tance of users, different relationships among users and actions they perform on objects can impact the final relevance ranking. After proposing the new social relevance model, we present a novel socio-textual relevance ranking technique to combine textual and social relevance rankings. Finally, in our experimental section, we evaluate the effectiveness of our proposed models and show that our new relevance ranking methods are effective and improve the accuracy of the returned results.

2. RELATED WORK

There are two groups of related work on the application of social networks in search. With the first group, people through their social networks are identified and contacted directly to answer web queries. In other words, queries are directly sent to individuals and answer of the queries are coming from people themselves [1, 16, 17]. In this approach called *search services*, people and their networks are indexed and a search engine has to find the most relevant people to send the queries/questions to.

The main focus of the second group is on the search process over social data (tags, users and objects) from sites/application with social aspect such as social tagging sites and (some) Web 2.0 applications. In [2], authors investigate a personalized social search engine based on users' relations. They study the effectiveness of three types of social networks: familiarity-based, similarity-based and both. In [4], which is a short paper, authors propose two search strategies for performing search on the web: textual relevance (TR)-based search and social influence (SI)-based search. In the former, the search is first performed according to the classical tf-idf approach and then for each retrieved document the social influence between its publisher and querying user is computed. The final ranking is based on both scores. In the latter, first the social influence of the users to the querying user is calculated and users with high scores are selected. Then, for each document, the final ranking score is determined based on both TR and SI. In both strategies, two separate costly steps are needed. Also, it is not clear how accurate are the ranking functions since there is no experimental evaluation for the effectiveness of the rankings.

In a set of similar papers [5, 6, 7], authors propose several social network-based search ranking frameworks. The proposed frameworks consider both document contents and the similarity between a searcher and document owners in a social network. They also propose a new user similarity algorithm (MAS) to calculate user similarity in a social network. In this set of papers, the focus is more on user similarity functions and how to improve those algorithms. Most of their experiments are limited to a small number of queries on YouTube videos with 3 users, 15 queries and small number of textual keywords. Relevant (interesting) result is a result (video) whose category is simialr/equal to the dominant category of videos that searcher has uploaded.

In a relatively older paper [8], authors explore the possibility of using online social networks to improve the search on the Internet. Although this paper is not very technical , it provides some interesting intuitions on integration of social networks and web search. With regards to commercial search engines, Bing and recently Google have started to integrate Facebook and Google+, respectively, to their search process. For some search results, they show query issuer's friends (from his/her social network) that have *liked* or *+1ed* that result. Their algorithms are not public and it seems that they only show the *likes* and *+1s* and the actual ranking is not affected.

There exists a relevant but somehow different topic of *folksonomies*. Tags and other conceptual structures in social tagging networks are called folksonomies. A folksonomy is usually interpreted as a set of user-tag-resource triplets. Existing work for *social search* on folksonomies is mainly on improving search process over social data (tags and users) gathered from social tagging sites [10][11][12]. In this context, relationships among users and tags and also among tags themselves are of significant importance.

Finally, there are few studies on the role of collaborative filtering in this new social context. Role of social networks on collaborative filtering is studied in [3] and [13]. It is shown that using social networks in collaborative filtering and recommendations makes the recommendations better in comparison with the traditional collaborative approaches. In another direction, [14] studies the application of collaborative filtering on a movie search engine. Authors propose to calculate documents (movies) authorities based on users' ratings (using collaborative filtering) instead of pagerank and other link-based authority measures. Social networks of users are non-existent in this study.

In contrast to the above, our notion of *social search* is to utilize exiting social networks to improve the accuracy and relevance of convention textual web search. For us, search still has its core textual dimension, represented by textual keywords/content in the query and the documents. In parallel to the textual dimension, (querying) user's social network is exploited to make the final search results more relevant. Our focus is mostly on finding/modeling effective measures to calculate the social relevance/ranking and combine it with the existing standard textual relevance rankings. We also take into consideration the *actions* users perform on documents (as described in Section 3).

3. DEFINITIONS AND FORMALIZATIONS

In this section, we formally define and formalize the problem of socio-textual search.

Objects: We assume a collection $O = \{o_1,o_2,...o_n\}$ of n objects (documents). An object can be a traditional web document such as a news page or a business home page or a Web 2.0 object such as a YouTube video, a tweet, a Facebook status or any other similar entity. An object o is composed of a set of textual keywords K_o and a set of users U_o associated with it. U_o is a set of users with some type of *actions* on the object o (see *actions* below).

Users and Social Network: There is a set $U = \{u_1,u_2,...u_m\}$ of m users using the system. We also assume a social network modeled as a directed graph $G = (V,E)$ whose nodes represent the users of the system and edges represent the ties (relationship) among the users. The most common type of relationship is the friendship relationship but other type of relationships can be also applied (e.g. *follow* relationship in Twitter).

Actions: There is a set $A = \{a_1,a_2,...a_l\}$ of l actions that users can perform on the objects. These actions represent the relationship between users and objects. For instance, in Twitter, users can perform the following actions on objects (tweets): publish a tweet, retweet a tweet or make a tweet as their favorite tweet.

Socio-Textual Query: A socio-textual query is defined as $Q = \langle K_q, S_q \rangle$, where K_q is the textual part of query specified as a set of keywords in the query and S_q is the social part of query specified as the user u_q issuing the query and the social network G. Since our social network is always G, it is sufficient to define the socio-textual query as $\langle K_q, u_q \rangle$. Note that while the textual part of the query is always explicit in the query, the social part is often implicit. In other words, we can safely assume that the system (search engine) knows the user issuing the query and also the underlying social network, hence the social part of the query can be automatically added to the textual part by the search engine.[1]

User relevance: User u_i is relevant to user u_j if the network distance from the node corresponding to u_i to the node corresponding to u_j is less than or equal to a system defined *threshold value* δ. The less the distance between two nodes, the more *(user) relevant* are those two nodes (users) [2]. Network distance can be any of the existing network distances in the literature. Two users with the network distance more than δ are considered non-relevant to each other.

Social relevance: Social relevance between the object o and the query q is defined based on the social relationship that exists between the querying user (u_q) and users associated with the object o (U_o). Object o and query q are socially relevant if at least one of the object's users (U_o) is *user relevant* with the user issuing the query. The larger the user relevance is, the more socially relevant o and q are. We denote social relevance of object o to query q by $socRel(o, q)$. We define social relevance in more details in Section 4.

Textual relevance: Object o is textually relevant to the query q if there exists at least one keyword belonging to both o and q, i.e., $K_q \cap K_o \neq \emptyset$. We represent textual relevance of object o to query q by $texRel(o, q)$. [3]

Socio-textual relevance: Object o is social-textual (socio-textual) relevant to the query q if it is both socially and textually relevant to the query q. Socio-textual relevance can be defined by a monotonic scoring function F of textual and social relevances. For example, F can be the weighted sum of the social and textual relevances:

$$F(o, q) = \alpha.socRel(o, q) + (1 - \alpha).texRel(o, q) \qquad (1)$$

. α is a parameter assigning relative weights to social and textual relevances. The output of function $F(o, q)$ is the socio-textual relevance score of the object o for the query q, and is denoted by $stRel(o, q)$. In Section 4 we show how to calculate socio-textual relevance.

Socio-textual search: A socio-textual search identifies and ranks all the objects that are *socio-textual* relevant to the query q. The result is the top-k objects sorted based on objects' socio-textual relevance scores. The parameter k is determined by the user.

4. SOCIAL RELEVANCE RANKING

In this section, we propose a new social relevance model to calculate the social relevance between users and objects. We also show how to combine the proposed social relevance model with an existing textual relevance model and introduce our socio-textual relevance ranking.

We first propose a new scoring approach to calculate the social relevance between an object o and a query q (issued by user q_u). Our social relevance ranking creates a new scoring framework to retrieve and rank objects based on the social dimension of the query and objects. In order to have an accurate scoring function and retrieve the most socially relevant results to the user, we consider three important factors: (1) relevance of each user to the query's user, (2) importance of each user in general, and (3) relationship between users and actions they perform on each object. In the following we discuss each measure.

User Relatedness. We measure the relatedness of a user to the querying user (and hence to the query itself) by the *user relatedness function* $urf(u_q, u_i)$. There are several measures to calculate the relatedness/closeness of two nodes in a graph/social network. Some of the approaches consider the distance between nodes, some look at the behaviors of users in a social network and some take into consideration number of mutual neighbors of two nodes. While the required data is available, any of the above methods or other exiting methods can be used for the user relatedness function as long as the following three constraints are satisfied: (1) $urf(u_i, u_i) = 1$, (2) $0 \leq urf(u_i, u_j) \leq 1$ and the more relevant the users, the higher the value, and (3) $urf(u_i, u_j) = 0$ when $urf(u_i, u_j) < \delta$. The first constraint states that each user is the most related user to herself. The second constraint normalizes this measure and also ensures that the more related users are assigned higher scores. Finally, third constraint filters out all relationships that their significance is below a certain threshold (δ). As a simple example satisfying all the above constraints and also capturing the relatedness among users, we can use an inverse of distance between users (nodes) in the social network (graph) as follows:

$urf(u_i, u_j) = \frac{1}{dist(u_i, u_j)}$

where $dist(u_i, u_j)$ is the number of edges in a shortest path connecting u_i and u_j.

User Weight. We quantify the overall (global) importance of each user by the *user weight function* $uwf(u_i)$. This measure quantifies the significance of a user in its social network. For instance, for Twitter, a user with many followers should be assigned a higher weight than a user with only few followers, or for Facebook, a user with more friends is more important to the social network than a user with fewer friends. In the field of graph theory and social networks this value is called *centrality* and there exist several approaches to measure it. Four popular methods to compute centrality are: degree centrality, betweenness, closeness, and eigenvector centrality. [9] is a good resource For a review of these methods and further reading. Similar to the user relatedness function, the user weight function is also general enough and most of the existing approaches can be applied to uwf. As an example for this function we can use the degree centrality of nodes (users) as an indication of their importance as follows:

$uwf(u_i) = \frac{deg(u_i)}{m-1}$

where $deg(u_i)$ is the number of edges incident upon u_i and m is number of nodes (users).

[1] Naturally, here and in other parts of this paper, we consider only users who willingly make their social information public to the system.

[2] For simplicity of presentation, from now on, we assume that users' social network is implemented as an undirected graph. Hence, user relevance and other relationships between users will be symmetric.

[3] In this paper, we do not focus on textual relevance models. We use popular tf-idf model when we need to calculate textual relevance.

User Action. The importance of each user for each object is directly related to the action(s)[4] user perform on each object. Publishing/owning an object by a user shows a higher weight/relevance between the object and the user than only commenting on the object. For instance, a user uploading (and thus owning) a YouTube video is more significant to that video than a user who only *comments* on that video. The importance/relevance of each user to an object is measured by the *user action function* $uaf(u_i, o_k)$ and is dependant on the type of action user u_i performs on object o_k. For each system, weight/significance of each action should be determined based on specific characteristics of that system. We normalize the value of uaf by assigning values between 0 and 1 (inclusive) to it. The higher the value is, the more important/relevant is the user to the object. With some systems, there exist actions that can be performed multiple times by a user on an object, and the more the action is performed the higher is the relevance between the user and that object. For instance, in an online radio website (e.g. last.fm[5]), action *listening* to a track can be done multiple times by a user. The more the user chooses to listen to a track, the more relevant/significant is that track to the user. Below, we show examples of different actions and their corresponding weights for four popular web 2.0. objects. Note that the assigned weights are only our suggestion, they can (and should) be easily changed for different applications and/or settings. Examples are as follows:

- YouTube videos: Actions $= \{own(publish) : 1, favorite : 0.9, like : 0.7, comment : 0.4\}$.
- Twitter tweets: Actions $= \{tweet(publish) : 1, favorite : 0.9, retweet : 0.5\}$.
- Facebook objects: Actions $= \{own(publish) : 1, like : 0.8, share : 0.6, comment : 0.4\}$.
- last.fm tracks (songs): Actions $= \{like : 0.8, tag : 0.5, comment : 0.4, listen : \frac{playcount}{max\ playcount}\}$.

Note that for last.fm, we see an example of an action (*listen*) that can be performed multiple times (keep in mind that many other actions in our examples also can be performed multiple times). The above model can be applied to other object types for different web, web 2.0 and non-web objects. It is simple, flexible and easy to update/change based on different applications and purposes. It shows and quantifies what users are important/relevant for each object and how much is this relevance/importance for each user/object.

Now, we propose the final scoring function to calculate the social relevance between object o and query q as follows:

$$socRel(o, q) = \sum_{v_i \in U_o} urf(u_q, v_i) \times uaf(v_i, o) \times uwf(v_i) \quad (2)$$

In Equation 2, u_q is the user issuing the query and U_o is the set of users with some *actions* on the object o. While in classical textual relevance models such as tf-idf, more weight is given to the objects (documents) with 1) more number of query keywords (tf), and 2) more important query keywords (idf), in our social relevance model, more weight is given to the objects with 1) more important actions 2) performed by more important users 3) whom are more related (closer) to the querying user.

4.1 Socio-Textual Search

In this section, we combine social relevance with an existing textual relevance model (tf-idf) to calculate the overall socio-textual relevance of the object o with query q with regards to both social and textual dimensions. Socio-textual relevance ranking considers both the textual relevance of the objects to the query and also the social relevance of the objects to the query. We formulate the socio-textual relevance ranking as follows:

$$\begin{aligned} stRel(o, q) = &\quad \alpha \times socRel(o, q) + (1 - \alpha) \times texRel(o, q) \\ = &\quad \alpha \times \sum_{v_i \in U_o} urf(u_q, v_i) \times uaf(v_i, o) \times uwf(v_i) \\ + &\quad (1 - \alpha) \times \sum_{t_j \in K_q} tf(o, t_j) \times idf(t_j) \end{aligned} \quad (3)$$

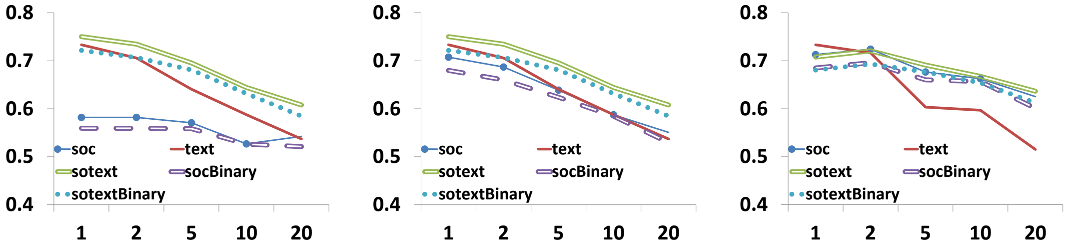
where $stRel(o, q)$ is the socio-textual relevance of object o to query q where user u_q is the query issuer; *socRel* and *texRel* are corresponding social and textual relevances for object o; urf, uaf and uwf are user-related functions as described above; K_q is set of query keywords (tags) and t_js are individual query keywords (tags); $tf(o, t_j)$ is *term frequency* function determining relevance of term t_j to object o; $idf(t_j)$ is *inverted document frequency* function determining the importance of keyword t_j in the entire collection; and α is a parameter giving relative weights to social and textual importance. Not only the implementation of urf, uaf and uwf functions are flexible (see Section 4), also the implementation of *texRel* is flexible. Although, we used the conventional *tf-idf* model for capturing the textual relevancy, any other textual relevance (similarity) function can be also used. Equation 3 provides the general framework for calculating socio-textual relevance and implementation and/or importance of each weight can be changed based on the context and users/applications needs.

5. EXPERIMENTAL EVALUATION

In this section, we evaluate the effectiveness of our proposed approaches. First, we describe the dataset, the settings and the queries used for the experiments. Next, we show and discuss the results.

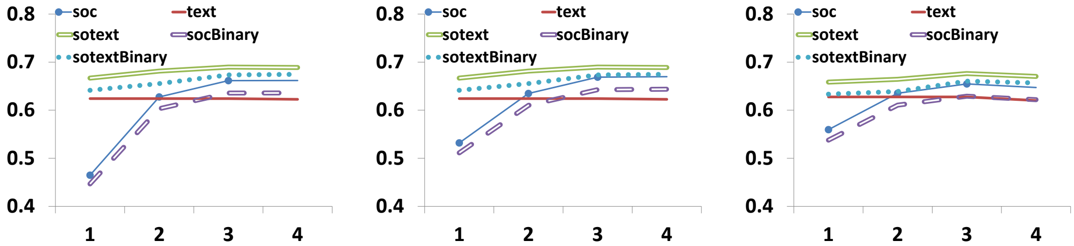
Data. There are very few publicly available datasets for experimentation that include both friendships (social network) and textual keywords (tags). One very good dataset is a dataset generated by [3] from a Web 2.0 website *last.fm*. Since this dataset has both social and textual components needed for our setting, we used this dataset. Last.fm is a music social network that allows users to listen to different music tracks, tag them with textual keywords and at the same time make friendships with other people on the network. While the users listen to a track they have the ability to either move to the next track of the playlist or keep listening to the same. These actions can be interpreted as explicit negative and implicit positive feedback respectively

[4]for simplicity, we assume that each user can perform at most one action on each object. However, our model can easily be generalized for multiple actions per user.
[5]http://www.last.fm/



(a) Setting 1 (b) Setting 2 (c) Setting 3

Figure 1: Impact of k on nDCG



(a) Setting 1 (b) Setting 2 (c) Setting 3

Figure 2: Impact of δ on nDCG

[3]. The dataset used contains 3148 users, 30520 tracks, 12565 tags and 5616 unique bonds of friendship among the users collected, which was made freely available by [3]. In our context (search), each track is a document (object) consisting of several textual keywords (tags). Users search for desired documents (tracks) by specifying one or more textual keywords (tags).

Actions. The only information available from last.fm website is the number of times a user listens to a track. This value is called *playcount* and is a very important indicator of relevance/importance between the user and the track. This is an action that can be performed multiple times on a track (user listening to a track multiple times). We use this action to calculate the value of *user action function* as follows: $uaf(u_i, o_k) = \frac{playcount(u_i,o_k)}{max-playcount(u_i)}$ where $playcount(u_i, o_k)$ is the number of times user u_i listens to track o_k without skipping the song and $max-playcount(u_i)$ is the maximum playcount value among all tracks that user u_i has listened to.

Queries. For each query, we randomly chose 1 to 3 textual keywords (tags) from the list of all the tags in our dataset, and one random user from the list of all users with the minimum of 4 friends in the system. We filtered out queries that did not generate any results. Queries are performed in *rounds*. Each round consists of 100 queries and is conducted for each input setting.

Ground Truth. To evaluate our results, we have to compare them with a ranking that is the most relevant ranking to the user (ground truth). Since *playcount* indicates the real interest of each user to each track, we leverage *playcount* to construct the most relevant list (ranking) for each query as follows: for each query, we return list of all textually relevant tracks (tracks contain one or more of the query keywords), order them based on the querying user's playcount values and return the top k results.

Approaches. We computed top-k query results for each query using the following five approaches: *soc*, *text*, *sotext*, *socBinary* and *sotextBinary*. *soc* approach generates the results based on the social relevance model only (presented in Section 4). *text* approach generates the results based on the conventional tf-idf relevance model only. *sotext* approach computes the results based on our socio-textual relevance model discussed in Section 4.1. Finally, *socBinary* and *sotextBinary* are simplified versions of *soc* and *sotext* approaches in which action *listening* only has the binary value 0 or 1 (instead of the actual playcount value). In other words, the value of *user action function* is calculated as follows:

$uaf(u_i, o_k) = 1$ if $playcount(u_i, o_k) > 0$, $uaf(u_i, o_k) = 0$ otherwise.

For each query, when using *soc*,*sotext*, *socBinary* and *sotextBinary* approaches, we do not use the existing information regarding the relationship/actions between the querying user and the objects (tracks). This is done to be fair and be able to evaluate the social component of the approaches based only on user's social network and friends.

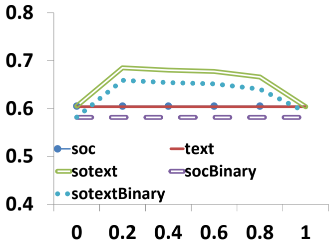
Settings. We evaluate the results for three different settings. *Setting1* is the default setting with all the details described so far. *Setting2* is a subset of *setting1* in which queries that generate fewer than k results are pruned (and not evaluated). Finally, *setting3* is a subset of *setting2* in which queries with querying users with less than 8 immediate neighbors are pruned.

Evaluation Metric. We evaluated the accuracy of the methods under comparison using the most common standard metric: nDCG (normalized discounted cumulative gain) [15]. When computing nDCG, we consider the playcount value as the relevance value. IDCG (ideal DCG) is the results generated by our ground truth.

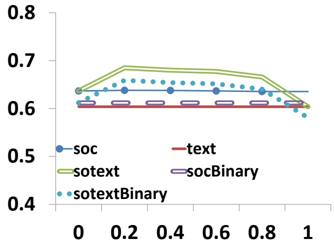
5.1 Results

5.1.1 Varying k

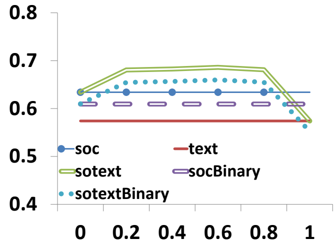
With the first set of experiment, we evaluate the effectiveness of the proposed approaches by varying number of requested results k. We report the average nDCG for each round. Here, we fix the number of keywords at 1, *alpha* at 0.5 and the threshold value δ at 2. The value of k varies from 1 to 20. The results for three settings are shown in Figures 1(a), 1(b) and 1(c), respectively. The first observation is that *sotext* is the most effective approach among all the three settings. This is very promising since it shows that



(a) Setting 1



(b) Setting 2



(c) Setting 3

Figure 3: Impact of *alpha* on nDCG

combining the textual relevance and social relevance using our model generates more relevant/accurate results in comparison with using only textual relevance or social relevance. The second observation is that *sotext* and *soc* are superior to their corresponding binary approaches (*sotextBinary* and *socBinary*). This implies that using a more detailed action model will improve the accuracy of the results. The third observation is that the results for our approaches are getting better from *setting1* to *setting2* and from *setting2* to *setting3*. This shows that 1) social-related approaches are even better for more realistic settings, and 2) socially-related approaches generate more accurate results when users have more neighbors (more socially connected).

5.1.2 Varying δ

In the second set of our experiments, we evaluate the impact of changing the threshold value δ. For different rounds, we set the threshold value to 1,2,3 and 4. In this set of experiments, we fix the number of query keywords at 1, k at 5 and *alpha* at 0.5. The results for three settings are shown in Figures 2(a), 2(b), and 2(c), respectively. Again, for all cases *sotext* easily outperforms the other four approaches. As expected, the accuracy increases for socially-related approaches as the threshold value increases (and obviously no change for *text* approach). Again, these figures confirm the observation that *sotext* and *sotextBinary* are superior than *sotextBinary* and *socBinary* approaches.

5.1.3 Varying *alpha*

In the final set of our experiments, we evaluate the impact of changing the value of *alpha* (relative weight of social and textual relevances) on the effectiveness of the proposed approaches. We vary the value of *alpha* from 0 (social only) to 1 (textual only). In this set of experiments, we fix the number of query keywords at 1, k at 5 and δ at 2. The results for all three settings are shown in Figures 3(a), 3(b), and 3(c), respectively. The obvious observation is that the results do not change for textual or social only approaches. The more interesting observation is the behavior of the two socio-textual approaches. While both show their poorest results on the boundaries (only social or only textual), they present their best accuracy in the middle of the range (when both textual and social relevance are considered almost equally). We have to note that for most cases, the best accuracy is achieved when the social relevance has a little more weight. Again, this set of experiments confirm the above observations regarding the superiority of *sotext* and also the improved accuracy of *setting2* and setiing3.

6. CONCLUSION

In this paper, we introduced the problem of ranking web documents based on both their social and textual features. We proposed a new scoring model to calculate social relevance between documents and users. The proposed social relevance ranking utilizes the querying user's social network and actions her friends perform on web documents (objects) to generate more accurate results for her (textual) searches. We also showed how to combine the new social relevance with the textual relevance model. We performed a set of experiments on the real dataset of last.fm and proved that the new approach is superior to the existing approaches.

7. REFERENCES

[1] D. Horowitz et al., The anatomy of a large-scale social search engine. WWW 2010.

[2] D. Carmel et al., Personalized social search based on the users social network. CIKM 2009.

[3] I. Konstas et al., On social networks and collaborative recommendation. SIGIR 2009.

[4] P. Yin et al., On top-k social web search. CIKM 2010.

[5] L. Gou et al., SNDocRank : Document Ranking Based on Social Networks. WWW 2010.

[6] L. Gou et al., SNDocRank: A social network-based video search ranking framework. MIR 2010.

[7] L. Gou et al., Social network document ranking. JCDL 2010.

[8] A. Mislove et al., Exploiting social networks for internet search. HotNets 2006.

[9] L.C. Freeman et al., Centrality in social networks: Conceptual clarification. Social Networks, 1(3), 215-239, 1979.

[10] M. Rawashdeh et al., Folksonomy-boosted social media search and ranking. ICMR 2011.

[11] R. Schenkel et al., Efficient top-k querying over social-tagging networks. SIGIR 2008.

[12] S.A. Yahia et al., Efficient network aware search in collaborative tagging sites. VLDB 2008.

[13] F. Liu et al., Use of social network information to enhance collaborative filtering performance. Expert Syst. Appl. 37, 7, 2010.

[14] S. Park et al., Applying collaborative filtering techniques to movie search for better ranking and browsing. KDD 2007.

[15] K. Jarvelin et al., Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems 2002.

[16] T. Yan et al., CrowdSearch: exploiting crowds for accurate real-time image search on mobile phones. MobiSys 2010.

[17] M.J. Franklin et al., CrowdDB: answering queries with crowdsourcing. SIGMOD 2011.