Tag Expansion Using Friendship Information: Services for Picking-a-crowd for Crowdsourcing

Bin Liang, Yiqun Liu, Min Zhang, Shaoping Ma, Liyun Ru, and Kuo Zhang

State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology
Tsinghua University, Beijing, China, 100084
mgigabyte@gmail.com, {yiqunliu,z-m,msp}@tsinghua.edu.cn,
{ruliyun,zhangkuo}@sogou-inc.com

Abstract. To address self-tagging concerns, some social networks' websites, such as LinkedIn and Sina Weibo, allow users to tag themselves as part of their profiles; however, due to privacy or other unknown reasons, most of the users take just a few tags. Self-tag sparsity refers to the problem of low recall obtained when searching for people on systems based on user profiles. In this paper, we use not only users' self-tags but also their friend relationships (which are often not hidden) to expand the tag list and measure the effectiveness of different types of friendship links and their self-tags. Experimental results show that friendship information (friendship links and profiles) can effectively improve the performance of tag expansion, especially for common users who have limited followers.

Keywords: Tag expansion, Self-tag mining, Energy function, Machine learning.

1 Introduction

With the boom in social networks, microblog services such as Twitter, Sina Weibo and LinkedIn have grown rapidly. On Mar. 21st, 2012, its sixth birthday, Twitter announced that it had 140 million users and 340 million tweets per day. On Feb. 20th, 2013, Sina Microblog announced that it had 500 million users and 46 million active users per day. With the development of social networks, many applications are based on the profiles and social relationships of these millions of users (G. A. Gupta 2013, Liang Bin 2014). In practice, users are willing to create profiles on these online social networks; the profiles consist of attributes such as location, hobbies, and sex. Additionally, users love to follow other users depending on their interests. This phenomenon is called *homophily,* which is a tendency that says "interpersonal similarity breeds connection" (M. McPherson 2011). Weng has reported that users who follow each other reciprocally usually share topical interests (Weng 2010).

It is, however, a heavy burden to require every user to create a complete profile. Although many social network sites allow users to tag themselves with a few

H.-Y. Huang et al. (Eds.): SMP 2014, CCIS 489, pp. 25–43, 2014.

keywords to reduce the burden, many users still take only a few tags for themselves, due to privacy or other reasons.

As a result, we have two types of data: user following data and user profile data. Because the former is sparse and the latter is rich, it guides us to apply user following data to predict user profiles. Actually, the ability to automatically predict user attributes could be useful for a variety of social networking applications such as friend and content recommendations (G. A. Gupta 2013) and our people search system[1], a research project on picking-a-crowd from social networks for crowdsourcing. Here, we would like to briefly introduce crowdsourcing. Current approaches to crowdsourcing are viewed as a type of pull methodology, where tasks are split and published on platforms where online workers can pick their preferred tasks. In fact, this type of approach has many advantages, such as simplicity and equality; however, it does not guarantee the assigning of tasks to suitable workers. We provide a service for crowdsourcing based on a push methodology that carefully selects workers to perform given tasks according to their profiles extracted from social networks. As a result, the self-tag sparsity problem makes it hard for our system to find enough candidate workers to complete a crowdsourcing task.

Workers on crowdsourcing platforms are neither celebrities nor well-known users but common users; therefore, expanding the tags of these users is a major challenge and must rely on automatic algorithms. It occurred to us that while common users may post a few microblogs and take a few tags, they usually maintain a good social network. Therefore, we mainly focus on expanding the tags of common users using their friendship information and their self-tags.

To summarize our motivation, our work focuses on expanding tags for users, especially common users, in social networks. Many users take only a few tags for themselves; this makes some of them unsearchable, and the quantity and quality of candidate workers in crowdsourcing systems is not up to the mark.

Before we introduce our model, we would like to briefly explain the differences between social tagging, people-tagging and self-tagging. Social tagging is a way for users to freely choose keywords to describe Internet content resources (Delicious and Flickr provide the service). People-tagging is a form of social bookmarking that enables people to organize their contacts into groups, annotate them with terms supporting future recall, and search for people by topic area (Bernstein 2009; Farrell 2007). Self-tagging is a way for users to tag themselves, for example, LinkedIn and Sina Weibo are services that allow users to only tag themselves but do not allow them to tag other users. Despite their differences, the above three concepts still have many similarities. Social tagging, People tagging and Self-tagging all aim at getting better descriptions of an object to make it easy to search and share. Muller's work shows that self-tags usually reflect the hobbies, knowledge-domain, location and social role of a user, which is the same as social tags (Muller 2006).

We now introduce the baseline and our model. In this paper, we employ association rules mining (Heymann 2008), a tag recommendation approach based on joint probability (Rae 2010) and the random walk algorithm (Li 2009) as our baseline algorithms. We do a survey to explore sources of tag expansion and discover that users'

[1] Our online people search system, `http://xunren.thuir.org/`

self-tags have the shortest KL divergence to the tags of their bidirectionally linked friends and the largest KL divergence to the keywords of microblogs that users post. We only choose as features the tag frequency of users' social relationships, the conditional probability of friends' tags given users' self-tags and the prior probability of tags and adopt an energy-based function to create our model and use negative log-likelihood loss as our loss function to train our model; this approach leads us to discover that *Precision* and *Recall* of tag expansion have improved significantly. Moreover, we also discuss the differences between these improvements for common users and celebrities.

We share the data related to our paper on a web page for researchers[2].
Finally, we sum up our contributions.

- We take users' self-tags and 3 types of users' friendship information into consideration to expand tags, as detailed in Section 4, and we show the effectiveness of different kinds of friendship information by experiments, as detailed in Section 5.
- We are the first to use different ranges of followers to delineate the model's performance on common users vs. celebrities, as detailed in Section 5, and we show that our algorithm is effective on common users who have fewer followers.
- We discuss the power of friendship information exacting on the performance of tag expanding.

The remainder of this paper is organized as follows: we introduce background information and related work in Section 2 and define the problem in Section 3. Then, Section 4 describes in depth our work including our survey, the related baseline chosen and our algorithms. Section 5 illustrates experiments showing the performance of each algorithm on sets having different ranges of users' followers. Section 6 includes some discussion on three questions on our algorithm and baseline. Finally, the summary is presented in Section 7.

2 Related Work

Many research efforts focus on tag expansion (tag recommendation or tag suggestion); however, these works mainly serve applications that make objects easy to be searched and shared, such as expanding tags of photos (Kucuktunc, 2008; Garg, 2008; Li, 2009), MP3s (Eck, 2007) and Blog posts (Sood, 2007). However, as far as we know, few works focus on expanding tags for linked people.

Many works in this area focus on social tags: Heymann (2008) proposed market-basket data mining to retrieve relevant tags. Agrawal (1993) used association rules that observe the relationships between tags from the co-occurrence relations of tags. Song (2011) proposed a general model of the description of tag expansion in a bi-graph. Rae (2010) mentioned a computing mode for predicting another tag *t* based on some known tags. Its basic idea is that the probability of the known tags generating

[2] Our data are shared on `http://xunren.thuir.org/share_EPSN/`

tag *t* is the joint probability of each known tag producing tag *t*. This approach not only takes into account the conditional probability but also observes the prior probability *P(t)* of the predictive tags, which is in favor of recommending tags with high frequency and therefore helps to solve Inter-User disagreement. Schenkel (2008) presented a computing mode for expanding a tag *t* based on a known document. The probability of a tag being expanded is computed by the maximum probability of a certain keyword in the document. Liu (2009) put forward a random walk model over a tag graph to improve the ranking of tags. The idea is based on the probability that a tag of the object is related to all of the keywords in the graph. Li (2009) proposed a neighbor voting algorithm that accurately and efficiently learns tag relevance by accumulating votes from visual neighbors. They used 3.5 million tagged Flickr images and concluded that the voting method is very efficient and effective. Szomszor (2008) presented a method for automatic consolidation of users who are active in two social networks to have more tags to model user interests, which is also an important approach to tag expansion.

Importing friendship information to find users' private attributes has also been explored by many research works (Linda mood, 2009; Zheleva, 2009; Mislove, 2010). Certain experimental results show that friendship information can leak private information to some extent (Zheleva, 2009), while other results show that certain user attributes can be inferred with high accuracy when given information on as little as 20% of the users (Mislove, 2010). However, these works focus more on privacy protection and on general profiles such as location, grade in school, etc.; thus, models of these works are more related to community detection, analysis of networks, and privacy-related topics.

To summarize, many works inspired us to solve the problem of expanding self-tags, especially the works on social tagging. However, we believe that our paper is the first research effort to focus on expanding tags for linked users by using their self-tags and their friendship information and is also the first one to measure the performance of the algorithms on common users and celebrities separately and on different types of social relationships.

3 Problem Definition

We define a social network as a directed graph *G (V, E, T)*, where *V* is a set of *u* nodes representing users in the social network, *E* is a set of following relations (the directed friendship links), T is a tag set of all users, and *t* (u) indicates self-tags that are viewed as a list of keywords *u* chosen from *T*. Finally, *G(u)* is a subgraph of *G(V,E,T)*, where *V*={*u*}, *E*={links related to *u*} and *T* = *t(u)*.

The problem of tag expansion of users in social networks can be generalized as solving the conditional probability of expanding tag *e* given the social networks *G*(u) and k self-tags($t(u)_1,\ldots t(u)_k$).

$$\mathrm{P}(\mathrm{e}|\{t(u)_1, \ldots t(u)_k\}, \mathrm{G}(\mathrm{u})), \mathrm{e} \neq \mathrm{t}(\mathrm{u})_1, \ldots t(u)_k$$

4 Our Work

4.1 Survey on the KL Divergences for Different Friendships

Traditional approaches such as joint probability and association rules mining only consider relations between tags or relations between users who tag the same content, while self-tags in social networks are given by users themselves instead of others, and as a result, there are no relations between users who tag the same users. Zheleva (2009) first proposed that user profiles can be inferred from their friendship and group information; therefore, we believe that tag expansion can also be inferred from users' relationship information, and the core of our survey methodology is to explore tags whose sources are the most similar to users' self-tags, their bidirectional following friends, followers, following sets or their own microblogs.

First, we define a basic function called *follow (u, x)*, which indicates the following relationship between *u* and *x*. If *u* follows *x*, then *follow(u,x) = true*; if not, then *follow(u,x) = false*. The social relationships of an objective user form a type of user set, and each user in this set has some relationship with the objective user. Based on the function of *follow*, we define 4 types of social relationships.

The Only-Following User Set of a user *u* (*OFS(u)*) contains the users followed by *u* instead of the users following *u*.

$$\mathbf{OFS}(\mathrm{u}) = \{\mathrm{x}|\mathrm{follow}(\mathrm{x},\mathrm{u}) == \mathrm{false}\ \Lambda\ \mathrm{follow}(\mathrm{u},\mathrm{x}) = \mathrm{true}\} \tag{1}$$

The Follower User Set of a user *u* (*FS(u)*) contains the followers of *u* instead of the users that *u* follows.

$$\mathbf{FS}(\mathrm{u}) = \{\mathrm{x}|\mathrm{follow}(\mathrm{x},\mathrm{u}) == \mathrm{true}\ \Lambda\ \mathrm{follow}(\mathrm{u},\mathrm{x}) = \mathrm{false}\} \tag{2}$$

The Bidirectional Following User Set of a user *u* (*BFS (u)*) contains the users following *u* and the users followed by *u*.

$$\mathbf{BFS}(\mathrm{u}) = \{\mathrm{x}|\mathrm{follow}(\mathrm{x},\mathrm{u}) == \mathrm{true}\ \Lambda\ \mathrm{follow}(\mathrm{u},\mathrm{x}) = \mathrm{true}\} \tag{3}$$

The ALL User Set of a user *u* (*ALL(u)*) contains the users who have at least a following link with *u*.

$$\mathbf{ALL}(\mathrm{u}) = \{\mathrm{x}|\mathrm{follow}(\mathrm{x},\mathrm{u}) == \mathrm{true}\ \vee\ \mathrm{follow}(\mathrm{u},\mathrm{x}) = \mathrm{true}\} \tag{4}$$

We randomly select 0.32 million users who are divided into 5 ranges based on the number of their followers. Then, we observe the KL divergences between users' self-tags and tags from *OFS*, *FS*, and *BFS*. In *BFS*, for example, we extract all of the tags of users in BFS(*u)* and draw a probability table of the occurrence of all of the tags that can compute the probability *P(t)* of any of the tags in the table. The KL divergence formula is shown below:

$$\mathrm{KL}(\mathrm{t}(\mathrm{u}),\mathrm{tag}(\mathrm{BFS}(\mathrm{u}))) = -\sum_{i=1}^{k}\frac{1}{|t(u)|}\log\big(P(t_i)\big)\ \ t_i\ \epsilon\ t(u) \tag{5}$$

The experimental results of Table 1 show that KL divergences between users' self-tags and tags of *BFS* are the shortest, with an average KL divergence of 5.84.

The divergence between users' self-tags and tags of *FS* reaches 6.74, which is the largest. This result indicates that users share more tags with their bidirectional following friends than with their followers. On the other hand, common users who have fewer than 1000 followers are the mainstream crowd (78% are common users in the random sampling), and only 1% of the users are celebrities who have more than 1 million followers. The KL divergences are small for the crowd and large for celebrities, showing that the friendship of the crowd is simpler than that of celebrities who have diversified circles.

Table 1. KL divergences between user's self-tags and different types of social relationships

Followers-range	# users in the range	BFS	OFS	FS
[1M,∞]	146	5.8021	5.7265	7.6361
[100k ,1M]	3058	5.9733	6.2122	7.2468
[10k ,100k]	15636	5.9910	6.2834	6.8481
[1k,10k]	55016	6.0472	6.2288	6.4336
[0,1k]	266093	5.3895	6.0332	5.5275
Average		**5.8406**	**6.0968**	**6.7384**

In addition, to understand the relationship between tags and users' microblogs, we select 1000 microblogs of each user and divide these microblogs into words (removing stop words and other meaningless words) that act as tags. The results show that the KL divergences between the tags and the microblog contents of users are huge, with the smallest KL divergence of 6.44 in Table 2. The reason is that most users usually do not talk about content related to their tags, such as nationality, sex, educational background, profession, etc., in their microblogs but tend to discuss other non-privacy-related topics such as news and constellations. We plot all of the KL divergences in Figure 1.

Table 2. KL divergences between tags and microblogs of users

Followers-range	# users in the range	Blog
[1M,∞]	146	7.7720
[100k ,1M]	3058	7.2440
[10k ,100k]	15636	6.9016
[1k,10k]	55016	6.7707
[0,1k]	266093	6.4472

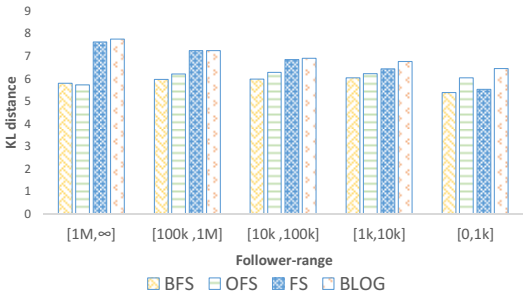

Fig. 1. KL divergences between tags of users and user sets of different types of social relationships

According to the survey in this section, we discover 4 basic facts:

- Tags of users and tags of their BFS have the maximum similarity.
- KL divergences between users' tags and the microblogs that they post are huge, especially for common users, who have no more than 1,000 followers.
- KL divergence greatly differs for different ranges of followers and is usually small for common users and large for celebrities.
- Common users are the mainstream users, and tag expansion should focus on common users who have fewer followers.

4.2 Our Baselines

As part of our research, we employ the approach of association rules mentioned by Heymann (2008) as our first baseline and consider the approach mentioned by Agrawal (1993) as our second baseline. Its formula is listed below:

$$\mathrm{P}\left(e\middle|\mathrm{t(u)}\right) = \mathrm{P}(e) * \prod_{t_i \in t(u)} \begin{cases} P(t_i|e), & if\ p(t_i|e) > 0 \\ \epsilon, & otherwise \end{cases} \tag{6}$$

We compute the association probability of each tag *e* on users' self-tag set *t(u)*. The higher the probability, the greater relevance *e* has. We adopt the approach mentioned by Liu (2009) to give a good rank as our third baseline. First, obtain the *BFS* of a user; then, use the probability of tags of *BFS* as the prior (v_j) and the joint probability of tag *i* and tag *j* as p_{ij}; and finally, use the formula below:

$$r_k(j) = \alpha \sum_i r_{k-1}(i) p_{ij} + (1-\alpha) v_j \tag{7}$$

4.3 Our Model

Machine Learning usually can be viewed as a method to create a connection between X (known variables) and Y (target variables). By capturing such dependencies, a

model can be used to answer questions about the values of target variables given the values of known variables. Energy-Based Models (EBMs) are a type of popular model that can capture dependencies by associating a scalar energy (a measure of compatibility) to each configuration of the variables and finding values of the target variables that minimize the energy. The process of Learning is to find an energy function that associates low energies with correct values of the target variables and higher energies with incorrect values (Yann LeCun 2006)

In this paper, the known X (u,t), a feature vector that represents the configuration of user *u* and tag *t*, is passed through a parametric function G_W, which produces a scalar output. The target variable $\hat{Y}(u,t)$ indicates whether user *u* regards tag *t* as a self-tag. The Energy function is the quadratic value of the difference between $G_W(X(u,t))$ and $\hat{Y}(u,t)$.

$$E\left(W,\hat{Y}(u,t),X(u,t)\right) = \left(G_W\left(X(u,t)\right) - \hat{Y}(u,t)\right)^2 \tag{8}$$

where $\hat{Y}(u,t) = \begin{cases} 1 & t \in tagset(u) \\ 0 & other\ wise \end{cases}$, tagset(u) is the set of self-tags of user *u*. The method of choosing the loss function is not the focus of our paper, so we just use negative log-likelihood loss, which works well in many architectures, as our loss function, and we omit *u* and *t* for conciseness

$$L(W,Y,X) = E(W,Y,X) + \frac{1}{\beta}\log\left(\int_{y\in Y} e^{-\beta E(W,y,X)}\right) \tag{9}$$

It is natural to compute the gradient for each record $<X^i, Y^i>$ in the corpus and generate update rules as follows:

$$\frac{\partial L(W,Y^i,X^i)}{\partial W_x} = \frac{\partial E(W,Y^i,X^i)}{\partial W_x} - \cdot \sum_{y\in Y} \frac{\partial E(W,y,X^i)}{\partial W_x} * P\left(y\middle|X^i,W\right) \tag{10}$$

$$W_x \leftarrow W_x - \eta \frac{\partial L(W,Y^i,X^i)}{\partial W_x} \tag{11}$$

Finally, we introduce the effective features of X(u,t) that we adopt in practice

- Prior probability of expanded tag *t*: P(*t*)
- Probability of expanded tag *t* generated by users' social graph *G(u)*: $P\left(t\middle|G(u)\right)$
- Probability of self-tag st_i, given expanded tag *t*: $P(st_i(u)|t)$

However, two obvious problems emerge:

- How do we construct the learning corpus?

There is a trick to constructing the learning corpus. First, we list each $P(t_i(u)|e)$ in the order of descending probability, say, for example, a user tags himself with *A, B* and *C*. Then, we construct a learning record of the expanded tag *t.* We just suppose that *P(B|e)>P(A|e)>p(C|e)* and $t \in t(u)$, so the pair of learning records is then X(u,t) = { P(B|e),P(A|e),p(C|e),P(e|G(u)),P(e)} and $\hat{Y}$ (u,t) = 1.

In summary, we define the following features in Table 3 and sort the conditional probability of tag *t* given self-tags of user *u* in descending order:

Table 3. Features and their definition

Features	Definitions
1st-related	The largest conditional probability of self-tag given expanded tag t.
…	
nth-related	The smallest conditional probability of self-tag given expanded tag t.
G-power	Probability of social relationship G producing tag e
priori	Prior probability of tag e

- How do we choose G(u) and calculate P(t|G(u))?

We just simply calculate *P(t|G(u))* as the frequency of the expanded tag *t* in all tags of users in *G(u)*; the reason for tf-idf-type approaches not being used is that the prior probability of tags has already been added to the regression calculations as a feature. Here, *G(u)* can be replaced by different types of social relationships, such as *BFS, OFS,* and *FS*, as defined in Section 4.

5 Experiments

5.1 Datasets and Tools

We have launched and led a crowdsourcing organization and crawled 0.25 billion users' profiles, including name, sex, tags, introduction, verification, mutual following relationships, as well as their microblog contents of over 15 billion since 2010. As a result, we can obtain the data of the users' following and followers, together with the contents of their microblogs posted in Sina Weibo. In our experiments, we choose 10,000 users and their friend links as a training set, and 320,000 users and their 65 million following links and 4 billion followers (including duplicate followers of different users) as our test set.

We adopt *THUIRDB(Liang 2013)*, which has a good performance of completing millions of queries per second, as our database, which can effectively help our computing by indexing the following user sets and the followers' user sets.

5.2 Research Questions

We would first like to propose two main research questions in this paper and then carry out our experiments and analysis with these questions in mind.

1) Will the performance of tag expansion improve after importing friend information, and what are the differences between the performances of tag expansion based on different types of friend information?

2) Is the performance improvement effective on both common users and celebrities?

5.3 Training

We randomly choose 10,000 users as our training user set, from which we generate 20-40M learning materials as our training set. For our model, all of the features are normalized; therefore, the weight associated with each feature can reflect the importance of the feature to some degree. After training with the *SGD* algorithm, we give the weight of each feature in Table 4.

Table 4. Weight of features

Friendship type	Intercept	1st -R	2nd -R	…	G	priori
BFS	-6.24	4.21	1.93		19.51	0.41
OFS	-6.04	3.67	2.18		13.24	11.03
FS	-5.88	4.13	2.07		13.77	5.09
ALL	-6.55	3.80	2.24		13.08	8.42

Table 4 shows that friend information exerts great influence on the end results because the weight of the feature is the largest among all of the features. Moreover, some relevant tags have certain weights, reflecting the effect of the conditional probability of these users' self-tags. Feature *1st-R* being greatly larger than feature *2nd-R* in most cases shows that the most relevant user self-tags exert more influence than the other tags. Case studies show that users' self-tags are usually diversified, while expanded tags are normally only relevant to 1 to 2 users' self-tags; therefore, this result is also in line with our case studies.

5.4 Evaluation and Analysis

We randomly choose 0.32 million users with 10 tags of themselves among 0.25 billion users as our Test Set. Then, we randomly hide 5 self-tags of each user and expand tags based on the rest of the tags and their friendship information. Then, we compare these expanded tags with hidden tags to observe the *Precision* and *Recall*. For a clear description, we list the algorithms in Table 5. To make the description convenient, we will abbreviate *RW+BFS* algorithms to *RW* as one of our baselines, *BT+BFS* into *BFS*, etc., in the rest of the paper.

First, we would like to answer the first research question: will the performances of tag expansion improve after importing friend information, and what are the differences between the performances of tag expansion based on different types of friend information?

Figure 2 plots the results of this experiment for all evaluated users. Bars are plotted for each algorithm, and height is with respect to the value of *precision* or *recall*. Three important results can be observed in this graph. First, we note that the *BFS* algorithm outperforms the best baseline algorithm by over 14.0% on *P@1*, 14.4% on *P@5* and 11.4% on *R@10*. In fact, considering our large test set (0.32 million users and 1.6 million tag comparisons), the significance of our result is reasonable. Second, it is reasonable that the winner is *BFS*; as we observed in Section 4, the quality of *BFS* is the best source, i.e., tags from *BFS* have the shortest *KL* divergence to users' self-tags.

When we import all of the friend information, the performance of *ALL* becomes much worse than that of *BFS*. Overall, this experiment shows that users' self-tags can be effectively inferred by friendship links, especially bidirectional following friendship links and their self-tags.

Table 5. Algorithms used in evaluation

Algorithm	Description
AR	Association Rules (Hemann,2008)
JP	Joint Probability (Rae,2010)
RW+BFS(RW)	Random Walk by using *BFS* (Liu,2009)
BT+BFS(BFS)	our algorithm by using *BFS*
BT +OFS(OFS)	our algorithm by using *OFS*
BT+FS(FS)	our algorithm by using *FS*
BT+ALL(ALL)	our algorithm by using all social relationships

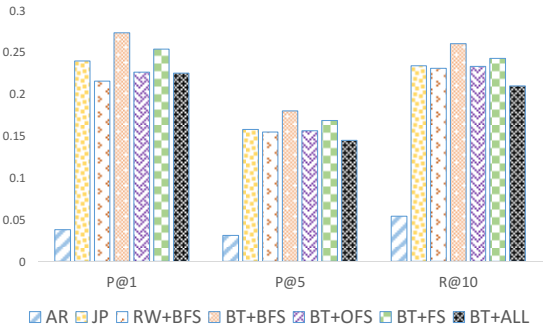

Fig. 2. Evaluations on all users

Additionally, Figure 3 plots the results of this experiment for the *Precision and Recall Curve* with each point (x=*P@k*, y=*R@k* | k=1, 2, 3, 4, 5), and then sets line properties that make the baselines look like dashed lines and our algorithms look like solid lines. This result is convincing because we usually limit the windows of expanded tags in practice; therefore, if *k* output windows are available, the performances of *P@k* and *R@k* are of great importance. We observe a more significant tendency in Figure 3, specifically that our algorithms are capable of expanding tags for various windows. From the perspectives of *Average Recall*, *Average Precision* and *F-Score* shown in Table 6, we can tell that *BFS* is also the best one.

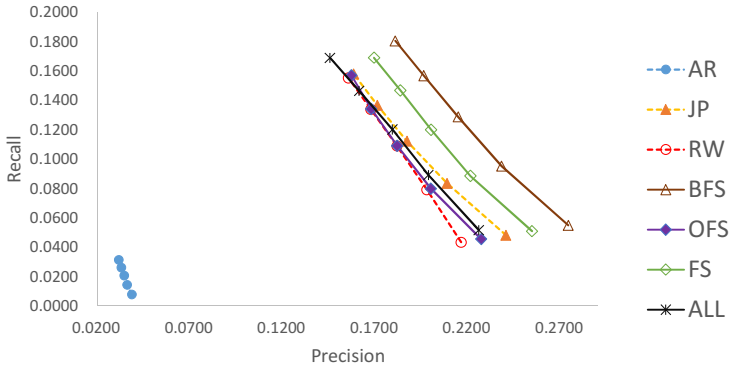

Fig. 3. Precision and Recall Curve

Table 6. Average Recall, Precision and F-Score

Algorithms	Average Recall	Average Precision	F-Score
AR	0.034	0.020	0.025
JP	0.192	0.107	0.138
RW	0.183	0.104	0.132
BFS	**0.220**	**0.123**	**0.158**
OFS	0.186	0.105	0.134
FS	0.205	0.115	0.147
ALL	0.206	0.115	0.148

To summarize our results for the first research question, performances of tag expansion have really been improved by importing friend information, whether viewed from the indexes *P@1, P@5* and *R@10* or from the perspective of *Precision and Recall Curve*, and the performance improves greatly, especially for *BFS* and *FS*. Other types of friend information are not as good as we imagined (they usually have a lot of noise), which is also consistent with our previous survey.

Next, let us come back to our second research question: is the improvement of performances effective for both common users and celebrities? We deliberately explore these results from the perspectives of P@1, P@5, and R@10.

P@1 is quite an important measurement because the performance of the best expanded tags usually represents the effectiveness of the algorithms on tag expansion. Figure 4 shows that the performance of the JP algorithm is the best baseline and that the BFS algorithm performs better than the best baseline algorithms in most cases. Case studies show that the tags of users with a large number of followers, e.g., invest and stock, tend to be subject-matter experts and hence lend themselves easily to association mining, while the tags of common users, e.g., Music, Runner, and Basketball, are usually high-frequency diversified words that are unlikely to have high-quality association rules; therefore, the performance of association mining will not be good enough. As for BFS, we discover that most common users with less than 1,000 followers are of high quality because they come from either the same school or the same

company and share many similar tags. The performance of tag expansion (BFS) of celebrities with more than 1,000,000 followers is also great, due to their bidirectional following friends having a close background and identification, which is often reflected in the friends' tags. For users with a number of followers in the middle, i.e., between that of common users and celebrities, the performance is relatively poor due to the diversity of social relationships. Case studies show that these users are usually journalists, politicians and social activists.

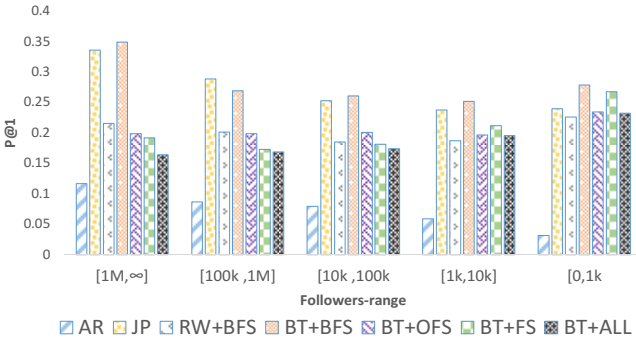

Fig. 4. *P@1* for different algorithms within different followers-ranges

Because we hide 5 tags, theoretically, if the expanded 5 tags are completely identical with the 5 hidden tags, then *P@5* may reach 100%. The experimental results in Figure 5 show that the *BFS* algorithm outperforms the baseline algorithms when the number of followers is smaller than 100,000. The result also shows that friendship information of common users is more effective, which is consistent with our survey in Section 4. For the *FS* algorithm shown by the green bar, due to the great differences in identification and background information between celebrities and their large number of followers, the performance of tag expansion is poor. However, identification and background information between common users are close to each other; hence, they share many similar tags, and the performance of the *FS* algorithm (green bar) becomes better and better with the reduction of the number of followers in Figure 5, which is in agreement with the findings of our survey in Table 1 in Section 4.

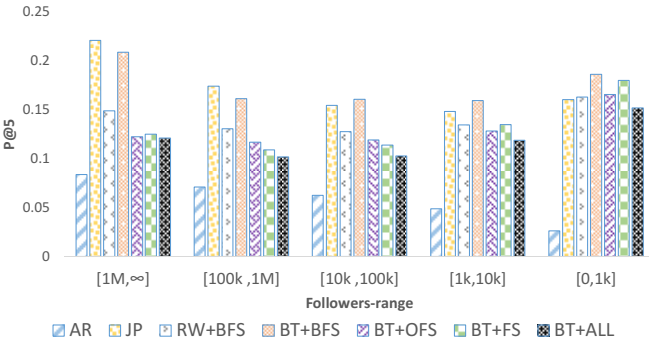

Fig. 5. P@5 for different algorithms within different followers-ranges

R@10 is also an important measurement to observe how much the rate of tags we hide can be recalled by each algorithm. The results in Figure 5 show that *BFS* can significantly outperform the baseline algorithms for common users. However, for celebrities, the performances of *JP* and *BFS* are similar. Because our experiment hides 5 of 10 tags, the *R@10* can be 50% at maximum; in fact, *R@10* of *BFS* and *JP* is more than 30% for celebrities, indicating that these algorithms can recall more than 3 tags that we have hidden before by generating 10 expanded tags. It is worth noting that Figures 4, 5 and 6 show the same phenomenon in that the red bar (*BFS*) and the green bar (*FS*) are similar in the range of followers of fewer than 1000. Case studies show that common users (who have no more than 1000 followers) have very few followers who usually follow back these users; in other words, *BFS* equals *FS* in most cases for common users. However, common users usually also follow a large amount of celebrities, which causes *OFS* to be different from *BFS* and *FS*.

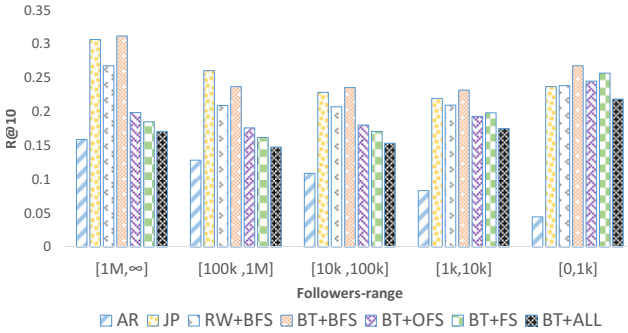

Fig. 6. R@10 for different algorithms within different followers-ranges

6 Discussion

After seeing the experimental results, 3 questions appear, which we discuss in depth in this section.

- **What Are the Qualities Tags Expanded by Different Algorithms Have?**

To find the answers to this question, we design another experiment and analyze the statistics on the average length and word frequency of tags recommended by different algorithms. The results are shown in Table 7.

After the calculation, we know that the average length of users' tags is 2.918758 and that the evaluation word frequency of users' tags is 0.036797. The results of Table 8 show that the tags expanded by algorithms AR and JP are most identical to the real tags of users, while RW and our methods prefer more hot tags.

However, the performance is not always better when the expanded tags are shorter and the word frequency is higher. For example, tags recommended by the ALL algorithm have the quality of short evaluation length and high evaluation word frequency but with poor performance. After taking following and followers into consideration,

we find that popular tags are more easily outstanding, thus diluting the proportion of tags more related to users. This indicates that being popular is not necessarily good, but it is important to have high relevance.

Table 7. Word frequency and length of tags

Algorithms	Average length	Average word frequency
AR	3.04328	0.00011
JP	2.62578	0.02907
RW+BFS	2.15382	0.11213
BT+BFS	2.11390	0.12047
BT+OFS	2.00250	0.12824
BT+FS	2.02436	0.12437
BT+All	2.00150	0.13100

- **What will Happen If We Only Adopt the Feature of Social Networks?**

In our algorithms, we take into account the prior probability of the expanded tags and the relevance between expanded tags and users' real tags. What will happen if we only consider the tag weight of social networks, i.e., expanding tags on the tags appearing most frequently in users' social relationships, instead of considering the two factors mentioned above? To find the answer to this question, we design a comparison experiment using the data of all users without dividing the range of followers. The result is shown in the following table:

Table 8. Performance of algorithms only considering the feature of social networks

Algorithms	P@1	P@5	R@10
AR	0.038	0.031	0.054
JP	0.240	0.158	0.234
BT+BFS	0.273	0.180	0.260
BT+BFS_ONLYG	0.211	0.070	0.111
BT+OFS	0.226	0.156	0.233
BT+OFS_ONLYG	0.167	0.059	0.099
BT+FS	0.254	0.169	0.243
BT+FS_ONLYG	0.200	0.067	0.107
BT+ALL	0.225	0.145	0.210
BT+ALL_ONLYG	0.193	0.065	0.104

Algorithms with the suffix ONLYG represent algorithms only considering the feature of social networks instead of other features. We discover that under this circumstance, the four main indexes decrease in an obvious fashion. For example, in BFS, P@1 decreases by 22%, P@5 decreases by 61%, and R@10 decreases by 57%.

This indicates that without considering the relevance between expanded tags and users' real tags, the performance of words with high frequency on social networks will decrease greatly.

- **Does Our Algorithms Perform Better Only If the Number of Followers Is Small?**

We suppose that this phenomenon may be related to the fact that the subjects of training samples in the process of training feature weights are users with a small number of followers. Therefore, we train users with over 10k followers to learn another set of parameters and observe the change in the performance. We only use the BFS algorithm to observe the trend.

Table 9. Weights of features for training of different users

	All users	Users with >10K followers
Intercept	-6.2415	-6.1342
1st –R	4.2188	4.4144
2nd -R	1.9302	2.3467
3rd -R	1.2862	0.5856
4th -R	0.2528	1.8256
5th -R	2.6454	2.0146
6th -R	0.9987	0.6409
7th -R	1.3642	1.0403
8th -R	3.2611	10.484
9th -R	-7.3255	-11.045
10th -R	-62.705	-117.535
G-power	19.5	27.7
priori	0.4121	-7.5465

From Table 9, we discover that the importance of the prior of the expanded tags is weakened and that the effect of the social networks is stronger when carrying out parameter training with users who have over 10k followers.

Table 10. The improvement by using training set of users with over 10k followers

Followers-range	**BT+BFS**			**BT+BFS-10K**		
	P@1	**P@5**	**R@10**	**P@1**	**P@5**	**R@10**
[1M,∞]	0.347	0.207	0.312	0.372	0.213	0.318
[100k ,1M]	0.270	0.161	0.237	0.283	0.173	0.255
[10k ,100k]	0.256	0.156	0.229	0.276	0.172	0.252
[1k,10k]	0.247	0.153	0.224	0.262	0.166	0.244
[0,1k]	0.265	0.175	0.252	0.273	0.180	0.262

We design the same experiment on this set of parameters and name it BFS-10K. The results in Table 10 show that nearly every measurement improves, especially when the number of followers is over 10k, indicating a great improvement in the performance after changing the training corpus. This fully indicates that the performance of expanded tags still has room for improvement and is closer to the real data when we use users with different ranges of followers for learning.

Finally, after some deep discussions, we draw several important conclusions:

- Tags expanded by our algorithms have relatively high word frequency and short length.
- If we only take into account the weight of social networks without considering the relevancy of the tags that users already have, the performance will decrease greatly.
- After using a new training set with users having over 10k followers on our algorithms, we discover that the performance of our algorithms increases greatly.

7 Conclusion

This paper puts forward and defines the problem of expansion of self-tags of users in social networks. Under our definitions of four types of social relationships (BFS, OFS, FS and ALL), we discover that users' tags are more similar to the tags of their bidirectional following friends (BFS). We only choose as features the tag frequencies of users' social relationships, the conditional probabilities of friends' tags given users' self-tags and the prior probabilities of tags and adopt energy-based learning to build a model and use a negative log-likelihood loss as the loss function. The experimental results indicate that our algorithm outperforms the best baseline algorithm by over 14.0% on P@1, 14.4% on P@5 and 11.4% on R@10. Moreover, experiments also show that BFS is better for celebrities and that BFS as well as FS is better for common users.

Compared with traditional methods, our method inherits the previous work and imports friendship information into the modeling to gain an obvious improvement, which encourages us to go further in this direction. Future work will include the following: 1) exploration of more complex algorithms; 2) prediction of users' expanded windows; 3) cross-social network tag expansion; and 4) expansion of tags from the contents of microblogs of users' friends.

Acknowledgments. This work was supported by Natural Science Foundation (60903107, 61073071), National High Technology Research and Development (863) Program (2011AA01A205) and Research Fund for the Doctoral Program of Higher Education of China (20090002120005).

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. ACM SIGMOD Record 22(2), 207–216 (1993)
2. Bernstein, M., Tan, D., Smith, G., et al.: Collabio: A game for annotating people within social networks. In: Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology, pp. 97–100. ACM (2009)
3. Bowman, S., Willis, C.: We media: How audiences are shaping the future of news and information (2003)
4. Eck, D., Lamere, P., Bertin-Mahieux, T., Green, S.: Automatic generation of social tags for music recommendation. In: Advances in Neural Information Processing Systems, pp. 385–392 (2007)
5. Farrell, S., Lau, T., Nusser, S., et al.: Socially augmenting employee profiles with people-tagging. In: Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology, pp. 91–100. ACM (2007)
6. Goel, A., Gupta, P., Lin, J., et al.: Wtf: The who to follow service at twitter. In: Proceedings of the 22nd International conference on World Wide Web, pp. 505–514 (2013)
7. Garg, N., Weber, I.: Personalized tag suggestion for flickr. In: Proceedings of the 17th International Conference on World Wide Web, pp. 1063–1064. ACM (2008)
8. Heymann, P., Ramage, D., Garcia-Molina, H.: Social tag prediction. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 531–538. ACM (2008)
9. Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag recommendations in social bookmarking systems. AI Communications 21(4), 231–247 (2008)
10. Kucuktunc, O., Sevil, S.G., Tosun, A.B., Zitouni, H., Duygulu, P., Can, F.: Tag suggestr: Automatic photo tag expansion using visual information for photo sharing websites. In: Duke, D., Hardman, L., Hauptmann, A., Paulus, D., Staab, S. (eds.) SAMT 2008. LNCS, vol. 5392, pp. 61–73. Springer, Heidelberg (2008)
11. Kullback, S., Leibler, R.A.: On information and sufficiency. The Annals of Mathematical Statistics 22(1), 79–86 (1951)
12. LeCun, Y., Chopra, S., Hadsell, R., et al.: A tutorial on energy-based learning. Predicting Structured Data (2006)
13. Liang, B., Liu, Y., Zhang, M., Ma, S., Zhang, K.: Predicting Tags for None-tagged Person on SNS. Journal of Computational Information Systems 10(8), 3123–3132
14. Liang, B., Liu, Y., Zhang, M., Ma, S.: THUIRDB: A large-scale, highly-efficient index, fast-access key-value store. Journal of Computational Information Systems 9(6), 2347–2355 (2013)
15. Li, X., Snoek, C.G., Worring, M.: Learning social tag relevance by neighbor voting. IEEE Transactions on Multimedia 11(7), 1310–1322 (2009)
16. Lindamood, J., Heatherly, R., Kantarcioglu, M., et al.: Inferring private information using social network data. In: Proceedings of the 18th International Conference on World Wide Web, pp. 1145–1146. ACM (2009)
17. Liu, D., Hua, X.S., Yang, L., Wang, M., Zhang, H.J.: Tag ranking. In: Proceedings of the 18th International Conference on World Wide Web, pp. 351–360. ACM (2009)
18. Mislove, A., Viswanath, B., Gummadi, K.P., et al.: You are who you know: Inferring user profiles in online social networks. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 251–260. ACM (2010)
19. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: Homophily in social networks. Annual Review of Sociology 27(1), 415–444 (2001)

20. Muller, M.J., Ehrlich, K., Farrell, S.: Social tagging and self-tagging for impression management. Submitted as an Interactive Poster to CSCW (2006)
21. Chodorow, K.: MongoDB: The definitive guide. O'Reilly Media, Inc. (2013)
22. Rae, A., Sigurbjörnsson, B., van Zwol, R.: Improving tag recommendation using social networks. In: Adaptivity, Personalization and Fusion of Heterogeneous Information, pp. 92–99. Le Centre de Hautes Etudes Internationales D'Informatique Documentaire (2010)
23. Schenkel, R., Crecelius, T., Kacimi, M., Michel, S., Neumann, T., Parreira, J.X., Weikum, G.: Efficient top-k querying over social-tagging networks. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 523–530. ACM (2008)
24. Song, Y., Zhang, L., Giles, C.L.: Automatic tag recommendation algorithms for social recommender systems. ACM Transactions on the Web (TWEB) 5(1), 4 (2011)
25. Sood, S., Owsley, S., Hammond, K.J., et al.: TagAssist: Automatic Tag Suggestion for Blog Posts. In: ICWSM 2007 (2007)
26. Szomszor, M., Alani, H., Cantador, I., O'Hara, K., Shadbolt, N.: Semantic modelling of user interests based on cross-folksonomy analysis. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 632–648. Springer, Heidelberg (2008)
27. Wal, T.V.: Folksonomy coinage and definition (2007), `http://www.vanderwal.net/folksonnomy.html`
28. Wal, T.V.: Explaining and showing broad and narrow folksonomies (2005), `http://wwww.vanderwal.net`
29. Weng, J., et al.: Twitterrank: finding topic-sensitive influential twitterers. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining. ACM (2010)
30. Zheleva, E., Getoor, L.: To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In: Proceedings of the 18th International Conference on World Wide Web, pp. 531–540. ACM (2009)