Personalized Recommendation of Social Software Items Based on Social Relations

Ido Guy, Naama Zwerdling, David Carmel, Inbal Ronen,
Erel Uziel, Sivan Yogev, Shila Ofek-Koifman

IBM Haifa Research Lab
Mt. Carmel, Haifa 31905, Israel

{ido, naamaz, carmel, inbal, erelu, sivany, shila}@il.ibm.com

ABSTRACT

We study personalized recommendation of social software items, including bookmarked web-pages, blog entries, and communities. We focus on recommendations that are derived from the user's social network. Social network information is collected and aggregated across different data sources within our organization. At the core of our research is a comparison between recommendations that are based on the user's *familiarity network* and his/her *similarity network*. We also examine the effect of adding explanations to each recommended item that show related people and their relationship to the user and to the item. Evaluation, based on an extensive user survey with 290 participants and a field study including 90 users, indicates superiority of the familiarity network as a basis for recommendations. In addition, an important instant effect of explanations is found – interest rate in recommended items increases when explanations are provided.

Categories and Subject Descriptors

H.5.3 Information Interfaces and Presentation: Group and Organization Interfaces – Collaborative computing; H.3.3 Information Search and Retrieval – Information filtering.

General Terms

Algorithms, Measurement, Experimentation, Human Factors.

Keywords

Recommender systems, social networks, social software, social media, personalization.

1. INTRODUCTION

Social software – software that allows users to share and interact – is proliferating [27]. Blogs, wikis, social bookmarking, and social network sites (SNSs) are a few prominent examples of social software applications (also commonly referred to as social media applications) that have been gaining a great amount of popularity in recent years.

Yet, as more and more social software sites keep popping up, it is becoming more challenging for them to attract and retain users, and for users to decide which sites to try out and keep track of. Furthermore, users are "flooded" with information from feed readers, news alert systems, blog services, and many other resources. The access to so much information, without knowing the validity of the content and the risk of misinformation, might lead to *Information Overload,* i.e., having more information available than one can readily assimilate.

One way to address these challenges is to provide users with personalized recommended items. The goal of such recommendations is to adapt the content of the site to the specific needs of the individual user, presenting to him/her the most (and only) attractive and relevant items.

In recent years, many third party services for personalized recommendations in leading social software sites have emerged. For example, InSuggest[1] provides personalized recommendations of bookmarks originating from the social bookmarking site Delicious[1], and Outbrain[1] provides personalized blog recommendations from leading blogging services, such as Blogger[1] and WordPress[1]. Following this trend, a few social software sites have lately added services for personalized item recommendations of their own. For example, social news aggregator service Digg[1] has launched a personalized recommender engine for presenting the most interesting stories to the user [20]. Social network news aggregator FriendFeed[1] has added personalized recommendations for showing the most interesting news items within the user's network [22].

In this work, we study personalized recommendations within an enterprise social software application suite. The fact that the application suite consists of different social software applications, each containing a different type of items, poses an even greater challenge in terms of information overload for users. We focus on recommending items of three applications: bookmarked web pages (bookmarks), blog entries, and communities.

The task of personalized recommendation requires the ability to predict which items will be considered interesting by the user. Such prediction is typically based on (1) content, i.e., recommending items with content that is "similar" to the content of the items already consumed by the users; (2) social networks, i.e., providing items related to people who are related to the user, either by explicit familiarity connection (e.g., by being connected

[1] {insuggest, delicious, outbrain, blogger, wordpress, digg, friendfeed}.com

in an SNS), or by some kind of similarity (e.g., by using similar tags, consuming similar documents, or having similar tastes as expressed in item rating). We note that we refer to social networks in their broad definition, i.e., networks of people; where connecting edges may represent any type of relationship, not only direct familiarity [30].

Content-based recommendation relies on the assumption that user interests are reflected by previous items they have consumed. This assumption has several drawbacks, among them the changes in user interests over time and the typical restriction to items similar to those already consumed [8]. In recent years, social network-based approaches, in particular collaborative filtering [7], have become more popular. These methods are based on the assumption that "similar" users share mutual interests. This leads to the question how similarity of interests can be measured. A key distinction made in this work is between the user's *familiarity* network and *similarity* network. The *familiarity* network consists of the people the user knows. The *similarity* network contains people whose social activity is found to be overlapping with the user's social activity.

To retrieve the user's social network we use the SONAR social network aggregation system [10]. SONAR extracts relationships between people from different sources across the organization and aggregates them to build the user's weighted social network. SONAR uses relationships such as a connection within an SNS, co-authorship of a wiki page, usage of the same tags, or bookmarking the same pages.

The comparison between the similarity and familiarity networks as a basis for recommendations is an intriguing one. On the one hand, people who share similar activity with you, not necessarily your friends, are the ones who are likely to indicate the most attractive items for you. Collaborative Filtering [7], one of the most popular techniques for recommendations, is based on user similarity. On the other hand, in real life people mostly seek advice from people they know. Within an enterprise, the people you are familiar with are typically colleagues with whom you work or have worked in the past, and are thus likely to be a source for interesting items. On top of that, explanation capabilities of recommender systems, if exist, play a key role in the comparison between similarity and familiarity. Clearly, in the case of familiarity, showing the list of people based on whom an item was recommended can help reasoning the recommendation, as the user is more likely to know their interest, tastes, and expertise. But showing similar people can also help justifying a recommendation, especially if the system can provide more information about their similar activity with the user. This is, for example, done in the recently announced personalized recommender system within the Digg site [20].

Our evaluation includes a user survey with 290 participants, who were randomly divided into three groups that received recommendations based on similarity network only, familiarity network only, and a network with both familiar and similar people. All groups received recommendations in two phases – without explanations and with explanations and participants were asked to provide feedback on their interest in recommended items. We also inspected our recommender system through a field study, which included 90 users and lasted three weeks. Our main contributions are (1) demonstrating that recommendation based on social relations is effective (In our study, the most effective configuration yields a 57% vs. 43% interesting to not interesting rating ratio); (2) showing that recommendations which are based on users' familiarity network are significantly more effective than those based on users' similarity network (and those based on a combination of both networks); and (3) suggesting a novel user-interface for people-based explanations and showing the instant effect of these explanations – adding them to recommended items increases users' interest in these items.

In the next section, we discuss how existing work relates to our research. We then present our recommender system, followed by a detailed description of our experiments and their results. We conclude by discussing our findings and suggesting future work.

2. RELATED WORK

Collaborative Filtering (CF) [7] has become a very common technique for providing personalized recommendations, suggesting items based on the similarity between users' preferences. One drawback of traditional CF systems is the need for explicit user feedback, usually in the form of rating a set of items, which might increase users' entry cost to the recommender system. Hence, leveraging implicit user feedback [4], such as views, clicks, or queries, has become more popular in recent CF systems. In this work, we leverage implicit social network information, which can be viewed as a variant of implicit user feedback. In the case of similarity network, the process is quite similar to CF with implicit user feedback, however we do not apply clustering or other computationally intensive methods after creating the user similarity network, as typically done in CF systems. This fact together with the use of public data only (bookmarks, tags, blogs) allows us to provide intuitive and simple explanations for each recommended item.

As mentioned in the introduction section, we also provide recommendations based on the user's familiarity network. Previous work on recommender systems has taken into account direct relationships between people. ReferralWeb [16] suggests a system to enhance searching for documents and people by combining collaborative filtering and social networks. Geyer et al. [6] compare methods of recommending 'About You' entries within an enterprise SNS. They show that a social network-based recommender outperforms a content-based one. Guy et al. [11] use aggregated familiarity relationships to recommend people to connect to within an enterprise SNS and show high impact of the recommender on the number of connections within the site.

The comparison between the similarity and familiarity networks as basis for recommendations is central to this work. A few previous papers have dealt with related questions. Groh and Ehmig [8] show that using explicit social network information (originating from a German SNS) improves regular CF techniques in taste related domains. Lerman [17] reveals that users of the Digg social news aggregator site are more interested in stories from their Digg friends than stories recommended using a CF system. In a lab study, Sinha and Swearingen [25] compare movie and book recommendations from friends with online recommender systems (such as Amazon.com) that use CF based on explicit user feedback. Friends' emails are manually provided by the study's participants. The authors note that in their experiment, users knew the origin of the recommendations, which created a probable bias in favor of the friends' recommendations. Bonhard et al. [3]

examine movie recommendations in a lab experiment and explore the effects of familiarity and similarity between the user and the people who have rated the items. Similarity is defined in two forms – profile similarity (age, hobbies, film genre preferences, etc.) and rating overlap. Familiarity is simulated through repeated exposure to profiles and thus does not reflect real acquaintance. They conclude that traditional recommender systems should be combined with functionality from social systems, such as SNSs, to achieve better results. They also provide explanations to recommended movies in the form of showing the profile of the recommending person and highlighting the overlaps with the user's profile. However, no quantitative experiment is conducted to evaluate the impact of these explanations.

In this work, we quantitatively measure the instant effect of explanations on users' interest in recommended items. Our focus is on people-based explanations. Herlocker et al. [12] emphasize the fact that most recommender systems are black boxes, providing no transparency into the working of the recommendation. They state that users will be more likely to trust a recommendation when they know the reasons behind it and suggest different interfaces for explaining movie recommendations in a traditional CF system. As recommendations are based on a complex mathematical model, explanations are far less intuitive and explicit than those suggested here and are mostly based on rating histograms of anonymous neighbors. Evaluation is based on general user satisfaction survey only and the hypothesis that explanation facilities can enhance filtering performance is left unproved. Guy et al. [11] suggest people-based explanations and show their value in the context of people recommendations through a user satisfaction survey and interviews. However, no experiment was conducted to prove the explanations' role in increasing acceptance rate. Surveys of explanations in recommender systems are provided by McSherry [18] and by Tintarev and Masthoff [28].

While the majority of the literature on recommender systems focuses on traditional taste domains such as music, movies, or books, in this work we recommend aggregated social software items that include bookmarks, blog entries, and communities. Some previous work has been done for recommending single types of social software items. For example, Dugan et al. [5] describe a prototype of a bookmark recommender system that uses a social game for encouraging users to manually recommend bookmarks to each other. Vatturi el al. [29] study personalized bookmark recommendations using a content-based approach that leverages tags. Java et al. [15] use feed subscription information from the Bloglines feed reader service to recommend feeds and blogs. Arguello et al. [2] propose a blog recommender system based on IR techniques and evaluate it over the TREC Blog06 dataset. Seth et al. [24] propose an approach to personalized recommendation of participatory media content (such as blogs and forums) using social networks and Bayesian user-model. Their evaluation is based on recommendation of messages within communities and focuses on the user-model. Finally, Spertus et al. [26] recommend online communities to users of the Orkut SNS and evaluate different similarity measures between communities, which are all based on user overlap between the communities. As aforesaid, in this work we study recommendations of three different types of social software items, and compare the recommender's impact on these three item types.

3. Recommender System Description

3.1 Lotus Connections (LC)

Lotus Connections (LC) [13] is a social software application suite for organizations that was introduced in 2007. It contains (as of version 2.0) five social software applications: profiles, activities, social bookmarks, blogs, and communities. We focus on recommending elements of the last three applications, which contain mostly public items. We disregard the other two as an activity is mostly restricted to a limited number of users and profiles pose a different challenge of people recommendations, which we dealt with in previous work [11]. Dogear [19], LC's social bookmarking application, allows users to store and tag their favorite web pages. Over 90% of the bookmarks are public (visible to all other users) and about half are intranet pages, while the other half are external internet pages. Dogear includes[2] 743,239 bookmarks with 1,943,464 tags by 17,390 users. Blog Central [14], LC's blogging system has[2] 16,337 blogs, 144,263 blog entries, with 69,947 users. LC's communities service contains[2] over 2,100 online communities, each with shared resources and discussions, with a total of over 50,000 members.

3.2 Social Network Aggregation

To generate the users' familiarity and similarity networks, we use *SONAR* [10], a system for collecting and aggregating social network information across different sources within the organization. Previous work [10] has indicated that in order to obtain a good representation of the user's familiarity network using public sources, information from a rich set of data sources needs to be aggregated. We thus aggregate relationships from different data sources across our organization that reflect familiarity relationships, similarly to the way it was done in previous papers [9][11]. In particular, we extract organizational chart relationships, direct (explicit) connections originating from two enterprise SNSs, direct tagging within a people tagging application, co-authorship in our organizations' projects-wiki, and co-authorship of papers and patents. Based on each of these relationships, a score between *0* and *1* is assigned to each pair of individuals (*0* stands for no relationship and *1* for the strongest relationship). The scores of the different relationships are averaged with an equal weight per relationship to generate a general familiarity relationship score. More details on the score calculation for each of the relationships can be found in [9].

For composing the similarity network, SONAR aggregates the following three types of relationships: co-usage of the same tag (where tags were originated from different applications, including the social bookmarking application, the people tagging application, and the blogging system); co-boomarking of the same web page; and co-commenting on the same blog entry. All of these relationships represent similar activity in the context of social software. Similarity score (in *[0,1]* range) for each of these three relationships is calculated using *Jaccard's index* (a.k.a Jaccard's similarity coefficient), i.e., by dividing the number of items in the intersection set by the number of items in the union set. For example, for bookmarks, the number of pages bookmarked by *both* users is divided by the number of distinct pages bookmarked by *any* of these two users. The overall

[2] As of March 17th, 2009

similarity score between two individuals is calculated by averaging the three similarity scores with equal weight.

Besides the familiarity and similarity networks, we also examine an *overall network*, by aggregating all types of relationships mentioned above (with equal weight per relationship) to create a network that includes both people who are familiar to the user and people who are similar to the user.

3.3 Item Recommendation Algorithm

In addition to using SONAR for aggregating people relationships, we use the unified search system [1], which maps users to related items, in our case bookmarks, blog entries, and communities. We use the following user-item relationship types and weights (extending the model given in [1]) : authorship (0.6), community membership (0.4), commenting (0.3), and bookmarking (0.3).

For each of the three networks - familiarity, similarity, and overall - we use SONAR to retrieve the list of top 30 related people and their corresponding relationship score to the user. We then suggest items to the user which are related to people within his/her network. The recommendation score of item *i* to user *u* is determined by the following formula:

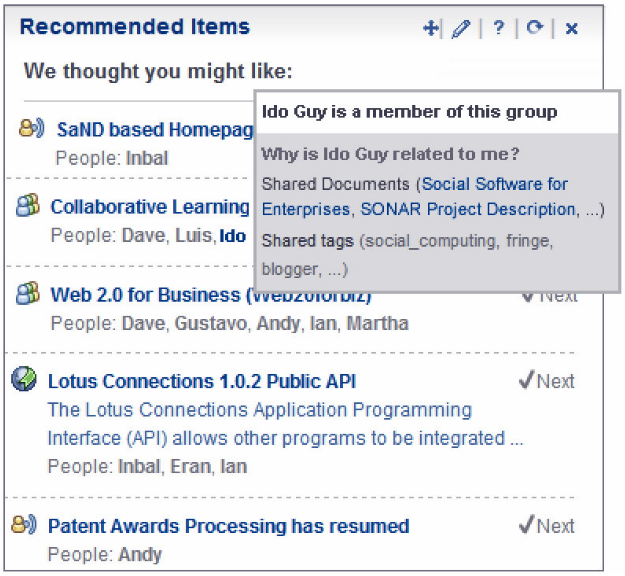
$$RS(u,i) = e^{-\alpha t(i)} \cdot \sum_{v \in N^T(u)} S^T[u,v] \sum_{r \in R(v,i)} W(r)$$

where *t(i)* is the number of days passed since the creation date of *i*; α is a decay factor (set in our experiments to 0.025); $N^T(u)$ is the set of users within *u*'s network of type *T*, *T∈{familiarity, similarity, overall}*; $S^T[u,v]$ is the SONAR relationship score between *u* and *v* based on the network of type *T*; *R(v,i)* is the set of all relationship types between user *v* and item *i*, given by the unified search system (authorship, membership, etc.); and *W(r)* *(r∈R(v,i))* is the corresponding weight for the user-item relationship type between user *v* and item *i* as described above. For example, given an item with *t(i)=1*; if only one person within the user's network is related to this item and this person is the author of the item, has bookmarked it, and has a 0.5 SONAR relationship score with the user, then the corresponding item score for the user will be calculated by *exp*(-0.025*1) **[0.5*(0.6+0.3)].*

Ultimately, the recommendation score of an item, reflecting its likelihood to be recommended to the user, may increase due to the following factors: more people within the user's network are related to the item, stronger relationships of these people to the user, stronger relationships of these people to the item, and freshness of the item. In addition, we exclude items that are found to be related to the user. For example, we will not recommend an item the user has already commented on or has already bookmarked.

3.4 Recommender Widget

Figure 1 depicts our widget for providing item recommendations based on the algorithm described in the previous section. The user is presented with five items consisting of a mix of bookmarked pages, communities and blog entries. Each item has a title which is a link to the original document and a short description if available. The icon to the left of each item symbolizes its type – the first item in Figure 1 is a blog entry, the second is a community, and the fourth is a bookmarked page. The user can remove an item in order to retrieve a new recommendation by clicking on the Next icon. Each recommended item includes a list of up to five person names that are related to the item. Each person provides an explanation of why the item is recommended (serving as an implicit recommender of the item). When hovering over a name, the user is presented with a popup detailing the relationships of that person to the user and to the item. In Figure 1 the recommended items are chosen according to the similarity network of the user. The popup indicates that Ido on the one hand is a member of the recommended community and on the other hand is similar to the user as they both share a set of documents and used the same tags. We assume that in the case of familiarity the names will mostly suffice as explanations, while in case of similarity, the popup will be used more often to inspect the common activity with a person.



Recommended Items

We thought you might like:

SaND based Homepag
People: Inbal

Collaborative Learning
People: Dave, Luis, Ido

Web 2.0 for Business (
People: Dave, Gustavo, Andy, Ian, Martha

Lotus Connections 1.0.2 Public API ✓Next
The Lotus Connections Application Programming Interface (API) allows other programs to be integrated ...
People: Inbal, Eran, Ian

Patent Awards Processing has resumed ✓Next
People: Andy

Ido Guy is a member of this group

Why is Ido Guy related to me?

Shared Documents (Social Software for Enterprises, SONAR Project Description, ...)

Shared tags (social_computing, fringe, blogger, ...)

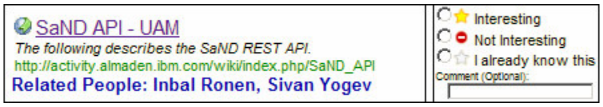
Figure 1. Item Recommendation Widget

4. USER STUDY

4.1 Methodology

We conducted an extensive survey with 290 users of LC in order to compare the familiarity and similarity networks as basis for recommendation, to examine the value of people-based explanations, and to compare the different types of recommended items. We considered users who had data for all three types of similarity relationships and at least two types of familiarity relationships. All of these users had at least 30 people in both their similarity and familiarity networks. We note that this sample does not represent the entire population of IBM employees, but rather active users of the LC system, who are the target population for our recommender system. We sent a link to the survey with a request for participation to 757 of these users and got a response from 290. On average, each participant had data originating from 3.08 types of familiarity relationships (stdev 0.93, median 3, max 6). Our survey participants originated from 28 countries, spanning the different organizational divisions: 33% Software, 21% Services, 17% Sales, 10% Research, 8% Headquarters, 6% Systems, and 5% others.

The survey consisted of two phases, each suggesting six LC items – two bookmarked pages, two blog entries, and two communities (all in a randomized order). Figure 2 demonstrates a recommended item presentation in the survey. Participants were asked to rate the items according to one of three options: *Interesting*, *Not Interesting*, or *I Already Know this*. Participants could also write free text comments per each of the recommended items and general comments at the end of the survey. Items were presented in a similar way to the widget described in Section 3.4, and included an icon representing the item's type, its title with a direct link to the item's page, a description (if existed), and its explicit URL. Participants were not explicitly told whether they need to click and further explore the item before rating it. The difference between the two phases of the survey was that one included explanations for its six items as previously described (showing related people and upon hovering their relationship to the user and to the item) and the other did not include any explanations. The order of the two phases was switched for each new user taking the survey.


SaND API - UAM
The following describes the SaND REST API.
http://activity.almaden.ibm.com/wiki/index.php/SaND_API
Related People: Inbal Ronen, Sivan Yogev

○ ★ Interesting
○ ⛔ Not Interesting
○ ☆ I already know this
Comment (Optional):

Figure 2. A recommended item from our survey

Each participant was assigned, in a round robin order, to one of three groups, corresponding to the three network types we set to examine: familiarity, similarity, and overall. All items in a user's survey were suggested according to the group s/he was assigned to (i.e., based on the corresponding network). We collected user feedback per item as well as clicks and hovers.

4.2 Results

User responses to our recommendations, as reflected in the survey's comments, were in general very positive, describing the recommendations as "*useful*", "*valuable*", "*relevant*", "*neat*", and "*right on spot*". One participant wrote "*It's great. I could read and find knowledge that I never search or even know about*". Another participant indicated that the recommendations triggered an extra action – bookmarking recommended pages: *"[…] just in the sample of 12 items, I found things I had to add to my personal bookmarks. Well done!"*. Another action that was triggered as a result of the recommendations was joining a recommended community, e.g.: "*I'm glad I took this survey as it made me join a couple of communities of which I was unaware*". Several users pointed out they would like to have more control over the list of people based on which items are suggested and over the actual content that mostly interests them. For example, one participant wrote "*I found myself thinking that I want to rate the related people: credible or not, talkative or not (tied to credible) […] or some other feedback like the thumbs up in Facebook*".

As aforesaid, our main goal of the survey was comparative. Table 1 summarizes the results of the survey for each of the two phases (with and without explanations) and per each phase, the rating percentage for each of the three groups (familiarity, similarity, overall). The last column shows the ratio between interesting and non-interesting items. The best ratio is achieved by the familiarity network with explanations. In the following we analyze those results in more detail.

One question that arises when analyzing the results is how to refer to the "already know" feedback. Since we did not have information about past user's views and clicks in the real LC system, we could only remove items related to the user by more explicit means, such as authoring or bookmarking. We therefore feel that hitting an item the user had already seen is an indication for the potential of the system to identify interesting items. However, since in an ideal recommender system these items would have been removed, we mainly focus on the ratio between interesting and not interesting ratings. We refer to the "already know" percentage as an indicator for recommendations' expectedness – the higher it is the more expected are the results. Another question related to result analysis is the treatment of click information on items. In a survey setup, users are likely to go over the items one by one and click on them to decide on the rating (e.g., one participant wrote that he "*assumed that it was needed to open a link to evaluate it*"). Hence, we cannot ascribe any particular significance to clicks in this survey. In a live system, however, click information is highly useful to reflect users' interest in items (see Section 4.3).

Table 1. Survey Results Summary (in %)

Phase	Group	Interesting	Not Interesting	Already Know	Int. Ratio
No Expl.	Familiarity	**44.01**	38.58	17.41	1.14
	Similarity	38.07	**48.82**	13.11	0.78
	Overall	**43.18**	39.34	17.48	1.10
Expl.	Familiarity	**47.32**	35.70	16.98	**1.33**
	Similarity	39.27	**47.52**	13.21	0.83
	Overall	**44.35**	37.94	17.71	1.17

4.2.1 *Familiarity or Similarity?*

Figure 3 depicts the overall rating results (over both phases) for all items in each of the three groups. The group who received recommendations based on the familiarity network has the best interesting to non-interesting rating ratio (45.7% interesting vs. 37.1% not interesting), followed by the overall network (43.8% vs. 38.6%), and finally the similarity network, with a negative ratio (38.7% vs. 48.2%). The familiarity's rating result (the percent of interesting items) is significantly better than the similarity's rating result (one-tailed unpaired t-test, p<0.01), but not significantly better than the overall's result.

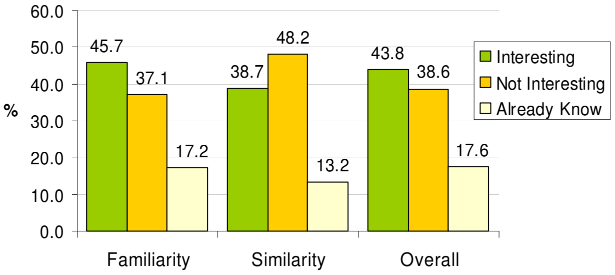


Figure 3. Rating results (over both phases) in each of the three groups

Inspecting Table 1, it can be seen that the differences in rating results between the groups are consistent, both when explanations are provided and when they are not provided. However, only in the case where explanations are provided the differences between the familiarity and similarity groups are significant (one-tailed unpaired t-test, p<0.05). The fact that differences exist between the familiarity and similarity networks without explanations indicates that there is a difference in the quality of items each network produces in favor of the familiarity network. Yet, when explanations are provided, these differences become significant, thus indicating that explanations for the familiarity network are more effective.

The ratings of the overall network are in-between similarity and familiarity (leaning towards familiarity), indicating that there is no effect of the whole being greater than the sum of its parts – considering the familiarity network on its own for recommendation is better than considering a network based on both familiarity and similarity relationships.

4.2.2 Explanations

Figure 4 shows the overall rating results (over all three groups) for all items presented with and without explanations. Items for which explanations were provided were rated with a better interesting to non-interesting ratio (43.7% interesting vs. 40.2% not interesting) than items for which explanations were not provided (41.8% vs. 42.2%). These differences are found to be non-statistically significant; however, as can be seen in Table 1, they are consistent over the three groups: interest ratio increases for all types of networks when explanations are added. This interesting result points at the immediate value of explanations in the form we provided them – despite the fact that the explanations have no direct effect on the relevance of items, they instantly raise the interest rate in the corresponding recommended items. This positive effect of explanations adds to longer-term benefits of explanations pointed out in previous works, in terms of understanding and trusting the system, and likelihood to reuse it [12]. Quite a few of the comments we received referred to the value of explanations. For example, one participant said that "*the explanations let me understand the reason for an item showing up even if I did not find it interesting*" and another wrote "*The people I knew helped a lot to add a frame of reference and a factor of pre-knowledge […]since I knew the person I had some idea of how the topic is related*".

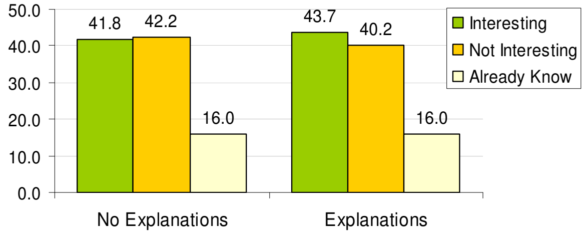


Figure 4. Rating Results (over all three groups) with and without explanations

The most evident effect of explanations is achieved for the familiarity group (interest ratio increases from 1.14 to 1.33). This again shows that familiar people have higher impact when presented as explanations than people with similar activity. Yet, there is some positive effect of explanation using similar people as well – interest ratio increases for the similarity network from 0.78 to 0.83. When inspecting usage of the hovering feature, more hovers were performed over similar people when they appeared as evidence than over familiar people (overall 738 hovers in the similarity group compared to 534 hovers in the familiarity group). This indicates that as we expected, people were using the hovering feature to explore people they do not know more often. It could be that this exploration contributed to the overall impact of explanations in the similarity group.

4.2.3 Bookmarks, blog entries, and communities

Figure 5 depicts the rating results for each of the three item types (over both phases and all three groups). Results show reasonable interest percentage across all three types, with bookmarks having the best interest ratio (1.18), followed by blog entries (1.05), and finally communities (0.93). The diversity between the types is most noticeable w.r.t the percentage of already known items: bookmarks have a very high percentage of known items - 27%, while communities have the lowest one – only 7.4%. This may indicate the potential "surprise effect" of community recommendations, possibly due to their relative low exposure to users. Quite a few of the survey participants indicated that due to the recommendations, they asked to join a community of interest they had not been aware of. One of them wrote: "*I was surprised to find out there is a Collaborative Learning community in our organization and asked to join it thanks to the survey*".

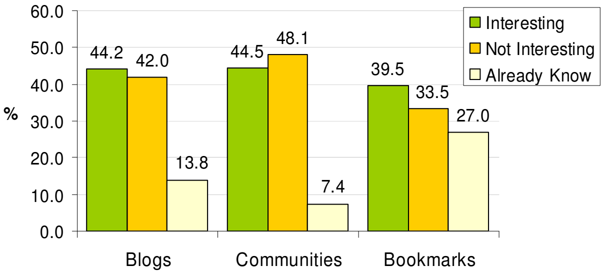


Figure 5. Rating results for each of the three item types (over both phases and three groups)

4.3 Field Study

We deployed the widget depicted in Figure 1 on the homepage of Lotus Connections within our organization for a period of three weeks. We assigned each user of the widget to one of six groups (in a round robin order) – getting recommendations based on either familiarity, similarity, or overall, with or without explanations. We refer to clicks over item links as an indication of users' potential interest in them.

Results, based on 90 users (due to technical constraints, we could not expose the widget to a larger audience), support the main findings from the user survey. Users that received recommendations with explanations clicked on substantially more items than users who received recommendations without explanations (109 clicks vs. 76 clicks in total). This behavior is consistent across all three network types – familiarity (44 vs. 32), overall (42 vs. 29), and similarity (23 vs. 15). This finding supports the important role of explanations in instantly attracting users to recommended items, as increasing the number of user clicks is the ultimate indication for successful recommendation.

Also supporting the survey's results is the fact that users who received recommendations based on familiarity network clicked on more items than users who received recommendations based on the similarity and the overall networks (76 vs. 38 and 71 respectively). This reinforces the conclusion about the superiority of the familiarity network as a basis for recommendations.

5. DISCUSSION

Results from both the large user survey and the field study show a clear superiority of the familiarity network over the similarity network when used for predicting a user's interests. As we conducted our experiments within an enterprise setting, it can be concluded that colleagues serve as better information filters for social software items than employees with high overlapping social software activity. This result is more expected in the case where explanations are provided, since users are exposed to the related people and, in case of familiarity, see familiar names of colleagues who are related to the items. Some users indicated in the survey that presenting the items with the related colleagues or friends, serves analogously to SNS's feature of showing news within your network (e.g., Facebook's well-known "News Feed" feature [21], or the key functionality of FriendFeed.com).

In the case where explanations are excluded, familiarity was still clearly favored over similarity, which is a less expected result. It indicates that the differences between familiarity and similarity stem also from the quality of the corresponding items they produce (i.e., their quality as information filters for social software items) and not just from the presentation of related people. Colleagues simply produce a larger portion of interesting items than people with similar activity. One participant who was assigned to the similarity group said "*It seemed many of the people were from my extended social network - in some ways I think this could be interesting, but also has a greater chance at being less relevant*". Another participant who belonged to the overall group explicitly referred to the differences between colleagues and similar people: "*Implicit vs. explicit social connections are very different beasts. I look to my explicit social connections to discover things about areas I'm already interested in, and look to implicit connections to provide things that I'd be interested in but might not yet know about*".

The last comment highlights the fact that people with similar activity may yield more unexpected items than colleagues, which is also reflected in the percentage of "already known" items in our survey – 13.2% for similarity vs. 17.2% for familiarity (one-tailed unpaired t-test , $p<0.05$). These findings resemble those of Sinha and Swearingen [25], who found that friends consistently provide better recommendations (book and movies in their case) than online CF-based recommender systems, but the latter provide items that are more "new" and "unexpected". Yet, the value of "surprising" items needs to be further studied and proven. One of our participants commented: "*Surprising recommendations should be a very low percentage and not always present such that they remain a surprise and do not become an annoyance*". In addition, our attempt to combine familiarity and similarity networks, which could have created a mix of expected and unexpected items, did not show to produce better results than familiarity only. It should be noted, however, that we only experimented with one specific method for combining the networks.

Another very interesting result is related to the role of explanations in instantly increasing users' interest in items. The benefit of explanations becomes very clear in our field study – they considerably increase the number of items clicked by users. Seeing related people names attracts users to specific items and to the recommender widget as a whole. This result should motivate recommender systems in the social software area to provide more transparency into the way they work, as this would pay off immediately and in the most valuable way – more clicks. Interestingly, the instant value of explanations was also reflected in our survey through user rating – items with explanations were rated more interesting. This means that users are not only attracted to further explore the items with explanations, but that the related people also help the users realize that an item is interesting for them. For example, one participant wrote that "*[this item is] only interesting because Matt is on my team*" and another wrote "*I checked this [item] out only because of the related people*".

The studies in this work are based on a specific configuration of familiarity and similarity relationships. It could be argued that adding more relationships that reflect familiarity (e.g., email correspondence) or similarity (e.g., common membership in a community) would affect the results. Yet, we believe that our aggregation is rich enough such that both the familiarity and the similarity networks are well represented and thus the results will not dramatically change, especially as the gaps between familiarity and similarity are very significant. In addition, other algorithms could be applied for inferring relationships, for example, clustering of users or items to yield richer similarity networks. We feel that the aggregation of rich information allowed us to provide recommendations of good quality based on a simple algorithm. The use of this simple algorithm allowed (1) a common basis for comparison between familiarity and similar activity, neutralizing the effect of the algorithm used, and (2) providing the intuitive explanations as described before. Further study should be conducted to test whether richer social networks or different algorithms can improve the recommendation results reported in this work.

The fact that we use aggregated social network information originating from different data sources lead to another important benefit of our recommendation method – being able to deal with the *cold start problem* of new users [23]. As relationships composing the familiarity network are taken from external sources to LC, like the organizational chart, a patent DB, or an external SNS, it is possible to provide recommended items to brand new users, who do not yet have any activity within the LC system.

Our evaluation is based primarily on a user survey, where we asked participants to rate items as interesting vs. not interesting. We chose the word "interesting" over other potential terms, such as "relevant", or "useful", as we believe it most reliably represents the goal of our recommender (e.g., one of the participants wrote "*it's *relevant* to me, but not necessarily interesting. I mean who's interested in Seedlist development?"*). The high correlation with the field study results, where user interest was measured by click-through analysis, supports this assumption.

6. CONCLUSIONS AND FUTURE WORK

We presented a system for recommending social software items based on aggregated social relationships. We furthermore demonstrated a method for explaining such recommendations by showing related people and their relationships to the user and to the recommended item. Our experiments indicated a clear superiority of the familiarity network over the similarity network as predictor for users' interests. Explanations were indicated to be effective in growing users' interest in items, especially in the case of familiar people. Accordingly, the most effective configuration was the one suggesting items based on the familiarity network with explanations, leading to a 57%-43% ratio between interesting and not interesting items that the users did not already know. All item types recommended – bookmarks, blog entries, and communities – aroused user interest. Bookmarks received the highest interest ratio, while communities were the least expected and often triggered additional user action – asking to join the community.

Several participants mentioned their desire for a feedback mechanism that would allow further tuning of their recommendations based on past behavior. We intend to add a feedback feature to our recommender system that would allow users to specify people they want more or less recommendations from. A few participants also pointed out that additional content-based analysis is required to further improve the accuracy of the recommendations. It is possible that the combination of social networks and content-based methods will be effective, especially over a rich set of aggregated data sources, and we plan to continue experimentation around this direction. We also plan to extend our recommended item types to other social software resources like wikis and shared files.

Other possible future work includes conducting an analogous experiment outside the firewall, where the familiarity network consists mostly of friends rather than colleagues, and the similarity network spans beyond the borders of one organization. Finally, the effects of social software item recommendation should be thoroughly examined along time – inspect whether users who were exposed to recommendations are likely to be more active by consuming and creating more content in the long term; and whether recommendations are an effective way to increase user retention rate of social software applications.

7. ACKNOWLEDGMENTS

We thank Tal Daniel and Sigalit Ur for designing and implementing the recommender widget and for their useful advice.

8. REFERENCES

[1] Amitay, E., Carmel, D., Har'el, N., Soffer, A., Golbandi, N., Ofek-Koifman, S., & Yogev, S. 2009. Social search and discovery using a unified approach. *Proc. HT'09,* 199-208.

[2] Arguello, J., Elsas, J., Callan, J., & Carbonell J. 2008. Document representation and query expansion models for blog recommendation. *Proc. ICWSM'08*.

[3] Bonhard, P. & Sasse, M. A. 2006. 'Knowing me, knowing you' -- Using profiles and social networking to improve recommender systems. BT Technology Journal 24, 3 (Jul. 2006), 84-98.

[4] Claypool, M., Le, P., Wased, M., & Brown, D. 2001. Implicit interest indicators. *Proc. IUI '01*, 33–40.

[5] Dugan, C., Muller, M., Millen, D.R., Geyer, W., Brownholtz, B., & Moore, M. 2007. The dogear game: a social bookmark recommender system. *Proc. GROUP'07*, 387-390.

[6] Geyer, W., Dugan, C., Millen, D., Muller, M., & Freyne, J. 2008. Recommending topics for self-descriptions in online user profiles. *Proc. RecSys'08*, 59-66.

[7] Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. 1992. Using collaborative filtering to weave an information tapestry. *Commun. ACM 35, 12* (Dec. 1992), 61-70.

[8] Groh, G., and Ehmig, C. 2007. Recommendations in Taste Related Domains: Collaborative Filtering vs. Social Filtering. *Proc. GROUP'07*, 127-136.

[9] Guy, I., Jacovi, M., Meshulam, N., Ronen, I., & Shahar, E. 2008. Public vs. private – comparing public social network information with email. *Proc. CSCW'08*, 393-402.

[10] Guy, I., Jacovi, M., Shahar, E., Meshulam, N., Soroka, V., & Farrell, S. 2008. Harvesting with SONAR: the value aggregating social network information. *Proc. CHI'08*, 1017-1026.

[11] Guy I., Ronen I., & Wilcox E. 2009. Do you know? recommending people to invite into your social network. *Proc. IUI'09*, 77-86.

[12] Herlocker , J. L., Konstan, J.A., & Riedl, J. 2000. Explaining collaborative filtering recommendations. *Proc. CSCW'00,* 241-250.

[13] IBM Social Software for Business – Lotus Connections: http://www-01.ibm.com/software/lotus/products/connections/.

[14] Jackson, A., Yates, J., & Orlikowski, W. 2007. Corporate blogging: building community through persistent digital talk. *Proc. HICSS'07*, 80.

[15] Java, A., Kolari, P., Finin, T., Joshi, A., & Oates, T. 2007. Feeds that matter: a study of Bloglines subscriptions. *Proc. ICWSM'07.*

[16] Kautz, H., Selman, B., & Shah, M. 1997. ReferralWeb: Combining social networks and collaborative filtering. *Commun. ACM 40*, 3 (Mar. 1997) 63-65.

[17] Lerman, K. 2007. Social networks and social information filtering on Digg. *Proc. ICWSM'07*.

[18] McSherry, D. Explanation in recommender systems. 2005. *Artif. Intell. Rev.* 24, 2 (Oct. 2005), 179-197.

[19] Millen, D.R., Feinberg, J., & Kerr, B. 2006. *Dogear*: Social bookmarking in the enterprise. *Proc. CHI'06*, 111-120.

[20] Offical Digg Blog: http://blog.digg.com/?p=127.

[21] Official Facebook Blog: http://blog.facebook.com/blog.php?post=2207967130.

[22] Official FriendFeed Blog: http://blog.friendfeed.com/2008/06/friendfeed-adds-personalized.html.

[23] Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. 2002. Methods and metrics for cold-start recommendations. *Proc. SIGIR'02*, 253-260.

[24] Seth, A., & Zhang, J. 2008. A social network based approach to personalized recommendation of participatory media content. *Proc. ICWSM'08*.

[25] Sinha, R. & Swearingen, K. 2001. Comparing recommendations made by online systems and friends. *Proc. DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries*.

[26] Spertus, E., Sahami, M., & Buyukkokten, O. 2005. Evaluating similarity measures: a large-scale study in the Orkut social network. *Proc. SIGKDD'05*, 678-684.

[27] Tepper, M. 2003. The rise of social software. *netWorker* 7, 3 (Sep. 2003), 18-23.

[28] Tintarev, N. & Masthoff, J. 2007. A Survey of explanations in recommender systems. *Proc. ICDEW'07*, 801-810.

[29] Vatturi, P. K., Geyer, W., Dugan, C., Muller, M., & Brownholtz, B. 2008. Tag-based filtering for personalized bookmark recommendations. *Proc. CIKM'08*, 1395-1396.

[30] Wasserman, S., & Faust, K. 1994. *Social Network Analysis*.