.

—$\phi_{17} = RpC(u, a)$, *reply count by user*: the number of a's tweets being replied by u in the past. Intuitively, if u replies frequently to a's tweets, u may have great interests to a's tweet.

—$\phi_{18} = RtR(u, a)$, *retweet ratio*: the percentage of a's tweets retweeted by u.

$$RtR(u, a) = \frac{RtC(u, a) + 1}{SC(a) + 1} \tag{1}$$

—$\phi_{19} = RpR(u, a)$, *reply ratio*: the percentage of a's tweets replied by u.

$$RpR(u, a) = \frac{RpC(u, a) + 1}{SC(a) + 1} \tag{2}$$

The latter two features, which may be perceived as the normalized retweet count and reply count, are used here to alleviate the bias introduced by absolute values in case a user has large number of updates.

4. REORDER MODEL

We adopt a supervised learning framework for building the tweet reordering model. With the features defined earlier, the remaining task is to generate a set of high-quality training data. For a tweet in a user's tweet stream, we need to assign a score to indicate the user's interest level to this tweet (relevance of the tweet to the user). Due to the personalized nature of the reordering, editorial labeling of the data is not feasible. In this study, considering the fact that Twitter data are public by default, we attempt to leverage the observable user behavior data on Twitter to build the training set. Next we first introduce a heuristic strategy to generate user sessions, from which we extract relevance scores in the training data, followed by the procedures for model training.

4.1. User's Session

The purpose of the tweet reordering model is to reorder the user's recently received (unread) tweets. So we have to find out what tweets the user u receives in the *k-th* visit and what tweets the user u is most interested in. Furthermore, we need to get the rank of each tweet in the timeline order at the time it was read in u's *k-th* visit. Because we have no access to the information of user's log-in and log-out activities on Twitter, the first step of our data preparation is to segment users' activities into sessions to identify the set of new tweets in each of the user's visits. We here present a heuristic method to approximately define a user session based on users' observable activities such as tweeting, replying, and retweeting. The relative relevance is then defined for tweets within the same session.

Let us denote the set of tweets received by the user u as S^u, where $S^u = \{m_i | i = 1, 2, \cdots, N\}$. For the rest of this section, we will drop the notation of the user u since we consider one user at a time for the purpose of session partition. The task is to partition S into multiple consecutive subsets so that each subset corresponds to a user session. The tweets in the *k-th* session are defined by

$$S(k) = \{m | B(k) \leq t(m) < B(k+1), \forall m \in S\}, \tag{3}$$

where $B(k)$ is the start time of the *k-th* session and $t(m)$ is the publishing time of the tweet m. We estimate $B(k)$ with observable user behavior data.

Let $A = \{a_{m_1}, a_{m_2}, \cdots\}$ denote the set of timestamps when any action was taken such as tweeting, retweeting, and replying. We estimate the starting time of a session as one of the timestamps in A. The underlying guideline is that the starting time for a session should be close to the time when the earliest action was taken after a set of tweets was posted by followees. All tweets published between two consecutive starting times belongs to the same session.

The session partition method is an intuitive way to approximate the sessions. For each received tweet m, denote the starting time of the session next to m's current session as $t^*(m)$. Then we have

$$t^*(m) = \min\{a | a \geq t(m), \forall a \in A\}. \tag{4}$$

If tweet m belongs to the *k-th* session, $t^*(m)$ will be the beginning time of the $(k+1)$-*th* session, that is, $B(k+1)$.

Table II provides an example to illustrate the preceding proposed approach for session partition. The events in Table II are ordered by the publishing time of the tweet (the second column). There are two categories of events: passively receiving a tweet and actively publishing through posting a new tweet, retweeting, or replying to a received tweet. The third column is the time when the tweet was retweeted/replied. If this tweet was not retweeted/replied, it would be N/A. The fourth column is the start time of the next session $t^*(m)$, where $t^*(m)$ depends on later active actions of the user. As shown in the fifth column, tweets with the same value of $t^*(m)$ belong to the same session.

Table II. An Illustration of the User Session Partition

Event	Time Stamp	Retweet/Reply Time	$t^*(m) = B(k+1)$	k-th session
Receive m_1	2010-07-18 07:10:12	N/A	2010-07-18 07:34:29	1
Receive m_2	2010-07-18 07:29:38	2010-07-18 07:34:29	2010-07-18 07:34:29	1
Receive m_3	2010-07-18 07:31:26	N/A	2010-07-18 07:34:29	1
Retweet m_2	2010-07-18 07:34:29			
Publish a tweet	2010-07-18 07:41:02			
Receive m_4	2010-07-18 15:58:39	N/A	2010-07-18 16:37:45	2
Receive m_5	2010-07-18 16:08:02	N/A	2010-07-18 16:37:45	2
Receive m_6	2010-07-18 16:10:01	N/A	2010-07-18 16:37:45	2
Receive m_7	2010-07-18 16:11:21	2010-07-18 16:40:35	2010-07-18 16:37:45	2
Receive m_8	2010-07-18 16:23:04	2010-07-18 16:38:17	2010-07-18 16:37:45	2
Receive m_9	2010-07-18 16:28:43	2010-07-18 16:37:45	2010-07-18 16:37:45	2
Retweet m_9	2010-07-18 16:37:45			
Reply m_8	2010-07-18 16:38:17			
Retweet m_7	2010-07-18 16:40:35			
Receive m_{10}	2010-07-19 11:26:25	2010-07-19 11:29:32	2010-07-19 11:29:32	3
Receive m_{11}	2010-07-19 11:26:32	N/A	2010-07-19 11:29:32	3
Receive m_{12}	2010-07-19 11:27:00	N/A	2010-07-19 11:29:32	3
Retweet m_{10}	2010-07-19 11:29:32			

The active actions of the user are marked in gray.

There are three sessions being detected in this example and the user performs actions on three tweets within the second session.

We take a user session (u,k-th) as our basic data units. Each data unit includes the tweets received within the current user session, such as m_{10}, m_{11}, m_{12} in u's 3rd session in Table II. It's to be noted that in the evaluation part: (1) every tweet will be given a score; (2) the tweets within the same user session/data unit will be compared with each other; (3) for every user's session, we will rank tweets according to their scores. So the partition of the user's session is important for both the training and the testing of the reordering model.

If new tweets arrive before user u ends his current session, according to the described rule, our partition method will assign the newly received tweets into the user's next session, and tweets in these two sessions will be treated separately.

Our partition method in general can provide a good approximation of the user sessions when the records of the user's active actions are adequate. However, if the user takes no action during a session, this silent visit will be merged to another visit. This kind of problem can be alleviated by the relative relevance labeler introduced in Section 4.2.

4.2. Preference Training Data

In our reorder model, a tweet is considered as a positive sample if it is relatively more relevant than other tweets within the same user visit. Here the definition of the relative relevance of a tweet is based on the following assumption.

Assumption. Within the same user's session, tweets being acted (retweeted/replied) by user u are considered to be relatively more relevant than other tweets in the session.

Although this assumption may not always be true, it is generally accepted that tweets acted upon are more relevant than those not acted upon. This assumption can be justified as follows. First, the relevance of a responded tweet is considered to be higher than that of the nonresponded tweets within the same user session. No

Table III. The Distribution of Responded Tweets of a Typical User

	No Response	Retweet	Reply	Total Relevant
Ratio	99.91%	0.05%	0.04%	0.09%

relevances are compared for tweets in different visits of this user. Secondly, a responded tweet is considered to be more relevant than the tweets nearby within a time window in the timeline. We conservatively assume that there is no relative relevance between two tweets that are far from each other in the time being posted, regardless of the user's responses to them. For example, a responded tweet with the rank of 11 is only considered to be more relevant than the nonresponded tweets with rank from $11 - WS$ to $11 + WS$ if available, where WS is the window size of the comparison.

We denote the reorder model as function $h(.)$. Given a tweet m, we calculate its features vector $\vec{\phi}_m = (\phi_1(m), \phi_2(m), \ldots, \phi_{19}(m))$ as described in Section 3. The relevance score of a given tweet m equals $h(\vec{\phi}_m)$. We use set $P(k) = \{(m_i, m_j)|h(\vec{\phi}_{m_i}) >= h(\vec{\phi}_{m_j}), |i - j| \leq WS, m_i, m_j \in S(k)\}$ to represent the pairwise training data. The idea of using pairwise preference as training data is similar to that of the click-through method in Joachims [2002].

The reason that we do not directly use the responded tweets as positive samples is that the ratio of the number of user's retweets to the number of tweets the user u received is quite low. According to our analysis (as shown in Table III) only 0.09% of received tweets are responded by a user. That means if we directly use those responded tweets as our positive samples, the positive samples will be quite scarce. The second point is that there are a variety of possible reasons behind a nonresponded tweet: (a) this tweet is lacking in information; (b) this tweet is informative, but the user is simply not interested in it; (c) the user somehow does not see this tweet at all; (d) although the tweet is interesting to him/her, the user deliberately chooses not to retweet it.

4.3. Reordering Model

With the personalized features and training data discussed before, the remaining question is how to learn a model from the data. We use one of the state-of-the-art machine learning methods: Gradient Boosted ranking (GBrank)[Zheng et al. 2007] as the underlying algorithm for our learning. Please note that other pairwise-preference-based learning methods are also applicable. But it is out of the scope of this article to compare different pairwise preference learning methods. GBrank is a general framework to learn ranking functions with any regression algorithm as the base learner. By minimizing the pairwise loss on the observed data generated, we model the relevance of tweets with user's retweeting or replying activities with no additional editorial effort required.

We denote the learned model as function $h(.)$. Essentially $h(.)$ is a nonlinear function, and has the ability to capture the nonlinearity of features to tweet relevance.

For user's *k-th* session, the training inputs of the GBrank module are the set of feature vectors $\{\vec{\phi}_{m_i}|m_i \in S(k)\}$ and the pairwise preference $P(k)$ from Section 4.2.

After training, we can obtain the target model $h(.)$. The ideal of the GBrank function $h^*(.)$ should satisfy $h^*(m_i) >= h^*(m_j)$, if (m_i, m_j) in $P(k)$(i.e., m_i is relevant), which means more relevant tweets will achieve higher scores than other tweets in the same user's session.

More details about how to train the GBrank function $h(.)$ are presented in the Appendix.

5. EXPERIMENTS

We compare three methods for reordering the tweets for a user. The first method under comparison is Twitter's default way of simply ordering the tweets by their publishing

time (thereafter denoted as TIMELINE). The second method is the proposed reordering method (denoted as GBREORDER) where tweets are ordered by their potential relevance to the user as predicted by the machine learned model. All features described in Section 3, that is, features ϕ_1 to ϕ_{19}, are used in the GBREORDER method. To test the effectiveness of the personalized features, we also consider a third method (NONPERSONALIZED) for comparison which is the same as the second method except that it excludes all personalized features, that is, the features ϕ_2, ϕ_{13}, ϕ_{14}, ϕ_{15}, ϕ_{16}, ϕ_{17}, ϕ_{18}, ϕ_{19}.

5.1. Twitter Data for Training and Evaluation

There are two types of Twitter data: publishing data (e.g., what and when are users posting); and subscription data (e.g., users have followed which of Twitter users).

For the purpose of tweet reordering, we are only interested in the active Twitter users in this experiment. We empirically define Twitter users who retweet or reply at least 5 times each day on average as active users. As a result, there are 51, 598 active Twitter users in July 2010 according to our data. Due to the limitation of resource, we randomly choose 816 users from them for this study. We crawled all the followee lists of these selected active users with Twitter API. The average number of followees for the selected users are 834.

The publishing data of these selected active Twitter users and their followees include their posts and received tweets as well as the retweeting and replying activities. Tweets published before 2010-07-11 00:00 are used for training. Tweets published after 2010-07-21 00:00 are left out for evaluation. The remaining tweets are used as validation set for parameter tuning.

5.2. Evaluation Metrics

In this work, we use pairwise reordering accuracy (ACC), Mean Reciprocal Rank (MRR), and precisions at different positions ($P@1$, $P@3$, $P@5$), that is, R-precision(RP), as our evaluation metrics. Given a user session, let N_r and N_{ir} denote the number of relevant tweets and the number of irrelevant tweets in the session, respectively. The pairwise reordering accuracy for a user session is then expressed by

$$ACC = \frac{\sum_{i=1}^{N_r} Correct(i)}{N_{ir} N_r}, \tag{5}$$

where $Correct(i)$ is the number of correctly ordered pairs for the $i\text{-}th$ relevant tweet, equivalent to the number of irrelevant tweets with rank lower than that of the $i\text{-}th$ relevant tweet. The ACC is then averaged over all user sessions.

MRR is the mean (calculated over all user session) of the reciprocal rank (RR) for each user session with the RR is defined as

$$RR = \frac{1}{\min_{i=1}^{N_r} rank_i}, \tag{6}$$

where $rank_i$ is the rank for the $i\text{-}th$ relevant tweet.

RP is the mean (calculated over all user session) of the precision at the $N_r\text{-}th$ position for each user session. Since N_r is the total number of relevant tweets in the user session, it's different for each session.

5.3. Results of the Reordering

We use the pairwise preference data as input for training. A tweet being responded by the user is considered to be more relevant than tweets not being responded nearby within a window size in timeline order. Denote the window size as WS. We tune this

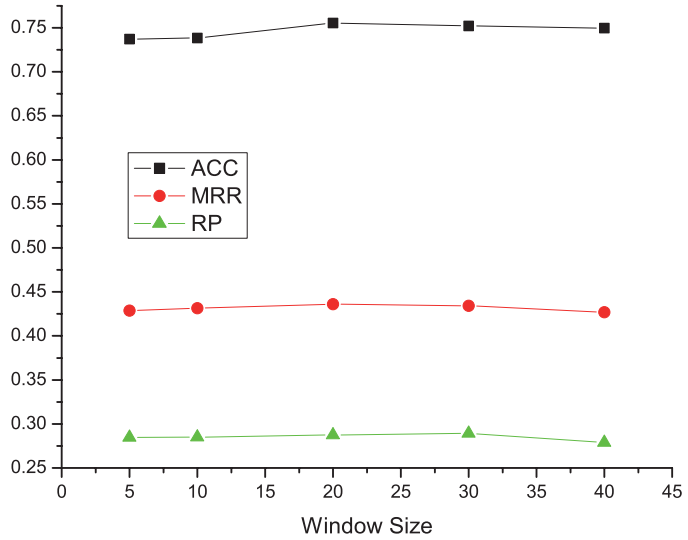


Fig. 1. Evaluation results under different window sizes.

parameter by varying WS within the set $\{5, 10, 20, 30, 40\}$. As shown in Figure 1, the reordering model achieves best performance on the validation set when WS is set as 20. In the rest of experiments, we set the window size WS to be 20.

Table IV and V provide a real case to illustrate the effectiveness of GBREORDER when compared with TIMELINE. After the reordering, more relevant tweets bubble up to top positions according to the relevance score (from rank 6, 16, 20 in the TIMELINE to rank 1, 3, 4), which enables users to find the interesting/important tweets more efficiently.

In Figure 2, we show the performance of GBREORDER, TIMELINE, and NONPERSONALIZED in terms of the three metrics. We can see that both GBREORDER and NONPERSONALIZED have outperform TIMELINE in terms of the three metrics. For example, the accuracy is improved by 34.5% when comparing the results of TIMELINE and GBREORDER. Moreover, compared with the NONPERSONALIZED results, GBREORDER achieves more performance gain. For example, there is around 3.8% improvement in terms of the prediction accuracy. This shows the effectiveness of the personalized features we proposed for tweets reordering.

5.4. Features Importance

As a property of the boosted trees (boosted trees are used in the GBrank), a greedy feature selection already happens in our model learning. As a byproduct, the GBrank provides a feature importance list, which is computed by keeping track of the reduction in the loss function at each feature variable split and then computing the total reduction of loss function along each feature variable [Friedman 2001].

The feature importance to our GBREORDER algorithm is listed in Table VI. Features not listed in Table VI are eliminated by the GBrank algorithm. Figure 3 shows the curves of accuracy, MRR, and RP scores of our GBREORDER model, when it's trained with different number of top features from the importance list in Table VI. The feature selection adopted here is straightforward but effective for the boosted trees algorithm such as GBrank.

The top two features are time freshness and rank freshness. Given users in general are interested in getting real-time update/information from their contacts, it seems

Table IV. A Case Study Part I: Before GBREORDER

Rank	Time	Action	Tweet
1	21:54:48		@TheLornaAlice Lol im pretty sure they have great stores though (: and nooo i dont. . . .
2	21:54:41		@AmoreSelly *stops thinking, getting weak, shivering*
3	21:54:39		Dear @NickJonas, I'm glad you've landed in Dallas safely. All us fans in England will . . .
4	21:54:36		@Ashl3yFaith nothing just here bored :)
5	21:54:34		@NickLuvIsMyDrug me too! and if you report it for spam it doesnt really do anythinf : /
6	21:54:30	Retweet	NICK IS IN TEXAS, NICK IS IN TEXAS, NIIIICK IS IN TEEEXAAAASSSS!!! :D -5 . . .
7	21:54:27		Where is facebook's 'Don't Care' button?
8	21:54:23		IF YOU'RE IN LOVE WITH ZACH PORTER FOLLOW @katie_zach_asw :) i know a . . .
9	21:54:21		4 more followers please.
10	21:54:19		Btw Guys, Who knows Miss Hooley from Balamory? well @Chelsiemillsx met her. . . .
11	21:54:16		Photo: thisiscritical: http://tumblr.com/xbxe9swn3
12	21:54:15		TV REMINDER: Jonas Brothers on the Paul McCartney Tribute this Wed (July 28) . . .
13	21:54:12		@JBLivingtheDrm Allll donnee (:
14	21:54:08		'i'm having trouble opening this jar.'
15	21:54:06		@Lballer3 mine are for what girls think & like & @boyfacts are for what boys thinks & like
16	21:53:59	Reply	@UBroughtAJonas yeah i do too its so annpying who are they anyway?
17	21:53:57		@_Crazy_Dona_ woah. . . obsessed! haha X). . . uhmm yep He's cause almost every song . . .
18	21:53:55		Sorry i didn't tweet earlier guys, Hadn't any time!
19	21:53:54		@JasmineVForLife oh haha lol cool! yeah i know its gonna take me awhile but its gonna . . .
20	21:53:48	Reply	Keep voting Kiss the amazing Kevin Jonas or Kiss Bieber? X)
21	21:53:47		@lilchrishardman Now what do i get for being her 100th follower? ;D
22	21:53:46		RT @CamilleRenae6: I see @catchjbfever online.I log on.Hope he notices me.I tweet. . . .

Three relevant tweets with rank 6, 16, 20 in the default TIMELINE.

reasonable to have freshness as the key factors for reordering. On the other hand, in the training sessions, the tweets are ordered by publishing time. It is generally agreed that a user's retweet activity is highly influenced by presentation: if a tweet is ranked high, then its probability of getting noticed by the user is more likely to be high. Hence the importance of freshness features could also be partially due to an artifact introduced together with the presentation bias in the training data. According to our evaluation, the TIMELINE method, which simply orders the tweets by their freshness, has a much more inferior performance than the GBREORDER method, indicating some other factors also played important roles in tweet reordering.

The next important feature is retweet count, which indicates that a widely spread tweet will attract more attentions. This feature is also a key feature in Twitter's default search engine. The retweet count by followee is a similar feature, but to our surprise it doesn't help in our model. Large retweet count usually corresponds to high quality. But if most of the user u's followees have already retweeted it, the tweet might not be retweeted by u. Therefore according to our evaluation this tweet might be considered not as important as other tweets and the feature ϕ_{13} retweet count by followee is not as effective as we expected.

Table V. A Case Study Part II: After GBREORDER

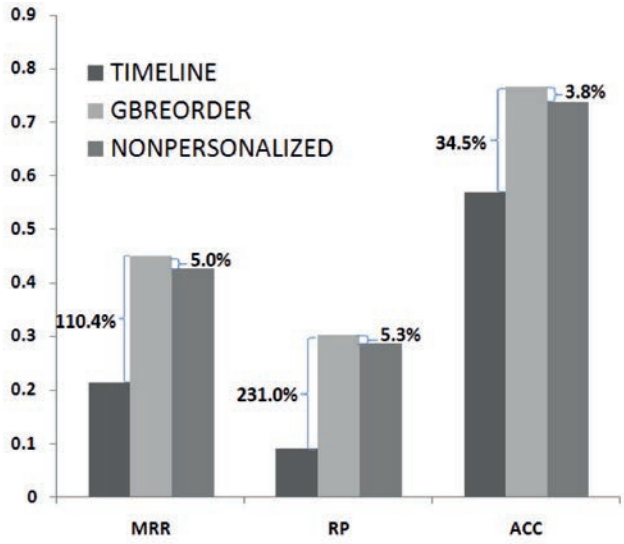
Rank	Score	Action	Tweet
1	0.5460	Retweet	NICK IS IN TEXAS, NICK IS IN TEXAS, NIIIICK IS IN TEEEXAAAASSSS!!! :D -5 . . .
2	0.1916		@NickLuvIsMyDrug me too! and if you report it for spam it doesnt really do anythinf : /
3	0.1801	Reply	@UBroughtAJonas yeah i do too its so annpying who are they anyway?
4	0.1307	Reply	Keep voting Kiss the amazing Kevin Jonas or Kiss Bieber? X)
5	0.0998		@Lballer3 mine are for what girls think & like & @boyfacts are for what boys thinks & like
6	0.0884		Photo: thisiscritical: http://tumblr.com/xbxe9swn3
7	0.0846		@JBLivingtheDrm Allll donnee (:
8	0.0818		Where is facebook's 'Don't Care' button?
9	0.0752		@TheLornaAlice Lol im pretty sure they have great stores though (: and nooo i dont. . . .
10	0.0453		@lilchrishardman Now what do i get for being her 100th follower? ;D
11	0.0443		'i'm having trouble opening this jar.'
12	0.0435		@Ashl3yFaith nothing just here bored :)
13	0.0411		@JasmineVForLife oh haha lol cool! yeah i know its gonna take me awhile but its gonna . . .
14	0.0367		Sorry i didn't tweet earlier guys, Hadn't any time!
15	0.0333		Btw Guys, Who knows Miss Hooley from Balamory? well @Chelsiemillsx met her. . . .
16	0.0301		Dear @NickJonas, I'm glad you've landed in Dallas safely. All us fans in England will . . .
17	0.0231		@AmoreSelly *stops thinking, getting weak, shivering*
18	0.0219		RT @CamilleRenae6: I see @catchjbfever online.I log on.Hope he notices me.I tweet. . . .
19	0.0215		TV REMINDER: Jonas Brothers on the Paul McCartney Tribute this Wed (July 28) . . .
20	0.0075		@_Crazy_Dona_ woah. . . obsessed! haha X). . . uhmm yep He's cause almost every song. . .
21	0.0071		4 more followers please.
22	-0.019		IF YOU'RE IN LOVE WITH ZACH PORTER FOLLOW @katie_zach_asw :) i know a . . .

Three relevant tweets are reordered from original rank 6, 16, 20 in the default TIMELINE to rank 1, 3, 4 by our reordering algorithm GBREORDER.

The feature follower count, representing the popularity of the author of the tweet, is also helpful. The importance of this feature is consistent with the observations in Nagmoti et al. [2010]. But the list count is not that useful, since Twitter's list is just an optional function and most users' list count is zero.

As we expected, the count and ratio of retweet and reply by user are important features for reordering. They provide useful guidelines for predicting whether the tweet will be retweeted/replied by the user or not.

The feature ϕ_{15}, the interest match between the author and the reader, also plays a role in our model (if we drop feature ϕ_{15}, the accuracy, MRR, and RP scores will decrease by -0.58%, 0.58%, and 2.1%, respectively). But the feature ϕ_{14}, the match of tweet topics and user interests, is not as effective as the interest match (if we drop feature ϕ_{14}, the accuracy, MRR, and RP scores will decrease by -0.20%, 0.33%, and 0.53%, respectively), because only about 10% of tweets' topics can be inferred, due to the fact that tweets have very limited length and the words used in Twitter are very informal. Many tweets are just noises with no meaningful topic [Kelly 2009]. The quality of the tweet's topics limits the use of this feature. The interest match between



	MRR	RP	ACC
TIMELINE	0. 2136	0. 0912	0. 5696
GBREORDER	0. 4494	0. 3019	0. 7663
NONPERSONALIZED	0. 4279	0. 2868	0. 7379

Fig. 2. Performance of three tweets ordering methods.

Table VI. Feature Importance for Our Reorder Model

Rank	Feature	Importance
1	ϕ_1 :Time freshness	100
2	ϕ_2 :Rank freshness	85.34
3	ϕ_{12}:Retweet count	71
4	ϕ_3 :Follower count	51
5	ϕ_{19}:Reply ratio	44.25
6	ϕ_{15}:Interest match of author and user	17.84
7	ϕ_6 :Status count	16.94
8	ϕ_4 :Followee count	13.14
9	ϕ_9 :Tweet length	9.7
10	ϕ_{17}:Reply count by user	8.27
11	ϕ_7 :Days	7.38
12	ϕ_{18}:Retweet ratio	6.12
13	ϕ_{11}:Hashtag count	3.95
14	ϕ_{14}:Match of tweet and user interests	3.2

the author and the reader still works, since we use all the user's tweets to learn the user's interests. The interests of users are more reliable than the inferred topics of tweets. This similarity between different users is in fact quite accurate, especially when the reader and the author share lots of common tweets.

5.5. Limitation Analysis

In both the relative relevance definition and the evaluation part, our work is built upon an assumption that retweeted/replied tweets are relatively more relevant than other tweets. This somehow brings some limitations to our method.

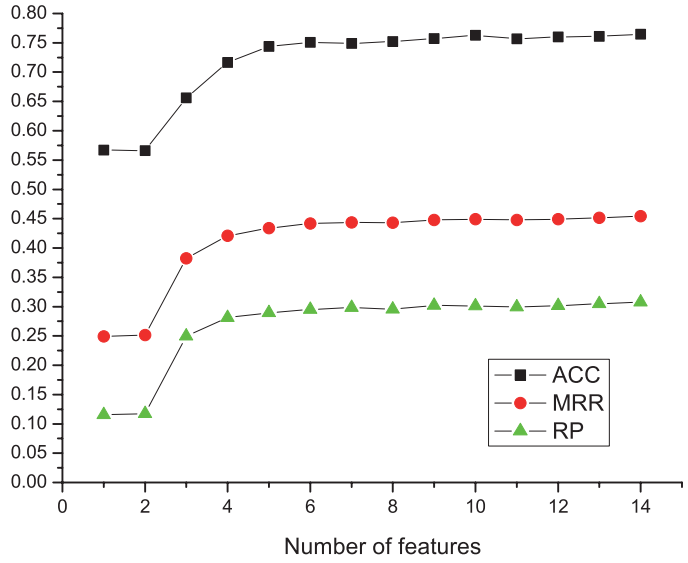


Fig. 3. Feature selection: trained with different number of top features.

—We assume that retweeting by the user u to a tweet suggests that to some extent, the user u likes this tweet and is willing to share it with u's followers. However, response does not necessarily equal to relevance. There might be other criteria besides the retweet/reply to judge whether a tweet is relevant to its reader.
—Some nonresponded but relevant tweets (i.e., some high-quality tweets which are not responded by user u) will hurt the accuracy for both training and evaluation because we treat them as irrelevant tweets by definition.
—We do not have the browsing information of the users. So we are not sure whether a tweet is actually read by the user u or not. This will affect the quality of the user session partition and the relative relevance definition to some extent.

In all, our method works for those Twitter users with active retweeting or replying activities. Applying the method to nonactive users may obtain unsatisfiable reordering relevance. Therefore, the more the user takes action in Twitter, the better the tweets' relevance would be by our method.

6. CONCLUSIONS

Few users are willing to scroll down the tweet list far enough to get all the interesting or important tweets. It is hence very important to reorder a user's unread tweets. According to our experiments, our GBREORDER method is a very effective personalized method, in comparison with the default TIMELINE method and the NONPERSONALIZED method. The direct application of our method could be a Twitter client that offers one a choice to reorder the tweets according to one's interests. With this Twitter client, we can get the users' real browsing behaviors and refine our model.

For now, we only evaluated our method based on user observable behaviors. We attempted to get real users' feedback by selecting 54 active users and asking their opinion on reordering their tweets. However, we did not get enough replies for a statistically meaningful evaluation. A larger scale of manual evaluation may be performed for future work.

APPENDIX

In this appendix, we describe the details of the GBrank algorithm with GBDT (Gradient Boosting Decision Tree) as the base regression learner for our tweets reordering task in Algorithm 1.

ALGORITHM 1: Gradient boosted tweets reordering

Require: Pairwise preference $P(k)$ for $u \in \mathcal{U}$ and $k \in \{1, \cdots, K\}$; The number of trees M to use in GBrank; Argument τ for the discriminative margin.
Ensure: The optimal reordering function h^*.
Start with a initial guess h_0;
for $t = 1, .., M$ **do**
(a) Use h_{t-1} as an approximation of h^* and we divide the P into two sets:

$$P_t^+ = \{(m_i, m_j) \in P(k) | h_{t-1}(m_i) \geq h_{t-1}(m_j) + \tau, \\ u \in \mathcal{U}, k \in \{1, \cdots, K\}\}$$

$$P_t^- = \{(m_i, m_j) \in P(k) | h_{t-1}(m_i) < h_{t-1}(m_j) + \tau, \\ u \in \mathcal{U}, k \in \{1, \cdots, K\}\}$$

(b) Fit a regression function g_t by GBDT based on the incorrectly ordered examples:

$$\{(m_i, h_{t-1}(m_j) + \tau), (m_j, h_{t-1}(m_i) - \tau) | (m_i, m_j) \in P^-(t)\} \tag{7}$$

(c) Update the approximate function:

$$h_t(x) = \frac{t h_{t-1}(x) + \eta g_t(x)}{t+1}, \tag{8}$$

where η is the learning rate.
end for
Output h_M as the optimal reordering function h^*.

The hyper-parameters in Algorithm 1 such as the number of tree M and the learning rate η are determined by cross-validation on the training dataset.

REFERENCES

Asur, S. and Huberman, B. 2010. Predicting the future with social media. Arxiv preprint arXiv:1003.5699.

Bollen, J., Mao, H., and Zeng, X. 2010. Twitter mood predicts the stock market. Arxiv preprint arXiv:1010.3003.

Castillo, C., Mendoza, M., and Poblete, B. 2011. Information credibility on twitter. In *Proceedings of the 20th International Conference on World Wide Web*. ACM, 675–684.

Chen, J., Nairn, R., Nelson, L., Bernstein, M., and Chi, E. 2010. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems*. ACM, 1185–1194.

Duan, Y., Jiang, L., Qin, T., Zhou, M., and Shum, H.-Y. 2010. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*. 295–303.

Friedman, J. 2001. Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 1189–1232.

Hofmann, T. 1999. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence, (UAI)*.

Joachims, T. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 142.

Kelly, R. 2009. *Twitter Study Reveals Interesting Results About Usage.* Pear Analytics, San Antonio, TX.

Kwak, H., Lee, C., Park, H., and Moon, S. 2010. What is Twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*. ACM, 591–600.

Nagmoti, R., Teredesai, A., and De Cock, M. 2010. Ranking approaches for microblog search. In *Proceedings of IEEE/WIC/ACM International Joint Conference on Web Intelligence*.

PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T. 1998. The pagerank citation ranking: Bringing order to the web. Tech. rep., Stanford Digital Library Technologies Project.

RITTERMAN, J., OSBORNE, M., AND KLEIN, E. 2009. Using prediction markets and twitter to predict a swine flu pandemic. In *Proceedings of the 1st International Workshop on Mining Social Media*.

SUH, B., HONG, L., PIROLLI, P., AND CHI, E. 2010. Want to be retweeted? Large scale analytics on factors impacting retweet in twitter network. In *Proceedings of the IEEE 2nd International Conference on Social Computing (SocialCom)*. 177–184.

TUMASJAN, A., SPRENGER, T., SANDNER, P., AND WELPE, I. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*.

WENG, J., LIM, E., JIANG, J., AND HE, Q. 2010. Twitterrank: Finding topic-sensitive influential twitterers. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*. ACM, 261–270.

YANG, Z., GUO, J., CAI, K., TANG, J., LI, J., ZHANG, L., AND SU, Z. 2010. Understanding retweeting behaviors in social networks. In *Proceedings of the 19th Conference on Information and Knowledge Management*.

ZHENG, Z., CHEN, K., SUN, G., AND ZHA, H. 2007. A regression framework for learning ranking functions using relative relevance judgments. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 287–294.

Received January 2012; revised September 2012; accepted October 2012