Incorporating User Grouping into Retweeting Behavior Modeling*

Author 1† Author 2‡

Abstract

Social media applications are emerging, with rapidly growing users and large numbers of retweeted blogs every minute. The variety among users makes it difficult to model their retweeting activities. Obviously, it is not suitable to cover the overall users by a single model. Meanwhile, building one model per user is not practical. To this end, this paper presents a novel solution, of which the principle is to model the retweeting behavior over user groups. Our approach, GruBa, consists of three key components for extracting user based features, clustering users into groups, and modeling upon each group. Particularly, we look into the user interest from different perspectives including long-term/recent interests and explicit/implicit interests, which results deep analyses towards the retweeting behavior and proper models in the end. We have evaluated the performance of GruBa by datasets of real-world social networking applications and a number of query workloads, showcasing its benefits.

1 Introduction

The rapid development of social network is accompanied by the generation of tremendous user-generated contents. People not only consume information but also publish or share messages. An interesting function in many existing Social Network Applications is reposting (e.g, reposting someone else's microblog on weibo.com). Modeling reposting behavior is of vital significance for Social Network Applications, which can help them to mine potential benefits, predict bursting events and so on.

Modeling reposting behavior has attracted attention from both academia and industry. However, many existing works have certain limitations. First, most of them mainly studied how to model reposting behavior for a single user or for the whole users. Considering the huge amount of users in Social Network , modeling for a single user is not easy. What's more, it can also make our model too particular. Modeling for the whole users is also not a good idea, which may get a inaccuracy model. Secondly, how to extract users's features, especially the extraction of users's interest, is a challenging problem.

To this end, we extract each user's features firstly, including user's basic information, behavior information and interests, then perform users clustering to divide users into groups and model reposting behavior of each user group respectively. The main contributions of this work can be summarized as follows:

- We construct a large collection of microblogs from a real microblog service. It contains users's information, microblog content and part of the social network information of related users.
- Instead of modeling for a single user or the whole users, we divide all users into several groups by users clustering and model for each group.
- We propose a noval method to extract user interests from user generated texts.
- We develop a demo system for visualization of single user's profiling and each group's information.

This article is organized as follows: In Section 2, we introduce the framework of our work. We then describe our methodology in section 3, including features extraction, how we cluster users into groups, and how to model reposting behavior. Section 4 presents and discusses the experimental results. Section 5 discusses related work followed by conclusions in Section 6.

2 GruBa Overview

2.1 Problem Formulation We consider people's retweeting behavior in social media. For simplicity, with a given user, we assume that blogs created or retweeted by his/her followees cover the overall candidates, from which the said user may retweet. All our results could straightforwardly generalize to alternative candidate scopes.

DEFINITION 1. *A blog $B = (O, T, M, R)$ consists of the owner O (a.k.a. user in this paper) to whom B belongs*

*Supported by NSFC (U1636210), 973 program(2014CB340300), NSFC (61322207&61421003), Special Funds of Beijing Municipal Science & Technology Commission, Beijing Advanced Innovation Center for Big Data and Brain Computing, and MSRA Collaborative Research Program.

†affiliation
‡affiliation

(either created or retweeted), the timestamp T showing when B is generated, the blog message M and a bit R denoting B is retweeted (1) or originally created (0) by the owner O.

DEFINITION 2. *A user $U = (B_s, R_s, E_s)$ consists of three sets regarding the user's blogs B_s, followers R_s and followees E_s separately. Each follower/followee per se refers to a user.*

The mapping between blog B and user U is a bilateral operation, i.e., $U = O(B)$ and $B \in B_s(U)$, through ID(s) of user and blog respectively.

Informally, providing a set of users $\{U\}$ and the associated blogs $\{B\}$, as well as a blog query b and a follower of $O(b)$ written as f, i.e., $f \in R_s(O(b))$, GruBa shall build a retweeting model for $(\{U\}, \{B\})$, upon which Y/N is returned regarding whether f shall retweet b.

2.2 GruBa Framework GruBa is designed from the ground up as a system for modeling users' retweeting behavior in social media. Figure 1 shows the architectural components of GruBa, comprising three subsystems: Data Storage, Processing Runtime and Profile Demonstrator.

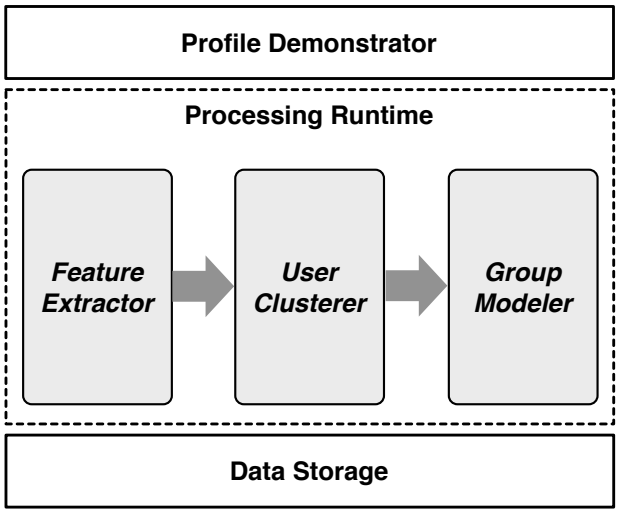


Figure 1: GruBa Architecture

Data Storage. The underlying Data Storage subsystem stores data to be processed by GruBa, i.e., data of blogs and users, as shown in *Definition 1* and *2*.

Processing Runtime. In the heart of GruBa lies the Processing Runtime subsystem, which consists of three major components as follows.

1. Feature Extractor: By coalescing the blog data, each user is depicted by a bunch of features, which are grouped into three categories. They are features of *Info* (e.g., the number of followers and followees), *Behavior* (e.g., the frequency and the popular slots of retweeting) and *Interest* (e.g., the long-term/recent interests, as well as the explicit/implicit interests). These features are extracted from the stored data by Feature Extractor and serve as the input of User Clusterer.
2. User Clusterer: Providing the user-based features, User Clusterer takes charge of the clustering task such that each user falls into a proper group.
3. Group Modeler: For each group obtained by User Clusterer, Group Modeler employs both positive and negative samples to train a model, over which the testing of users' retweeting behavior is performed.

Profile Demonstrator. At the top layer of GruBa, it is the Profile Demonstrator subsystem for visualization. For the time being, Profile Demonstrator presents [To Be Completed].

3 Feature Extractor

With the underlying data in Data Storage subsystem, Feature Extractor is responsible for "mining" the user characteristics, resulting three features for each user. These features are *Info Feature*, *Behavior Feature* and *Interest Feature*, which constitute the *Feature Data* in GruBa, i.e., *Feature Data* = {*Info Feature*, *Behavior Feature*, *Interest Feature*}.

3.1 Info Feature *Info Feature* employs a vector I to cover the basic info of user.

$$I = (\#R_s,\ \#E_s,\ R_{ee},\ \#B_s,\ R_{oc},\ U_t) \tag{3.1}$$

where each variable is illustrated in Table 1. Specifically, we use $\#(B_s|R(B) == 1)$ and $\#(B_s|R(B) == 0)$ to represent the number of retweeted blogs and blogs that are originally created by the user.

3.2 Behavior Feature Unlike *Info Feature*, *Behavior Feature* shows several statistics regarding the user's retweeting behavior. Such statistics include:

- a value showing the average number of retweeted blogs per week: $\#W_r$
- a normalized vector regarding the time distribution of a user's retweeting behavior: $P_t = (p'_0,\ p'_1,\ ...,\ p'_{11})$, where p'_0 is the probability that the retweeting activity happens from 0am to 2am, p'_1 is the probability that the retweeting activity happens from 2am to 4am, and so on.

Table 1: Illustration of Variables in Info Feature

variable	illustration
$\#R_s$	number of followers
$\#E_s$	number of followees
R_{ee}	a ratio defined as $\frac{\#R_s}{\#E_s}$
$\#B_s$	number of blogs owned by the user
R_{oc}	a ratio defined as $\frac{\#(B_s \mid R(B)==1)}{\#(B_s \mid R(B)==0)}$
U_t	user type (as detailed in Table 2)

Table 2: Category of Info

value	illustration
0	$\#E_s \leq 50$ & $\#R_s \leq 50$
1	$\frac{\#E_s}{\#R_s} \geq 5$
2	$\frac{\#R_s}{\#E_s} \geq 5$
3	other cases

- a normalized vector with respect to the gap distribution of a user's retweeting behavior: $P_g = (p_0'', p_1'', ..., p_5'')$, in which p_0'' is the probability that the gap between two retweeted blogs is within 1 min. Ditto for p_1'' (1 min to 1 hour), p_2'' (1 to 12 hours), p_3'' (12 to 24 hours), p_4'' (24 to 48 hours) and p_5'' (more than 48 hours).

Hence, *Behavior Feature* H per user comes with:

$$H = (\#W_r,\ P_t,\ P_g) \tag{3.2}$$

where $\#W_r$, P_t and P_g are illustrated as above.

3.3 Interest Feature Different from the straightforward notions of *Info Feature* and *Behavior Feature*, *Interest Feature* involves a process of labeling users by their interested topics. In short, with a given lexicon consisting of several *topics*, the interest feature of a user is a normalized vector, in which each dimension refers to the probability that the said user matches a *topic*.

DEFINITION 3. *A lexicon $L = \{t\}$ consists of a set of topics while each topic $t = \{c\}$ is associated with a list of cell words $\{c\}$. Each cell word c refers to one unique word in L.*

DEFINITION 4. *With a given user u, each blog $b \in B_s(u)$ could be decomposed into a set of words $\{w\}$, i.e., $b = \{w\}$.*

DEFINITION 5. *The interest feature of a given user is a normalized vector*

$$P_f = (p_0,\ p_1,\ ...,\ p_{x-1}) \tag{3.3}$$

in which the said user matches x topics in lexicon and p_i refers to the similarity of the user and each matched topic (interest). The definition of such similarity shall be detailed in each scenario (explicit/implicit interest analysis, towards words/topics, etc).

In GruBa, a word, either in the form of c or w, acts as the minimum unit for analysis. Hence, the similarity of a word pair $(w,\ c)$, i.e., $sim(w, c)$, could be generalized to the similarity of a blog against one topic $sim(b, t)$, and finally to a user versus each topic in lexicon $sim(u, \{t\})$; topics with similarity satisfying certain thresholds are allocated to the user u and constitute the interests of u.

For instance, the following steps depict the "mining" process of the interest feature P_f for a user u.

Step 1: Each blog of u is decomposed into a word set, i.e., $b = \{w\}$ where $b \in B_s(u)$.

Step 2: Explicit interests are explored. Specifically, every word w is sent to match each cell word c of lexicon topics. If w and c are identical, $sim(w, c) = 1$. Otherwise, $sim(w, c) = 0$. As to the similarity of b against a lexicon topic t, it is:

$$sim(b, t) = \sum_{i,j} sim(w_i, c_j) \tag{3.4}$$

where $sim(w_i, c_j)$ refers to the similarity of a word pair.

If $sim(b, t)$ satisfies a certain threshold, topic t is labeled to blog b; the user u is then discovered having an explicit interest (topic) t. Thus, by looking into the similarity of b against all topics in lexicon, the explicit interests of u is returned, in the form of interest feature (see Definition 5).

If none of $sim(b, t)$ could meet the threshold, i.e., explicit interest discovery over user u fails, go to *Step 3* and *Step 4* in parallel, so as to "mine" the implicit interests of u.

Step 3: A metric *TF-IDF weight* W_f is computed, i.e., employing TF-IDF [reference] to calculate the weight distribution of words in blog b:

$$W_f = \{(w_i, p_i)\} \tag{3.5}$$

where w_i refers to a single word, of which the weight is p_i, with $\sum_i p_i = 1$.

To compute such weight p_i for word w_i, a metric p_i'' is first calculated as:

$$p_i'' = \frac{|b_i|}{|b|} * log(\frac{|D_i|}{|D|}) \tag{3.6}$$

in which we use the operator $|\ |$ to measure the cardinality, such that $|b_i|$ is the occurrences of word w_i in blog b and $|b|$ the total occurrences of all words in b. Ditto for $|D_i|$ and $|D|$, except that the scope is the overall dataset, rather than a single blog b.

Hence, each word w_i shall get an initial weight of p_i'', upon which the normalization is performed and p_i is obtained, resulting the *TF-IDF weight* (see Definition 3.5). Go to *Step 5*.

Step 4: Similarly, another metric *Twitter-LDA weight* W_w is obtained, i.e., using Twitter-LDA [reference] to result the word weight distribution of blog b. Unlike TF-IDF [reference], Twitter-LDA [reference] first trains the overall blogs, allocating each blog with a *tag*. The structure of *tag* is as follows:

$$W_t = \{(w_i', p_i')\} \tag{3.7}$$

where w_i' refers to a word in *tag* W_t, and p_i' is the probability that w_i' appears in blogs with the said *tag*, with $\sum_i p_i' = 1$. Subsequently, W_t are leveraged to conclude W_w, i.e., $W_w = W_t$, which shares the format with that of W_f. Go to *Step 6*.

Step 5: TF-IDF [reference] based similarity is calculated. For example, the similarity (in the form of a value) of W_f over a single topic t in lexicon, written as $sim(W_f, t)$, is defined as:

$$sim(W_f, t) = \sum_i p_i * sim(w_i, t) \tag{3.8}$$

where $W_f = \{(w_i, p_i)\}$, $t = \{c_j\}$, and:

$$sim(w_i, t) = \sum_j sim(w_i, c_j) \tag{3.9}$$

Go to *Step 7*.

Step 6: Accordingly, Twitter-LDA [reference] based similarity is available. Again, a single topic t in lexicon is used for yardstick and the similarity of W_w over t, written as $sim(W_w, t)$, is defined as:

$$sim(W_w, t) = \sum_i p_i' * sim(w_i', t) \tag{3.10}$$

where $W_w = \{(w_i', p_i')\}$, $t = \{c_j\}$, and:

$$sim(w_i', t) = \sum_j sim(w_i', c_j) \tag{3.11}$$

Go to *Step 7*.

Step 7: Hence, the similarity of a blog b against a lexicon topic t is given by:

$$sim(b, t) = \alpha * sim(W_f, t) + (1 - \alpha) * sim(W_t, t) \tag{3.12}$$

where the α is a parameter by which GruBa could set flexible priorities between TF-IDF [reference] and Twitter-LDA [reference]. Go to *Step 8*.

Step 8: Repeat the above steps (Step 1 to Step 7) for the blog b over every topic in lexicon, i.e., $\forall t_k \in L$ results one similarity value of $sim(b, t_k)$. Such computation further extends to all the blogs owned by user u, such that: $\forall b_m \in B_s(u)$, $\forall t_k \in L$, there exists a similarity of $sim(b_m, t_k)$. Hence, the overall similarity of user u over lexicon topics $\{t\}$ (i.e., L), written as $S(u,\ L)$, could be denoted by a vector:

$$S(u, L) = (s_0,\ s_1,\ ...,\ s_{n-1}) \tag{3.13}$$

where n refers to the cardinality of L (i.e., number of topics in L) and s_k is the overall similarity of user u over topic t_k, which is given by:

$$s_k = \sum_m sim(b_m, t_k) \tag{3.14}$$

Among the n dimensions of $S(u,\ L)$, those with top x similarity values are selected to label the implicit interests of user u, which results an x dimensional vector P_f as described in Definition 5. Similarly, interest features of all users are returned.

As a result, the *Feature Data* for every user u, written as $F(u)$, is given by:

$$F(u) = (I,\ H,\ P_f) \tag{3.15}$$

where I, H and P_f refer to *Info Feature*, *Behavior Feature* and *Interest Feature* separately (see formulas 3.1, 3.2 and 5). And it could be written as a vector:

$$F(u) = (\#R_s, \#E_s, R_{ee}, \#B_s, R_{oc}, U_t, \#W_r, P_t, P_g, P_f) \tag{3.16}$$

where each dimension refers to a data item of GruBa.

4 User Clusterer

Providing the *Feature Data*, User Clusterer takes the charge of grouping each user concerned into a proper cluster. Algorithm 1 illustrates such overall procedure.

The idea is to enumerate a number of clustering trials (line 4) and select the optimal solution with the best coefficient value (v in line 14). In principle, each trial (referred by t in line 4) first performs a clustering task (line 5; to be detailed in section 4.1), resulting a cluster (by $l(u)$) for each user u (line 6); then, each user obtains a coefficient value $v(u)$ stemmed from the

Algorithm 1 User Clustering in GruBa

1: Input: *Feature Data* of users $\{F(u)\}$, the minimum/maximum number of clusters N_i and N_a
2: Output: Optimal user clustering result R
3:
4: **for all** $t \in [N_i, N_a]$ **do**
5: group users $\{u\}$ into t clusters by $\{F(u)\}$
6: clustering result $R'(t) = \{(u, l(u))\}$ with cluster info $l(u)$ for each user u
7: **for all** $u \in \{u\}$ **do**
8: in-cluster distance $d_i(u)$
9: out-cluster distance $d_o(u)$
10: coefficient value $v(u) = \frac{(d_o - d_i)}{max(d_o,\ d_i)}$
11: **end for**
12: $v(t) = Avg\{v(u)\}$
13: **end for**
14: **if** $v(a) == Max\{v(t)\}$ **then**
15: $R = R'(a)$
16: **end if**
17: **return** R

in/out-cluster distances (lines 8–10; shall be illustrated in section 4.2); finally, the averaged coefficient value of all users serves as the coefficient value of the current trial, written as $v(t)$ (line 12), by which the said selection process is conducted (line 14).

Next, we shall now first detail how GruBa performs the clustering task and subsequently illustrate the computation regarding the metric of coefficient value.

4.1 Clustering in GruBa In GruBa, the clustering rests on an optimized K-Prototype [reference] algorithm, named K-Gru in this work. Similar as K-Prototype, K-Gru randomly selects the cluster kernels among samples and employs the minimum distance between them to determine an initial result, upon which the clustering tasks are iterated until the results are stable.

Unlike K-Prototype that supports vector samples in which each dimension is of numerical/categorical, K-Gru could also handle the case where a dimension is one normalized vector. Recall the sample data for User Clusterer, i.e., *Feature Data* in form of vectors (see formula 3.16), of which the data type regarding each dimension is shown as Table 3.

As aforementioned, the clustering of K-Gru rests on the distance between vector samples, where the dimensions are combined with numbers, categories and normalized vectors. For simplicity, we shall first illustrate the distance calculation of the simple vectors with mono data type on each dimension and then demonstrate that of complex vectors in K-Gru.

Table 3: Types of Dimensional Data in *Feature Data* Vector

type	data dimensions
numerical data	$\#R_s$, $\#E_s$, R_{ee}, $\#B_s$, R_{oc}, $\#W_r$
categorical data	U_t
normalized vectors	P_t, P_g, P_f

Given two numerical vectors $Y' = (y'_0, y'_1, ...)$ and $Z' = (z'_0, z'_1, ...)$, the O's Distance [reference] between Y' and Z' is given by :

$$D_n(Y', Z') = \sum_e (y_e - z_e)^2 \tag{4.17}$$

As to the categorical vectors $Y'' = (y''_0, y''_1, ...)$ and $Z'' = (z''_0, z''_1, ...)$, the H's Distance [reference] of Y'' and Z'' is:

$$D_h(Y'', Z'') = \sum_e H_e \tag{4.18}$$

where H_e refers to the H's Distance over each dimension, with $H_e = 1$ if y''_e and z''_e share the identical value, and $H_e = 0$ otherwise.

Regarding two vectors where each dimension is a normalized vector per se, Cosine Similarity [reference] is leveraged to compute the distance. Then, the distance between such two vectors $Y^* = (Y^*_0, Y^*_1, ...)$ and $Z^* = (Z^*_0, Z^*_1, ...)$ is:

$$D_v(Y^*, Z^*) = \sum_e Y^*_e \cdot Z^*_e \tag{4.19}$$

where $\cdot$ refers to the dot product operation between two normalized vectors Y^*_e and Z^*_e.

Hence, the said distance regarding the complex vectors ($Y = (Y_0, Y_1, ...)$ and $Z = (Z_0, Z_1, ...)$) in K-Gru, named G's Distance, could be deduced as:

$$D_g(Y, Z) = \sum_e G_e \tag{4.20}$$

where the distance on each dimension G_e is given by:

$$G_e = \begin{cases} (Y_e - Z_e)^2 & \text{if } Y_e/Z_e \text{ is numerical} \\ H_e \ (1 \ or \ 0) & \text{if } Y_e/Z_e \text{ is categorical} \\ Y_e \cdot Z_e & \text{if } Y_e/Z_e \text{ is of normalized vector} \end{cases} \tag{4.21}$$

4.2 Coefficient Metric Computation In GruBa, coefficient value serves as the fundamental criteria for the optimal clustering selection. Providing a clustering

result, each user is associated with a cluster. For a given user u of cluster l, we employ the vector Y to denote the *Feature Data* as in formula 3.16.

DEFINITION 6. *The in-cluster distance $d_i(u)$ is the average distance to all the other users in the same cluster, i.e., $\forall u'' \in l$ & $u \neq u''$:*

$$d_i(u) = Avg\{D_g(Y_u, Y_u'')\} \tag{4.22}$$

DEFINITION 7. *The out-cluster distance $d_o(u)$ is measured as the minimum of the distances $\{d^*\}$ between u and other clusters ($\forall l' \neq l$), i.e.:*

$$d_o(u) = Min\{d^*(u, l')\} \tag{4.23}$$

where d^* is given by:

$$d^*(u, l') = Avg\{D_g(Y_u, Y_u')\}\ \forall u' \in l' \tag{4.24}$$

DEFINITION 8. *The coefficient value $v(u)$ is thus concluded:*

$$v(u) = \frac{(d_o - d_i)}{max(d_o,\ d_i)} \tag{4.25}$$

Intuitively, a good clustering solution should result bigger d_o and smaller d_i, such that samples with obvious differences go to various clusters and vice versa. When d_o is far more than d_i, coefficient value approaches to 1. Hence, the larger coefficient value is, the better clustering performs, by which the optimal solution is selected.

5 Group Modeler

Recall the central problem of GruBa, where the retweeting behaviors of users are modeled. Specifically, such model is built by Group Modeler for each user group and thus named as group model. To avoid ambiguity, we shall use the term of *items* to denote the data for training the group model. A given *item* is either positive or negative.

DEFINITION 9. *An item E involves a blog b and a user f such that $f \in R_s(O(b))$, i.e., f is a follower of b's owner.*

$$E \in \begin{cases} positive\ items & if\ f\ retweeted\ b \\ negative\ items & if\ f\ did\ not\ retweet\ b \end{cases} \tag{5.26}$$

And the data of item E could be further divided into three parts.

- *User Part* contains a list of aforementioned metrics $\{\#R_s, \#E_s, R_{ee}, \#B_s, \#W_r\}$.
- *Blog Part* consists of a metric C_h, referring to the correlation between blog contents and recent events returned by Ring [reference]. C_h is in the form of a normalized vector with each dimension represents one event (similar as P_f in formula 3.3). Specifically, each event could be viewed as a topic t, over which the correlation of a blog b could be obtained by formula 3.12.
- *Interaction Part* includes three correlation metrics. They are of blog b versus the user u's *Interest Feature* $P_f(u)$ (a.k.a. long-term/stable interest in this work), b versus u's short-term interest $P_s(u)$ that is mined from u's recent blogs (e.g., within 30 days) in the same manner of $P_f(u)$, and b's timestamp versus the time distribution of u's retweeting behavior P_t.

As a result, the obtained group model could learn what does a positive/negative item look like over each metric mentioned above.

6 Performance Evaluation

In this section, we validated the effectiveness of our methods, including features extraction, users clustering and reposting behavior modeling.

6.0.1 Experimental Setup The data we used in this study was crawled from Sina Microblog, which allows users to follow with each other. When user A follows B, A is called the follower of B and B is called the friend of A. User B's behaviors such as posting and reposting will be visible to A. User A can choose microblogs to repost which was posted (or reposted) by user B. Our data set was crawled by the following ways. Firstly, we selected 43500 users randomly from the public Microblog stream. We crawled the basic information of these users, including gender, province, number of microblogs, city, number of followers, number of friends, signature, nickname, etc. In addition, we also crawled all the microblogs the user posted or reposted. The microblog information including the number of it has been reposted, the number of it has been commented, the number of it has been liked, type of the microblog (be posted or reposted by the user), the location where the microblog was posted, the content of the microblog, the device the microblog was posted by, the time when the microblog was posted and so on. In addition, we can get part of one user's friends from the microblogs that user reposted. So we also collected some users's friends's information and microblogs. Table 4 lists the statistics information of the data set. Another dataset we used here was 1000 labeled microblogs selected from the microblogs we crawled.

We labeled each microblog with categories in the cell lexicon.

Table 4: statistics information of Dataset

DataSet	#Users	#Microblogs	#Posts	#Reposts
WeiBo	410000	157943530	77234797	80708733

Features Extraction One user's basic and behavioral information can be obtained either from the data set straightly or from the statistics information of user's microblogs. As for user's interests, we proposed a novel method to do interest extraction. Here we compared our method using different parameters with method that just using TF-IDF or Twitter-LDA. To evaluate the performance of the method, we labeled 1000 microblogs with categories of the cell lexicon. One microblog may belong to one or several categories. Considering it's a multi-label problem, our evaluation metrics here are accuracy, precision and recall as described in [**?**]. Let $S = (x_i, Y_i)|1 \leq i \leq p)$ be the test set and $m(\cdot)$ be the mapping method. The accuracy, precision and recall are defined as equation (6.27), (6.28) and (6.29).

$$accuracy_m = \frac{1}{p}\sum_{i=1}^{p}\frac{\mid Y_i \cap m(x_i) \mid}{\mid Y_i \cup m(x_i) \mid} \tag{6.27}$$

$$precision_m = \frac{1}{p}\sum_{i=1}^{p}\frac{\mid Y_i \cap m(x_i) \mid}{\mid m(x_i) \mid} \tag{6.28}$$

$$recall_m = \frac{1}{p}\sum_{i=1}^{p}\frac{\mid Y_i \cap m(x_i) \mid}{\mid Y_i \mid} \tag{6.29}$$

Users Clustering We used the data generated by features extraction to perform users clustering. Each sample contains user's basic features, behavioral features and interests features. Silhouette coefficient [**?**] was used as our evaluation metric here. For each sample i, let $a(i)$ be the average dissimilarity of i with all other data within the same cluster. Let $b(i)$ be the lowest average dissimilarity of i to any other cluster, of which i is not a member. We now get the silhouette of sample i as equation (6.30). Then silhouette coefficient of the result with N samples can be got as equation (6.31).

$$s(i) = \frac{b(i) - a(i)}{max\{b(i), a(i)\}} \tag{6.30}$$

$$silhouette = \frac{1}{N}\sum_{i=1}^{N} s(i) \tag{6.31}$$

Reposting Behavior Modeling For analyzing reposting behavior, we firstly divided users into groups by users clustering. For each group, we randomly selected microblogs reposted by users of the group as positive samples, and selected microblogs ,which were reposted by a friend of one user in the group but not reposted by this user, as negative samples. Then we generated final samples with features described in subsection **??**. We have carefully chosen algorithm to compare with our method. Here we compared our method with [20], which studied a state-of-art idea of social influence locality for modeling users reposting behaviors in the microblog network. We use accuracy, recall and f-measure as our evaluation metrics.

All experiments were conducted on a machine with 2 Intel Xeon E5C2630 2.4GHz CPUs and 64 GB of Memory, running 64 bit Windows 7 professional system. Each experiment was repeated 5 times, and the average is reported here. In all the experiments, we fixed parameters to their default values. The parameters with their descriptions and default values are presented in Table 5.

Table 5: Parameters used in the experiments with their descriptions.

Parameters	Descriptions	Default
ϵ	threshold for selecting the categories when considering the number of overlapped words.	3
α	weight for mapping value by performing Procedure 1	0.7
k	threshold number for selecting categories when performing mapping algorithm.	3
k_1	min number of clusters	2
k_2	max number of clusters	10

6.0.2 Performance and Analysis

Features Extraction To accurately extract user's interests feature, we proposed a novel method that combined the method of Twitter-LDA [23] and TF-IDF to tag the microblog with 512 categories. We performed an experiment to compare three methods which only using Twitter-LDA, only using TF-IDF, and combining the two with different value of α. Fig 2 shows the results of the experiment. From the results, we can see that when combining the two, we got a better performance. We also developed a demo system for visualization of users's features, fig. 3 shows one case of a user.

User Clustering We performed users clustering on the 43500 users we selected through the method described in subsection **??**. In addition, we reconstructed

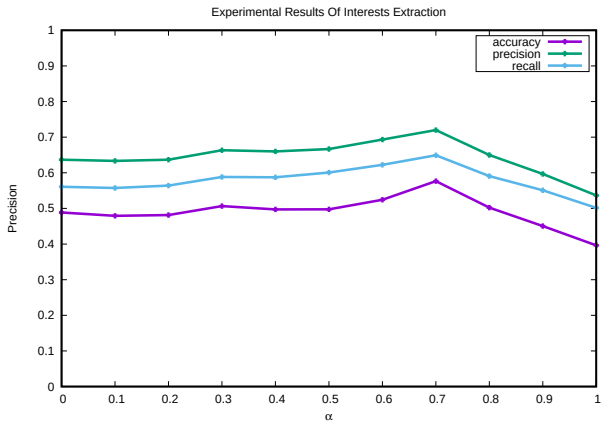


Figure 2: Visualization for User Features.

Figure 3: Visualization for User Features.

five datasets which were randomly selected from the 43500 users and each dataset had 10000 users. The results are list in Fig. 4. From the results, we can draw the following observations: (1) On the whole dataset with 43500 users, the clustering method got its largest coefficient when the number of clusters was four. (2) On the five datasets with 10000 users, the clustering method got its largest coefficient on four datasets when the number of clusters is four and one dataset when the number of clusters is three. Based on the observations, we thought it a reasonable result dividing 43500 users into four groups.

We carefully analyzed the four groups we obtained through users clustering. (1)Most of users of the group one are male, and they are mainly distributed in Beijing. What's more, most of these users have similar amount of friends and followers. The most active time of this group are mainly distributed between 10 am and 12 am. (2)The users of group two are mostly female, they are also mainly distributed in Beijing. Most of users also have similar amount of friends and followers as group one. The most active time of this group are mainly distributed between 22 pm and 24 pm. (3)The users of group three are mostly male, and they are distributed in a wide range of provinces. Most of users have the number of followers far more than friends. Users of this group are active both in 10-12 am and 22-24 pm. (4)Most of users of group four are female, and they are distributed in a wide range of provinces as group three. Most of these users have the number of followers far more than friends. And the most active time of this group are mainly distributed between 22 pm and 24 pm. We also found that the distribution of the number of users's microblogs of each group is similar. We visualized the users's interests using words cloud, the difference between groups is obvious, excepting that group one and group three have similar interests.

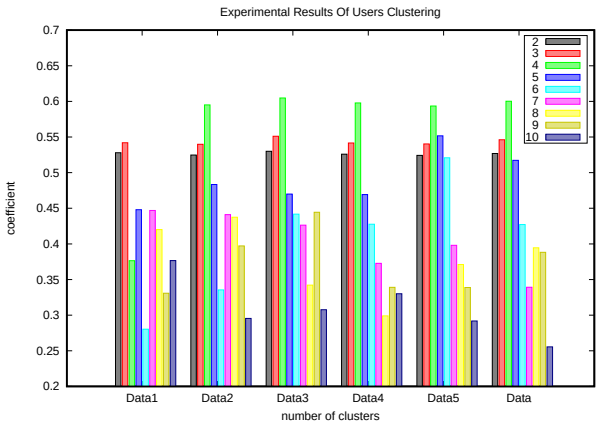


Figure 4: The Results of Users Clustering Performed on Six Datasets

Reposting Behavior Modeling By performing users clustering, we got four user groups. We generated instances by the method as described in subsection **??**. However, we observed that positive instances (reposting) and negative instances (un-reposting) are much unbalanced. Thus we randomly select a subset

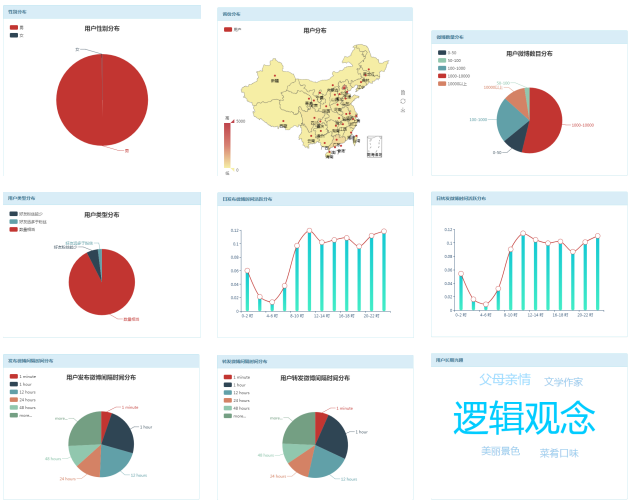


Figure 5: The Statistics of User Group One

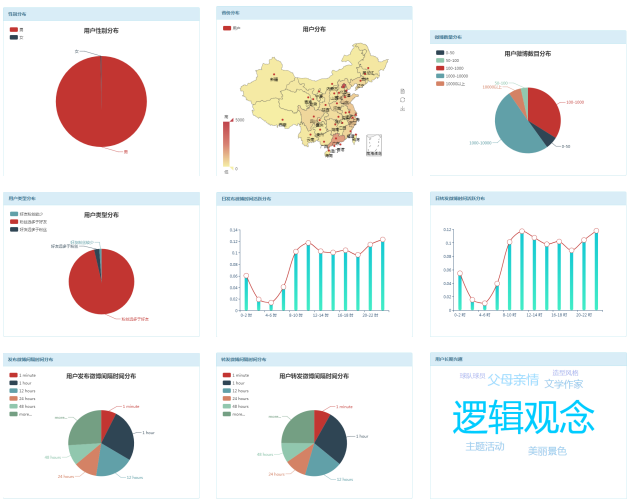


Figure 7: The Statistics of User Group Three

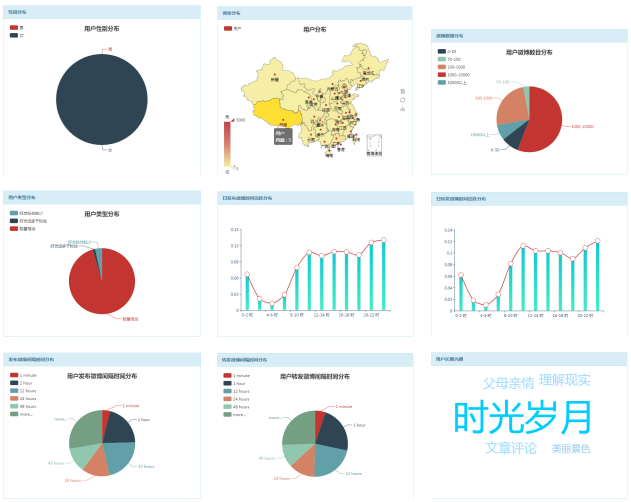


Figure 6: The Statistics of User Group Two

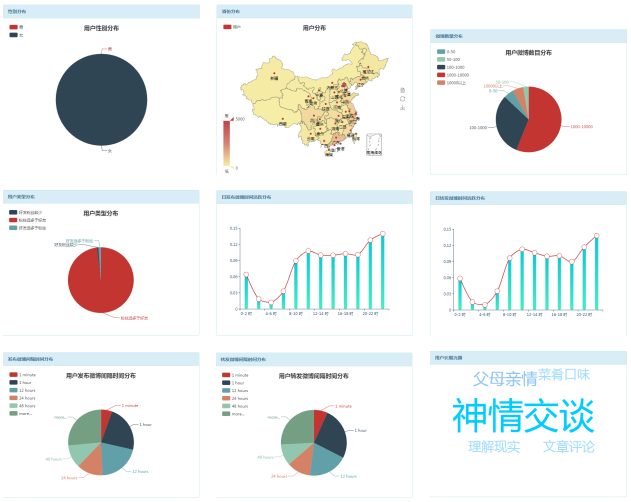


Figure 8: The Statistics of User Group Four

of negative instances with the equal number of positive number for experiments. We compared our method with [20], the results are list in figure 9, here we call our method as reposting model for group(RMG). The result suggests that our method had a better performance than LRC-BQ in most cases.

We further try different features and their combinations for modeling reposting behaviors, they are user-based information(UI), microblog-based information(MI), interaction information(II) and their combinations. The results are list in in figure 10. The result suggests that the MI and UI perform similarly for modeling group one and group three, MI performs better than UI and II on group four. Combining all the three types of features can get similar performance with the combination of UI and MI on group two and group three, and the two combinations perform better than all other combinations. As for group one and group four, using UI and MI can get the best performance, which indicates that there in deed exists strong correlation between the reposting behavior and the two types information.

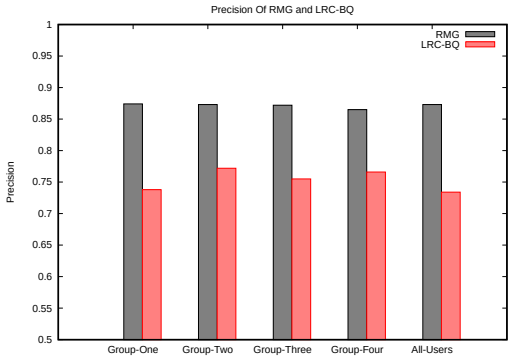
Time Report

7 Related Work

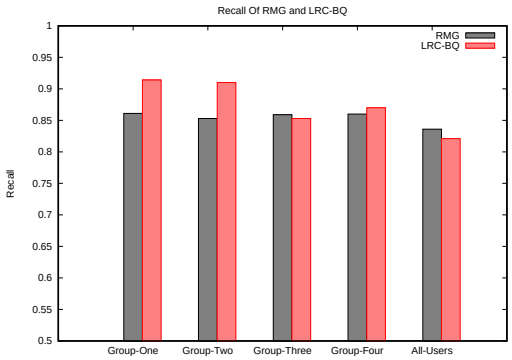
The rapid development of social network is accompanied by the generation of tremendous user-generated contents. Message forwarding (e.g., reposting on weibo.com) is one of the most popular functions in many existing social networks. Behavior modeling of social media users has received great attention in recent years, and has become a research hot spot of academies and industries. In this section, we will introduce the related work of user modeling in social network from the aspects of user features analysis, user groups mining and user behaviors modeling.

Social media users's features we mentioned here include users's basic information, behavior features and interests features. Users's basic information include the users's gender, age, region, occupation and other personal information. There are many researches for analyzing users's basic information, such as users's racial information analysis [14], users's gender inference [2], users's actual age inference [**?**], user political tendency analysis (e.g. [14], [17]), user policy orientation analysis[10], user's geo-location and occupation mining[3, **?**], etc. Most of these methods analyze the unknown attributes by using classification or regression model.

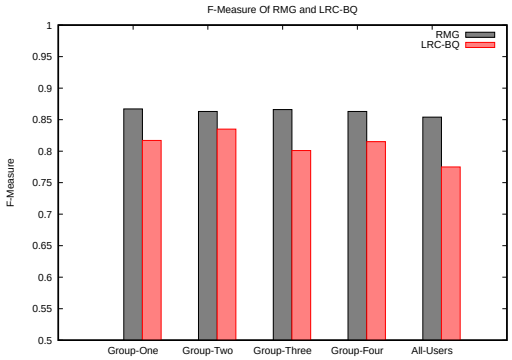
Social medias users's behaviors mainly refers to the reposting, posting and commenting behaviors. Users's behaviors also have certain characteristics and regularity. As mentioned in [9], it found that the behavior of



(a) Precision Of RMG and LRC-BQ

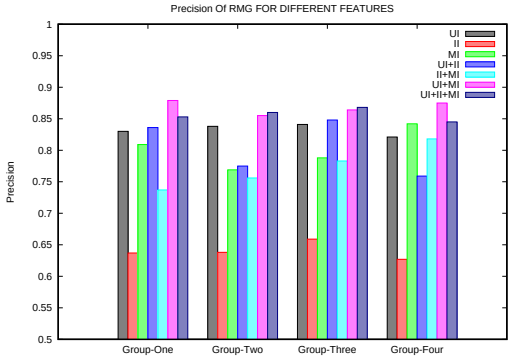


(b) Recall Of RMG and LRC-BQ

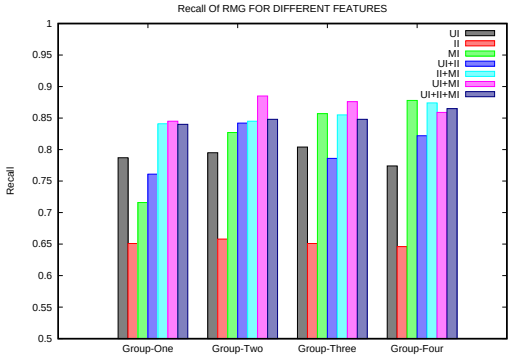


(c) F-Measure Of RMG and LRC-BQ

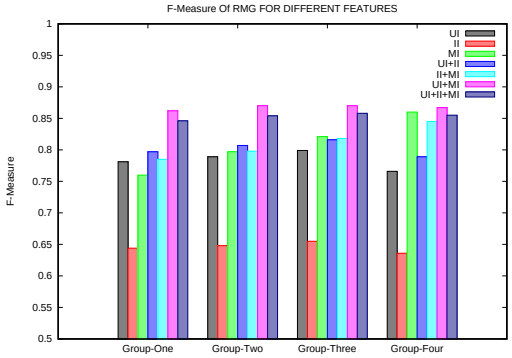
Figure 9: Comparison Of RMG and LRC-BQ



(a) Precision of RMG Using Different Features



(b) Recall of RMG Using Different Features



(c) F-Measure of RMG Using Different Features

Figure 10: Performance Of RMG Using Different Features

a user exhibits the power-law distribution at the time interval, and the power-law distribution is related to users's schedule. Guo Z et al.[5] analyzed the behavior of microblog users, the difference between the activity of users in different periods, and obtained the distribution of individual behavior on time.

Users's behaviors are largely driven by user interests, so it's of vital importance to model users's interests. Zhiyuan et al.[12] modeled users's interests through mining keywords. They extracted keywords from the users's microblogs by the combination of words frequency and translation model. Xu Z et al.[18] proposed a method which extended user topic model to analyze users's interests. Michelson M et al.[13] analyzed interests based on a knowledge base. They used a knowledge base to identify and classify the entities in twitters of one user, then generate the user's interests category subtree to express his interests. Lim K H et al.[11] analyzed the celebrities a user mentioned, then they got the preference degree of the user in different interests categories. Bhattacharya P et al.[1] proposed a method to extract a user's interests by analyzing the experts he followed. By digging a list of certain topics of the custom experts the user follows, they got the user's interests profiling. Wei Feng et al.[**?**] studied the methods mapping tweets to hashtags to get users's preferences for hashtags.

Considering the fact that the amount of users in Social Medias is so huge, modeling for each user is not a good idea. In addition, modeling for a single user may make the model too particular. There are also some studies on the user groups analysis. Some researches studied how to classify users under a specific situation. For instance, Marco Pennacchiotti et al.[14] classified users by race, political tendencies and so on. In addition, research for user community mining is a hot spot. Jaewon Yang et al.[19] proposed a method based on non-negative matrix decomposition to mine user community. Yiye Ruan et al.[15] considered user's friends and user's text content into their method when measuring the similarity between users for clustering. He et al. [6] only considered the relationship between friends, and used the edge aggregation coefficient as a measure of clustering. Hiroaki Shiokawa et al.[16] used the modular degree as a clustering standard and proposed an incremental algorithm to mine user community.

As for behavior prediction, Jing Zhang et al.[21] proposed a method of analyzing users's reposting behavior from the perspective of influence. Some researchers (e.g. [4], [20]) find that the user's reposting behavior is largely influenced by the relationship between friends, so they consider user's friend structure information into their models. Bo Jiang et al. [8] proposed a method that

combined matrix decomposition and microblogs clustering to analyze the user's reposting behavior. Zhang et al.[22] proposed non-parametric statistical models to combine structural, textual, and temporal information together to predict reposting behavior. Considering the situation that users's not reposting does not mean that users are not interested in it, the users may just did not see it, [7] applied a collaborative filtering method to the reposting behavior analysis.

8 Conclusions

In this paper, we studied to model reposting behavior of group users in microblog. We first extracted users's features including basic features, behavior features and interests features. Meanwhile, we proposed a novel method to extract user interests from the user generated texts. By performing users clustering, we divided users into several groups and modeled reposting behavior of each group users respectively. The final experiment shows that our model have strong predictive power. What's more, we developed a demo system to display the profiling for users and groups. As future work, it is interesting to study other methods to improve the accuracy of interests extraction.

Acknowledgment

References

[1] P. Bhattacharya, M. B. Zafar, N. Ganguly, S. Ghosh, and K. P. Gummadi. Inferring user interests in the twitter social network. In *Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014*, pages 357–360, 2014.

[2] M. Ciot, M. Sonderegger, and D. Ruths. Gender inference of twitter users in non-english contexts. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1136–1145, 2013.

[3] Q. Fang, J. Sang, C. Xu, and M. S. Hossain. Relational user attribute inference in social media. *IEEE Trans. Multimedia*, 17(7):1031–1044, 2015.

[4] W. Feng and J. Wang. Retweet or not?: personalized tweet re-ranking. In *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*, pages 577–586, 2013.

[5] Z. Guo, Z. Li, H. Tu, and L. Li. Characterizing user behavior in weibo. In *Third FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing, MUSIC 2012, Vancouver, Canada, June 26-28, 2012*, pages 60–65, 2012.

[6] C. He, H. Ma, S. Kang, and R. Cui. An overlapping community detection algorithm based on link clustering in complex networks. In *Military Communications Conference (MILCOM), 2014 IEEE*, pages 865–870. IEEE, 2014.

[7] B. Jiang, J. Liang, Y. Sha, R. Li, W. Liu, H. Ma, and L. Wang. Retweeting behavior prediction based on one-class collaborative filtering in social networks. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 977–980, 2016.

[8] B. Jiang, J. Liang, Y. Sha, and L. Wang. Message clustering based matrix factorization model for retweeting behavior prediction. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1843–1846, 2015.

[9] Z. Jiang, Y. Zhang, H. Wang, and P. Li. Understanding human dynamics in microblog posting activities. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(02):P02006, 2013.

[10] M. Kosinski, D. Stillwell, and T. Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.

[11] K. H. Lim and A. Datta. Interest classification of twitter users using wikipedia. In *Proceedings of the 9th International Symposium on Open Collaboration, Hong Kong, China, August 05 - 07, 2013*, pages 22:1–22:2, 2013.

[12] Z. Liu, X. Chen, and M. Sun. Mining the interests of chinese microbloggers via keyword extraction. *Frontiers of Computer Science in China*, 6(1):76–87, 2012.

[13] M. Michelson and S. A. Macskassy. Discovering users' topics of interest on twitter: a first look. In *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data, AND 2010, Toronto, Ontario, Canada, October 26th, 2010 (in conjunction with CIKM 2010)*, pages 73–80, 2010.

[14] M. Pennacchiotti and A. Popescu. A machine learning approach to twitter user classification. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*, 2011.

[15] Y. Ruan, D. Fuhry, and S. Parthasarathy. Efficient community detection in large networks using content and links. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 1089–1098, 2013.

[16] H. Shiokawa, Y. Fujiwara, and M. Onizuka. Fast algorithm for modularity-based graph clustering. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA.*, 2013.

[17] S. Volkova, G. Coppersmith, and B. V. Durme. Inferring user political preferences from streaming communications. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL*

2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers, pages 186–196, 2014.

[18] Z. Xu, R. Lu, L. Xiang, and Q. Yang. Discovering user interest on twitter with a modified author-topic model. In *Proceedings of the 2011 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2011, Campus Scientifique de la Doua, Lyon, France, August 22-27, 2011*, pages 422–429, 2011.

[19] J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*, pages 587–596, 2013.

[20] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li. Social influence locality for modeling retweeting behaviors. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 2761–2767, 2013.

[21] J. Zhang, J. Tang, J. Li, Y. Liu, and C. Xing. Who influenced you? predicting retweet via social influence locality. *TKDD*, 9(3):25:1–25:26, 2015.

[22] Q. Zhang, Y. Gong, Y. Guo, and X. Huang. Retweet behavior prediction using hierarchical dirichlet process. In *AAAI*, pages 403–409, 2015.

[23] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *European Conference on Information Retrieval*, pages 338–349. Springer, 2011.