Social Networking for Scientists Using Tagging and Shared Bookmarks: a Web 2.0 Application

Marlon E. Pierce[1], Geoffrey C. Fox[1,2], Joshua Rosen[1], Siddharth Maini[1], and Jong Y. Choi[1,2]

[1]Community Grids Laboratory, Indiana University, Bloomington, IN 47404, USA
[2]Department of Computer Science, Indiana University, Bloomington, IN 47404, USA

Web-based social networks, online personal profiles, keyword tagging, and online bookmarking are staples of Web 2.0-style applications. In this paper we report our investigation and implementation of these capabilities as a means for creating communities of like-minded faculty and researchers, particularly at minority serving institutions. Our motivating problem is to provide outreach tools that broaden the participation of these groups in funded research activities, particularly in cyberinfrastructure and e-Science. In this paper, we discuss the system design, implementation, social network seeding, and portal capabilities. Underlying our system, and folksonomy systems generally, is a graph-based data model that links external URLs, system users, and descriptive tags. We conclude with a survey of the applicability of clustering and other data mining techniques to these folksonomy graphs.

***Index Terms*—Web 2.0, Social Networks, Folksonomies, Collaboration**

I. Introduction

The proliferation of online communities and social networks such as Facebook, LinkedIn and many others, with memberships numbering in the millions, has reinvigorated the Web by making it a participatory entity with blurred lines between users and developers. These social networking systems are part of a larger activity that is collectively labeled "Web 2.0" [1]. Although Web 2.0 is an uncoordinated activity when compared to Web Services or Grid computing, its disparate activities collectively define a comprehensive distributed computing approach [2][3]. As such, it is challenging many of the architectural foundations of cyberinfrastructure and e-Science.

This paper describes our work to build a social networking portal that is geared toward enabling faculty and researchers to find both useful online resources and also potential collaborators on future research projects. We are particularly interested in helping researchers at Minority Serving Institutions (MSIs) connect with each other and with the education, outreach, and training services that are designed to serve them, expanding their participation in cyberinfrastructure research efforts. This portal is a development activity of the Minority Serving Institution-Cyberinfrastructure Empowerment Coalition (MSI-CIEC). The portal's home page view is shown in Figure 1.

Online bookmarking was pioneered by such sites as del.icio.us, Connotea, Digg, Slashdot, and CiteULike, among others (for a summary, see [4]). These sites vary in purpose. General-purpose bookmarking sites such as del.icio.us can bookmark any link and have a time-independent view of the URLs. Digg and Slashdot, on the other hand, are geared toward tagging and rating news links and more ephemeral subjects. Connotea and CiteULike both cater to academic citation links and provide additional tools (such as automatic metadata fill-in with a provided Digital Object Identifier). In related work, our lab's IDIOM project [5] seeks to couple tagging of academic material with scholarly search engines such as Google Scholar and Microsoft Live Academic.

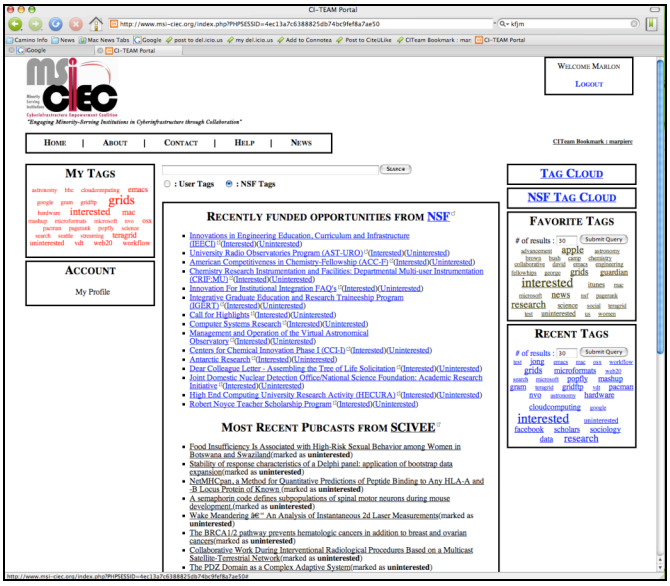


Figure 1 The MSI-CIEC social networking Web portal combines social bookmarking and tagging with online curricula vitae profiles. The display shows the logged-in user's tag cloud ("My Tags" on left), taggable RSS feeds (center), and tag clouds of all users ("Favorite Tags" and "Recent Tags" on the right). Users may search tags (including researcher names, NSF directorates, and TeraGrid allocations) using the center text field.

Apart from their utility, social bookmarking and tagging are interesting for Computer Science research because they create usage-driven descriptions of URLs (and potentially any URIs). Such descriptions are known as *folksomomies* and superficially resemble more structured ontology approaches pursued by the Semantic Web activity. As we discuss below, folksonomies are in fact graphs (as are RDF and OWL-represented ontologies). Unlike ontologies, folksonomies lack the expression of logical associations in the arcs of the graph. This does not allow, for example, logical inferences to be made in the relationships in the graphs (as is the goal of

Semantic Web ontologies), but it does indicate that a wealth of data mining algorithms may be applied to discover interesting emergent relationships in the data in place of designed-in and derived relationships. We conclude with a discussion our initial survey of these problems.

II. MSI-CIEC Networking Portal Features

Our portal is designed to support academic user communities through a combination of online user profiles and shared online bookmarks that are described with keyword tags. The system capabilities include the following:

- Users can create public profiles of themselves to describe their research interests, provide their publication lists, academic and professional training, and other curricula vitae.
- Profiles are also decorated with the user's tag cloud (see Figure 1).
- By importing RSS feeds, users can further enhance their profiles with other information, such as Connotea publication feeds, SciVee videos, etc.
- Users can bookmark any URL during normal browsing and have it stored in the MSI-CIEC portal database. Users describe bookmarks with one or more keyword tags.
- Users can search their own bookmarks by navigating tags, and they can also search publicly tagged URLs from other users.
- Researchers can also "click tag" featured RSS feeds, such as NSF Recently Announced Funding Opportunities [6]. Click-tagging allows a user to label entries in the feed with "interesting" or "uninteresting" tags. Users can later view their own and public "interested" tags.
- Users can search award funding and project data. We currently import data and auto-generate tags from the NSF's awards database and the NSF TeraGrid's allocations database. These tags can be searched and navigated just as normal, user-generated tags.

Social networking sites depend upon a minimal number of users and richness of data to be self-sustaining, so our initial capabilities have been chosen to support uncorrelated usage. Bookmarking, NSF award navigation, and click-tagging are all applications that are independent of the number of users. The social networking properties (such as joining groups, finding most interesting tags, and viewing other users' profiles) are emergent capabilities of the system that become richer as more people use the system. We now review these capabilities in more detail through example usage scenarios in the following section.

III. Usage Scenarios

A. *Creating an Online Profile*

As described above, one of the portal's primary functions is to provide online, customizable and extensible curricula vitae for users. In addition, users with previous NSF awards have automatically generated profile stubs that they can enhance. Figures 2-4 display the sections of this form. Figure 2 shows a display of basic portal information (the user's name and profile tags). Forms for updating the user's professional preparation are shown in Figure 3.

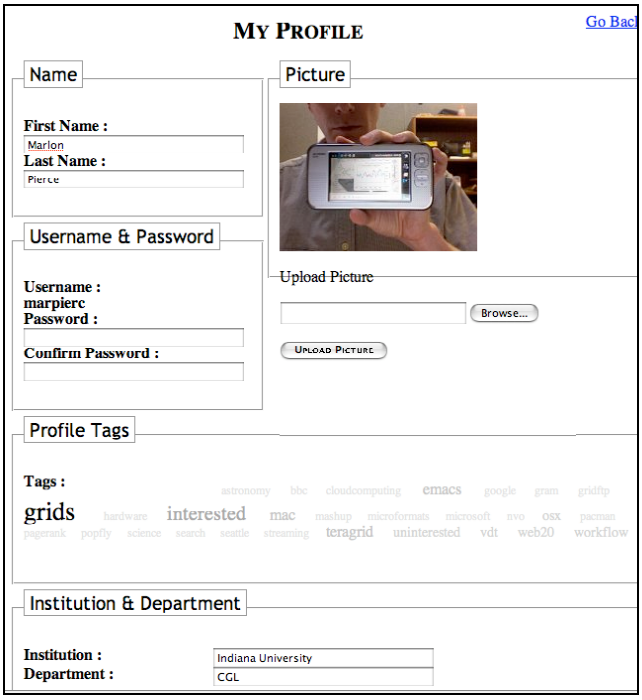


Figure 2 Logged-in users can edit basic user information. Autogenerated information may also be provided. "Profile Tags" section shows the results of a user's interaction with the system.

Figure 4 shows the user's network of friends, list of NSF collaborators, and list of NSF awards. Award and collaborator information sections are automatically created from publicly available data, harvested as described below. Although we have concentrated on NSF data sources in our implementation, we believe the approach can be adapted to other, similar data sources.

As we discuss below, these profiles are discoverable through tag navigation. To illustrate this from the profile point of view, we can see in Figure 2 that the user has a tag cloud resulting from his interactions with the system. The tags such as "Grids" can be used by others while searching and walking the tag graphs underlying the display in Figure 1. These will eventually take users to profile pages such as Figures 2-4.

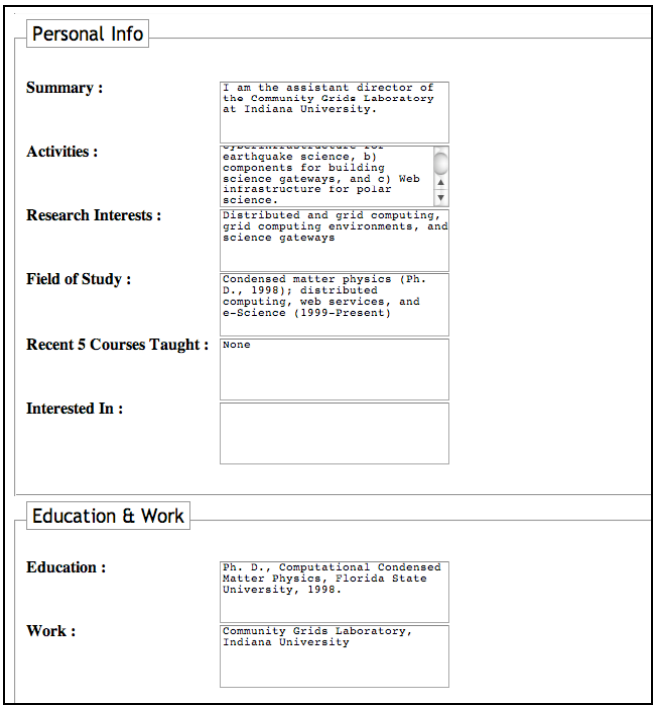


Figure 3 Additional forms allow users to describe professional preparation and research.

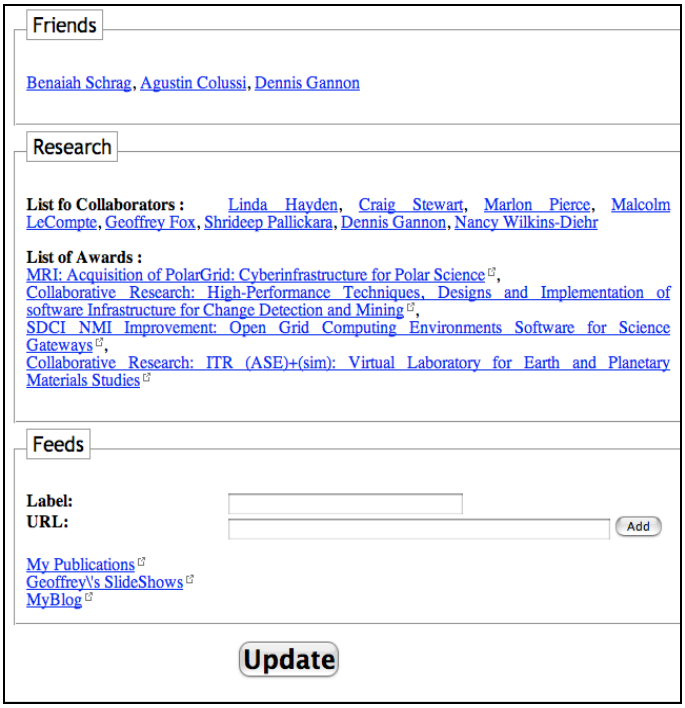


Figure 4 Social networking information, including lists of friends (links to other profiles), collaborators, and funded projects. Users can decorate their profiles with arbitrary RSS feeds such as Connotea publication lists.

B. Tagging a URL

As shown in Figure 1, users' profiles include their tag clouds. These are keyword links to external URLs that a user has found useful or interesting. Bookmarking a link is done in an unobtrusive manner using a small JavaScript bookmarklet that a user drags into the bookmark toolbar (see Figure 5).

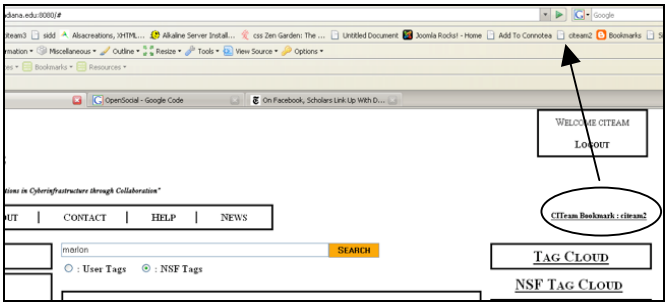


Figure 5 A logged-in user can drag the bookmarklet into the bookmark toolbar.

During usual browsing, a user can click this bookmarklet to post the URL to the portal, along with descriptive tags and keywords. This information is supplied through a popup window. See Figure 6.

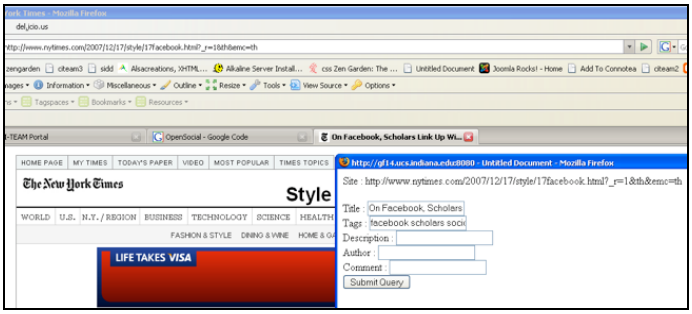


Figure 6 By clicking the portal bookmarklet, a user can tag a particular URL while browsing. The user specifies tag keywords through a popup window (lower right). These are used to generate the tag clouds in Figure 1.

C. Click Tagging a Featured RSS Feed

Although the portal is can be used to bookmark any URL, it is intended to foster research collaborations. To encourage this, we provide relevant RSS feed displays through the portal, such as recent funding announcements from the NSF (Figure 7).

We reformat RSS feeds to allow a user to quickly tag the individual feed entries as "interesting" or "uninteresting". These tags will appear in the user's tag cloud. The list of all such feeds tagged as "interesting" are also available from the system tag cloud, such as shown in Figure 8.

This approach can be used to convert any RSS feed. Unfortunately, not all information on funding is currently available in RSS or Atom syndication formats: grants.gov provides a prominent example. We can convert these sites into RSS feeds using tools such as OpenKapow's RoboMaker (see http://openkapow.com/).

Figure 7 Portal displays of RSS feeds may be "click tagged" as interesting or uninteresting. Tagged material will be displayed in tag clouds.

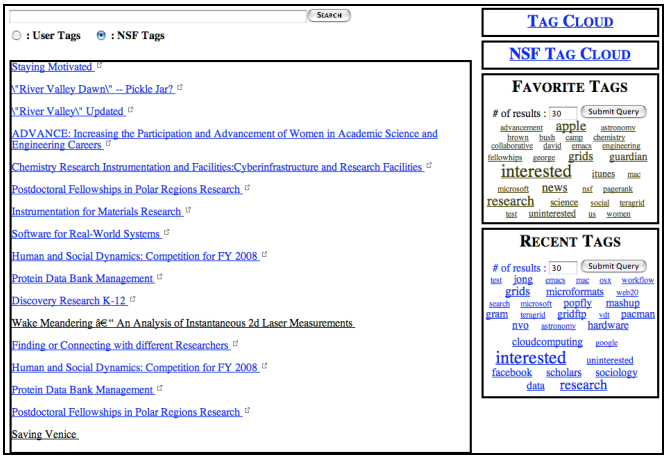


Figure 8 Users can see all recent funding announcements that have been tagged as interesting by clicking the "Interested" tag in either the "Favorite Tags" or "Recent Tags" clouds on the left.

D. Searching NSF Awards

As described below, we populated the system by harvesting publicly available data from sources include the NSF awards database and the TeraGrid allocations database. This information results in several automatically generated tags that are summarized in Table 2.

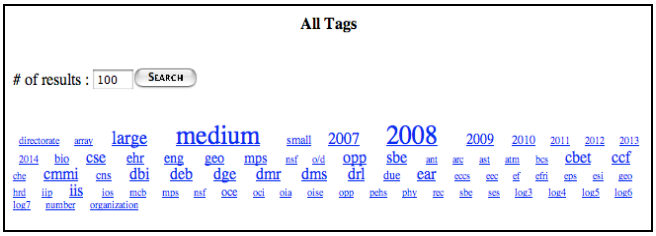


Figure 9 Cloud of all NSF namespace tags.

There are several pathways through this data in the portal. One option is for the user to click the "NSF Tag Cloud" link (left side of Figure 1). This will display the cloud of NSF-namespaced tags in the central display (Figure 9).

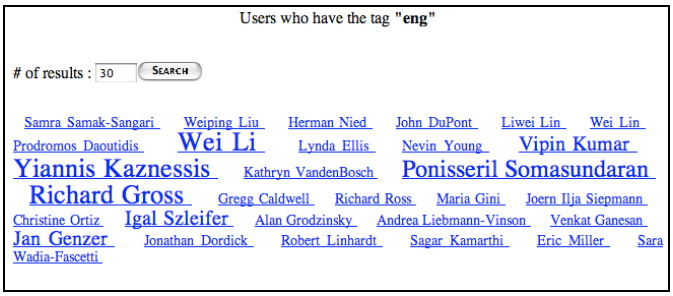


Figure 10 A tag cloud of users funded through NSF ENG.

"Small", "medium", and "large" tags refer to the size of the grant. Years ("2007", "2008", etc) refer to project end dates. Other tags ("cse", "eng", etc) refer to NSF divisions or directorates. Clicking one of these ("eng") produces a cloud of researchers funded through this division (Figure 10).

Figure 11 Tag cloud and funded projects for the user "Wei Li".

By selecting a name from the above cloud ("Wei Li"), a user can see this researcher's tag cloud and list of funded projects. The funded project links are URLs to the appropriate NSF award abstract page.

IV. Implementation Details

A. User Interface Design

The blueprint of our design was distilled from use case scenarios acquired through interviews and discussions with MSI-CIEC team members. In the design phase, content analysis was used to do content mapping where content chunks are formed and then mapped onto the different positions on the web pages.

As shown in Figure 1, the portal is divided into different content components. This has helped in the design and development of the wire-frame. The components are divided

into 4 content areas:

- Header: This contains the logo, title info of the portal, and the login area. The login area uses an Ajax updater library that gives a slide-down effect.
- Footer: this contains redundant navigational links and funding agency acknowledgments.
- Content: The center is the main content area where most of the content is dynamically generated using Ajax libraries imported from Scriptaculous (http://script.aculo.us). Some example content chunks are NSF Tag Clouds, User Tag Clouds, Profile Information, Search Results, RSS Feeds, etc.
- Navigation: The navigational structure is composed of Global, Sub-Global, and contextual navigation. This type of navigation is often described as an *embedded navigation system*. Such navigation helps users in understanding where they are and where they can go on a website. The global navigation in this case consists of global links, namely *Home, News, Contact, Help, and About.* The sub-global navigation on the left consists of a drop down menu for *My Tags* and *My Account.* The right navigational structure consists of modules that are Tag Clouds. There are four different tag cloud structures:
    - User Tag Cloud: containing tags tagged by real users
    - NSF Tag Cloud: containing self-generated tags imported from NSF awards.
    - Favorite Tags: containing the list of favorite tags of all users.
    - Recent Tags: containing tags recently generated by all users.

We implemented the portal with numerous third party tools. These are summarized in Table 1.

Table 1 Third party tools and technologies used in the portal.

Tools / Technologies	Uses
PHP / PEAR	Backend database programming, function calls, creating rss feeds etc.
Scriptaculous Javascript Libraries	Animated visual effects such as drop-downs, draggable and droppable menus, etc.
Adobe Photoshop, Illustrator	Graphic design for the portal, wire frames
Adobe Dreamweaver	HTML/PHP/CSS Editor
MySQL /phpmyadmin utility	Database creation, updating etc.
Google Analytics	Analyze traffic patterns, finding sources where the users come from etc.

B. Grant Information Harvesting

The NSF maintains a publicly searchable online database of awards (see http://nsf.gov/awardsearch/). The online forms use HTTP GET URLs and support several output formats (including XML, text with comma-separated values, and Microsoft Excel spreadsheets) in addition to HTML. This provides us with a REST-like (if undocumented) programming interface that we can use for development. Information retrieved in this fashion includes the following fields:

- Project name,
- Award size,
- Organization,
- Directorate, and
- Co-investigators.

In order to download and incorporate this data into our portal and our tag data model, we decided to use a crawling approach seeded with researcher names. The co-investigators returned in the HTTP response message were used in the next round of searches. Co-investigators were then harvested from those projects and were added to a queue where the same information was downloaded for them. We have currently harvested over 8,600 researchers in this fashion.

We next must convert this information into tags. The NSF query responses are obviously tabular data (see list above for column headings), so these can be converted into tag families, or namespace groups. We convert the individual table entries (such as award size and date for a particular entry) into tags. For entries with ranges of values (award sizes, for example), we have defined tags (i.e., small, medium, and large) with range values. These are summarized in Table 2.

Tags gleaned in this way are prepended with a namespace value (nsf.*). This prevents tag name collisions with user-supplied tags (i.e. "small" may be a user-supplied tag irrelevant to award sizes). It also provides us with a simple organizational label that can be used for separating out the NSF tags into separate clouds.

Table 2 Harvested NSF award and allocation data are converted into tags. We use namespaces to distinguish these tags from user-supplied keywords. Namespaces are not displayed by the portal (i.e. "nsf.date.2008" is displayed as "2008" in a tag cloud.)

Tag Format	Tag Description	Example Tag
nsf.investigator.	The name of an investigator of this project	nsf.investigator.firstname.geoffery nsf.investigator.lastname.fox
nsf.date.	End year of this project	nsf.date.2008
nsf.number.	Award number	nsf.number.0407040
nsf.award	Award size	nsf.award.medium
nsf.organization.	Associated NSF organization	nsf.organization.ast
nsf.directorate.	Associated NSF directorate	nsf.directorate.mps
nsf.tghours.	Allocated teragrid hours (in a log10 format)	nsf.tghours.log6

V. Tagging and Folksonomies

Development of the MSI-CIEC Networking Portal is motivated by a real application, but it also provides us with a test bed for investigating interesting computer science research issues, particularly the application of data mining and clustering techniques to folksonomies.

A. *Exploring communities in collaborative tagging systems*

Collaborative tagging systems have been drawing wide attention as an open medium to freely share information on the Internet. The key aspect of such systems is that objects such as URLs and URIs can be simply tagged by a list of keywords provided by any user. Due to its semantic-free format, collaborative tagging systems have intrinsically a low barrier to promote a user's participation.

Community activities are also an important aspect in most collaborative tagging systems. A user may want to see other people who have tagged on the same object that he or she tagged. A user may want to find a group of people who might have the same interest and look at their bookmarks or resources. To help such users to discover unexposed communities and explore them efficiently in the system, we need to develop and apply data mining algorithms. In the following, we describe the model of tagging system and discuss possible solutions for supporting community exploring.

B. *Models of collaborative tagging system*

The main elements of collaborative tagging systems consist of tags, resources, and users. In most scenarios of using collaborative tagging systems, a user uses tags – which can be keywords, terms, or neologisms – to tag a resource that is normally an URL but generally can include an URI. We can represent those tagging activities as a tuple consisting of a user, a set of tags, and a resource. Alternatively, we can use graphical connections in a tripartite graph where links are drawn between three domains of users, tags, and resources (see Figure 12) [7]. In general, the purpose of such systems is to find specific resources tagged collaboratively by multiple users and retrieve information about resources or users, entangled in the mesh of tags and resources by using query tags.

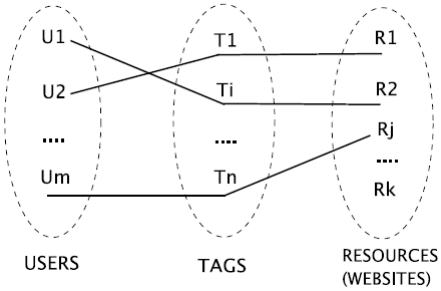
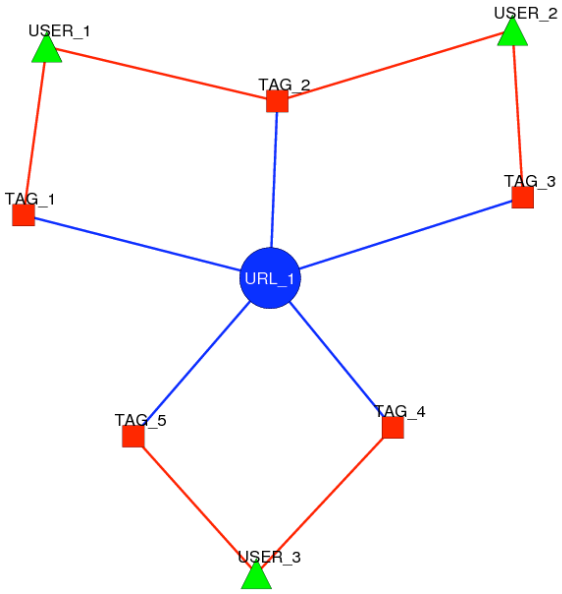
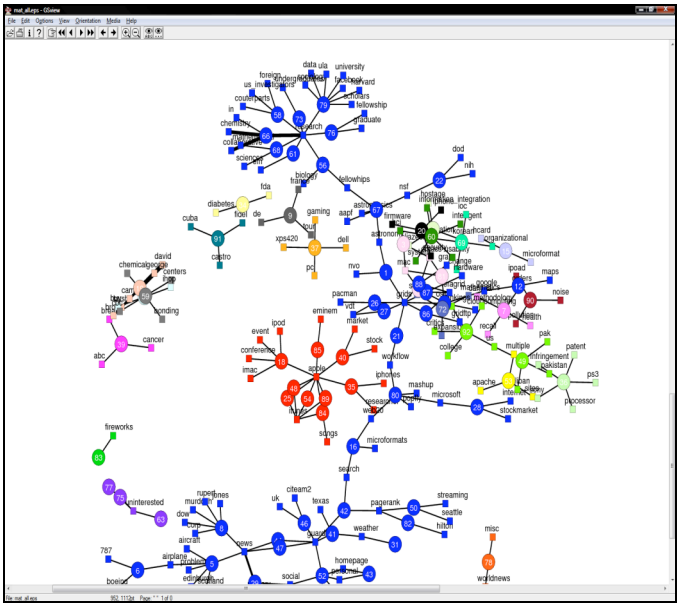


Figure 12. Tripartite graph of a tagging system.

To build a system for this purpose, we can use two different models: a vector space model and a graph model. Although the two models can be convertible to each other in general, they are distinct in their ways of representations and usages. While the vector space model uses vectors in an orthogonal basis tag space, the graph model exploits graph structures of three elements — tags, users, and resources. The vector space model considers the frequencies of tag occurrences for searching, but the graph model focuses on graphical characteristics such as paths and the degree of connectivity between nodes. The vector space model has been widely developed and applied in many different ways in the field of conventional information retrieval for its simplicity, and the graph model has become popular in the areas such as the Internet search engines and social network analysis.



(a)



(b)

Figure 13 (a) A tag graph example and (b) a part of the tag graph of MIS-CIEC portal. Tags, resources (URLs), and users are represented as a square, a circle, and a box respectively. (b)

shows only resource-tag graphs and each independent network (connected graph) is assigned to a unique color.

More precisely, in the vector space model, a resource (or a user)[1] is represented as a vector of tag occurrences in a tag space. For example, a resource tagged by 2 occurrences of tag_1 and 1 occurrence of tag_2 can be expressed as a vector <2, 1>. A dimension is often used to describe the size of a tag space, which equals the number of total tags used in the system. Thus, <2, 1> is a 2 dimensional vector.

In reality, the dimension of these tag vector spaces is huge. Connotea and del.icio.us have tens of thousands of dimensions, and the dimension of our MIS-CIEC portal is about 180. In the vector space model, queries are also given as tag vectors in the same space and then searching is a process to find the exact or, more likely, the most similar vectors.

In practice, since searching a space of tens of thousands dimension is a daunting task, we can use dimension reduction schemes for decreasing dimensions to search by removing noisy and unrelated tags. Latent Semantic Analysis (LSA) and Principal Component Analysis (PCA) are the well-known algorithms for this purpose. We can use the vector space model for finding specific frequency patterns. For example, finding a group of people who share specific set of tags of interest, finding a person whose tags are similar with mine, and so on.

In contrast, the graph model takes advantage of graph structural relationships between tags, resources, and users. Those relationships can be depicted in a graph, which is known as a tag graph, where each tag, a resource and a user are represented as a node and a relationship between them as an edge (see Figure 13(a)). Tag graphs of real systems are more complicated, consisting of thousands of thousands nodes. A part of the tag graph of our MSI-CIEC portal is shown in Figure 13(b).

In this model, graph properties such as connectivity, hop distance, and strength are the important figures to measure, and thus searching is a task to find specific properties in the graph. For example, to find strong relationships between two nodes is to identify a path that consists of a high degree of connectivity but with short hop distance. In this way, we can use the graph model to investigate more sophisticated relationships between tags, resources, and users. Examples of complicated questions we can have include finding a person who is related with my friend, discovering a group of people who is working on the same topic, and so on.

Table 3 A summary of potential questions for discovering communities in a collaborative tagging system and the appropriate algorithm.

Questions	Technology or Algorithm
Who is sharing a similar interest with me?	LSA, TagRank, Graph-based algorithm, Clustering
Which group of people is working on a specific topic?	LSA, Graph-based algorithm
What are the characteristics of a community group?	Clustering algorithm, Graph-based algorithm
Who is the most influential in a community?	LSA, TagRank
What kind of recommendations can I obtain?	TagRank, Graph-based algorithms
How similar is two different communities?	Graph-based algorithm
What is the most outstanding trend?	TagRank

C. Discovering Communities

Tagging a resource that has already tagged by other users, watching other user's tagging activities, and expressing one's interests though tags can be the most common examples of social activities in a network. Now the most of collaborative tagging systems explicitly support community activities by enabling users to create a new community or to join other communities of interest. By doing so, users can actively collect more valuable information by contacting other people in the net who share the same interest of the users.

In this situation, finding a group of people who are working on the same topics or interests, which we call discovering a community, will be an inevitable task in the systems. For example, users may ask to the system; "Who is sharing a similar interest with me?", "Who is the most influential in a community?", or "What kind of recommendations can I obtain?" A list of feasible questions users might ask and the appropriate technique is summarized in Table 3. Considering the size of such networks, solutions for those problems are not trivial and will require efficient, parallel algorithms.

Depending on the models discussed in the previous section, we can classify potential solutions into two main categories: frequency analysis that rely on the vector space model and structural analysis that are based on the graph model.

D. Frequency Analysis and Clustering

Based on the vector space model, the frequency analysis can be performed over the frequencies of tag occurrences in a system. In this analysis, the more frequently used tags, the more referenced resources, and more actively involved users are considered to be more significant and thus have stronger impacts in a system.

Preparing data is relatively simple: one makes a frequency matrix by counting the number of tag occurrences with respect to each resource or each user. Instead of simple counting, more sophisticated methods, such as entropy or scores, can be used. Finding specific patterns means matching a target vector in the frequency matrix. Latent Semantic Analysis is one of the most popular algorithms among many conventional algorithms used in the field of information retrieval [9] for these types of problems.

Latent Semantic Analysis (LSA) has been developed since 1990 for the use of information retrieval [8]. The key ability of LSA is to eliminate statistically unrelated tags from a frequency matrix, which is also known as dimension

reduction, and enable users to compare them with only the most significant components. As a result, LSA helps users to recover "latent" core tags obscured by "noisy" (or the less significant) tags and thus can give more insightful perspectives regardless of presence of noises.

Clustering is another prominent method used for frequency analysis. Discovering communities of similar interests can be performed by identifying clusters of users based on their tag patterns. Well-known clustering algorithms can be applied for this purpose. Hierarchical clustering [11], k-means clustering [12], and deterministic annealing algorithms [13] are good candidates. Other clustering algorithms can be found in [14].

When selecting algorithms, performance is also a critical issue in the frequency analysis since the dimension of frequency matrix will exceed tens of thousands or even more. Indeed, a few clustering algorithms have been designed to deal with high-dimensional problems by exploiting parallelism. For example, parallel hierarchical clustering [15] and parallel k-means method [16] can be found in literature. However, although the most recent advent of multi-core technologies now supports intra-chip parallelisms, very little research has been done so far [17][18] on parallelizing and optimizing these algorithms on these new chip architectures. More performance gains can be obtained in the frequency analysis by adapting those algorithms – LSA and various clustering algorithms – to the multi-core environments.

E. Structural Analysis

In contrast to the frequency analysis, which uses tag frequencies in a vector space, the structural analysis considers the tagging activities as a graph, described in the graph model, and utilizes graph-structural properties in the tag graph for discovering communities in a collaborative tagging system. Compared with the data used in the frequency analysis (i.e., frequency matrix), the representation of data in a graph structure is more intuitive and human-understandable. For this reason, structural analysis may help users to find other information that is not obtainable when performing the frequency analysis. Example properties we may want to find out are connectivity, connection distances between users, size of communities, and the degree of strength of a connection.

In literature, many graph-based algorithms can be found for the structural analysis. Among them, FolkRank [10] and graph-based clustering algorithms, as shown in [19][20][21], are applicable in our purpose.

The concept of FolkRank algorithm, which is a variant of well-known PageRank algorithm of Google, is to assign each node – which is a tag, a user, or a resource – a system-wide numeric score, also known as a rank, by measuring contributions or a degree of importance in system. To obtain such rank scores, the algorithm starts with random seeds of nodes and recursively follows sub-graphs by utilizing the graph structures of tagging. This process is iteratively repeated until the scores converge to a certain threshold.

Like other clustering algorithms used in the frequency analysis, graph-based clustering algorithms, as shown in [19][20][21], can be used for identify or searching similar group of people in a system. Similarly, there is very little study in literature on parallel graph-based clustering algorithms working on multi-core environments, so these remain open and important problems.

VI. Summary and Future Work

This paper describes the design and implementation of the MSI-CIEC Networking Portal, a Web 2.0-style tagging and social network style application. This work is motivated by the need to support social networks of researchers, particularly at minority serving institutions.

In addition to this practical motivation, we hope also to use the portal as a laboratory for core computer science work on social network analysis. As described in Section V, we are researching the application of various techniques for clustering and mining the data we are harvesting. Although some form of this is quite familiar from many social Web sites, we hope to put the techniques on a firm, open academic footing, avoiding proprietary and ad hoc algorithms.

The key problem with most social network applications is the lack of interoperability, but this fortunately is beginning to change. The major social network activities, Facebook and the Google-led Open Social consortium, are both providing programming APIs that allow developers to embed applications in existing social networks and, conversely, allow embedding social network tools into other Web sites. It will be crucial for our project, in the next phase, to establish interoperability with these social networking tools.

Acknowledgment

The MSI-CIEC portal is supported by the National Science Foundation's CITEAM project, Award Number SCI-0537498.

References

[1] O'Reilly, Tim, "What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software." 2005. Available from http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.

[2] Marlon E. Pierce, Geoffrey Fox, Huapeng Yuan, and Yu Deng Cyberinfrastructure and Web 2.0 Proceedings of HPC2006 July 4 2006 Cetraro Italy.

[3] Geoffrey C. Fox, Rajarshi Guha, Donald F. McMullen, Ahmet Fatih Mustacoglu, Marlon E. Pierce, Ahmet E. Topcu, and David J. Wild, "Web 2.0 for Grids and e-Science." INGRID 2007: Instrumenting the Grid. 2nd International Workshop on Distributed Cooperative Laboratories, S.Margherita Ligure Portofino, ITALY, April 18 2007.

[4] Geoffrey Fox, "Some Comments on CiteULike, Connotea, and Related Tools." Community Grids Laboratory Technical Report, January 1 2006. Available from http://grids.ucs.indiana.edu/ptliupages/publications/ToolsEvaluation.doc.

[5] Geoffrey Fox, Ahmet Fatih Mustacoglu, Ahmet E. Topcu, Aurel Cami: SRG: A Digital Document-Enhanced Service Oriented Research Grid. IRI 2007: 61-66.

[6] NSF Upcoming Funding Announcements RSS Feed: http://www.nsf.gov/rss/rss_www_funding_upcoming.xml

[7] Halpin, H., Robu, V., and Shepherd, H. 2007. The complex dynamics of collaborative tagging. In Proceedings of the 16th international Conference on World Wide Web (Banff, Alberta, Canada, May 08 - 12, 2007). WWW '07. ACM, New York, NY, 211-220.

[8] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41(6), 391-407.

[9] David A. Grossman and Ophir Frieder. Information Retreival – Algorithms and Heuristics. Springer. 2004
[10] A. Hotho, R. Jaschke, C. Schmitz and G. Stumme, "Information retrieval in folksonomies: Search and ranking," The Semantic Web: Research and Applications, vol. 4011, pp. 411-426, 2006.
[11] Y. Zhao, G. Karypis and U. Fayyad, "Hierarchical Clustering Algorithms for Document Datasets," Data Mining and Knowledge Discovery, vol. 10, pp. 141-168, 2005.
[12] C. Ding and X. He, "K-means clustering via principal component analysis," ACM International Conference Proceeding Series, 2004.
[13] K. Rose, "Deterministic Annealing for Clustering, Compression, Classification, Regression and Related Optimization Problems," Proc. IEEE, vol. 86, pp. 2,210-2,239, 1998.
[14] R. Xu and D. Wunsch, "Survey of clustering algorithms," IEEE Trans.Neural Networks, vol. 16, pp. 645-678, 2005.
[15] E. Dahlhaus, "Parallel Algorithms for Hierarchical Clustering and Applications to Split Decomposition and Parity Graph Recognition," Journal of Algorithms, vol. 36, pp. 205-240, 2000.
[16] K. Stoffel and A. Belkoniene, "Parallel K-Means Clustering for Large Data Sets," Proceedings Euro-Par, vol. 99, pp. 1451-1454, 1999.
[17] C.T. Chu, S.K. Kim, Y.A. Lin, Y.Y. Yu, G. Bradski, A.Y. Ng, K. Olukotun and R. Inc, "Map-Reduce for Machine Learning on Multicore," Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference, 2007.
[18] X. Qiu, G.C. Fox, H. Yuan, S.H. Bae, G. Chrysanthakopoulos, H.F. Nielsen and W. Redmond, "High Performance Multi-Paradigm Messaging Runtime Integrating Grids and Multicore Systems,", in proceedings of eScience 2007 Conference, Bangalore, India, 2007.
[19] G. Karypis, E.H. Han and V. Kumar, "Chameleon: hierarchical clustering using dynamic modeling," Computer, vol. 32, pp. 68-75, 1999.
[20] E. Hartuv and R. Shamir, "A clustering algorithm based on graph connectivity," Information Processing Letters, vol. 76, pp. 175-181, 2000.
[21] G. Karypis, E.H. Han and V. Kumar, "Chameleon: hierarchical clustering using dynamic modeling," Computer, vol. 32, pp. 68-75, 1999.