Social Lens: Personalization Around User Defined Collections for Filtering Enterprise Message Streams

Elizabeth M. Daly, Michael Muller, David R. Millen
IBM Research
Cambridge, MA 02142, USA
dalyeliz, michael_muller, david_r_millen@us.ibm.com

Liang Gou
Pennsylvania State University
University Park, PA, 16802, USA
lug129@ist.psu.edu

Abstract

Social media has led to a data explosion and has begun to play an ever increasing role as a valuable source of information and a mechanism for information discovery. The wealth of data highlights the need for methods to filter and sort information in order to allow users to discover useful information. Most traditional solutions focus on the user, either the user's social network, or a form of personalization based on collaborative filtering or predictive user modeling. This paper presents a novel algorithm to view information through a lens based on a user defined collection while *excluding* the attributes of the user from the analysis. As a result, the lens is transparent, tunable and sharable amongst users and, additionally allows both a reduction in information overload while discovering new related content.

Introduction

Social media has become an important mechanism for people to share and discover information. On sites such as Twitter, Digg, Slashdot, and Facebook, users post comments, status messages and links. People are relying more and more on these event streams as a source of information (Java et al. 2007). However, with sites like Twitter generating an estimated 17,000 posts a minute, the challenge for a user to filter and process this information is overwhelming. In some settings, messages come not only from people, but also from the objects with which people interact, adding to this problem (e.g., Geyer et al. 2007; Mathioudakis, Koudas, and Marbach 2010).

Proposed solutions include highlighting content deemed popular by the crowds, such as Twitter featuring Trending Topics, a list of terms that have been growing in frequency over the last day, week, or month; or a promoted front page for popular stories, such as in Digg. Personalization techniques may be used to reduce the overwhelming content by recommending items of interest to the user based on preferences, past behavior or their social network. User defined feeds, such as watchlists or following feeds, allow users to manually select sources they are interested in receiving events from. Faceted browsing allows users to explore content by pivoting around specific attributes such as tags, topics or sources (e.g., Hong et al. 2010).

We propose a novel mechanism to support users in the following goals: to enable a combination of both *filtering* and *discovery*; to explore new interests quickly without requiring historical data; to provide transparency in personalization decisions; to allow users influence and tune the view; and to separate their interests into appropriate "working spheres" as defined by Mark et al. (González and Mark 2004; Mark, González, and Harris 2005; Mark, Gudith, and Klocke 2008) which then may be shared amongst users.

This paper presents a novel algorithm which allows each user to create one or more "lenses" - i.e., personally specified filters that focus on a topic of the user's interest. To create the lens, the user 'specifies by example' and lists a small number of persons and entities from a social software suite related to a given topic (such as a file, a blog, a community, a wiki page, etc.). We compute the lens (the filter) by traversing three networks related to the user defined collection: the person-to-person social graph, the entity-to-entity object graph, and a more complex graph that relates persons to objects through links of creators (e.g., files, wiki pages) or commenters (e.g., blog comments) or membership (e.g., communities). This novel algorithm reconciles two often opposing goals: to reduce the volume of update messages (the information overload) and to discover highly related persons and entities that are beyond the graph of the user's network. Additionally, most traditional algorithms focus on the user, either the user's social network , or a form of personalization based on collaborative filtering or predictive user modeling. These algorithms cannot be shared from one user to another, because the filter is based on each users own metadata. In contrast, our algorithm focuses purely on a user-specified collection of persons and objects, and is divorced from the metadata of the individual user. It thus provides the same results no matter who uses it. Moreover, our solution can be used by a new user who does not yet have either a profile or a history (which is required for personalization-based filters).

We report a small-scale user study aimed at understanding if personalization around a user-defined collection of persons and objects aids in reducing information overload while also allowing users to discover related content. This study is conducted in a complex enterprise social software suite, in which messages may come from other users, online communities, blogs, activities, files, and wikis.

The paper is organized as follows. The next section surveys related work and existing solutions. We then briefly describe the formal attributes of the enterprise social software suite, as parameters of the problem. After that, we describe our solution: the core algorithm, the experimental architecture, and the user interface. In the evaluation section, we report on a small user study of our new approach, focusing on reducing the volume of update messages and on the discovery of previously unknown resources. We conclude by placing our work back into the context of related work, and we close with challenges for the future.

Related Work

As discussed in the previous section, the problem of information overload can be tackled through providing a user with a subset of content on the system in the form of recommendations, personalized views etc. These techniques for recommendations and filtering attempt to identify items of interest to users.

Personalization

Content-based approaches match items based on the text similarity to a content-based model of the user's interests. For example, Phelan et al. propose recommendations based on the co-occurrence of terms in a users twitter friend stream and the RSS feeds they subscribe to (Phelan, McCarthy, and Smyth 2009). *Collaborative-filtering* techniques match items that similar users have previously found interesting and in essence try to match similar users assuming they will be interested in the same items. For example, Stewart et al. exploit cross domain tagging behavior and collaborative filtering to generate personalized tag based recommendations (Stewart et al. 2009). A significant disadvantage of the above two solutions is that users may be similar in some topics, and divergent in others. In the case of content-based filtering, a user who has a predominant interest in one topic, may easily overwhelm any interest in alternative topics that are also of relevance. In the case of collaborative-filtering, a user may be similar to other users in some topics, but divergent in others.

A more recent approach is to harness social relationships to improve relevance to the individual user. Song et al. use a combination of user history, i.e., their social network, to capture information between users in order to provide personalized recommendations (Song et al. 2006). Gürsel and Sen (2009) attempt to deal with information overload by recommending items of interest to the users from items recently posted by other users in their social network. Similarly, Guy et al. (2009) show items that are strongly related to people in the user's social network and go on to exploit indirect relationships such as overlapping tagging behavior between users to provide recommendations (Guy et al. 2010). Adams et al. (2009) provide a blog feed reader which visualizes the social relationships between blogs to allow users to navigate and explore their intrinsic relationships. Chen et al. (2010) explored 12 different Twitter based recommendation algorithms using content similarity, topic modeling, social voting and popularity. They found that a combination of user topic modeling and social voting which proved promising for personalized recommendations.

There are a number of drawbacks regarding these existing solutions. Methods that model the users interest require historical data and take time to adapt to new interests. Personalization results in the lack of ability to share views between users, i.e., users who collaborate around specific tasks may have very divergent views of the updates that have occurred. Additionally, users have little or no ability to tune the recommendations presented to them. As a result, the lack of control and transparency makes it difficult for users to organize their streams in a meaningful way.

Supported Faceted Browsing

Some tools allow different ways to partition the message stream. FriendFeed allows users either to filter by people or to use a form-based search tool[1]. Eddi provides a Twitter interface that clusters a user's feed into topics supporting topic browsing (Bernstein et al. 2010). FeedWinnower allows for faceted exploration of feeds based on topic, people, source or date (Hong et al. 2010). Similarly, VisGets supports faceted browsing based on time, location and tags (Dork et al. 2008). O'Connor et al propose TweetMotif which groups tweets into subtopics where users can explore content by theme (O'Connor, Krieger, and Ahn 2010). These methods are designed to allow users to browse and explore content through their different facets, rather than provide a reusable aggregated view that can be shared.

User Defined Feeds

Lerman and Ghosh performed an analysis of Digg and Twitter and found that users actively created social networks by designating users whose activities they want to follow and used these networks to discover information (Lerman and Ghosh 2010). Twitter[2] has more recently provided a list mechanism, where users can follow groups of related people by creating user defined collections. In this manner, users divide their attention into different topics of interest with the additional advantage that other users may reuse lists they have determined to be a good source of information.

User defined feeds give control back to the user, at the sacrifice of discovering serendipitous and potentially relevant content. Users are limited to the view they have defined, and the discovery of new related content may be limited.

Social Lens Filtering for Enterprise Message Streams

We present a novel mechanism for filtering enterprise message streams based on user-defined topics of interest. Our enterprise social software environment includes applications for social-networking, social-bookmarking, social file-sharing, blogging, wikis, online communities, and shared project/task-management. Each user may issue status updates from the social-networking application. Each data object may issue updates when changed by a user.

[1] http://friendfeed.com/search/advanced

[2] http://www.twitter.com

The framework of the social lens is shown in Figure 1. The social lens is populated with a user-specified collection of people and objects from the social software environment, such as communities, blogs, wikis, activities and files.

This user-defined collection is deemed related to their particular topic, and the user's own metadata and history is excluded from the analysis. The aim is to rank this collection and also to find and rank related resources and people, using a combination of social similarity and topic similarity around the collection.

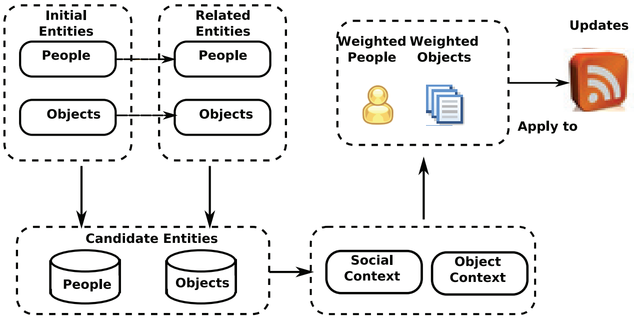


Figure 1: The framework of Social Lens.

From this initial collection, we identify (a) objects related to the specified people and (b) people related to the specified objects. Examples of *objects related to people* are communities they are a member of, and blogs they have written or commented on. The *people related to objects* are identified in a similar manner, e.g. through authorship, comments, and so on.

These *related* entities form a candidate set of entities for the filter. We score these candidate entries with our weighting model. The weighting model assigns relevance weights to the candidate entities derived from *social context* and *object context* in the social software. Social context indicates how people are connected in a certain way, e.g., two persons may be linked by an explicit friendship in their social-networking profile or by commenting on each others' blog or by sharing a file. Object context refers to how objects relate to each other in terms of both similarity of content and how they are related through people, e.g. two documents are related if they are shared by the same person. The algorithm is based on the premise that, a person may be socially connected to the users and the resources, but have little or no relationship to the topic. Similarly, resources that are a close match in terms of topic, could be more relevant to the user if they are socially connected to the resources and people in the initial collection.

Finally, the *initial* people and objects and the high-scoring *related* people and objects are used to filter and discover within the information streams. The weighting model in the Social Lens has two modules: people-weighting and object-weighting. Each module includes two phases: (1) weight the related people/objects and (2) weight the initial people/objects. The related people and objects are used to assign weights of relative importance to the initial objects. As a result, it is necessary to execute these two phases in this non-intuitive order (*related* before *initial*). When weighting initial people/objects, we follow a pseudo feedback strategy which is widely used in information retrieval (Baeza Yates and Neto 1999): We only use the top N weighted related people/objects as the reference to weight the initial people/objects. The rationale is that the top N weighted related people/objects are highly relevant to the topic of interest and provide more information to weight the initial entities.

Calculate Related Entities

The related entities are calculated by exploring the relationship graph between people and objects.

Step 1. Calculate *related* people. The objects related to the initial people I_p are collected, and the people linked to them are retrieved. Additionally, the people connected to the initial objects I_o are retrieved. Both sets of people are added to the candidate list of related people C_p.

It should be noted that the same person may appear multiple times in C_p. The score of each person p is weighted as function of frequency of appearance f, the Friend-of-Friend (FoF) similarity s with the initial people and the content similarities c with the initial objects. The content similarity is calculated by the cosine similarity with tf-idf schema of two documents (Salton and Buckley 1988).

$$Score(C_p, I_p, I_o) = func(f, s, c). \qquad (1)$$

Step 2. Calculate *related* objects. The related people are assumed to have high relevance to both the initial people and the initial objects. As a result the top M related people (M=50 in our algorithm) are used as a starting point to derive the candidate related objects. The objects linked to the initial people similar to the previous step. Additionally, the objects linked to the top M related people. Both sets are added to the candidate related objects list C_o.

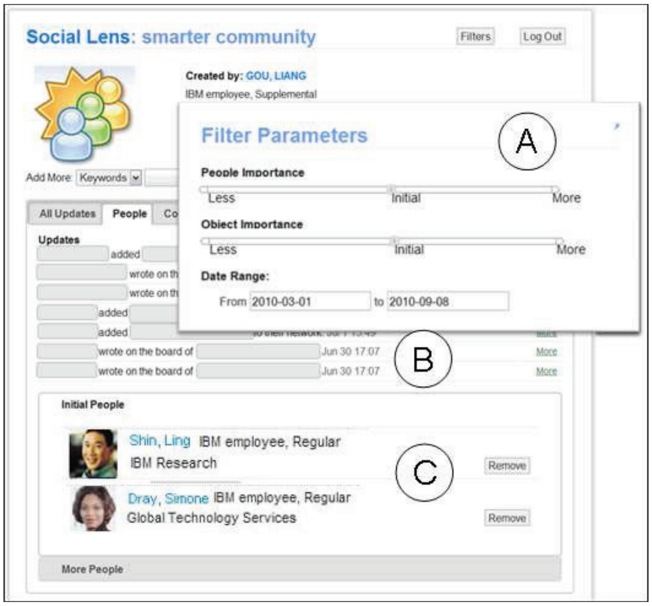
As with candidate people, objects may appear several times in the candidate entry set and are therefore weighted according to their occurring frequency. Each candidate object C_o is weighted as a function of the frequency of appearance f, the Friend-of-Friend (FoF) similarity s of the person that links the object to the initial people and the content similarities c with the initial objects.

Rank Initial Entities

Pseudo-relevance feedback is used to rank the initial entities based on the premise that the top-rank results are the most relevant (Baeza Yates and Neto 1999). Therefore, in our case, the top N (N=10 in our algorithm) related people and top N related objects are the most relevant to the user defined collection. The initial collection is then scored based on their relevance score to the top N related items. The final weights of the initial people and objects are the calculated score plus 1, so that the weights of initial collection are always larger than 1. The weight of the related collection is always less than 1.

A Social Lens

Figure 2 shows how to use the filters to adjust the views of updates in a social lens. The user defines a lens by creating


Social Lens: smarter community

Created by: GOU, LIANG
IBM employee, Supplemental

Filter Parameters

People Importance

Object Importance

Date Range:
From 2010-03-01 to 2010-09-08

Updates
added
wrote on th
wrote on th
added
added
wrote on the board of Jun 30 17:07
wrote on the board of Jun 30 17:07

Initial People

Shin, Ling IBM employee, Regular
IBM Research

Dray, Simone IBM employee, Regular
Global Technology Services

Figure 2: User-adjustable filters for the output of one Lens. (a) A stream of update messages, including (b) the original two ”seed” people who were specified for the lens. (c) Slider controls for the relative importance of people and of data objects.

a collection of people and resources they deem relevant to a given topic. This collection is used to ”seed” the social lens which results in a collection of initial entities and related entities. When a user views the social lens, the updates and events associated with the collection are retrieved and shown to the user in a time ordered manner (Figure 2b). These views can be private or publicly shared with other users. A simple filtering option (Figure 2a) is provided through which the user can move sliders to adjust the threshold associated with the collection of objects and the collection of people, respectively. When the user increases the person-threshold to above 1, then only those updates associated with the initial people are displayed. Conversely, when the user decreases the threshold below 1, then updates of related people are also included. In this manner the user has a simple yet powerful mechanism to tune the amount of information they retrieve and the strength of their relatedness to the topic of interest. Additionally, the user is show the list of initial people and objects along with the newly discovered related people and objects. The user may promote these related items to become part of the initial social lens, or remove the item from lens (Figure 2c). As a result, the user has transparency as to what defines the view and the ability to tune the collection further.

Experimental Study

To evaluate our algorithm, we asked users to compare the results of the Social Lens with two existing streams of items in the social software application suite.

Participants

We requested 20 informants to participate in our first evaluation. Each informant was invited to provide, via email, a description of up to five lenses. Each lens consisted of a title and a collection of items from the social software application suite defined by the user. These items could be persons, communities, activities, bookmarks, files, blogs, or wikis.

Twelve people (60%) responded within our initial evaluation time frame, resulting in 34 lens specifications. A total of ten informants (83% of the initial timeframe participants) responded during our second evaluation time frame, to our request to rate the update items and the secondary objects, and to experiment with the sliders. The number of lenses evaluated per informant ranged from 1-3.

Evaluation Metrics

In order to evaluate whether the updates shown through a social lens are of interest to the user, we compared the performance of our algorithm to two existing streams available in the enterprise system:

- The first is referred to as **Top** updates, which shows the user all updates from people the user has connected to in a colleague social network. This is a personalized list of updates based on the user’s ego-centric social network.
- The second is referred to as the **Discover** updates, which shows all updates throughout the entire system and is designed to all users to browse and discover what is going on in the system. This is a non-personalized list, the same for all users.

Interestingness: For each lens, we collected 20 updates from the top-update’s stream, the discover-update’s stream, and our algorithm’s social-lens stream (total of 60 updates). Updates that appeared in more than one stream were summarized to a single instance to avoid showing the users duplicate updates. Nine percent of updates were duplicates.

For a “blind” user test, all identifying information about the source of each update was removed, so that informants could not tell whether a particular update was from the Social Lens, the Top-updates, or the Discover-updates stream. The combined list of 60 updates for each lens was then randomly shuffled in order to reduce user bias based on list position. Users were presented with the randomized list for each of their own lenses (one lens at-a-time), and were asked to provide ratings of interestingness on a five point scale (from “very interesting” to “very uninteresting”).

We selected the metric of *interestingness* to make it a “fair” test, because (unlike the Social Lens) the items in the Top and Discover streams were not intended to be related to any particular topic. Additionally, we assumed that users will still rate updates related to their lens topic as interesting.

Relatedness: Within the more restricted domain of the Social Lens, we wanted to evaluate the utility and relatedness of the related people and objects derived for each lens. We asked for ratings of relatedness on the related people and the related objects (as calculated by our algorithm, described above). The collection of related persons and objects was shown to the users and they were asked to rate each item in order of relatedness, on a five point scale (from "very related" to "very unrelated").

User Feedback: Finally, the users invited to give feedback and comments of the user experience, including the filtering experience.

Results

Interestingness

Which streams were more successful at showing the most interesting updates? Figure 3 shows the percentage breakdown of the users "interestingness" ratings of the update streams from the three sources: social-lens-updates, top-updates, and discover-updates. The social-lens updates had the highest percentage of interesting updates ("very interesting" + "somewhat interesting" = 42%), and the lowest percentage of uninteresting updates ("somewhat uninteresting" + "very uninteresting" = 38%). By contrast, the top-updates and discover-updates had at least 53% uninteresting updates.

To test these results, we randomly selected one lens from each informant (for statistical independence), and we calculated the mean rating (across all update items) for Social Lens, for Top-updates, and for Discover-updates, respectively, for each informant. We then had 30 means (10 informants x 3 streams). We subjected these means to a repeated measures analysis of variance. Despite the small sample size, differences among the three streams were significant ($F_{2,18}$ = 9.048, p <.02), with a significant linear trend in the hypothesized direction of Social Lens more interesting than Top-updates, followed by Discover-updates ($F_{1,9}$ = 13.503, p <.005). In pair-wise post-hoc LSD tests, each stream was significantly different from each other stream at p <.05 or better (for summary means and standard errors, see Table 1). These results show that informants found the Social Lens results to be the most interesting, followed by the Top-updates stream, and followed lastly by the Discover-updates stream.

Relevance

In view of the relatively strong performance of the social lens algorithm, we wanted to know if the related-persons and related-objects had helped us. We compared the informants' ratings of Relatedness with the Social Lens calculated score of relevance, as shown in Figure 4. In general, our calculated relevance score appears to be a good predictor of informants' relatedness ratings. Thus, the calculated object score is a good representation of the relevance of each object, for both persons and entities.

Stream	Social Lens	Top	Discover
Mean	3.250	3.945	4.400
Std. Error	0.322	0.297	0.204

Table 1: Means and standard deviations for the three streams, across ten informants. The rating scale ran from 1="very interesting" to 5="very uninteresting."

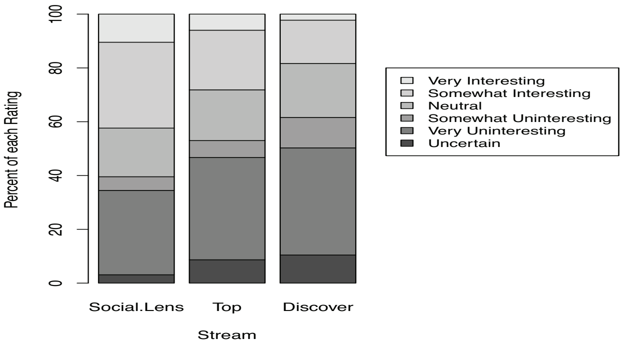


Figure 3: Percentage of ratings for each stream of updates.

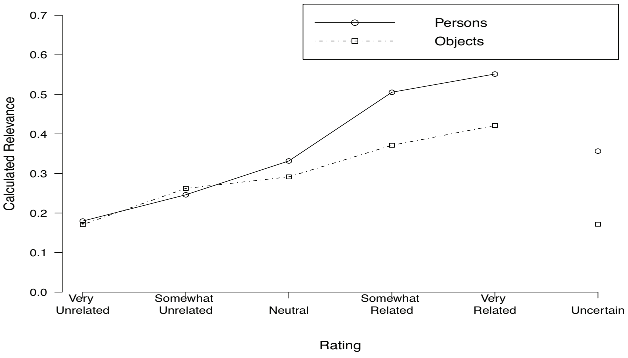


Figure 4: Comparison of the Social Lens calculated relevance score with the informant-rated Relatedness score, for Persons and for Entities.

The Effect of Removing the Ego from Personalization

In order to explore the effect of personalization around a collection rather than the user, we examine the relationship between the user and the results through the user's ego-centric network when exploring the social graph and the object graph. The aim is to evaluate whether personalization techniques based around the user's social graph or object relationship graph (e.g. EdgeRank[3]), would have resulted in excluding results the user found relevant and would otherwise have remained undiscovered.

Direct Colleagues: We examined the users who were calculated to be relevant to each informants lens. The percentage that are *not* part of the informant's social network was 47% of the users calculated to be related. Thus, we achieved high serendipity. These people proved to be relevant to the informants with 62% of the unlinked users rated as "very relevant," and 88% of the unlinked users rated as "very" or "somewhat relevant." Similarly, when examining the objects that users found "very" or "somewhat relevant", 62% of the objects were authored or owned by users that were not a part of the users social network and therefore would not have been found if using the ego-centric graph alone.

[3] http://techcrunch.com/2010/04/22/facebook-edgerank/

Social Similarity: We examined the social overlap similarity of the users connections in the social graph and found that 90% of the related people had at least 1 social contact in common. However, figure 5 shows the social overlap between the ego user and the discovered related people and the owner/author of the related objects. As can be seen the items and people that were rated "very" or "somewhat relevant" include a large number of users with a low social overlap similarity, indicating social similarity alone would have missed a large portion of related items.

Resource Overlap: Figure 6 shows the resource overlap between the ego user and the discovered related people and the owner/author of the related objects. 73% of the related people and 69% of the owners/authors of the related objects had at least one resource in common with the ego users. However, as can be seen in figure 6 the resource overlap is relatively low. When examining the related items that had no overlapping resources 48% of the related people and 37% of the related objects were rated "very" or "somewhat relevant".

As a result, we determine that personalization around a collection of resources rather than a user has resulted in discovering content that is unconnected to the user either through the social graph or the object graph.

User Feedback

Finally, we examined informants' written comments (see the "Filtering" part of our evaluation procedure, above). This is a preliminary study with only ten people, so we do not take these comments as definitive. We performed a quick open coding on the comments, and found the following themes:

- **Importance of Viewing Person information.** The social software environment stores a great deal of data, in the form of files, wikis, bookmarks, etc. However, many informants emphasized their interest in other users, rather than in the artifacts created by those users: *"Objects are not interesting in and of themselves", "the information I need is more closely associated with people than with an object", and "I think of my learning in terms of incidental learning I do from other people"*.
- **Repetitive update messages.** Informants complained that we had not successfully filtered out redundant update messages about related actions: *"Could be more concise, though", and "when someone frequently posts changes to a wiki page, you get the same person/action/object set repeatedly. A better design would exclude asking repeatedly. Should I repeat that? ⌣̈?"* We note that the repetition occurred in the source update streams that we were filtering. Because the stream contains person identifiers and entity-identifiers, it will be easy to remove these kinds of duplications in future versions.
- **Need for more context to the update messages.** The update message streams report only the fact that an update has occurred. Some informants wanted more: *"seeing that people wrote on someone else's board (with no details) doesn't mean a lot to me" and "Many [updates] were [that] X wrote on the board of Y but what is more interesting is WHAT is written"*. We will need to balance this kind of need for context and depth, with a need for terseness and ease of scanning multiple events. This trade-off may lead to another user specifiable display parameter, similar to the controls for person-threshold and object-threshold in Figure 2.

Discussion

Limitations

We have already noted the small sample size in our preliminary user study. We will need to expand the number of users, in order to make a stronger test of the advantages of our social lens filter. We will also need to diversify our user population beyond the knowledge workers who participated in our study. Our preliminary results are encouraging.

We have shown some evidence that our social lens approach functions well in one enterprise social software environment. We hope to test our algorithm in other social software environments in the future.

Reducing Information Overload

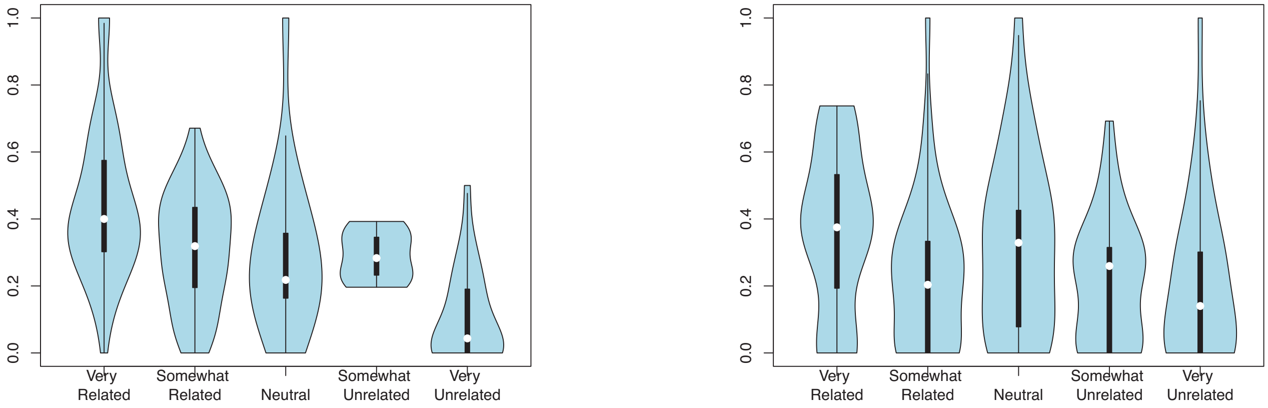
There are many approaches to reducing information overload. Previous research has explored content-based filtering and social recommendations/filtering (Geyer et al. 2007; Gürsel and Sen 2009; Adams, Phung, and Venkatesh 2009; Hong et al. 2010). In this paper, we have explored a hybrid approach based on a combination of persons and content entities. Initial results suggest that our approach performs better than conventional social recommendation/filtering. Future tests will compare our approach with content-based filtering, as well.

Topic-Specific Filtering Defined By User

Our approach differs from conventional holistic user modeling by focusing on specific topics selected and described by individual users. We believe that we have implemented, as it were, one embodiment of Mark's concept of working spheres (González and Mark 2004; Mark, González, and Harris 2005; Mark, Gudith, and Klocke 2008). An immediate advantage of our topic-specific filters is that they can be shared from one user to another. If this sharing is successful, then we may eventually be able to contribute to work-oriented theory by extending the concept of a working sphere from an individual's attention focus into a collaborative perspective and/or artifact (a "sharable lens"). Our filters provide not only a view on the current stream of updates, but also a view into the past of that stream (assuming a persistent store of updates). If our new approach is successful within organizations, we will be interested to see how users make use of these filters that can replay their own history.

Discovery as well as Overload Reduction

Our approach does more than reduce the volume of incoming updates, which is the common goal of must filters. We also showed a large degree of serendipity, i.e., the ability to find previously unlinked persons who are rated as highly relevant to the user's particular interest. While reductions



(a) Social Overlap of Ego User to Related People.

(b) Social Overlap of Ego User to Author/Owner of Related Objects.

Figure 5: Social Overlap of Discovered Related Items.

in message volume are important, the discovery of new information may also be crucial in certain jobs. Informants' comments in our small user study suggested that users may want to see more information about the topics that are within their focus (i.e., within their current working sphere). If other users make similar requests, we will explore further methods to tune the contextual "depth" as well as the topic specific "breadth" of our filters.

Conclusion

We have proposed a solution to the problem of information overload, which is already occurring in enterprise social software services, and in social media services on the Internet. Our Social Lens approach was written in the context of the theory of working spheres, and attempts to address this challenge through several innovations. First, we use a *hybrid* filtering model that combines persons and content related to a particular topic. Second, we invite users to define their own filters by naming familiar people and objects as the initial "seeds" of the filter. Third, we use those initial seed objects not only to reduce the load of incoming information, but also to discover unlinked resources.

We believe the concept of allowing users to view information through a "lens" of their defining, allows for quick exploration of new topics of interest without previous history, with the benefit of transparency and tunability. We have tested some of our claims in a small user study, with promising outcomes and challenges for future work. Finally, we hope that an implementation of aspect of the concept of working spheres may prove useful for further development of that theory, and that it will lead to better tools for managing individuals and groups *information overload*.

References

Adams, B.; Phung, D.; and Venkatesh, S. 2009. Social reader: following social networks in the wilds of the blogosphere. In *Proceedings of the first SIGMM workshop on Social media*, WSM '09, 73–80. New York, NY, USA: ACM.

Baeza Yates, R. A., and Neto, B. R. 1999. *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

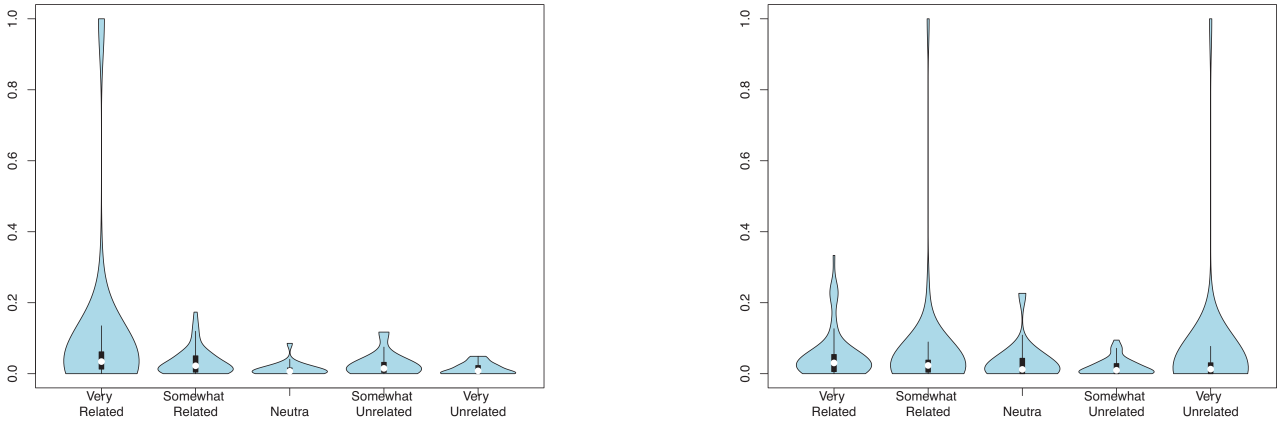
Bernstein, M. S.; Suh, B.; Hong, L.; Chen, J.; Kairam, S.; and Chi, E. H. 2010. Eddi: interactive topic-based browsing of social status streams. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, 303–312. New York, NY, USA: ACM.

Chen, J.; Nairn, R.; Nelson, L.; Bernstein, M.; and Chi, E. 2010. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, 1185–1194. New York, NY, USA: ACM.

Dork, M.; Carpendale, S.; Collins, C.; and Williamson, C. 2008. VisGets: Coordinated Visualizations for Web-based Information Exploration and Discovery. *IEEE Transactions on Visualization and Computer Graphics* 14(6):1205–1212.

Geyer, W.; Brownholtz, B.; Muller, M.; Dugan, C.; Wilcox, E.; and Millen, D. R. 2007. Malibu personal productivity assistant. In *CHI '07 extended abstracts on Human factors in computing systems*, CHI '07, 2375–2380. New York, NY, USA: ACM.

González, V. M., and Mark, G. 2004. "Constant, constant, multi-tasking craziness": managing multiple working spheres. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, 113–120. New York, NY, USA: ACM.



(a) Resource Overlap of Ego User to Related People.

(b) Resource Overlap of Ego User to Author/Owner of Related Objects.

Figure 6: Resource Overlap of Discovered Related Items.

Gürsel, A., and Sen, S. 2009. Producing timely recommendations from social networks through targeted search. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, 805–812. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Guy, I.; Zwerdling, N.; Carmel, D.; Ronen, I.; Uziel, E.; Yogev, S.; and Koifman, S. O. 2009. Personalized recommendation of social software items based on social relations. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, 53–60. New York, NY, USA: ACM.

Guy, I.; Zwerdling, N.; Ronen, I.; Carmel, D.; and Uziel, E. 2010. Social media recommendation based on people and tags. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, 194–201. New York, NY, USA: ACM.

Hong, L.; Convertino, G.; Suh, B.; Chi, H.; and Kairam, S. 2010. FeedWinnower: Layering Structures Over Collections of Information Streams. In *CHI '10 Proceedings of the 28th international conference on Human factors in computing systems*.

Java, A.; Song, X.; Finin, T.; and Tseng, B. 2007. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, SNA-KDD '07, 56–65. New York, NY, USA: ACM.

Lerman, K., and Ghosh, R. 2010. Information Contagion: an Empirical Study of the Spread of News on Digg and Twitter Social Networks. In *Proceedings of 4th International Conference on Weblogs and Social Media (ICWSM)*.

Mark, G.; González, V. M.; and Harris, J. 2005. No task left behind?: examining the nature of fragmented work. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '05, 321–330. New York, NY, USA: ACM.

Mark, G.; Gudith, D.; and Klocke, U. 2008. The cost of interrupted work: more speed and stress. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, 107–110. New York, NY, USA: ACM.

Mathioudakis, M.; Koudas, N.; and Marbach, P. 2010. Early online identification of attention gathering items in social media. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, 301–310. New York, NY, USA: ACM.

O'Connor, B.; Krieger, M.; and Ahn, D. 2010. TweetMotif: Exploratory Search and Topic Summarization for Twitter. In Cohen, W. W.; Gosling, S.; Cohen, W. W.; and Gosling, S., eds., *ICWSM*. The AAAI Press.

Phelan, O.; McCarthy, K.; and Smyth, B. 2009. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, 385–388. New York, NY, USA: ACM.

Salton, G., and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.* 24(5):513–523.

Song, X.; Tseng, B. L.; Lin, C. Y.; and Sun, M. T. 2006. Personalized recommendation driven by information flow. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, 509–516. New York, NY, USA: ACM.

Stewart, A.; Aviles, E. D.; Nejdl, W.; Marinho, L. B.; Nanopoulos, A.; and Lars, S. T. 2009. Cross-tagging for personalized open social networking. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, HT '09, 271–278. New York, NY, USA: ACM.