Modeling User Posting Behavior on Social Media

Zhiheng Xu, Yang Zhang, Yao Wu and Qing Yang
Institute of Automation
Chinese Academy of Sciences
Beijing, 100190, China
{xuzhiheng,yagzag,wuyao,qyang}@nlpr.ia.ac.cn

ABSTRACT

User generated content is the basic element of social media websites. Relatively few studies have systematically analyzed the motivation to create and share content, especially from the perspective of a common user. In this paper, we perform a comprehensive analysis of user posting behavior on a popular social media website, Twitter. Specifically, we assume that user behavior is mainly influenced by three factors: breaking news, posts from social friends and user's intrinsic interest, and propose a mixture latent topic model to combine all these factors. We evaluated our model on a large-scale Twitter dataset from three different perspectives: the perplexity of held-out content, the performance of predicting retweets and the quality of generated latent topics. The results were encouraging, our model clearly outperformed its competitors.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.3.3 [**Information Search and Retrieval**]: Information filtering—*performance measures*

General Terms

Algorithms, Experimentation

Keywords

Twitter, user modeling, user behavior, topic model

1. INTRODUCTION

With the rising popularity of social media, better understanding of user posting behavior has become crucial for many personalization and information filtering applications, as well as for better site design and advertising policies. Towards this goal, existing works have examined the workloads of various social media websites [7, 13, 21], aimed at providing a global picture of user activity patterns on these websites. There are also studies focused more on individual user behavior, by analyzing the content users have created [30, 37, 41] or inferring from their social friends [9, 14, 40], to help users find interesting information or people.

While previous works on individual user behavior [9, 14, 30, 37, 40, 41] have simply assumed that users tend to publish content they are interested in or make friends with similar interest, however, reality is much more complicated due to different usage patterns and user intentions. For example, it is reported that users are easily attracted by breaking news [1, 27], and are likely to create conversation with their intimate friends [15, 20]. On the other side, friendship on social media does not necessarily indicate similar interest, since it may arise from different sources such as influence, homophily, environment and reciprocity [4]. All these problems require a more comprehensive model of user behavior on social media, which is the task we deal with in this paper.

Inspired by those early works [1, 15, 20, 27], we believe that when a user publishes a post, he is probably influenced by three factors: breaking news happens at that moment, posts published by his friends recently and his intrinsic interest. In light of this, we equally divide our experimental time period into time intervals, and within each time interval, we compute the distribution of breaking news and friends' timeline for each user, which are assumed as two external factors that might affect his posting behavior in the same time interval. Subsequently, by modeling user interest as a distribution over latent topics, we use a mixture latent topic model to represent user posting behavior, and present the inference of our model based on collapsed Gibbs sampling.

Our experiment is based on Twitter, a popular social media website. Since Twitter has attracted thousands of individuals and organizations with business intents (e.g., news channels, online brands and social spammers), we first built a dataset of 11,358 common users, and then collected all tweets published by those users and samples of their social friends during a 70-days experimental time period. We tested our model on this dataset and showed its superiority over the competitors. Although our work has been done in the context of Twitter, we expect the same results would hold for many other similar applications, such as Facebook updates and Google Buzz.

The main contributions of our work include:

1. We build a large and reasonable dataset for analyzing user posting behavior on Twitter.

2. A simple but effective method is used to recognize breaking news from Twitter streams in a certain time period.

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.
Copyright 2012 ACM 978-1-4503-1472-5/12/08 ...$10.00.

3. We analyze the influence of different social relationships on user posting behavior, and quantitatively measure the influence between users.

4. We propose a mixture topic model to analyze user posting behavior, and demonstrate the superiority of our model from three different tasks.

The remainder of this paper is organized as follows: section 2 provides a brief review of related work, section 3 describes the way to build our dataset, section 4 formally presents our mixture model, followed by the results of our experiments in section 5. Finally we conclude in section 6.

2. RELATED WORK

2.1 Social Media

Social media has become indispensable to users recently. A rich set of studies has been conducted on various forms of social media, such as blogs, photo and video sharing communities, question/answering portals and social bookmarking sites, focused on different properties and applications of them. For example, Gruhl et al. studied the dynamics of information propagation in blogspace [12], Leskovec et al. analyzed the network structure and evolution on different information networks [24, 25], Agichtein et al. introduced a classification framework to extract high quality content on question/answering portals [3] and Benevenuto et al. tried to discover spammers on a video sharing community [6].

Among the various successful social media websites, Twitter, a microblogging service, has attracted considerable attention from research area recently. With a limit of 140 characters for each message, Twitter enables an even faster mode of communication and information propagation. Early works [15, 20] examined the usage patterns and network properties of Twitter, and revealed that Twitter was mainly used in two different ways: as an information platform or as a social network. Subsequently, to better leverage its great wealth of both textual and social information, researchers have used Twitter to discover breaking news [28, 36], detect natural disasters [35, 39], improve realtime web search [11], characterize media events [10] and identify influential users [41] or interesting content [9].

2.2 User Modeling

The massive amount of data generated by social media users has provided researchers with insights into user behavior. For instance, by analyzing workloads from three information networks, Guo et al. showed that users' posting behavior exhibited strong daily and weekly patterns. They also pointed out that different types of content would have different characteristics [13]. Benevenuto et al. used clickstream data from a social network aggregator to compare user behavior across different online social networks, and they further investigated social interactions on those networks [7]. These macroscopic analysis of user behavior provided interesting observations about general usage patterns on social media websites, but they might lack interpretations at an individual level. To reach a better understanding of individual user behavior, work [32] investigated the causality between individual behavior and social influence by observing the information diffusion among users, work [27] predicted a user's news interest from the user activities and the news trends, work [37] proposed a user interest model based on tags generated by users and their social friends, and the SVM classification framework was leveraged in [5, 6, 23] to detect spammers and content promoters on social media.

Within the research area of Twitter, few have been done to systematically analyze individual user behavior. Previous efforts about user modeling on Twitter simply built a "bag-of-words" profile for each user based on his tweets, and extracted key words [9], entities [2], categories [30] or latent topics [17, 41] for that user. Although existing works can to some extent help recognize important information about users, however, they failed to capture the real motivation of users to publish content, as user behavior can easily be affected by some external factors other than user interest. To reach a comprehensive model of user behavior, we propose a mixture model which incorporates three important factors that might trigger user posting behavior, namely breaking news, friends' timeline and user interest. Our model is under the framework of latent topic models, since the entity-based and category-based user modeling frameworks would require external knowledge bases such as Wikipedia and AlchemyAPI [1], which are time and resources consuming. Inspired by previous works on multiple text streams modeling [8, 18, 33], we present the inference of our model based on collapsed Gibbs sampling, and further test it on a large-scale Twitter dataset from three different tasks.

3. DATASET PREPARATION

We started by using Twitter's streaming API [2] to collect a random sample of the public tweets from March 10, 2011 to May 19, 2011 (the streaming API would return about 1% of all tweets each day, and is widely used for analyzing news on Twitter). After removing non-English tweets, the stream dataset contained 56,415,430 tweets published by 9,292,345 users, with an average of 805,935 tweets each day. This stream dataset was used to extract breaking news for each time interval on Twitter. Since Twitter imposes a rate limit on crawling posts of a specific user, it is difficult for us to analyze large amount of users. Thus we would like to build a relatively small dataset of common active users.

Specifically, we assumed that a user was common and active if he had (100-3000) friends/followers, (10-200) tweets per week and has been listed (1-50) times. Most common and active Twitter users were believed to fall into this category. We randomly picked 11,358 ordinary users as our experimental users, and crawled all their tweets during the 70-days experimental time period, yielding a dataset of 7,843,190 tweets. For each user, we crawled his entire social graph, including his followees, listers (people in the user-generated lists) and listfollowees (people in the user-followed lists). As it was difficult for us to collect tweets from all those friends, we created a sample of friends for each user, including all friends that have been retweeted or mentioned more than 4 times by him (on average 30 friends were chosen for each user), and 5 randomly picked followees, listers and listfollowees respectively. Finally the entire dataset of social friends contained 179,456 unique users, and we crawled all their posts during the experimental time period, yielding a dataset of 86,815,267 tweets.

Admittedly, our samples of social friends only contain small proportion of users' friends. However, as presented

[1] http://www.alchemyapi.com/

[2] https://dev.twitter.com/docs/streaming-api/

later in this paper, users' top retweeted/mentioned friends are much more influential than other social friends. Thus we believe that our dataset (which includes users' top retweeted/mentioned friends) is still to some extent reasonable for analyzing the influence of social friends.

4. MIXTURE MODEL OF USER BEHAVIOR

Imagine the situation when a common user publishes a post about iphone, the reason behind this behavior might be: (1) he is a fan of smartphones and has a long time focus on iphone (2) he is reminded by some big events about iphone, such as the release of Iphone 4S (3) he is attracted by a discussion about iphone raised by his close social friends. In light of this, user posting behavior can be represented as a mixture model of three different factors: breaking news, posts from social friends and the user's intrinsic interest. Specifically, given a user a in time interval T, the likelihood for him to generate a word w is regarded as a sample of the following mixture model (based on the bag-of-words assumption).

$$\begin{aligned} p_{Ta}(w) = \lambda_B p(w|\theta_B) + (1-\lambda_B)(\lambda_{a2} p_T(w|\theta_N) \\ +\lambda_{a3} p_T(w|\theta_{aF}) + \lambda_{a0} p(w|\theta_{aI})) \end{aligned} \quad (1)$$

In the formula above, θ_B is the background smoothing model using word frequency from the entire dataset, and λ_B is the mixing weight of the background model θ_B. $p_T(w|\theta_N)$ denotes the distribution of breaking news in time interval T, and $p_T(w|\theta_{aF})$ is the distribution of friends' timeline for user a in time interval T. All breaking news and friends' posts in the same time interval are assumed to have influence on user posting behavior, since it will be computationally expensive to consider what has happened before each user behavior. $p(w|\theta_{aI})$ means the distribution of a's interest, and is represented by a distribution over latent topics in this paper. λ_{a2}, λ_{a3} and λ_{a0} are the mixing weights of breaking news, friends' timeline and user interest for a respectively.

Notice that, for each user, the model uses different mixing weights, considering the difference between users in usage patterns. For instance, some users regard Twitter as an instant messaging tool and use it to communicate with their friends (where λ_{a3} should be big). On the other hand, some people consider Twitter to be an information platform and use it to release or seek information they are interested in (where λ_{a0} should be big). As presented later in this section, all mixing weights can be automatically learned during the training process.

In the following of this section, we first separately analyze the influence of breaking news and friends' timeline on user posting behavior, and compute their corresponding distributions. Then we view user interest as a distribution over latent topics, and use a latent topic model framework to represent our mixture model of user posting behavior.

4.1 Influence of Breaking News

To identify emerging news on Twitter, we borrow the idea from TwitterMonitor [28], where news is represented by a group of bursty keywords that suddenly appear in tweets at an unusually high rate. For each time interval T, a set of bursty keywords is extracted using equation (2):

$$score(w) = \frac{NT(w)}{NT}\frac{N}{N(w)}\sqrt{NT(w)} \quad (2)$$

where $NT(w)$ represents the number of tweets containing word w in time interval T, NT is the number of tweets in T, $N(w)$ is the number of tweets containing word w in the entire dataset, and N is the number of tweets in the entire dataset. $\sqrt{NT(w)}$ is used to promote the scores of high frequency words and punish low frequency words. We set a threshold S for $score(w)$ (300 is chosen when T is 24 hours), and discard words below the threshold. It is worth to mention that we also try to set different threshold S, and find that there are no obvious change for the experimental results when S is larger than 50.

Table 1: Bursty Words in 3 Time Intervals

Time interval	Bursty Words
2nd	tsunami japan earthquake 90999 quake redcross hawaii tsunamis prayers tokyo
50th	kate royal wedding william middleton abbey prince duchess mcqueen wills
53th	osama laden bin obama dead abbottabad death obl islamabad 1945

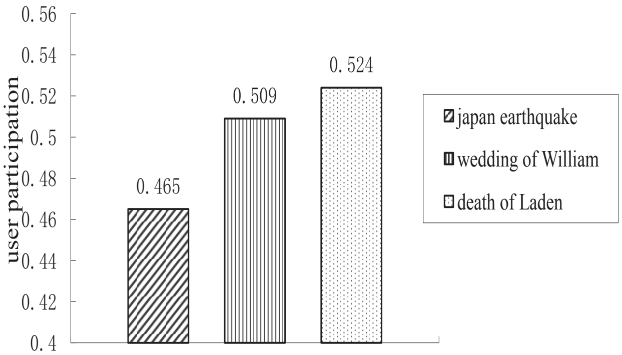

Figure 1: User Participation in 3 Events.

Table 1 gives the top 10 bursty words in the 2nd, 50th and 53th time intervals when T is set to be 24 hours, well represent 3 real world events: japan earthquake, wedding of prince William and the death of Osama bin Laden. Figure 1 shows user participation in the 3 events. Among the 11358 experimental users, 46.5% of users published tweets about japan earthquake in the 2nd time interval, 50.9% of users talked about the wedding of prince William in the 50th time interval and the death of Osama bin Laden attracted 52.4% of users in the 53th time interval, which demonstrate that breaking news has great impact on user posting behavior.

For each time interval T, we model the distribution of breaking news according to equation (3), where w' stands for any word that meets $score(w') \geq S$.

$$p_T(w|\theta_N) = \begin{cases} \dfrac{score(w)}{\sum_{w'} score(w')} & score(w) \geq S \\ 0 & score(w) < S \end{cases} \quad (3)$$

4.2 Influence of Social Friends

Twitter introduces a directed social relationship named "follow", which enables users to follow others to receive their tweets. To further help users organize their followees and filter incoming tweets, Twitter has launched another feature

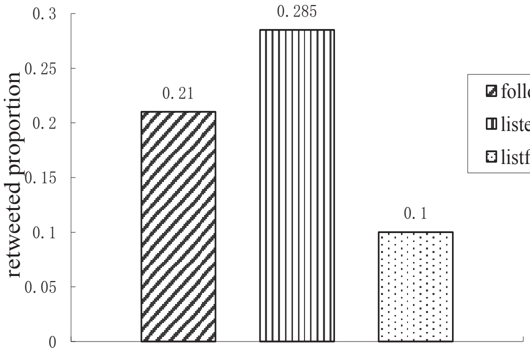

Figure 2: Influence of Friends on Retweet Behavior.

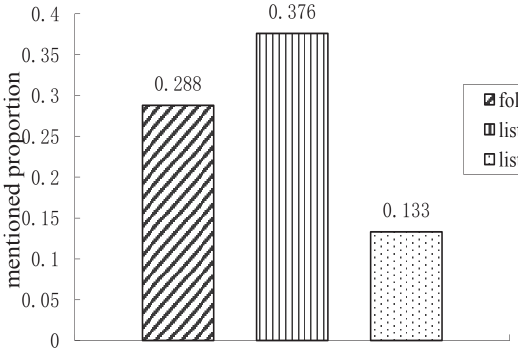

Figure 3: Influence of Friends on Mention Behavior.

named "list" since November 2, 2009, which can group sets of users into categories. Users can create their own lists, add and delete list members, or just follow other users' lists. Besides these two explicit relationships, there are two implicit relationships indicated by tweets, namely "retweet" and "mention". The retweet operates as a citation of another user's tweet, with the form "RT @username", while mention acts as a response to another user's tweet, with the form "@username". Both retweet and mention are strong signals of social influence [22].

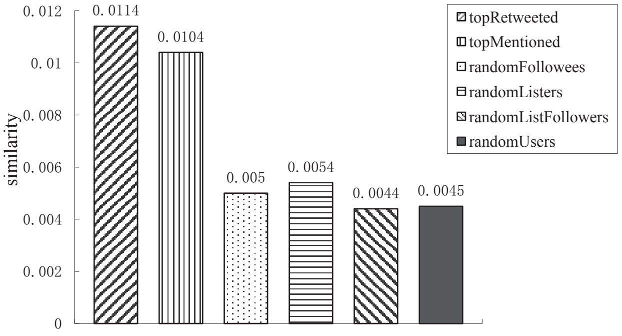

Figure 4: Similarity between Friends.

In our dataset, on average, each user has 644 followees. 39.8% of users have created at least 1 list, with an average of 169 people in each list, 37.2% of users have followed at least 1 list, with an average of 577 people in each list. 44.7% of users do not use list, which means list has not been used as widely as follow yet. Figure 2 analyzes the influence of followees, listers and listfollowees on user retweet behavior. During the 70-days experimental time period, on average, 21% of users' followees have been retweeted by them, 28.5% of listers have been retweeted and only 10% of users' listfollowees have been retweeted by them. The influence on user mention behavior is similar, as reported in figure 3, 28.8% of followees, 37.6% of listers and 13.3% of listfollowees have been mentioned respectively. The results show that listers have a little greater impact on users than followees, but listfollowees are far less important. However, most of users' social friends have not been retweeted or mentioned, which means that the explicit relationships on Twitter do not necessarily indicate strong influence [19]. As influence mainly exsits in the form of retweet and mention on Twitter, we assume that for each user, the more times a friend is retweeted or mentioned by him, the more influence that friend has on the user. To approximately verify this assumption, we build a "bag-of-words" profile for each user based on his tweets, and use TF-IDF algorithm to determine the word weight. Only the top 200 words are selected. For each experimental user, we compute the cosine similarities with his top retweeted friends, top mentioned friends, the random sample of his friends in section 3 (i.e., 5 listers, 5 followees and 5 listfollowees) and 5 random users that are not directly connected with him. As demonstrated in Figure 4, on average, the similarities with top retweeted and top mentioned friends are clearly higher than other friends, which proves that our assumption is reasonable. On the other side, the similarities with random listers, followees and listfollowees are almost the same as random users, which is consistent with our previous conclusion that explicit relationships on Twitter are not strong symbols of influence. Based on the assumption, we use equation (4) to measure the influence of friend j on user i:

$$Influence(j,i) = \frac{X_{j,i}}{max_{j'} X_{j',i}} \tag{4}$$

Where $X_{j,i}$ is as:

$$X_{j,i} = \frac{NR(j,i) + NM(j,i) + 1}{log(N(j) + 2)} \tag{5}$$

Here $NR(j,i)$ is the number of times friend j is retweeted by user i, $NM(j,i)$ is the number of times friend j is mentioned by user i and $N(j)$ is the total number of tweets posted by user j. Due to the similar performance in figure 4, we view retweet and mention the same in equation (5).

Our measure of social influence is computationally efficient, and generally captures the strength of communications between friends, which is shown to accurately reflect the strength of relationship between friends [26, 40]. Admittedly, there are many other works on measuring influence in social networks. Since we mainly focus on modeling user posting behavior rather than computing social influence between friends, we leave it as future work to compare different measures of social influence.

For each user i in time interval T, we compute the distribution of his friends' timeline as:

$$p_T(w|\theta_{iF}) = \frac{\sum_j influence(j,i) * N_{jT}(w)}{\sum_j \sum_{w'} influence(j,i) * N_{jT}(w')} \tag{6}$$

In the equation above, $N_{jT}(w)$ means the number of times word w is tweeted by friend j during the time interval T.

If word w has never been tweeted by any friends during T, equation (6) is set to be 0.

4.3 Mixture Latent Topic Model Framework

We use a latent topic model framework to represent our mixture model of user posting behavior, where user interest is represented as a random mixture over latent topics, and can be automatically inferred during the training process.

Figure 5 shows the Bayesian graphical framework of the proposed model. The model can be viewed as an extension of author-topic model [34], a widely used variation of Latent Dirichlet Allocation (LDA) [29] to integrate authorship information of documents into topic modeling. The author-topic model assumes that each author in the document collection is represented by a distribution over topics, and each word is associated with two latent variables: an author and a topic. To generate each document from a document collection, it first chooses an author from a document's author list, samples a topic from topic distribution associated with the selected author, and then picks a word from the topic specific word distribution. As each tweet has only 1 author, the author-topic model here acts as to collect a document for each user based on all his tweets, and uses LDA to extract the topic distribution of this document.

The proposed model has a similar general structure to the author-topic model, but with additional machinery to handle the distribution of breaking news, friends' timeline and background words respectively. In particular, a latent random variable x is associated with each word, acts as a switch to determine whether the word is generated from the distribution of background model, breaking news, posts from social friends or user's intrinsic interest. x is sampled from a user-specific multinomial distribution λ_a, which in turn has a symmetric Dirichlet prior, η. A indicates the set of authors, T_1 is the set of latent topics, N_d means the length of tweet d and D is the set of tweets. The generative process of this model is as follows:

1. For each topic k, draw φ_k from Dir(β)
2. For each author a, draw θ_a from Dir(α)
3. For each author a, draw λ_a from Dir(η)
4. For the ith word w_i posted by a during T
   (a) Sample x_i from Multinomial(λ_a)
   (b) If $x_i = 0$
      A. Sample topic z_i from Multinomial(θ_a)
      B. Sample w_i from Multinomial(φ_{z_i})
   (c) Else

$$\text{Sample } w_i \text{ from } \begin{cases} p(w|\theta_B) & x_i = 1 \\ p_T(w|\theta_N) & x_i = 2 \\ p_T(w|\theta_{aF}) & x_i = 3 \end{cases}$$

4.4 Model Inference

Our inference of the latent variable x is inspired by previous works on multiple text streams modeling [8, 18, 33], where a word is "split" into different streams, and a latent variable is sampled to indicate which stream the word belongs to. For each word in a document, the assignment of the latent variable is decided by two factors: the distribution of streams in the document, and the importance of the word in each stream. Based on this idea, we view the distribution of background model, user interest, breaking news and friends' timeline as four different streams, and apply collapsed Gibbs sampling using the following updating rules:

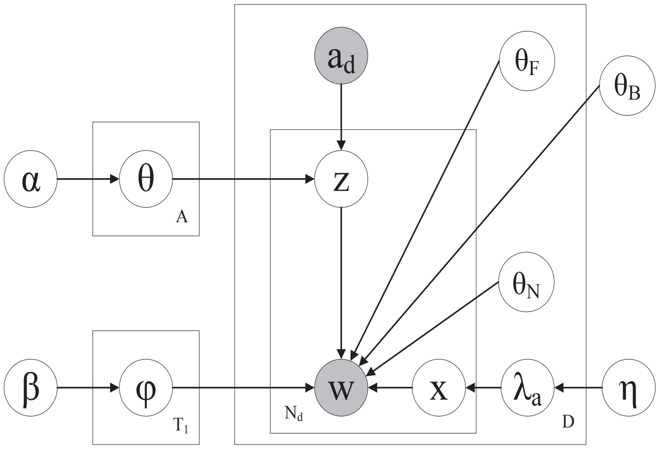

Figure 5: Bayesian Graphical Framework of the Mixture Model.

$$p(x_i = 0, z_i = t) \propto$$
$$(1 - \lambda_B)\frac{e_{a,0,-i} + \eta}{e_{a,-i} + 3\eta}\frac{m_{a,t,-i} + \alpha}{m_{a,-i} + \alpha K}\frac{n_{t,w_i,-i} + \beta}{n_{t,-i} + \beta W} \quad (7)$$

$$p(x_i = 1) \propto \lambda_B p(w_i|\theta_B) \quad (8)$$

$$p(x_i = 2) \propto (1 - \lambda_B) p_T(w_i|\theta_N)\frac{e_{a,2,-i} + \eta}{e_{a,-i} + 3\eta} \quad (9)$$

$$p(x_i = 3) \propto (1 - \lambda_B) p_T(w_i|\theta_{aF})\frac{e_{a,3,-i} + \eta}{e_{a,-i} + 3\eta} \quad (10)$$

where $e_{a,j,-i}$and $e_{a,-i}$ are computed as:

$$e_{a,j,-i} = \frac{c_{a,j,-i}}{\sqrt{|\theta_j|}} \qquad (j = 0, 2, 3) \quad (11)$$

$$e_{a,-i} = \sum_j e_{a,j,-i} \qquad (j = 0, 2, 3) \quad (12)$$

Here $c_{a,j,-i}$ is the number of words written by a assigned to stream j (excluding the ith word), $|\theta_j|$ is the size of stream j, which means the number of non-zero words in the distribution of stream j (for the stream of user interest, we use $c_{a,0,-i}$ as an approximation). While previous works in [8, 18, 33] simply use $c_{a,j,-i}$ to denote the importance of stream j, in our work we smooth it with the size of the corresponding stream, since different streams have different size. $m_{a,t,-i}$ is the number of words posted by a assigned to topic t (excluding the ith word) and $n_{t,w,-i}$ is the number of times word w assigned to topic t (excluding the current one). m_a denotes the total number of words posted by a and n_t is the total number of words under topic t. K means the number of latent topics and W is the number of words.

The other parameters can be estimated as follows:

$$\theta_{a,t} = \frac{m_{a,t} + \alpha}{m_a + \alpha K} \quad (13)$$

$$\varphi_{t,w} = \frac{n_{t,w} + \beta}{n_t + \beta W} \quad (14)$$

$$\lambda_{a,j} = \frac{e_{a,j} + \eta}{e_a + 3\eta} \qquad (j = 0, 2, 3) \quad (15)$$

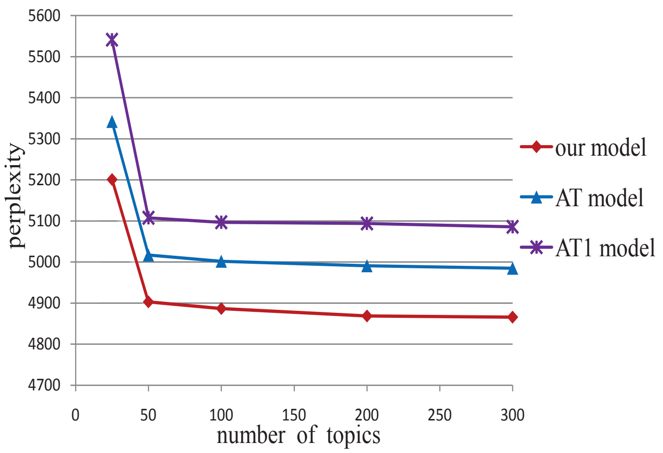

Figure 6: Perplexity of Held-out Content.

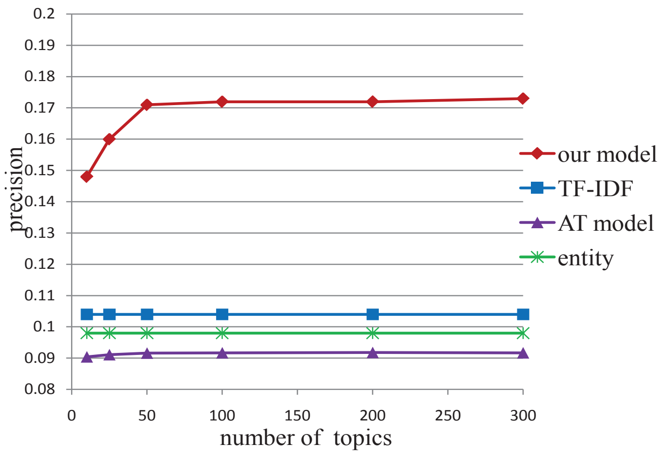

Figure 7: Precision of Predicting Retweets.

$$p(w|\theta_{aI}) = \sum_t \theta_{a,t} * \varphi_{t,w} \tag{16}$$

Hyper-parameters like α, β and η can be estimated using standard methods introduced in [31]. It is worth to mention that we also try to compute the weight of background model automatically using formula similar to (15), however the result is not as good as to set a reasonable weight before training starts (but it still outperforms our baseline).

5. EXPERIMENT

We examine the proposed model from three different perspectives: the perplexity of held-out content, the performance of predicting retweets and the quality of generated latent topics.

5.1 Perplexity of Held-out Content

The perplexity in this study means the performance of prediction for new tweets, which is a widely used method to judge the performance of a topic model. We compare the perplexities of two topic models: our model and the author-topic model. We randomly split the tweets of each user into 90% training tweets and 10% test tweets, and compute the perplexity of all test tweets according to:

$$exp(-\frac{1}{\sum_a m_a^{test}} \sum_a \sum_{w^{test}} log(p_a(w^{test}))) \tag{17}$$

where the predictive probability of a test word is denoted by $p_a(w^{test})$, and is computed by equation (1) in our model. A lower perplexity indicates better performance. We run each model five times and the perplexity of each model is average value. Figure 6 shows the results for the proposed model and author-topic model (AT model) with different number of topics. T is set to be 24 hours and λ_B is set to be 0.3 in the mixture model. As a result, the proposed model outperforms the author-topic model. We also add tweets published by users' close friends (top retweeted/mentioned friends) into the training data of author-topic model, however, as denoted by AT1 model, directly use posts from social friends even lower the performance. Notice that, the perplexities of both models do not change apparently when the number of topics is greater than 50.

5.2 Performance of Predicting Retweets

The main purpose of user modeling is to help users find interesting information from the overwhelming information streams. While in the context of Twitter, retweet is the most important signal of user interest, as users are prone to broadcast their favorite tweets to their followers. Thus, the performance of predicting retweets is a good standard to judge the performance of a user model.

Specifically, for each user in every time interval T, we randomly select a tweet that is retweeted by him (if exists), and mix it with 10 other tweets that are not retweeted by him. All the 11 tweets are published by his top retweeted/mentioned friends in the same time interval. This experiment can be seen as a real information filtering application: when a user is viewing his Twitter stream consists of 11 new tweets, he might find a tweet interesting and retweet it to his followers, on the other side, the other 10 tweets are not important to him relatively (since all the 11 tweets are published in the same time interval by his close friends, it is reasonable to assume that the user can see all of them at once). The task of a user model is to accurately predict which tweet can attract the user's attention and will be retweeted by him.

5.2.1 Compared with other user models

We first compare the performance of our model with three user models: the author-topic model in [41], the TF-IDF algorithm used in [9] and the entity-based user profile in [2] (The AlchemyAPI is used to extract entities). The predictive probability of each tweet d is computed according to equation (18) in our model, and the one with the highest probability is predicted as the retweet. T is set to be 24 hours and λ_B is set to be 0.3 again. For the three competitors, tweets are ranked based on their cosine similarities with user profiles. We repeat the experiment five times on different random sample sets and the results are the average value. Figure 7 shows the predictive precision of our model and all the other competitors.

$$p(d) = \frac{1}{N_d} \sum_{w \in d} p_a(w) \tag{18}$$

As a result, about 17.2% of retweets are correctly predicted in our model, which is clearly better than our competi-

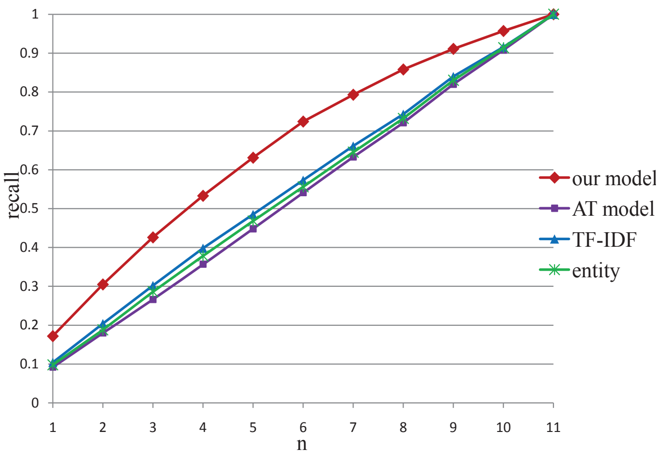

Figure 8: Recall of Predicting Retweets (5.2.1).

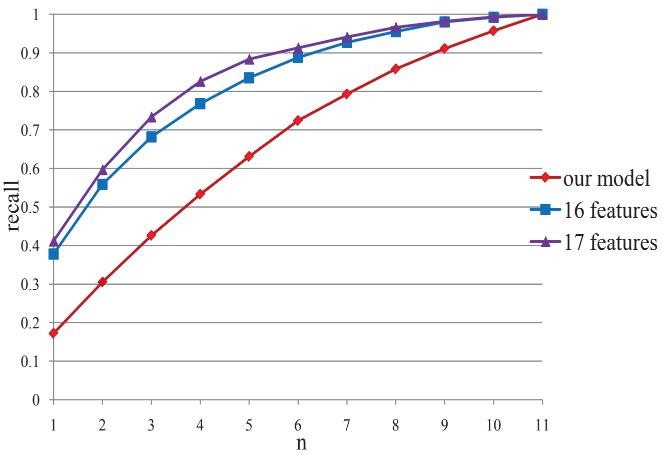

Figure 9: Recall of Predicting Retweets (5.2.2).

tors. The precision of TF-IDF algorithm, entity-based user profile and author-topic model are 10.4%, 9.8% and 9.2% respectively, which are only little better than random selection (9.1%). However, we must point out that the task is difficult since all candidate tweets are published by users' closest friends. Furthermore, all the four methods tend to reflect the similarities of tweets with users' general posting behavior, which are not especially intended for retweet prediction settings. Thus the relatively low precision of all models are reasonable. Nevertheless, the result still shows that our model can reach a better understanding of user posting behavior than the three competitors.

To avoid missing too much retweets, we should provide more candidates to increase the recall of all models. Assume that each time we provide the top n results returned by all methods, figure 8 gives the recall of retweets with different n. The number of topics is set to be 50 in our model. As demonstrated, our model outperforms the three competitors for all different n.

5.2.2 *Compared with a retweet prediction model*

We compare the performance of our model with a retweet prediction model. While retweet is recognized as the key mechanism for information diffusion on Twitter, a rich set of studies has been conducted to predict retweets [16, 38], mainly based on classification frameworks which incorporate different features related to tweets or authors.

We use logistic regression to build a retweet prediction model, leveraging 16 different features that are found to be important in previous retweet prediction models [16, 38], including author-based features (i.e., # of followers, # of followees, # of times listed, is he a verified user, his account age, # of tweets published totally and # of tweets published per day), tweet-based features (i.e., # of urls, # of hashtags, # of users mentioned, # of words, is the tweet a reply, is the tweet a retweet itself) and content-based features (i.e., the TF-IDF, entity and latent topic similarities of tweets with users' past tweets, which are the three competitors in 5.2.1).

We use the same test set in 5.2.1, and build another different training dataset based on the same method: for each group of 11 tweets, the retweet is labeled as positive example and the other 10 tweets are viewed as negative examples. To predict retweets, tweets are ranked based on their retweet probabilities returned by logistic regression.

Figure 9 gives the recall of predicting retweets with different n. The performance of our model remains unchanged to figure 8 because of the same test set. As shown in figure 9, the retweet prediction model (denoted by 16 features) indeed has large superiority over our model, since the proposed mixture model is not quite intended for retweet prediction tasks (our model only tries to reflect the likelihood of users to generate content, where other important factors associated with retweets are not considered, such as the global influence of the author and the syntactic features of tweets).

However, our mixture model can still provide an important feature for retweet models. As denoted by 17 features, the performance of the retweet model is improved after using the predictive probability of our model as a new feature (e.g., the recall of retweets is improved from 37.8% to 41.2% for top 1 result), which means that our model is of great importance for retweet prediction models. Notice that, in the 17-features retweet model, if we remove 64% tweets from a user's Twitter stream, we can still reach a recall of retweets over 80%, which is quite meaningful for social media websites like Twitter, where information overwhelming has already become a serious problem.

5.3 Impact of Model Parameters

We investigate the impact of λ_B and T on the model performance. Figure 10 shows the results of different λ_B when T is set to be 24 hours. The vertical axis on the left side of the graph is the perplexity of held-out content and the one on the right side means the precision of predicting retweets. As shown in figure 10, the model performance is generally similar when λ_B is small, and drops dramatically when λ_B is greater than 0.5. The best value for λ_B is between 0.2 and 0.3. Even if we set λ_B to be zero, the result is still satisfactory, which means the background model is not quite important to our method. We further fix λ_B to be 0.3 and analyze the influence of different time interval T in figure 11. When T is 24 hours, the model performance is the best. With T get longer, the model becomes coarse-grained and the performance drops. On the other side, the performance will also drop when T is shorter than 24 hours. According to

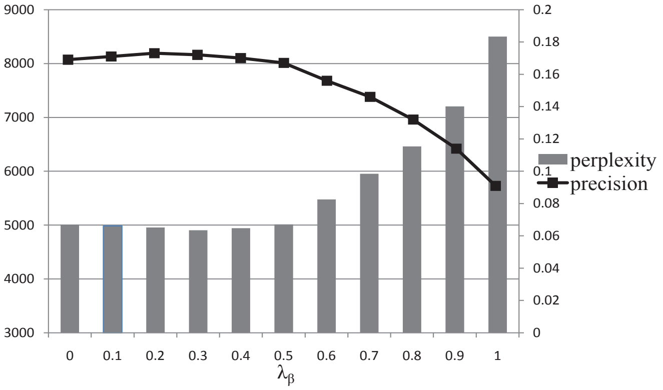

Figure 10: Influence of λ_B.

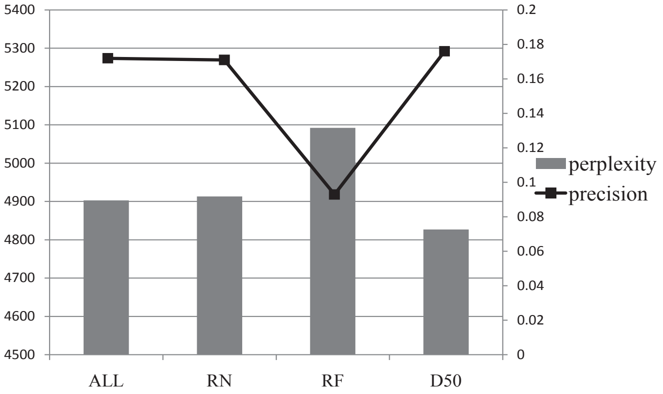

Figure 12: Importance of News and Friends' Posts.

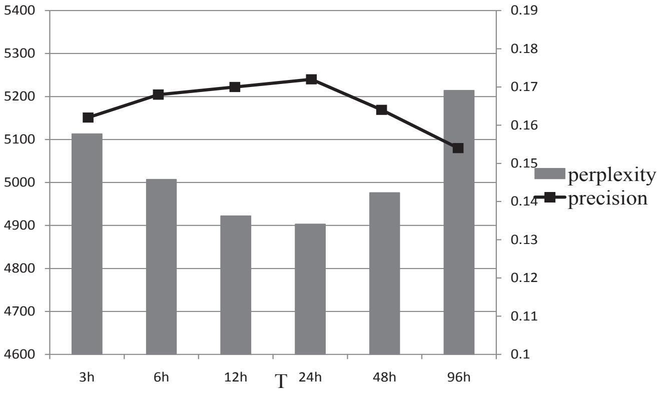

Figure 11: Influence of T.

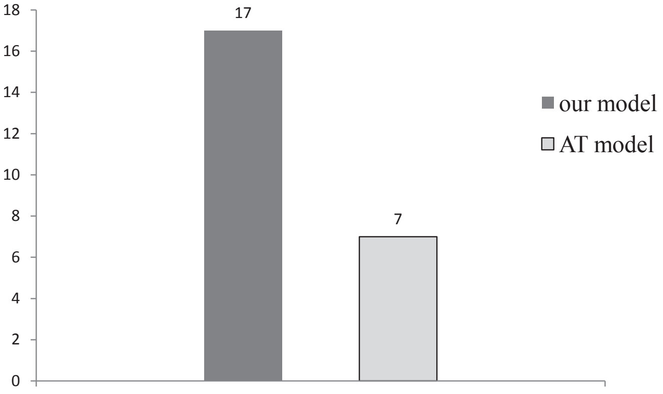

Figure 13: Labeled Result.

our observation, it might probably due to the data sparsity problem in each short time interval.

We further analyze the importance of breaking news and friends' timeline in our model respectively. As shown in figure 12, when removing breaking news from our model (denoted by RN), the performance remains almost the same, and when removing friends' timeline from our model (denoted by RF), the performance drops dramatically. This indicates that the distribution of friends' timeline contributes a lot to the mixture model, but the distribution of breaking news is of little importance. The little impact of breaking news might due to two reasons: 1. the distribution of breaking news is really sparse 2. it is quite possible that users' friends will publish posts about breaking news, which might also lower the importance of breaking news. However, the existence of breaking news is still meaningful, for example, as denoted by D50, the performance of our model is clearly better than average on the 50th day, when the wedding of prince William happened.

5.4 Quality of Latent Topics

Another typical method to judge the performance of topic models is to print top words for the latent topics and judge them by experience. We design an experiment to compare the latent topics generated by author-topic model and the proposed model. Specifically, we set the number of topics to be 50 and manually extract the same salient topics for both models. As a result, 8 latent topics are extracted, and the rest of latent topics are either meaningless topics or different topics between two models. We present the top 50 words of each latent topic to three labelers and ask them to label which one can better represent a topic. Due to the limit of space, table 2 only displays the top 10 words for each salient topic, and figure 13 gives the labeled result. On average, 71% of topics generated by the proposed model are labeled better, which shows that our model can reach a better understanding of latent topics behind Twitter streams.

5.5 Discussion

The results of our experiments prove that the proposed model clearly outperforms other user models. By comparing the perplexity of held-out content and the quality of generated latent topics, our mixture model is shown to be better than the traditional author-topic model, and within the task of predicting retweets, its superiority over other competitors is still obvious. Although our model is not comparable to retweet models in predicting retweets, however, it can still provide an important feature for them.

To reach a microscopic understanding of our mixture model, we empirically analyze how different factors work in our model based on a random sample of 30 users. First, we find that words associated with breaking news are generally well recognized. Second, for users with obvious interest (11 users are found to often publish tweets about some areas), our model can discover words related to user interest very well. Take a technology fan as an example, words such as "ipad" and "android" are easily assigned to his interest by our model. On the other side, if a user does not show strong

Table 2: Top 10 Words for Latent Topics

Topic 0		Topic 1		Topic 2		Topic 3	
mixture model	AT model	mixture model	AT model	mixture model	AT model	mixture model	AT model
film	film	libya	libya	canada	canada	game	state
movie	movie	war	people	election	vote	team	game
films	game	forces	news	canadian	election	coach	coach
screening	awesome	arab	israel	toronto	toronto	fans	draft
director	games	killed	killed	vote	harper	play	nfl
trailer	comic	gaddafi	forces	campaign	party	players	team
movies	time	military	police	government	campaign	sports	football
festival	trailer	israel	military	politics	canadian	games	players
interview	play	libyan	arab	harper	debate	basketball	ncaa
comedy	video	security	bin	vancouver	labour	season	big
Topic 4		Topic 5		Topic 6		Topic 7	
mixture model	AT model	mixture model	AT model	mixture model	AT model	mixture model	AT model
app	app	music	music	social	social	food	food
ipad	google	song	show	media	media	dinner	wine
iphone	ipad	album	tonight	twitter	great	wine	dinner
google	iphone	video	album	facebook	business	eat	beer
apple	apple	show	playing	blog	top	chocolate	eat
android	mobile	playing	band	post	stories	lunch	great
apps	twitter	songs	song	web	daily	cheese	lunch
data	android	listen	tour	marketing	twitter	restaurant	chocolate
code	facebook	listening	night	online	blog	delicious	love
web	video	tour	great	google	post	chicken	cheese

interest in any area, words assigned to user interest will be less reasonable, since it is difficult to model his topics of interest accurately. Third, the factor of social influence mainly captures words that are recently published by users' friends (and are not quite related to users' interest). Most words inspired by friends can be captured successfully, despite there are also many noise words included. Finally, some words out of all three factors can be handled by the background model, most of which are high-frequency words appear in daily life posts and conversations.

What is the potential value of our model for some real tweets recommendation systems? Admittedly, as demonstrated in the low accuracy of predicting retweets, it might still not be accurate enough to recommend tweets only based on our model. We believe that a good tweets recommendation system should based on classification frameworks which incorporate various of features (just like the retweet model in 5.2.2), and our model can be used as one feature to reflect the similarities of tweets with users' general posting behavior, as well as other user modeling frameworks. Furthermore, there are also many other important factors should be considered, such as the geographic information of users and their entire social graph, which we leave as future work due to the restriction of our current dataset.

Finally, we must point out that it is quite a difficult task to model user posting behavior on Twitter, since users can easily generate content with any intentions at different time and place. On account of this, our model is still a macro-level modeling of user behavior, as only three factors are included in it. Thus, the interpretation of our model at micro-level might not be good enough sometimes, especially for users with no obvious interest. Nevertheless, as shown in the results of our experiments, our model can still reach a better understanding of user posting behavior in most cases, and lay out a foundation to personalization and information filtering applications on Twitter.

6. CONCLUSION

In this paper, we propose a mixture model to analyze user posting behavior on social media. By assuming that user behavior is mainly influenced by three factors: breaking news, posts from social friends and user's intrinsic interest, our method is able to reach a more comprehensive model of user posting behavior on social media. We demonstrate the superiority of the proposed model on a popular social media website, Twitter, from three different tasks: the perplexity of held-out content, the performance of predicting retweets and the quality of generated latent topics. The results are satisfactory and our model clearly outperforms other traditional methods.

Our future work lies in several areas. First, our basic idea is not limited to latent topic models, and it will be an interesting direction to test it under other frameworks, such as the entity-based or category-based user modeling frameworks. Second, while it is widely agreed that user interest will change with time, our method does not model the change of user interest explicitly since the experimental time period is relatively short. However, to achieve a long time understanding of user posting behavior, it is necessary to incorporate time factors into user interest model. Third, there might be some special terms besides words in tweets, such as URLs, hashtags and usernames. We tend to investigate whether the presence of those terms can help improve our model performance. Finally, the distribution of breaking news and friends' timeline computed in our model are simple, and more accurate methods are worth further exploration. For example, to compute breaking news around users' geographic location, or try different measures of social influence in computing the distribution of friends' timeline.

7. REFERENCES

[1] F. Abel, Q. Gao, G.-J. Houben, and K. Tao. Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web. In *Proc. of WebSci*, 2011.

[2] F. Abel, Q. Gao, G.-J. Houben, and K. Tao. Analyzing user modeling on twitter for personalized news recommendations. In *Proc. of UMAP*, 2011.

[3] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding high-quality content in social media. In *Proc. of WSDM*, 2008.

[4] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *Proc. of KDD*, 2008.

[5] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *Proc. of CEAS*, 2010.

[6] F. Benevenuto, T. Rodrigues, and V. Almeida. Detecting spammers and content promoters in online video social networks. In *Proc. of SIGIR*, 2009.

[7] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida. Characterizing user behavior in online social networks. In *Proc. of IMC*, 2009.

[8] C. Chemudugunta, P. Smyth, and M. Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. In *Proc. of NIPS*, 2006.

[9] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. H. Chi. Short and tweet: Experiments on recommending content from information streams. In *Proc. of CHI*, 2010.

[10] N. A. Diakopoulos and D. A. Shamma. Characterizing debate performance via aggregated twitter sentiment. In *Proc. of CHI*, 2010.

[11] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, and H. Zha. Time is of the essence : Improving recency ranking using twitter data. In *Proc. of WWW*, 2010.

[12] D. Gruhl, R.Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *Proc. of WWW*, 2004.

[13] L. Guo, E. Tan, S. Chen, X. Zhang, and Y. Zhao. Analyzing patterns of user content generation in online social networks. In *Proc. of KDD*, 2009.

[14] J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proc. of RecSys*, 2010.

[15] C. Honeycutt and S. C.Herring. Beyond microblogging: Conversation and collaboration via twitter. In *Proc. of HICSS*, 2009.

[16] L. Hong, O. Dan, and B. D. Davison. Predicting popular messages in twitter. In *Proc. of WWW*, 2011.

[17] L. Hong and B. D. Davison. Empirical study of topic modeling in twitter. In *Proc. of SOMA*, 2010.

[18] L. Hong, B. Dom, S. Gurumurthy, and K. Tsioutsiouliklis. A time-dependent topic model for multiple text streams. In *Proc. of KDD*, 2011.

[19] B. A. Huberman, D. M. Romero, and F. Wu. Social networks that matter: Twitter under the microscope. 2008.

[20] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: Understanding microblogging usage and communities. In *Proc. of WEBKDD*, 2007.

[21] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proc. of WWW*, 2010.

[22] A. Leavitt, E. Burchard, D. Fisher, and S. Gilbert. New approaches for analyzing influence on twitter. *a publication of the Web Ecology project.*

[23] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: Social honeypots + machine learning. In *Proc. of SIGIR*, 2010.

[24] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *Proc. of KDD*, 2008.

[25] J. Leskovec, K. J.Lang, A. Dasgupta, and M. W.Mahoney. Statistical properties of community structure in large social and information networks. In *Proc. of WWW*, 2008.

[26] C.-Y. Lin, K. Ehrlich, V. GriffithsFisher, and C. Desforges. Smallblue: People mining for expertise search. *IEEE Multimedia Magazine*, 2008.

[27] J. Liu, P. Dolan, and E. R. Pedersen. Personalized news recommendation based on click behavior. In *Proc. of IUI*, 2010.

[28] M. Mathioudakis and N. Koudas. Twittermonitor: Trend detection over the twitter stream. In *Proc. of SIGMOD*, 2010.

[29] D. M.Blei, A. Y.Ng, and M. I.Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003.

[30] M. Michelson and S. A. Macskassy. Discovering users' topics of interest on twitter: A first look. In *Proc. of AND*, 2010.

[31] T. P. Minka. Estimating a dirichlet distribution. 2009.

[32] M. Papagelis, V. Murdock, and R. van Zwol. Individual behavior and social influence in online social systems. In *Proc. of HT*, 2011.

[33] M. Paul and R. Girju. Cross-cultural analysis of blogs and forums with mixed-collection topic models. In *Proc. of EMNLP*, 2009.

[34] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proc. of UAI*, 2004.

[35] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proc. of WWW*, 2010.

[36] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D.Lieberman, and J. Sperling. Twitterstand: News in tweets. In *Proc. of GIS*, 2009.

[37] J. Stoyanovich, S. Amer-Yahia, C. Marlow, and C. Yu. Leveraging tagging to model user interests in del.icio.us. In *Proc. of AAAI*, 2008.

[38] B. Suh, L. Hong, P. Pirolli, and E. H. Chi. Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In *Proc. of SocialCom*, 2010.

[39] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proc. of CHI*, 2010.

[40] Z. Wen and C.-Y. Lin. On the quality of inferring interests from social neighbors. In *Proc. of KDD*, 2010.

[41] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: Finding topic-sensitive influential twitterers. In *Proc. of WSDM*, 2010.