Whom to Mention: Expand the Diffusion of Tweets by @ Recommendation on Micro-blogging Systems

Beidou Wang†, Can Wang†, Jiajun Bu†, Chun Chen†, Wei Vivian Zhang♯, Deng Cai‡, Xiaofei He‡

†Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, China
♯Microsoft Corporation, Redmond, WA, USA
‡State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, China
†{beidou,wcan,bjj,chenc}@zju.edu.cn, ♯wzha@microsoft.com,
‡{dengcai,xiaofeihe}@cad.zju.edu.cn

ABSTRACT

Nowadays, micro-blogging systems like Twitter have become one of the most important ways for information sharing. In Twitter, a user posts a message (tweet) and the others can forward the message (retweet). *Mention* is a new feature in micro-blogging systems. By mentioning users in a tweet, they will receive notifications and their possible retweets may help to initiate large cascade diffusion of the tweet. To enhance a tweet's diffusion by finding the right persons to mention, we propose in this paper a novel recommendation scheme named as *whom-to-mention*. Specifically, we present an in-depth study of *mention* mechanism and propose a recommendation scheme to solve the essential question of whom to mention in a tweet. In this paper, *whom-to-mention* is formulated as a ranking problem and we try to address several new challenges which are not well studied in the traditional information retrieval tasks. By adopting features including user interest match, content-dependent user relationship and user influence, a machine learned ranking function is trained based on newly defined information diffusion based relevance. The extensive evaluation using data gathered from real users demonstrates the advantage of our proposed algorithm compared with the traditional recommendation methods.

Categories and Subject Descriptors

H.3.5 [**Online Information Services**]: Web-based services

General Terms

Theory

Keywords

Micro-blogging Systems; Twitter; Mention; Recommendation; Information Diffusion; Information Retrieval

1. INTRODUCTION

With more than 140 million active users and over 340 million messages posted per day, Twitter has become one of the most influential media for spreading and sharing breaking news, personal updates and spontaneous ideas. In Micro-blogging systems like Twitter, users tweet about any topics within the 140-character limit and follow others to receive their tweets. Furthermore, with *retweeting* (forward a tweet), information can be effectively relayed beyond adjacent neighbors, virtually giving every user the power to spread information broadly.

However, recent studies [2][31][37] show that the diffusion power of tweets from different users varies significantly: 0.05 percent of Twitter users attract almost 50 percent of all attention within Twitter and the spread of a tweet from an ordinary user is rather limited, with an average retweet rate of 0.11. This suggests a very limited diffusion for most tweets.

Fortunately, as a new feature in Micro-blogging systems, *Mention* can help ordinary users to improve the visibility of their tweets and go beyond their immediate reach in social interactions. *Mention* is enabled in a tweet by adding "@username". All the users mentioned by a tweet will receive a mention notification (*e.g.* by an e-mail) and are able to retrieve the tweet from their personal mention tab. By using *Mention*, one can draw attention from a specific user, or highlight a place or organization anytime. Properly using mention can quickly help an ordinary user spreading his tweets:

1. By mentioning a non-follower of the tweet author, the non-follower may retweet it to his followers and spread the tweet to a new group of users, which usually leads to further cascade diffusion.

2. By mentioning a follower of the author, the mention serves as a useful notification, especially when the follower follows a large number of other users and a tweet can be easily swamped in the enormous number of tweets. It's also critical for a tweet to be viewed promptly as 25% replies to a tweet happen within 67 seconds, 75% within 17 minutes and 75% message flow lasts less than an hour [35]. So, without proper notification, a tweet may easily be neglected as one's followers fail to read it in time.

Despite the significance of the mention feature, to the best of our knowledge, *Mention Recommendation* is seldom studied in previous works. To better help an ordinary user spreading their thought in Micro-blogging systems, we propose in this paper a novel Mention Recommendation algorithm named *whom-to-mention*, in which we help a tweet to reach more

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

people by recommending proper users to be mentioned before publishing it.

The recommendation task can be formulated as a ranking problem. Traditionally, one can rank users based on the similarity between a tweet and a user's profile (e.g. the aggregation of all the tweets posted by a user) and recommend the top ranked users to be mentioned. However, there are several challenges which make the traditional recommendation methods fail:

Information Diffusion Based Relevance: In classic information retrieval tasks (e.g. TREC *adhoc* retrieval tasks), relevance is usually interpreted as topical relevance, which stands for to what extent the topic of a result matches the topic of the query. However, the goal of *mention recommendation* is to find candidates who can help spread a tweet. Instead of topical relevance, the information diffusion power should be considered in the relevance judgement.

Content-dependent User Relationship Model: In traditional social network recommendation, user relationship is usually modeled as a weighted graph with edges indicating the bonds between two users based on explicit social relationship. The interactive functions (e.g. retweet, reply, mention) in micro-blogging systems allow us to adopt the implicit network derived from user's interactive behaviors to achieve more precise user relationship predictions. Moreover, it brings in new features for modeling user relationship, as users' interactions are usually content (topic) related, which makes the user relationship model content-dependent. For instance, a user may selectively retweet sport news from another user while ignoring other contents such as movie comments from the same user. Modeling the content-dependent user relationship based on the implicit network of user interactions thus remains as a challenge.

Recommendation Length Restriction: Due to the strict length restriction of a tweet, only a small number of users can be mentioned in a tweet. Moreover, a tweet mentioning a lot of users is likely to be treated as a spam tweet, which will decrease others' interest in retweeting it. Thus, to accomplish the mention recommendation task, the algorithm needs to be optimized for mentioning only a small number of users.

Recommendation Overload Problem: Traditional recommendation systems such as those used in Amazon may recommend one item to large numbers of users, which results in popular products. However, in the mention recommendation system, a user being recommended too many times will suffer from the severe mention overload problems. Tons of mention notifications will not only interrupt user‘s daily use of microblogs, but also result in frustration and decrease user's interest in retweeting.

To cope with all the above mentioned challenges, *whom-to-mention* is proposed in this paper. We use a machine learning approach to train a ranking model which consists of three parts: ranking features, relevance and a ranking function [10]:

We adopt a series of new *features* to deliver more precise mention recommendation, including: the match of the given tweet and interest profiles of the recommended users, the user relationship between the recommended users and the author of the tweet, and the influence of the recommended users. Furthermore, we manage to model user relationship based on the implicit network derived from user retweet interactions, which we name as *user social ties* model. We take advantage of the content-dependent feature of user social ties and make use of the content feature of the tweets one user has retweeted from another in a user social tie.

Instead of the classic topical relevance model, the relevance in *whom-to-mention* is redefined as the potential diffusion a user may bring to a tweet, estimating by the expectation user coverage, which will be further explained in section 4.2.

A Support Vector Regression (SVR) based ranking function is then trained to calculate the relevance of a candidate user to a tweet and ranks the most relevant candidates on the top of the recommendation list. Constraints are carefully designed in the ranking process to avoid the recommendation overload problem.

It is worthwhile to highlight the following three aspects of our *whom-to-mention* recommendation scheme in this paper.

1. We present the first in-depth study of mention feature in microblogs by resolving the most essential problem of whom to mention. Instead of passively waiting to be retweeted by others, our novel recommendation scheme allows users to improve the diffusion of their tweets by reaching out to the right person with the help of mention recommendation.

2. We formulate the mention recommendation as a ranking problem and to find the most proper users to be mentioned, a ranking function is learned with a novel information diffusion based relevance, incorporating with new features including user interest match, user social ties and user influence. We model user relationship based on the implicit network derived from user's retweet interactions and take fully exploit of its content-dependent features.

3. Our method is thoroughly evaluated on a real life dataset. *Whom-to-mention* algorithm is proved highly effective compared against a large number of baselines. We analyze how different features affect the recommendation performance with aborative designed comparison experiment. New issues like recommendation length restriction and recommendation overload problem is careful evaluated and discussed.

2. RELATED WORK

2.1 Recommendation Approaches

Using information retrieval approaches to recommend documents, users or items has been a fertile area of research. Content-based recommendation systems like [25][27], recommend items similar to those that a user liked in the past. Though the use of information retrieval on recommendation has been studied for a long time, new studies keep emerging to solve all kinds of new challenges [33]. For instance, Diaz *et al.* make the first in-depth study of information retrieval approaches applied to match-making systems and study unique problems like two-sided and subjective relevance[10]. In our work, the information diffusion based relevance, new features like *user social ties* , new challenges like the recommendation length restriction and overload problem all make our *whom-to-mention* different from previous information retrieval approaches.

Most of the prior work on social network recommendation mainly focuses on recommending interesting users or contents [11]. Hsu *et al.* address the problem of link recommendation in weblogs and similar social networks by proposing an approach based on collaborative recommendation using the link structure of a social network and content-based recommendation using mutual declared interests [17]. Chen *et al.* study people recommendations designed to help users find known, offline contacts and discover new friends on social networking sites [7]. Guy *et al.* study personalized recommendation of social software items and make a comparison between recommendations that are based on the user's familiarity network and his similarity network[13]. None of previous works can be directly applied to *whom-to-mention* task and solve all the new challenges.

2.2 Learning to Rank

Learning to rank has been a popular research area. Existing approaches can be roughly divided into three categories: pointwise approaches[21][26] in which the learning-to-rank problem can be approximated by predicting the score of a single query-document pair and various regression and ordinal regression algorithms can be adopted in this kind of approaches; pairwise approaches [4][5], in which the ranking problem is reduced to pairwise classification and the goal is to minimize average number of inversions in ranking; listwise approaches [28][36], in which the value of the evaluation measures is optimized directly, averaged over all queries in the training data. The ranking algorithm from our work belongs to the pointwise approaches.

2.3 Studies on Micro-blogging Systems

With the launch of Twitter in 2007, microblogs become highly popular and large numbers of researches have been done. Our research is involved with structure and user relationship analysis of microblogs, user interest modeling, recommendation, information diffusion and influential users identification on micro-blogging systems.

The characteristics of network structure and user relationship of microblogs have attracted much attention in the past few years. Kwak *et al.* make the first quantitative study on the information diffusion on Twitter by studying the topological characteristics of Twitter and provide lots of statistic details of Twitter[20]. Besides the network based on user's explicit following network of Twitter, analysis of the users' interactions in the implicit network of Twitter has been an emerging area [30][18]. Sousa *et al.* analyze replies of a specific Twitter dataset and a slight tendency for people selectively choosing whom to reply based on the topic of the tweets is found [30]. Jang *et al.* propose an egocentric semantic social network based on user reply interactions on Twitter, but the strength of user relationship is not considered and user bonds from different user pairs are incomparable[18].

A lot of works have been done on user interest modeling, Michelson *et al.* discover users' topics of Twitter by categorizing the entities in the tweets and developing a user profile by adopting the categorization result[23]. Hong *et al.* evaluate how the restricted length of the tweets can limit the potential of traditional topic models and the authors also show that training a topic model on aggregated messages can help to significantly enhance the experiment performance[16].

Information diffusion and influential user identification on Twitter have been extensively studied. Ye *et al.* first explore the propagation patterns of general messages on Twitter and how to measure social influence on Twitter [35]. Bakshy *et al.* study the attributes most relevant to the influence of Twitter users [3]. Cha *et al.* make an in-depth comparison of three measures of influence: indegree, retweets, and mentions and investigate the dynamics of user influence across topics and time [6]. Bakshy *et al.* find that predictions of which particular user will generate large cascades are relatively unreliable and word-of-mouth diffusion can only be harnessed reliably by targeting large numbers of potential influencers [2]. Romero *et al.* model the global influence of a node on the rate of diffusion through the network based on a Linear Influence Model [34].

Several researches have focused on recommending who to follow or what to read on Twitter. Hannon *et al.* recommend Twitter users to follow using content and collaborative filtering approaches [15] and Chen *et al.* recommend interesting content from information streams on Twitter considering features including content sources, topic interest models for users, and social voting [8]. To the best of our knowledge, recommendation on whom to mention in Twitter has never been studied in previous work.

3. PROBLEM DEFINITION

We formalize *whom-to-mention* into a retrieval scenario consisting of a set of users, U, each of whom maintains a user interest profile and a user influence profile. For a user $u \in U$, a user interest profile r_u, consists of a set of descriptive attributes and tf-idf features extracted from a modified bag of words model used on u's recent tweets. A user influence profile s_u is made up of attributes related to user's influence on Twitter. For users $u, v \in U$, there exists a social tie $tie_{u,v}$ based on the retweeting interactions between u and v, which includes a scalar attribute indicating the strength of bonds between u and v and tf-idf features extracted from the tweets u retweets from v. A query q consists of tf-idf features extracted from a specific tweet.

For each query (tweet) q from user u, we would like to rank all the other users $v \in U - u$ based on features including user interest match, user social ties and user's influence, so that the relevant candidates occur above non-relevant candidates.

4. RECOMMENDING WHOM TO MENTION BY LEARNING A RANKING MODEL

The key of *whom-to-mention* is to rank the candidate users given a specific tweet and we use a machine learning approach to train a ranking model for our recommendation task, which is made up of three parts: ranking features, relevance and a ranking function[10]. Ranking features include all the attributes which may influence the score of a candidate match. Based on our recommendation task, relevance refers to the potential diffusion a user could bring to a specific tweet. A ranking function is a machine learning model which predicts the relevance given observable ranking features. We will discuss the details of the three parts in this section.

4.1 Ranking Features

4.1.1 User Interest Match

The match of a tweet and the candidate's interest is an intuitively important feature for *whom-to-mention*. When mentioning a candidate in a tweet, a candidate interested in it is more likely to retweet it.

To calculate the match, the largest challenge is to generate the user interest model on micro-blogging systems, which differs from traditional user interest models because users' tweets are limited to only 140 characters in length, covering a wide variety of topics, as well as often presented with shorthands and special formats such as hash tags. Moreover, the nature of our recommendation task requires capturing more detailed aspects of interest. For instance, a football fan may be assumed interested in sports based on topic modeling technics like LDA. However, it is not a good interest match, if we mention the football fan in a tweet talking about a basketball match (because the tweet is also considered talking about sports, which makes a match for the tweet and the candidate).

Based on previous studies [16] , topic modeling techniques like LDA may not fit the short-length, ambiguous, noisy data feature in Twitter. Consequently, we use a modified bag-of-words model to generate the user interest model.

To begin with, we aggregate a user's recent tweets. For a candidate user u, we define d_u as the set of recent tweets for u; in this work, we will assume that d_u is u's 1000 most recent tweets. We also extract the words from the *hash tag topics*, which we name as h_u and they are important because they are usually used to identify a topic or an event. Besides the tweets, we also consider all the attributes from the user profile page, including user's full name, the location, a short biography and tags. For a user u, we choose the short biography feature f_u and tag feature tag_u for the interest modeling. A user interest profile r_u is then defined as $r_u = \{d_u, h_u, f_u, tag_u\}$ and $R = \{r_u | u \in U\}$. In this way, R provides us the basis for user interest modeling.

To cope with the short noisy text , we first analyze around 50,000 hot short queries (popular words or phrases) based on a latest search engine query log covering a lot of new words and words in short-hand format and we denote these words as *Dict*. In this way, popular phrase like "Big Bang Theory" is considered as a word in *Dict*. We filter the text in R, eliminating all the stop words, only keeping a word if it's either identified as a noun or a word from *Dict*.

The name entity recognition for tweets is conducted with the help of ICTCLAS [1] (a toolkit used for word split and name entity recognition) and the query log is provided by Sogou [2] (a leading search engine company in China).

Given a query (tweet) q_u from user u, we apply the same word parsing strategy as mentioned above and represent q_u and R as tf.idf-based term vectors, which are further used to estimate the user interest match. With the help of Lucene, a proven, robust and scalable indexing and retrieval platform, the match score between a query q_u and a user interest profile r_v can thus be defined as:

$$iscore(q_u, r_v) = coord(q_u, r_v) \cdot queryNorm(q_u) \cdot \sum_{t \in q_u} (tf(t \in r_v) \cdot idf(t)^2 \cdot norm(t, r_v)) \tag{1}$$

The $tf(t \in r_v)$ correlates to the term's frequency :

$$tf(t \in r_v) = \sqrt{n_{t,r_v}} \tag{2}$$

where n_{t,r_v} is the frequency of term t in r_v and normalization of the document length is defined in $norm(t, r_v)$ for efficiency consideration.

$idf(t)$ stands for the *Inverse Documentary Frequency* defined as:

$$idf(t) = 1 + log(\frac{|R|}{|r : t \in r|}) \tag{3}$$

$norm(t, r_v)$ is a normalization factor defined as:

$$norm(t, r_v) = lengthNorm(r_v) \cdot boost(t) \tag{4}$$

where $lengthNorm$ is a length normalization factor which ensures short document contributes more to the score. We also consider boost factors that terms from different sources own different weights (e.g. a term from tag_u is more important than one from d_u). According to evaluation on training data, we set the boost $boost(t)$ as:

$$boost(t) = \begin{cases} 2 & \text{if } t \in h_u, f_u, tag_u \\ 1 & \text{if } t \notin h_u, f_u, tag_u \end{cases} \tag{5}$$

$coord(q_u, r_v)$ is a score factor based on how many query terms are found in document r_v and $queryNorm(q)$ is a normalization factor used to make scores between queries comparable. They are implemented using Lucene's function which details can be found here [3].

4.1.2 User Social Tie Modeling

User relationship plays an important role in *whom-to-mention* task, an acquaintance is usually more likely to retweet compared with a total stranger. Previous studies [20][19] mainly study explicit social connections based on the follow relationship of Twitter. However, according to a study of Facebook [1], people only communicate with a few of their explicit declared friends. So modeling user relationship based on some implicit networks can be better indicators of the actual social relationships between users.

In our work, user relationship model is based on implicit connections derived from users' retweet activities in micro-blogging systems, which we name as *user social tie model*. Though lots of work on retweet behaviors have been done, they are usually in the information diffusion perspective instead of modeling user relationships [2][3][6].

We make two assumptions in modeling user social ties. First, user social ties can be derived from the retweet interactions between two users and frequency of interaction can be used to quantify the strength of a social tie. Second, the social tie between two users is content-dependent. Thus in our model, a user social tie consists of three parts: nodes

[1] http://ictclas.org/

[2] http://www.sogou.com/labs/dl/w.html

[3] http://lucene.apache.org/core/old_versioned_docs/versions/3_0_0/api/core/org/apache/lucene/search/Similarity.html

including the two users of a tie, a strength score indicating how strong two users are bonded in a tie and a content vector indicating topics the user interested in retweeting. The details are explained as follows.

For users $u, v \in U$, we define tweet set $rt_{u,v}$ as:

$$rt_{u,v} = \{tw | tw \text{ is a tweet } u \text{ retweets from } v\} \quad (6)$$

We define the social tie strength as $str_{u,v}$

$$str_{u,v} = |rt_{u,v}| \quad (7)$$

We filter $rt_{u,v}$ with the same method mentioned in section 4.1.1 and define user social tie between user u and v as:

$$tie_{u,v} = \{rt_{u,v}, str_{u,v}\} \quad (8)$$

It is important to notice that $tie_{u,v} \neq tie_{v,u}$. Given a query q_u from user u, we can calculate the user relationship score by multiplying the strength of the social tie with the similarity between q_u and $rt_{u,v}$:

$$tscore(q_u, rt_{u,v}) = str_{u,v} \cdot coord(q_u, rt_{u,v}) \cdot queryNorm(q_u) \cdot \sum_{t \in q_u} (tf(t \in rt_{u,v}) \cdot idf(t)^2 \cdot norm(t, rt_{u,v})) \quad (9)$$

All the factors in formula (9) are defined the same as in formula (1).

4.1.3 User Influence Modeling

Intuitvely, user influence is also important to the performance of our recommendation task. If two users are both likely to retweet the tweet, the more influential one could help it reach more people by initiating a larger cascade of retweet. Given a user u, we summarize a series of statistical indicators which may indicate his influence in Tabel 1.

We can define u's influence profile s_u as:

$$s_u = \{\text{Follower}(u), \text{Avg\_retweet}(u), \text{Avg\_reply}(u), \text{Avg\_coverage}(u)\} \quad (10)$$

4.2 Relevance

In traditional text retrieval tasks (*e.g.* search engine retrieval tasks), relevance always refers to the topical match between the query and the document[10]. When interpreted in this way, we can always rely on editors to manually assess the relevance based on their experience and expertise. However, when it comes to our recommendation task, editors have to compare a query (tweet) with user profiles made up of thousands of tweets and analyze hundreds of content-based user relationship bonds, which makes the process time-consuming and result inaccurate.

Instead, we can calculate the relevance based on user behavioral information. Our recommendation scheme aims to spread a tweet to more people by mentioning proper users in it. So we can define the relevance between a query (tweet) and a user as the diffusion the user brings to it. Intuitively, the diffusion can be easily estimated by how many retweets a user initiates by retweeting the tweet. However, for ordinary users, the retweet cascades of their tweets are usually very small. For instance, given a tweet, if one user can results in 3 retweets each by a user with 100 followers and another user brings it 2 retweets each by a user with 1000 followers, the latter user obviously helps it to reach more people (2000 vs. 300). Thus, it's more accurate to estimate the relevance based on the number of users a candidate helps the tweet to reach, which we name as *coverage*. We denote the relevance of query q and user v as $rel(q, v)$ and define it as:

$$rel(q, v) = \{\sum \text{Follower(u)} | u \in \text{the retweet cascades initiate by } v\} \quad (11)$$

Table 1: Statistical Indicators on Modeling User Influence

Denotation	Explanation
Follower(u)	The number of followers of user u, one of the most popular metrics on estimating a user's influence.
Avg_retweet(u)	The average number of retweets for each tweet from u.
Avg_reply(u)	The average number of replies for each tweet from u.
Avg_coverage(u)	The average number of users a tweet from u can reach. The coverage of a tweet is defined in details in section 4.2.

4.3 Ranking Function

Many machine learning models can be used as a ranking function for our *whom-to-mention* recommendation task. We adopt a machine learned ranking function based on support vector regression (SVR) , because it is a sophisticated proven regression algorithm which is adaptive to complex systems, robust in dealing with corrupted data and with a good generalization ability [32].

Given a query q_u from user u and a candidate match v, we use SVR to compute a score to serve as the relevance $rel(q_u, v)$. We define $x_{q_u,v}$ as the feature vector corresponding to the pair (q_u, v).

$$x_{q_u,v} = \{iscore(q_u, r_v), tscore(q_u, rt_v), s_v\} \quad (12)$$

The set of training data is as $\{(x_1, y_1), ..., (x_n, y_n)\}$, where $x_i \subset R^m$ stands for the feature vector for a pair of query and candidate in which m is the number of feature dimensions, and $y_i \subset R$ stands for the corresponding relevance value.

A generic SVR estimating function is with the form as:

$$f(x) = (w \cdot \phi(x)) + b \quad (13)$$

$w \subset R^m$,$b \subset R$ and ϕ stands for a nonlinear transformation from R^m to high-dimensional space. The core goal of SVR is to learn the value of w and b to minimize risk of regression.

$$Risk(f) = C \sum_{i=0}^{n} L(f(x_i) - y_i) + \frac{1}{2}||w||^2 \quad (14)$$

$L(\cdot)$ is a loss function and C is a constant used to determine penalties to estimation errors which is determined with grids search and cross-validation techniques. We experiment the performance of different kernel functions and choose kernel function with best performance (RBF kernel). Details of SVR can be found in [29].

4.4 Recommendation Overload Problem

One new issue of our *whom-to-mention* task is that the recommendation may concentrate on a few popular users, which causes mention overload (users get too many mention notifications from the recommendation system). Moreover, different users may respond differently to the overload. For instance, some users may not want to receive any mention notification from the recommender at all, while some others may feel okay even if mentioned 100 times in a day.

In our recommendation framework, we carefully cope with this problem by allowing users to freely set an up-limit of recommended times per day. After ranking phase, all the candidates with recommended times up-limit reached are eliminated and the top k of the remaining candidates are then recommended. In real application, within a day, our recommendation scheme follows a *first publish, first to choose* policy and recommend the next best candidate once a user's recommendation up-limit is reached.

In our evaluation, since our test tweets are published over a period of time, we set the up-limit for mentioning at 25, which as a matter of fact, is a quite strict constraint.

5. EXPERIMENT SETTING

We design the experiment with 4 goals:(1) To evaluate how our proposed algorithm performs compared with other base-line algorithms;(2)To test how different features we considered affect the recommendation performance;(3) To examine how different ranking functions affect the results;(4) To consider how new challenges like the recommendation length restriction and recommendation overload affect the performance of our algorithm.

The key challenge of the experiment design lies in evaluating the information diffusion (coverage of users) resulted by mentioning a user in a tweet. Instead, we make an approximate estimation using the user's retweet behavior. For example, if user A retweets a tweet t and helps t reach 500 people, it's reasonable to assume that A will retweet it if we mention A in t. So in our evaluation, by mentioning A in t, the user coverage A brings to t is 500. If user B has never retweeted t, we assume B will not retweet t when mentioning him in t and the user coverage B brings to t thus is considered to be 0.

5.1 Data Collection

We collected data from Sina Weibo, a Twitter-like microblogging system in China with more than 400 million registered users and over 100 million messages posted per day. Different from Twitter's API, which is restricted in retrieving mention and retweet timelines, Weibo's API allows us to get all the tweets from a user's different timelines. Moreover, we obtained authorizations from over 5000 real Weibo users, who grant us full access to all the authentication-protected user data, including user profiles, tweets, the retweet timeline, the reply and mention timeline, and accurate reply and retweet number for each tweet. We parse 48,000 tweets published by the authorized users, only keeping tweets being retweeted more than 5 times, which leaves us 132,796 retweet records and 7800 tweets to serve as the training and testing tweets.

Based on the retweet records, 52,468 users participate in retweeting and are considered as our recommendation candidates. We collect the most recent 1000 tweets from these users (around 46 million in total) and record their personal information including the full name, the location, user biography *etc.* Average retweet rate and relpy rate for each user are calculated based on the most recent 200 tweets (around 11 million in total). 97,164 user social ties are established based on retweet interactions. In our experiment, we split the parsed tweets into training and testing data set with an 80/20 proportion and cross-validation is used.

5.2 Evaluation Metrics

We evaluate the results using both standard information retrieval metrics[14][9] and metrics featuring on measuring information propogation[35]. In particular, we use the following metrics: precision (P), average precision at K ($AP@K$), retweet times (RT), user coverage (Cov) and normalized user coverage (Cov_N), which are defined as,

$$P = \frac{N_{hit}}{m} \tag{15}$$

$$AP@K = \frac{\sum_{i=1}^{K}(P(i))}{N_{hit}} \tag{16}$$

$$RT = \{\sum_{u \in R} |t| \ |t \in T_{t,u}\} \tag{17}$$

$$Cov = \{\sum_{u \in R} \sum follower(v) | v \in U_{t,u}\} \tag{18}$$

$$Cov\_N = \{arctan(\sum_{u \in R} \sum (follower(v))) | v \in U_{t,u}\} \tag{19}$$

where m is the size of the recommendation list, N_{hit} is the number of users in the recommendation list belonging to the top m relevant matches and $P(k)$ means the precision at cut-off k in the recommendation list. For a user u, a tweet t and the recommendation list R, we define $T_{t,u}$ as all the retweets from the retweet cascades initiated by u retweeting t and $U_{t,u}$ as all the users from the retweet cascades initiated by u retweeting t.

Retweet times stands for the number of hops in a tweet propagation and each hop increases the chance for the tweet to reach more users. User coverage is a more intuitional metric which is the cumulative number of users that a tweet has reached due to the mention recommendation. In the normalized user coverage, we normalize the coverage with an $arctan()$ function, to make the coverage number from different algorithms more comparable.

Due to the length restriction, only a limited number of users can be mentioned in a tweet and thus we set the length of recommendation list as 5 in our evaluations. We also test how the algorithm performs when we only recommend 1∼4 users to mention.

5.3 Comparison Algorithms

To the best of our knowledge, no previous studies have been done on the *whom-to-mention* task. Though the task is with lots of new challenges, we try our best to adapt several classic recommendation algorithms to this new problem to serve as baseline comparison algorithms.

- Content-based Recommendation (**CR**). A content based recommendation algorithm similar to [12] is carefully designed. User profile are based on the content of tweets and attributes from user profile page. A specific tweet is considered as an item, illustrated by its content. Both the user profile and item are modeled as tf.idf-based vectors and we recommend users by ranking the cosine similarity scores of user profile and item.

- Content-boosted Collaborative Filtering Recommendation (**CCFR**). For our task, recommendation is conducted before a tweet is published and there thus exist no user interaction behaviors like retweet and reply to serve as ratings, so the recommender is always in a cold start state. We choose Content-boosted Collaborative Filtering Recommendation[22] which copes with the cold start problem of traditional Collaborative Filtering. A tweet is viewed as an item and a candidate is regarded as a user. When a new item (tweet) needs recommendation, we find 5 most similar items from training data based on content similarity and recommend users by combining the recommendation results from the similar items.

- Bonds-based Recommendation (**BR**). In BR, we recommend candidates to a tweet based on the social bonds between candidates and the tweet author, which means the closer a candidate is linked to the author, the more likely he will be recommended. The social bond is modeled based on users' retweet interactions.

- Influence-based Recommendation (**INFR**). In INFR, we recommend candidates based on their influence, which is a linear combination of influence features mentioned in section 4.1.3. We try to recommend the most influential users to mention given a tweet.

- Random Recommendation (**RR**). In RR, 5 users are randomly chosen from the candidates to generate the recommendation list.

- Whom-to-mention with different Ranking function. To evaluate how different ranking functions affect the recommendation result, we compare the performance of WTM by using three different ranking algorithms as the ranking function, including using Support Vector Regression ($\mathbf{WTM}_{SVR}$), using Linear Regression ($\mathbf{WTM}_{LR}$) and using Gradient Boosted Decision Trees [24]($\mathbf{WTM}_{GBDT}$).

- Twitter and Weibo. Based on statistics from previous studies[2] [37], we get the average retweet rate and coverage of a tweet in Twitter. With the help of the data we collect for user influence modeling (11 million tweets from Weibo), we calculate the average retweet rate and coverage for a tweet from Weibo. These numbers show the general average diffusion of a tweet in a Micro-blogging system.

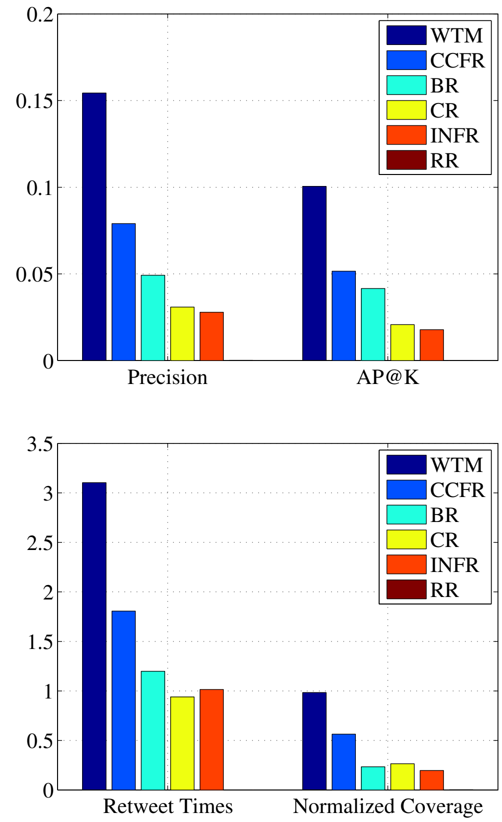

Figure 1: Performance Comparison of WTM and Baseline Algorithms

6. RESULTS AND ANALYSIS

6.1 Algorithm Performance Evaluation

As shown in table 2 and figure 1, our *whom-to-mention* approach (WTM) significantly improves the diffusion of a tweet in all the metrics. We draw the following conclusions from these results.

First, Random Recommendation (RR) barely shows any effect, which makes it clear that simply mentioning some users has little effect in improving the diffusion of a tweet. Second, the poor performance of Influence-based Recommendation (INFR) is because influential users may be neither interested in the tweet, nor share any social ties with the author. Moreover, mention notifications may be easily neglected by the influential users since they usually receive thousands of mention notifications per day. Third, Content-based Recommendation (CR), although effective, is not as good as those based on user relationships like BR and C-CFR; this is partly attributed to the noise and ambiguity existing in the tweet-based user profiles and item profiles. Fourth, the performance of Bonds-based Recommendation (BR) shows users who share strong social ties with the author are more likely to help him retweet the tweet and it is in accordance with our daily experience. Furthermore, Content-boosted Collaborative Filtering Recommendation (CCFR) shows the best performance in all of our comparison algorithms, owing to both its adoption of sophisticated CF recommendation scheme based on the implicit retweeting interaction network and the incorporation of content-based features during the recommendation.

Finally, our SVR based *whom-to-mention* recommendation (WTM) outperforms all the comparison algorithms. Even comparing with CCFR, it shows 70% increase in precision, a 94% increase in $AP@5$, an 72% increase in retweet rate and a 51% increase in normalized coverage of users. Our algorithm benefits from the exploitation of all the new features, a careful design of relevance model and a ranking function based on machine learning techniques. Moreover, our algorithms results in a 2821% and 389% increase of retweet rate and a 338% and 523% increase of coverage compared with an ordinary tweet from Twitter and Sina Weibo,

Table 2: Result Comparison of WTM and Baseline Algorithms

	WTM	CR	CCFR	BR	INFR	RR	Twitter	Weibo
Precision	**0.1343**	0.0309	0.079	0.0492	0.0279	1.47E-04	-	-
AP@5	**0.1005**	0.0207	0.0515	0.0416	0.0178	4.91E-05	-	-
Retweet Times	**3.1026**	0.9395	1.8058	1.1990	1.0147	0.0015	0.110	0.798
Normalized Coverage	**0.8525**	0.2649	0.5640	0.2349	0.1969	0.0023	-	-

Table 3: Comparison on How Different Features Affect the Performance of WTM

	ALL	NO_Interest	No_Influence	No_Ties	No_ContentInTie	Twitter	Weibo
Precision	**0.1342**	0.1328	0.1319	0.0658	0.1171	-	-
AP@5	0.1005	0.0985	**0.1129**	0.0410	0.0869	-	-
Retweet Times	**3.1026**	3.0559	3.0359	1.7540	2.6770	0.110	0.798
Coverage	**3716**	3643	3592	2185	3239	1100	711

which further confirms the effectiveness of our algorithm on boosting the diffusion of a tweet.

6.2 Feature Importance Evaluation

To analyze how features used in our proposed algorithm contribute to the learned model, we design this contrast experiment by eliminating one feature at a time and observe how the performance of our model changes. Furthermore, since we assume user social ties in micro-blogging systems are content-dependent, we design a contrast algorithm by eliminating all the content information from our user social ties, leaving only the number of interaction times to indicate the strength of social ties. All the results are listed in Table 3.

We note that when eliminating user interest match score (No_Interest), $AP@5$ suffers from a 2.0% decline and the coverage of users suffers from a 2.0% decline. Similar to user interest, the coverage of users decreases 3.4% after excluding user influence features (No_Influence) from our model. When we eliminate the user social ties feature (No_Ties), the model suffers a 60% decline of $AP@5$ and a 41% decline of coverage. This result is in accordance with the results in section 6.1, which shows although user interest match and user influence help to improve the recommendation result, content-dependent user social ties play a much more significant role in the recommendation. It's worth noting that AP@5 exhibits the best performance after eliminating the influence features, indicating that not all influential users are interested in the tweet and many pay little attention to mentions since they may receive hundreds, or even thousands per day. However, the influence features do help to expand the retweet rate and user coverage because the influence brought by influential users outweighs the precision loss.

Furthermore, after removing all the content feature from the social ties (No_ContentInTie), a 14% decline in $AP@5$ and a 13% decline in coverage prove that content feature in social ties plays an important part in the recommendation and user social ties are content-dependent.

6.3 Ranking Function Evaluation

Various machine learning models can be used as ranking functions for our task and we explore three most commonly used ones. The result is listed in Table 4. We can see that our SVR based model(WTM_{SVR}) outperforms the linear regression (WTM_{LR}) and GBDT (WTM_{GBDT}) based models and we attribute it to the kernel function feature used in SVR which helps us to map the data from the input space into a higher dimensional space.

Table 4: WTM with Different Ranking Functions

	WTM_{SVR}	WTM_{LR}	WTM_{GBDT}
Precision	**0.1343**	0.0877	0.0769
AP@5	**0.1005**	0.0613	0.0492
Retweet Times	**3.1026**	2.2342	0.8997
Normalized Coverage	**0.8525**	0.5837	0.7986

Table 5: WTM and CCFR with 1 or 3 Users Recommended

	WTM_1	CCFR_1	WTM_3	CCFR_3
Precision	**0.0988**	0.0358	**0.1263**	0.0718
AP@k (k=1 or 3)	**0.0988**	0.0368	**0.1021**	0.0509
Retweet Times	**1.1313**	0.4945	**2.3307**	1.2913
Normalized Coverage	**0.5003**	0.2569	**0.7224**	0.4587

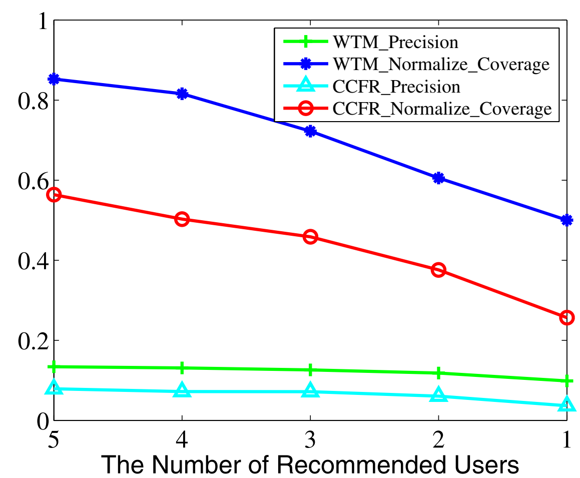

Figure 2: Results with Limited Recommended Users

6.4 Limited Recommended Users

The tweet-length limitation makes it hard to mention many users at the same time and moreover, mention too many users may results in the tweet looking suspicious as a spam tweet. We choose to recommend 5 users in our evaluation

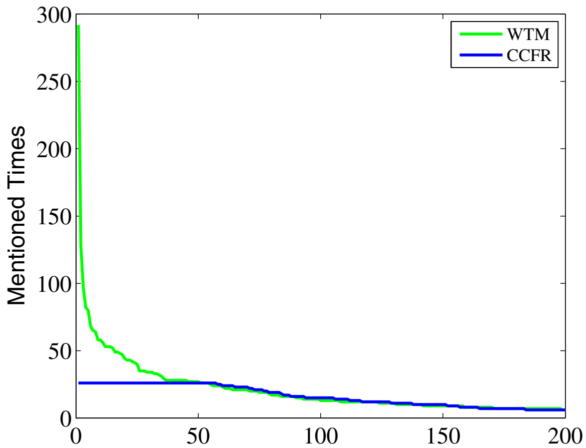

Figure 3: Recommendation Density Comparison Between WTM and CCFR (200 most recommended users)

because we believe 5 is the up-limit of mentioning users in one tweet and in practice use, a user can choose a subset of the 5 users to mention. We also test the performance of our algorithm when only mention 1∼4 users in a tweet and compares it with our best comparison algorithm CCFR, which are shown in table 5 and figure 2.

Our algorithm outperforms CCFR based on all metrics. For instance, when only mentioning one user, our algorithm shows a more than 200% remarkable improvement on all the metrics. Furthermore, compared with CCFR, the performance decline rate of our algorithm is much less than CCFR's when reducing the number of recommended users. For instance based on normalized coverage user metric, the average decline rate of our WTM is 31%, while the average decline rate of CCFR is 51%, which confirms our WTM performs much better when only a few users are recommended.

The precision drops slightly when recommending fewer users, showing that expanding the retweet is a quite difficult task and recommending only few users will incur higher miss rate, leading to the slight precision drop.

6.5 Recommendation Overload Evaluation

If everyone uses the *whom-to-mention* system, recommendation overload may occur and a popular user may receive tons of mention notification from the recommendation system which will result in a severe interruption. We show how many times a user is recommended in our evaluation in a descending order in figure 3. From the figure the recommendation distribution of WTM is more smooth compared with our best comparison algorithm CCFR. It is also worth noting that in CCFR, there exist users recommended hundreds of times which may lead to potential mention overload while our algorithm avoids the overload problem by setting the constraints based on user's free will.

7. DISCUSSION

The experiment results may seem a bit low, which is in accordance with our expectation. On one hand we ascribe it to we performing an off-line evaluation by using user's retweet log to estimate the possible information cascade and a perfect recommendation match in real world may be regarded as a miss in the evaluation as a result of lack of retweet log given the tweet. However, by comparing our algorithm with a set of carefully designed comparison algorithms, we believe our algorithm performs well based on the remarkable improvement on all metrics. On the other hand, attracting others to retweet is not an easy job and comparing with the average retweet rate 0.11 on twitter (0.78 on Weibo), our average 3.1 retweet rate shows a notable improvement.

Based on our comparison evaluation, it shows the content-dependent user social tie feature plays a much more important role compared with user interest match and user influence. We proposes 3 reasons for this phenomenon: First, though with careful pre-processing, the ambiguity and noise in the tweets still decrease the accuracy of user interest match. As a matter of fact, even though both are content features derived from user's tweets to model user's interest, the content feature from user's social ties shows more effectiveness compared with content feature from user's interest model, because the former feature is with less noise (users usually prefer to choose a well written tweet with a clear topic to retweet). Second, though intuitively influential users can lead to a larger diffusion of the tweet, they are usually mentioned by large numbers of people everyday, which makes them more easily to ignore the mention notifications. Third, the content-dependent retweet social tie is a strong indication. A user retweeting another user's tweet usually indicates a close user relationship and people who are close are more likely to retweet a tweet from each other. Moreover, retweet shows a strong interest on the topic of the tweet, so the user will be very likely to retweet the tweet with the same topic again in the future.

8. CONCLUSIONS

We offer the first in-depth study on Mention Recommendation and propose a new recommendation scheme to expand the diffusion of tweets by recommending proper users to mention. We formulate this new problem as a ranking problem and use new features, new relevance and a machine learned ranking function to solve it.

We find that the best performance of the algorithm is achieved when all the new features, including user interest match, user social ties and user influence, are used. A relevance defined by the coverage of users and an SVR based ranking function also help to improve the performance. Based on our comparison experiment, we also find that user relationship based features play a more important role than the content based features. Furthermore, we confirm that the content-dependent feature in user relationships is of high effectiveness in our recommendation model.

Many future works can be further explored. For instance, we use a post-processing step to solve the recommendation overload problem while constrained optimization can be tried to address this issue in the future. It's also interesting to study on how the proportion of strangers and friends in the recommendation list affect the tweet diffusion.

9. ACKNOWLEDGMENTS

This work was supported by the National Basic Research Program of China (973 Program) under Grant 2013CB336500, National Nature Science Foundation of China (Grant Nos: 61173185, 61173186, 61125203, 61222207), National Key Technology R&D Program (Grant Nos: 2012BAI34B01, 2008BAH26B00), Zhejiang Province Key S&T Innovation Group Project (Grant No.2009R50009), Program for New Century Excellent Talents in University (NCET-09-0685).

10. REFERENCES

[1] F. Abel, Q. Gao, G. Houben, and K. Tao. Analyzing user modeling on twitter for personalized news recommendations. *User Modeling, Adaption and Personalization*, pages 1–12, 2011.

[2] E. Bakshy, J. Hofman, W. Mason, and D. Watts. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of WSDM 2011*, pages 65–74, 2011.

[3] E. Bakshy, J. Hofman, W. Mason, and D. Watts. Identifying influencers on twitter. In *WSDM 2011*, 2011.

[4] B. Bartell, G. W. Cottrell, and R. Belew. Learning to retrieve information. In *Proceedings of the Swedish Conference on Connectionism*, 1995.

[5] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking svm to document retrieval. In *Proceedings of SIGIR 2006*, pages 186–193, 2006.

[6] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *4th International AAAI Conference on Weblogs and Social Media (ICWSM)*, pages 10–17, 2010.

[7] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy. Make new friends, but keep the old: recommending people on social networking sites. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 201–210, 2009.

[8] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1185–1194, 2010.

[9] M. Chen, J. Han, and P. Yu. Data mining: an overview from a database perspective. *Knowledge and data Engineering, IEEE Transactions on*, pages 866–883, 1996.

[10] F. Diaz, D. Metzler, and S. Amer-Yahia. Relevance and ranking in online dating systems. In *Proceeding of SIGIR2010*, pages 66–73, 2010.

[11] Z. Guan, C. Wang, J. Bu, C. Chen, K. Yang, D. Cai, and X. He. Document recommendation in social tagging services. In *Proceedings of the 19th International Conference on World Wide Web*, pages 391–400, 2010.

[12] I. Guy, I. Ronen, and E. Wilcox. Do you know?: recommending people to invite into your social network. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 77–86, 2009.

[13] I. Guy, N. Zwerdling, D. Carmel, I. Ronen, E. Uziel, S. Yogev, and S. Ofek-Koifman. Personalized recommendation of social software items based on social relations. In *Proceedings of the third ACM conference on Recommender systems*, pages 53–60. ACM, 2009.

[14] D. Hand, H. Mannila, and P. Smyth. *Principles of data mining*. The MIT press, 2001.

[15] J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 199–206, 2010.

[16] L. Hong and B. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, pages 80–88, 2010.

[17] W. Hsu, A. King, M. Paradesi, T. Pydimarri, and T. Weninger. Collaborative and structural recommendation of friends using weblog-based social network analysis. In *AAAI Spring Symposium Series*, 2006.

[18] J. Jang, J. Choi, G. Jang, and S. Myaeng. Semantic social networks constructed by topical aspects of conversations: An explorative study. In *Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

[19] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD*, pages 56–65, 2007.

[20] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of WWW2010*, pages 591–600, 2010.

[21] P. Li, C. J. C. Burges, and Q. Wu. Mcrank: Learning to rank using multiple classification and gradient boosting. In *NIPS'07*, 2007.

[22] P. Melville, R. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the National Conference on Artificial Intelligence*, pages 187–192, 2002.

[23] M. Michelson and S. Macskassy. Discovering users' topics of interest on twitter: a first look. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, pages 73–80, 2010.

[24] A. Mohan, Z. Chen, and K. Q. Weinberger. Web-search ranking with initialized gradient boosted regression trees. *Journal of Machine Learning Research, Workshop and Conference Proceedings*, 14:77–89, 2011.

[25] R. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204, 2000.

[26] R. Nallapati. Discriminative models for information retrieval. In *Proceedings of SIGIR 2004*, pages 64–71, 2004.

[27] M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine learning*, 27(3):313–331, 1997.

[28] T. Qin, X.-D. Zhang, M.-F. Tsai, D.-S. Wang, T.-Y. Liu, and H. Li. Query-level loss functions for information retrieval. *Information Processing and Management*, 44(2):838–855, 2008.

[29] A. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[30] D. Sousa, L. Sarmento, and E. Mendes Rodrigues. Characterization of the twitter@ replies network: are user ties social or topical? In *Proceedings of SMUC 2010*, pages 63–70, 2010.

[31] S. Wu, J. Hofman, W. Mason, and D. Watts. Who says what to whom on twitter. In *Proceedings of WWW 2011*, pages 705–714, 2011.

[32] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. McLachlan, A. Ng, B. Liu, P. Yu, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14:1–37, 2008.

[33] B. Xu, J. Bu, C. Chen, and D. Cai. An exploration of improving collaborative recommender systems via user-item subgroups. In *Proceedings of the 21st international conference on World Wide Web*, 2012.

[34] J. Yang and J. Leskovec. Modeling information diffusion in implicit networks. In *Data Mining , 2010 IEEE 10th International Conference on*, pages 599–608, 2010.

[35] S. Ye and S. Wu. Measuring message propagation and social influence on twitter. com. *Social Informatics*, pages 216–231, 2010.

[36] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *Proceedings of SIGIR 2007*, pages 271–278, 2007.

[37] W. Zhao, J. Jiang, J. Weng, J. He, E. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. *Advances in Information Retrieval*, pages 338–349, 2011.