Expertise Networks in Online Communities: Structure and Algorithms

Jun Zhang
School of Information
University of Michigan
junzh@umich.edu

Mark S. Ackerman
Dept. of EECS and School of Information
University of Michigan
ackerm@umich.edu

Lada Adamic
School of Information
University of Michigan
ladamic@umich.edu

ABSTRACT

Web-based communities have become important places for people to seek and share expertise. We find that networks in these communities typically differ in their topology from other online networks such as the World Wide Web. Systems targeted to augment web-based communities by automatically identifying users with expertise, for example, need to adapt to the underlying interaction dynamics. In this study, we analyze the Java Forum, a large online help-seeking community, using social network analysis methods. We test a set of network-based ranking algorithms, including PageRank and HITS, on this large size social network in order to identify users with high expertise. We then use simulations to identify a small number of simple simulation rules governing the question-answer dynamic in the network. These simple rules not only replicate the structural characteristics and algorithm performance on the empirically observed Java Forum, but also allow us to evaluate how other algorithms may perform in communities with different characteristics. We believe this approach will be fruitful for practical algorithm design and implementation for online expertise-sharing communities.

Categories and Subject Descriptors

H.5.3. [**Information Interfaces and Presentation (e.g., HCI)**]: Group and Organizational Interfaces – *collaborative computing, computer-supported cooperative work, theory and models, web-based interaction.* J.0 [Computer Applications] General.

General Terms

Human Factors, Algorithms, Experimentation.

Keywords

Social network analysis, expertise finding, expert locators, help seeking, online communities, simulation

1. INTRODUCTION

Steve is a Java programmer who just started working on a project using Java Speech on a new mobile platform. But he cannot run his first Java Speech program on the new platform and needs some help. Steve is unable to tell whether the problem has arisen because he does not understand how to use the Java Speech package, or because Java Speech does not support the mobile platform well.

WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

It can be difficult to get a satisfactory answer to Steve's problem by searching Google directly. Instead, he may prefer to find and ask someone who has related expertise or experience, and online communities have emerged as one of the most important places for people to seek advice or help. The topics range from advice on medical treatment, programming, software, building a computer from scratch to repairing the kitchen sink. These communities are usually bound by shared professions, interests, or products among their participants. For instance, the Sun Java Forum has thousands of Java developers coming to the site to ask and answer questions related to Java programming every day. The Microsoft TechNet newsgroup is a major place for programmers to seek help for programming questions relating to Microsoft products. Even though users in these online communities usually do not know each other and are identified using pseudonyms, they are willing to help each other for various reasons, such as altruism, reputation-enhancement benefits, expected reciprocity, and direct learning benefits [16, 18].

This work seeks to enhance online communities with expertise finders. Expertise finders, or expertise location engines, are systems that help find others with the appropriate expertise to answer a question. These systems have been explored in a series of studies, including Streeter and Lochbaum [24], Krulwich and Burkey [17], and McDonald and Ackerman [2] as well as the studies in Ackerman et al. [3]. Newer systems, which use a social network to help find people, have also been explored, most notably in Yenta [12], ReferralWeb [14], and most recently commercial systems from Tacit and Microsoft. These systems attempt to leverage the social network within an organization or community to help find the appropriate others.

Aside from relying on social networks, another interesting characteristic of these systems is that they tend to blur the dichotomy between experts and seekers. They treat one's expertise as a relative concept [3]. In reality, relatively few people will claim themselves as an expert, but many people agree that they have some measure of expertise in some area. These systems allow everyone to contribute as they can.

For these expertise finder systems to be of significant assistance, they must effectively identify people who have expertise in the area desired by the asker. Most current systems use modern information retrieval techniques to discover expertise from implicit or secondary electronic resources. A person's expertise is usually described as a term vector and is used later for matching expertise queries using standard IR techniques. The result usually is a list of related people with no intrinsic ranking order or ranks derived from term frequencies. It may reflect whether a person knows about a topic, but it is difficult to distinguish that person's relative expertise levels. Relying on word and document frequencies has proven to be limited [19] .

To ameliorate this, Campbell et al. [8] and Dom et al. [9] used graph-based ranking algorithms in addition to content analysis to rank users' expertise levels. This work, done at IBM Research, applied several graph-based algorithms, including PageRank and HITS, to both a synthetic network set and a small email network to rank correspondents according to their degree of expertise on subjects of interest. They found that using a graph-based algorithm effectively extracts more information than is found in content alone. However, there is a weakness in these studies. The size of their networks is very small and does not reflect the characteristics of realistic social networks.

As a result, we wished to revisit the possibilities of using graph-based algorithms on social networks of users in online communities. In this study, we analyze a large online help seeking community, the Java Forum, using social network analysis methods. We then test a set of network-based algorithms, including PageRank and HITS, on this large size social network. Using a set of simulations, we explore how various network structures affect the performance of these algorithms. We find a small number of structural characteristics in the social networks that we believe lead to differences in the algorithms' performance for online communities. We expect that not only will these characteristics be fruitful for practical algorithm design and implementation, but that they will offer new research insights for others to explore.

The paper proceeds as follows. In Section 2, we introduce the community expertise network and briefly review related work. In section 3, we describe the network characteristics of our test online community, the Java Forum. In section 4, we describe some expertise ranking algorithms. In section 5, we present an evaluation comparing the rankings produced by human raters and by the algorithms. In section 6, we then explore the network characteristics that affect the performance of these algorithms using a simulation study. And finally, we summarize our findings in Section 7.

2. EXPERTISE NETWORK IN ONLINE COMMUNITIES

Online communities usually have a discussion thread structure. A user posts a topic or question, and then some other users post replies to either participate in the discussion or to answer a question posed in the original post. Using these posting/replying threads in a community, we can create a post-reply network by viewing each participating user as a node, and linking the ID of a user starting a topic thread to a replier's ID, as shown in Figure 1.

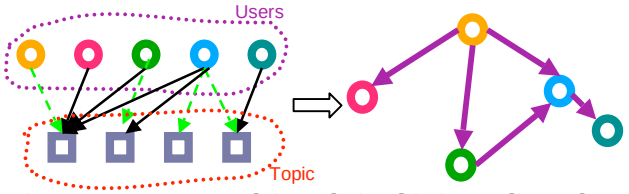


Figure 1: We map a replying relationship into a directed graph. On the left we have a bipartite graph of users (circles) and the discussion threads (squares) they participated in. This is transformed to a directed graph where an edge is drawn from the user making the initial post (the dashed edge shown in green) to everyone who replied to it.

This post-reply network has some interesting characteristics. First, it is not intentionally built by its users for the purpose of forming ties. Thus, it is not a network focused on social relationships. Instead, it reflects community members' shared interests. Whether it is a community centered on questions and answers, social support, or discussion, the reason that a user replies to a topic is usually because of an interest in the content of the topic rather than who started the thread. This indirectly reflects a particular shared interest between the original poster and the repliers (although the repliers' sentiment about the topic may differ).

Furthermore, in a question and answer community, the direction of the links carries more information than just shared interest. A user replying to another user's question usually indicates that the replier has superior expertise on the subject than the asker. The distribution of expertise, along with the network of responses, is what we will call the *community expertise network (CEN)*. It indicates what expertise exists within an online community, as well as how it is distributed in practice.

The full dynamic of a CEN may be much complex in some communities. For example, there may be trolls, spammers, etc. An answer thread to a question can be the result of a complex social process and the first few replies may actually not answer the question but try to clarify the problem. The network could be weighted according to the frequency of how often a user helps another. We will discuss these issues in later sections.

Structural Prestige in Social Networks

Expertise is closely related to structural prestige measures and rankings in social network studies. In directed networks, people who receive many positive choices are considered to be prestigious, and prestige becomes salient especially if positive choices are not reciprocated [25].

Researchers in various fields have applied these prestige ideas to different types of networks. Fisher et al. [11] used social network visualization and analysis on the patterns of replies for each author in selected newsgroups to find different types of participants. For instance, they used the indegree (how many people a user replied to) and outdegree (how many people replied to the user) of a user's egocentric network to identify the roles within the group (e.g., general asker or replier). Bollen et al. [5] used a similar ranking measure to evaluate the prestige of academic journals. Liu et al. [20] used it to evaluate the impact of an individual author in a co-authorship network. And, of course Page et al. [22] used PageRank to rank web pages.

In online help-seeking communities, the social network is an expertise network. Because the way links are constructed, the prestige measure of the network is highly correlated with a user's expertise. Thus, this hints that there are opportunities to make use of such network structures to rank people's expertise in online communities, and build related applications/systems that further improve the expertise sharing in the online world.

Next we turn to the investigation of an expertise network in one online community, the Java Forum.

3. EMPIRICAL STUDY OF AN ONLINE COMMUNITY

3.1 The Java Forum

The Java Developer Forum is an online community where people come to ask questions about Java. It has 87 sub-forums that focus

on various topics concerning Java programming. There is a large diversity of users, ranging from students learning Java to the top Java experts. Users usually can get an answer relatively quickly because of the large number of participants. In this study, we used the Java programming sub-forum (called here "Java Forum"), which is a place for people to ask general Java programming questions. The Java Forum had a total of 333,314 messages in 49,888 threads.

We used the network constructed upon these threads to evaluate the usefulness of our expertise-ranking algorithms. The Java Forum network had 13,739 nodes and 55,761 edges.

The next section describes the characteristics of the Java Forum network. This will provide both a test bed for the algorithms and, later in the paper, will help in understanding the underlying network characteristics that expertise ranking algorithms operate upon.

3.2 Characterizing the Network

3.2.1 The Bow tie structure analysis

Not all users in the Java Forum ask questions, nor do all users answer questions. Using a bow tie structure analysis, we examine the general structure of the Java Forum network.

The bow tie structure, first proposed by researchers at IBM, AltaVista, and Compaq, yields insights into the complex organization of the Web network structure. Its key idea is that the web is a bow tie and has four distinct components: Core, In, Out, and 'Tendrils' and 'Tubes' (see Broder et al. [7]). In our bow tie model, a central core contains users that frequently help each other. It is a strongly connected component (SCC), meaning that one can reach every user from every other by following questioner-answerer links. The 'In' component contains users that usually only ask questions. The 'Out' consists of users that usually only answer questions posted by users in the Core. Other users, the 'Tendrils' and 'Tubes', connect to either the 'In' or 'Out' clusters, or both, but not to the Core. They are users who only answer questions posed by 'In' users or whose questions are only answered by 'Out' users.

Figure 2, 3 and Table 1 compare the bow tie structure of the Java Forum network with that of the Web (as reported in [7]).

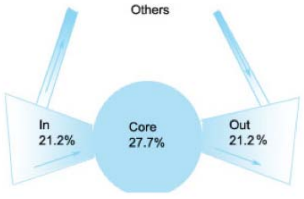


Figure 2: The web is a bow tie

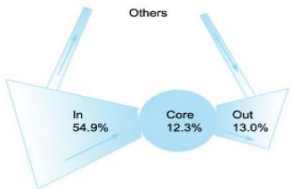


Figure 3: The Java Forum network is an uneven bow tie

Table 1: Comparison of bow tie analysis between Web and the Java Forum network

	Core	In	Out	Tendrils	Tubes	Disconnect
Web	27.7%	21.2%	21.2%	21.5%	0.4%	8.0%
Forum	12.3%	54.9%	13.0%	17.5%	0.4%	1.9%

These results show the Java Forum network looks much different from the Web. The Java Forum has a much bigger 'In' component and a relatively smaller Core than the Web. This indicates that in this online community, only about 12% of users actively ask and answer questions for each other. More than half of the users usually only ask questions, and about 13% users usually only answer questions. This result also indicates that instead of being a public place where people help each other reciprocally, this online help seeking community is more closely a place where askers come to seek help from volunteer helpers.

3.2.2 Distribution of degree

We can use the bow tie structure to show the role of users in the network, but it does not capture the level of their interaction. Looking at degree distributions is a general way to describe users relative connectedness in a large complex network [21]. The degree distribution is a function describing the number of users in the network with a given degree (number of neighbors). An interesting common feature of many known complex networks is their scale-free nature. In a scale-free network, the majority of nodes are each connected to just a handful of neighbors, but there are a few hub nodes that have a disproportionately large number of neighbors. Figure 4 shows the indegree distribution histogram for the Java Forum network. It is highly skewed (and in fact scale-free except for a cutoff at very high degrees), similar to a distribution observed for Web pages and for co-authorship networks. The scale-free degree distribution is a reflection of the highly uneven distribution of participation. Instead of everybody helping each other equally, in the Java Forum, there are some extremely active users who answer a lot of questions while a majority of users answer only a few. Likewise, many users ask only a single question, but some ask a dozen or more.

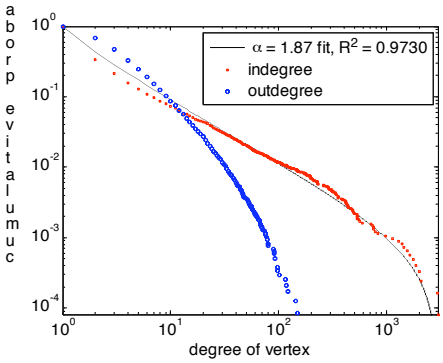


Figure 4: Degree distribution of the Java Forum network

3.2.3 Degree correlations

While the indegree distribution shows how many people a given user helps, it gives no information about those users' own tendency to provide help. For example, one might like to know whether high volume repliers only reply to newbies, or if they mostly talk to others similar to themselves. We can answer both of these questions by looking at the correlation profile (see Maslov et al. [23]) Here we consider a simplified correlation profile that for each asker-replier pair counts the indegree of the replier versus the indegree of the asker, as shown in Figure 5. We also report a simple correlation coefficient between the askers' and helpers' indegree.

Positive assortativity is common in social networks, where people with many connections tend to know other people with many

connections while hermits tend to know other hermits. We find however, that the Java Forum is far from an exclusive club where high volume repliers correspond with other high volume repliers, leaving the newbies to talk to one another. Rather, the Java forum is neither assortative nor disassortative. The correlation coefficient is ever so slightly negative at -0.013, and the correlation plot shows that the highest degree nodes (usually the experts) tend to answer questions across the board from whoever asks them. As one might expect, low degree users (ones who probably lack the expertise to answer others' questions) typically do not reply to high-degree users.

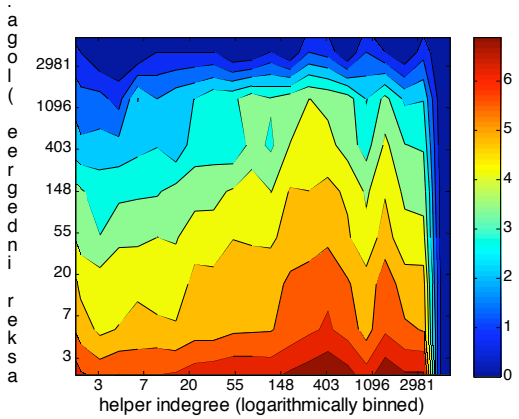


Figure 5: The correlation profile of the Java Forum network. The color corresponds to the logarithm of the frequency of such degree pairings.

In summary, from these network analyses, we can see that the Java Forum network has some unique characteristics, including:

- Different groups of users fall into structurally distinct parts of the network: There is a big 'In' group and relatively small Core and 'Out' groups.
- The users' indegree distribution is skewed, with few users answering a large number of questions while the majority of users only answer a few.
- Top repliers answer questions for everyone. However, less expert users tend to answer questions of others with lower expertise level.

Since these characteristics are different from the World Wide Web graph, they can potentially affect the performance of various expertise ranking algorithms, as we will discuss next.

4. EXPERTISE RANKING ALGORITHMS

After constructing an expertise network from the post-reply patterns in the online community, and having discovered interesting regularities in the structure of the network which might correlate with a user's expertise, we now present several algorithms designed to automatically infer a user's expertise level. After presenting the algorithms, we will provide the results of their tests.

4.1 Simple Statistical Measures

We surmise that if a person answers a lot of questions on a topic, it is often the case that he or she knows the topic well. Exceptions include spammers who may be posting advertisements or trolls who may be making inflammatory or otherwise disruptive posts. We found little trolling or spamming behavior on the Java Forum. However, our observations here would also be applicable to forums where spamming is more prevalent, but can be curbed or identified through users' relevance feedback. Returning to the Java Forum, the simplest method for evaluating a user's expertise may be counting the number of questions answered. We call it the "AnswerNum" measure.

A slightly different measure is counting how many other users a user helped. Some users may have a big AnswerNum but all these replies are answering questions repeatedly from several specific users. On the other hand, a user who posts fewer answers, but in the process helps a greater number of users, could have broader or greater expertise. Thus, counting how many people one helps may be a better indicator than counting the number of replies. In a social network, this could be calculated using the indegree of a node.

4.2 Z-score Measures

While replying to many questions implies that one has high expertise, asking a lot of questions is usually an indicator that one lacks expertise on some topics. Thus, we propose the "z-score" as a measure that combines one's asking and replying patterns, as shown in following formula: If a user makes $n=q+a$ posts, q of them questions and a of them answers, we would like to measure how different this behavior is from a 'random' user who posts answers with probability p = 0.5 and posts new questions with probability 1-p = 0.5. We would expect such a random user to post n*p = n/2 replies with a standard deviation of $\sqrt{n*p*(1-p)} = \sqrt{n}/2$. The z-score measures how many standard deviations above or below the expected 'random' value a user lies:

$$z = \frac{a - n/2}{\sqrt{n}/2} = \frac{a - q}{\sqrt{a+q}}$$

If a user asks and answers about equally often, their z-score will be close to 0. If they answer more than ask, the z score will be positive, otherwise, negative. We calculate the z-score for both the number of questions one asked and answered and the number of users one replied to and received replies from, denoted separately as "Z_number" and "Z_degree".

4.3 ExpertiseRank Algorithm

There is a potential problem in counting the number of answers one posted or the number of people one helped. A user who answers 100 newbies' questions will be ranked as equally expert as another user who answers 100 advanced users' questions. Obviously the latter usually has greater expertise than the former.

The well known PageRank algorithm, proposed by Page et al. [22] for ranking web pages, improves this. It provides a kind of peer assessment of the value of a Web page by taking into account not just the number of pages linking to it, but also the number of pages pointing to those pages, and so on. Thus, a link from a popular page is given a higher weighting than one from an unpopular page. Intuitively, the ranking in PageRank corresponds to the fraction of time a random walker would spend 'visiting' a page by iteratively following links from page to page. There are various versions of PageRank or similar measures; for an overview, see [4, 6].

We propose using a PageRank-like algorithm to generate a measure that not only considers how many other people one

helped, but also whom he/she helped. We call it "ExpertiseRank". The intuition behind ExpertiseRank is that if B is able to answer A's question, and C is able to answer B's question, C's expertise rank should be boosted not just because they were able to answer a question, but because they were able to answer a question of someone who herself had some expertise. In a sense, ExpertiseRank propagates expertise scores through the question-answer network.

Table 2 lists the ExpertiseRank algorithm that is similar to PageRank.

Table 2: Basic ExpertiseRank algorithm

Assume User A has answered questions for users $U_1 \ldots U_n$. , then the ExpertiseRank (ER) of User A is given as follows:

$$ER(A) = (1-d) + d\,(ER(U_1)/C(U_1) + \ldots + ER(U_n)/C(U_n))$$

$C(U_i)$ is defined as the total number of users helping U_1, and the parameter d is a damping factor which can be set between 0 and 1. We set d to 0.85[2] here. The damping factor allows the random walker to `escape' cycles by jumping to a random point in the network rather than following links a fraction (1-d) of the time.

ExpertiseRank or ER (A) can be calculated using a simple iterative algorithm.

Note that an expertise network could be weighted. For instance, we can add values to edges by how frequent one replies another. We can also weight each ask-reply occurrence differently based on how many replies there are in a question thread. It is straightforward to extend the notion of Expertise rank to incorporate the weights of the edges by substituting ER(Ui) with $ER(U_i)*w_{iA,}$ where w_{iA} is the number of times i was helped by A and $C(U_i)=\Sigma w_{ij}$. In our particular study, we found that weighting does not improve the accuracy of our results, so for simplicity we treat the networks as unweighted, although weights can easily be reintroduced for other applications.

4.4 HITS Authority

Another ranking algorithm similar to PageRank is HITS ("Hypertext induced topic selection") [15]. It also uses an iterative approach, but assigns two scores to each node: a hub score and an authority score. In our context, a good hub is a user who is helped by many expert users, and a good authority (an expert) is a user who helps many good hubs. The definition is recursive and converges after a few iterations. In our study, we used the Authority value of HITS to correspond to the expertise rank of the user.

5. EVALUATION

Since there was no explicit user-supplied expertise ranking data in the Java Forum, we needed to use human raters to generate a "gold standard" for comparison. Because it was not possible for us to rate a large number of these users, we randomly selected 135 users from the network for use as a comparison sample. By omitting those users posting fewer than 10 times, we ensured that the sampled users had generated enough Forum content for a reviewer to evaluate their expertise levels.

While some of the ranking algorithms such as ExpertiseRank and HITS can in principle produce continuous values that can potentially differentiate between all users, it is very difficult for humans to sort 135 users into a ranked list. Raters must read from ten to hundreds of messages posted by a user to evaluate his/her expertise level. It is also difficult to compare two users when they both have posted many messages but have not replied to each other.

Based on our observation of the forum and the results of a pilot rating set, we decided to categorize the users into 5 expertise levels instead of a complete ranked list. Table 3 displays details of these categorizations.

Table 3: Five levels of expertise rating

Level	Category	Description
5	Top Java expert	Knows the core Java theory and related advanced topics deeply.
4	Java professional	Can answer all or most of Java concept questions. Also knows one or some sub topics very well,
3	Java user	Knows advanced Java concepts. Can program relatively well.
2	Java learner	Knows basic concepts and can program, but is not good at advanced topics of Java.
1	Newbie	Just starting to learn java.

We found two raters who are Java programming experts to rate the 135 users' expertise. (These experts were not part of the research team; they were independent consultants.)

5.1 Statistical Metrics

Two of the most frequently used correlation measures between two ranks are Spearman's rho and Kendall's Tau [10, 13].

Both of these metrics have their limitations. The Spearman correlation does not handle weak orderings well (weak ordering means that there are multiple items in the ranking such that neither item is preferred over the other) and our rankings have a lot of weak orderings because multiple users are assigned the same rating. Kendall's Tau, on the other hand, gives equal weight to any interchange of equal distance, no matter where it occurs. For instance, an interchange between rank 1 and 2 will be just as bad as interchange between rank 100 and rank 101. Kendall's Tau may be a better metric for our purpose. Nevertheless, for the evaluation, we present both Kendall's Tau and Spearman's rho. Furthermore, we have also added a "TopK" metric, which calculates a Kendall's Tau for only the highest 20 ranks.

After each human rater submitted his ratings, we tested the reliability of raters by looking at their inter-rater correlation. The Kendall's Tau distance between the two human raters was 0.736, and the Spearman's rho correlation coefficient was 0.826 ($p<0.01$), a sufficiently high rate of inter-rater correlation.

5.2 Results

To have a conservative measurement of the possible performance for the automatic algorithms, we further removed 10 samples whose ratings have more than 1 level difference between the two raters. The Spearman's rho is 0.832 and Kendall's Tau is 0.796

[2] We tried various values (such as 0.95 and 0.70), but it did not make a significant difference.

between the two raters for the 124 users left. (One user was not rated because raters reported that they didn't have enough evidence.) Therefore, we may expect that any automated algorithm would at best achieve around a 0.8 correlation with the human raters. For each of these users, in the data analysis below, we summed the ratings from the two raters together as the standard human rating (HR).

Figure 6 shows the statistical correlations between various algorithms and the human ratings of the 124 users. (A sensitivity analysis including all 134 users showed insignificant differences.)

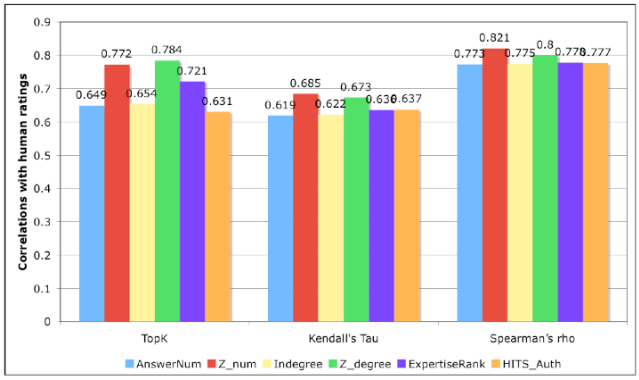


Figure 6: The performance of various algorithms in different statistical metrics

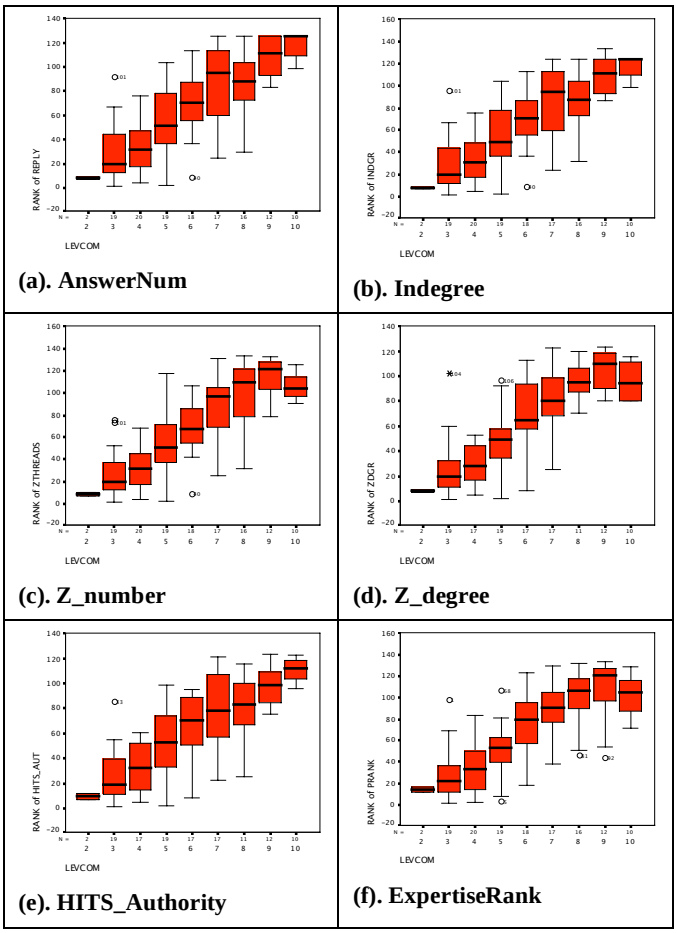
From the figure, one can see that all of these ranking algorithms give a relatively high correlation with the human-assigned ratings. This tells us that, indeed, structural information could be used to help evaluate users' expertise in online community networks.

Surprisingly, contrary to what Campbell et al. [8] and Dom et al. [9] found in their simulation studies, we found that, in this real network data set, ExpertiseRank actually does not perform better than other simpler methods. Instead, the z-score-based ranks tend to produce slightly better results than other methods. We will return to this in the subsequent analysis, where we try to find social network features that explain this result.

We can also see that different correlation metrics produce different results when comparing the same data. For instance, while Z_degree shows the highest correlation with the TopK metric, it is the Z_number that shows the highest correlation with the complete Kendall's Tau and Spearman's Rho metrics. In many applications, we may care more about whether the algorithm can identify the top K experts, rather than whether it can rate everyone's relative expertise. Being aware of these differences in metrics can help one choose an appropriate algorithm depending on whether it is the top experts one is after.

We further looked at the distribution of automatic rankings (summarized by the box plots shown in figure 7) corresponding to the human rating levels[3]. From these box plots, we can see the results are consistent with what we found in Figure 6. We can see that the Z_number, Z_degree, and ExpertiseRank all have a slightly smaller inter-quartile range at each human rating level, which indicates that they typically have smaller errors.

[3] We use the rating combination of two raters here, so there is a total of 10 categories.



(a). AnswerNum **(b). Indegree**

(c). Z_number **(d). Z_degree**

(e). HITS_Authority **(f). ExpertiseRank**

Figure 7: Box plots of algorithm rankings vs. human ratings

While it is interesting to look at the details of these results, it is more important to think about the big picture. We have observed a network structure different from the Web, and we have also seen that some algorithms, such as PageRank and HITS, which excel at ranking Web pages, do not outperform simpler algorithms in this network. The key to understanding the performance of the algorithms is in understanding the human dynamics that shape an online community. This understanding will then help select algorithms that may be more appropriate for other online communities where the dynamics may be different from the Java Forum. The approach we took was simulation: taking the simplest set of interaction rules that both replicated the observed structure and the relative performance of various algorithms.

We next present the results of those simulations.

6. SIMULATIONS

Much recent work on modeling of complex networks in social, biological and technological domains has focused on replicating one or more aggregate characteristics of real world networks, such as scale-free degree distributions, clustering, and average path lengths[21]. For instance, the preferential attachment network growth model of Barabasi et al. [1], where new nodes joining preferentially connect to well connected nodes, yields scale-free degree distributions.

Here, we take a different approach. We place an emphasis on studying the various factors that possibly affect the structure of the network. Instead of having a targeted network to generate, we let various factors determine the growth of the network and observe how changes in those factors affect the structure of the network. Figure 8 shows a snapshot of the simulator we developed to study how these various network characteristics (the corresponding controls are hidden in the figure) will affect the structure of the network in an online help-seeking community and in turn how they affect the performance of various ranking algorithms (shown in the plots and tables adjacent to the network layout). Details of this simulator can be found in [26].

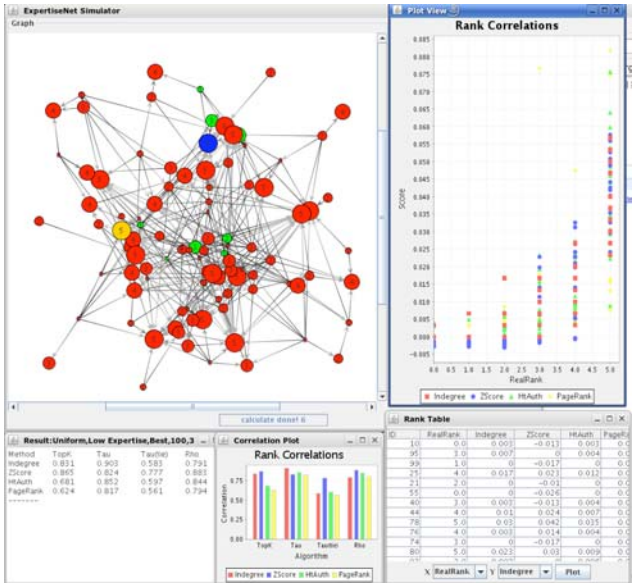


Figure 8: Snapshot of the network simulator interface

6.1 Modeling Java Forum's Network

From the empirical analysis of the Java Forum, we incorporated the following dynamics governing the forum into our model:

- The majority of users made few posts, either because they were new or had low expertise.
- There were a number of experts who mainly answered others' questions and seldom asked questions themselves.
- Users seemed to answer others' questions according to their own ability corresponding to their level of expertise.

First, we initialized the community with 1,374 users in the community (one-tenth of the observed population of the Java Forum) with a power law distribution for the levels of expertise. There were many level 1 (novice) users and relatively few level 5 (expert) users.

Second, we modeled that low-level users have high probabilities to ask questions. A user ***u*** with expertise level ***L(u)*** has the probability to ask questions $P_A(u)$ determined by the formula below:

$$P_A(u) = \frac{(L(u)+1)^{-1}}{\sum_v (L(v)+1)^{-1}}$$

Third, we modeled which users were most likely to answer a question posed by a user ***a*** with expertise level ***L(a)*** by using a "best preferred expert" rule, where the probability $P_H(u,a)$ of replying increases exponentially with the expertise level difference between the two users:

$$P_H(u,a) = \frac{\text{Exp}(L(u)-L(a))}{\sum_v \text{Exp}(L(v)-L(a))}$$

Note that according to this formula, even a user with a lower level of expertise than the asker has a small probability of answering the question, just as is the case in the actual Java Forum.

After setting up the model, we ran the simulation to generate networks. At each step, an asker was picked to ask a question and a helper was picked to answer based on the related probabilities.

After we ran the simulation for 5576 steps, we got a network with the same average degree as the Java Forum network. From scaled down versions, shown in Figure 8 and Figure 13, one can see that in this model, most of links are from low expertise (small nodes in the network visualization) to high expertise (big nodes).

Then, we analyzed the degree distribution of the simulated network to test whether it was similar to the Java Forum network. By comparing Figure 9 with Figure 4, one can see that while the indegree distribution replicates the heavy skew of the empirical network, the outdegree distribution does not. There are not as many single-post askers with low outdegree (0, 1, etc) in the simulated network. This is to be expected, since we are not modeling the growth dynamics where newcomers, by virtue of not being in community long enough to ask a large number of questions, contribute to the lower end of the distribution. When we updated our simulation to allow users to join the community with some probability at in each step, we were able to replicate the outdegree distribution (shown in Figure 10).

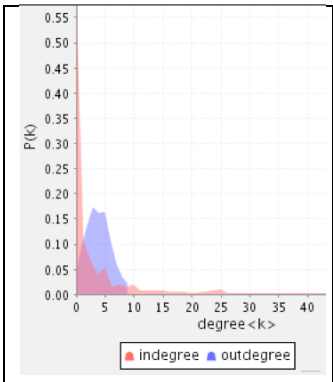


Figure 9: Simulated degree distributions with 'best preferred' helpers

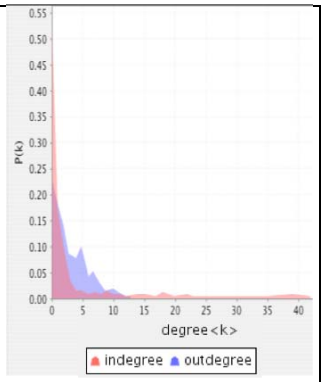


Figure 10: Simulated degree distributions with a growing network

We further looked at other characteristics of the network. Table 4 shows that the bow tie structure of the simulated network is similar to the Java Forum network. The only significant difference is that we have a relatively larger portion of disconnected users. This is because in the simulation, we built the network based on posting-replying patterns, but in the Java Forum, the lurkers (corresponding to disconnected nodes in our network) do not post in the community and therefore are not part of the empirical network.

Table 4: Bow tie structure of the 'best preferred' network

Core	In	Out	Tendrils	Tubes	Disc
13.8%	59.7%	3.6%	5.1%	1.0%	13.7%

Figure 11 shows that the indegree correlation profile fits rather closely with that of the Java Forum network. The correlation between asker and helper indegree is indistinguishable from 0 ($\rho = 0.009$, $p = 0.35$)

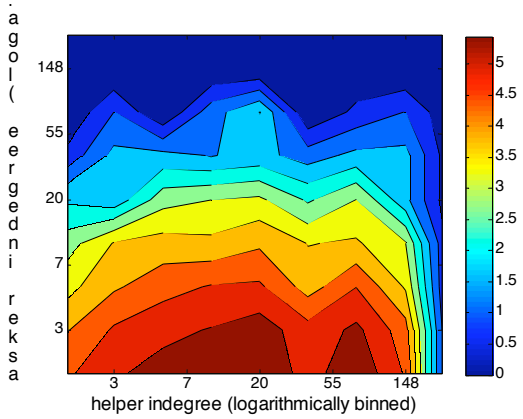


Figure 11: Degree correlation profile of the "best preferred" network

We tested various algorithms in this network and compared their ranks with the nodes' assigned rank in the simulation process. Figure 12 displays the result.

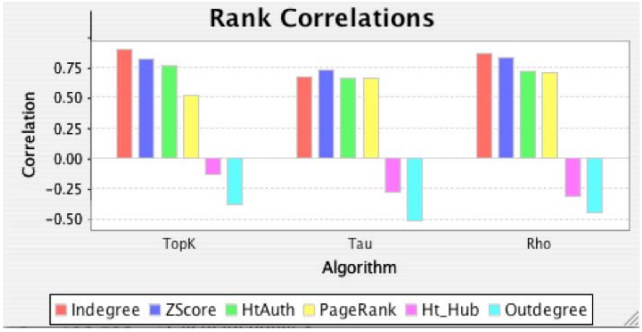


Figure 12: Performance of expertise-detection algorithms on the 'best preferred' network

From this figure, one can see that algorithms like ExpertiseRank and HITS do not perform better than simpler methods like indegree and z-score, much like what we found empirically in the Java community. This confirms our intuition that structural differences may be a major reason why complex algorithms like ExpertiseRank do not always work well in various network structures.

6.2 An Alternative Network Model

As we saw in the previous section, our simple model dynamics capture both the structural features and expertise ranking algorithm performance of the actual Java Forum. However, not all online expertise communities will follow the same dynamics as the Java Forum. We can glean useful insights by modeling different dynamics and then evaluating the expertise ranking algorithms on the models they create. For example, in other communities, especially ones that may be situated within an organization, experts may be under time constraints and choose to answer only those questions that make best use of their expertise. They would therefore be more likely to answer the questions of those slightly less expert than themselves. It may be the best way for people to make use of one another's time and expertise [2]. Such user behavior was not modeled in our "best preferred" model.

We thus constructed an alternate model, where users who have a slightly better level of expertise than the asker have a higher probability of answering the question, rather than those with a much larger difference in expertise. This model uses a "just better" rule, where a user ***u***'s probability of answering a question posed by user ***a*** is decided by the formula below:

$$P_H(u,a) = \frac{\mathrm{Exp}(L(a) - L(u))}{\sum_v \mathrm{Exp}(L(a) - L(v))} \quad \text{when } L(u) > L(a)$$

Figure 14 shows a network generated using this model. In contrast to the "best preferred network" shown in Figure 13, we can see that the links are not all pointing to the highest experts. Rather, questions are answered by users with higher, but not highest, expertise.

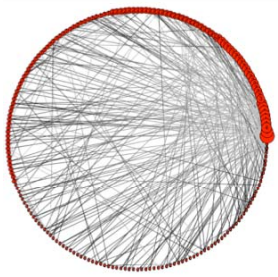


Figure 13: 'best preferred' network'

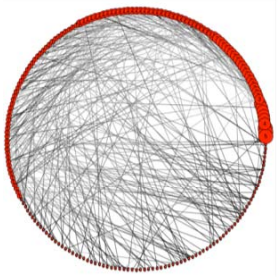


Figure 14: 'just better' network

Figure 15 shows the degree distribution of the network and Table 5 shows the bow tie structure analysis result. They are not very similar to Java Forum (note the very tiny Core in the bow tie structure), but some patterns are close (such as the highly skewed degree distribution and the biggest bow tie part being "In").

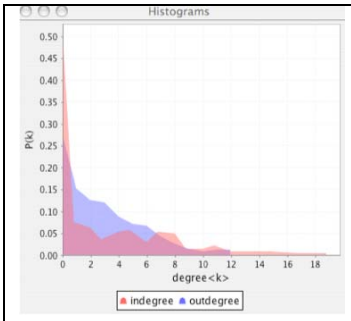


Figure 15: Simulated degree distributions with 'just better' helpers

Table 5: bowtie analysis of the 'just better' network

Core	1%
In	53.8%
Out	9.2%
Tendrils	9.5%
Tubes	17.4%
Disc	14.4%

Figure 16 shows the degree correlation profile, with an interesting appearance of strong correlation along the diagonal where users are helping those slightly less expert than themselves. At 0.14, the correlation coefficient is positive in contrast to the lack of correlation observed in both the empirical network and the "best preferred" model.

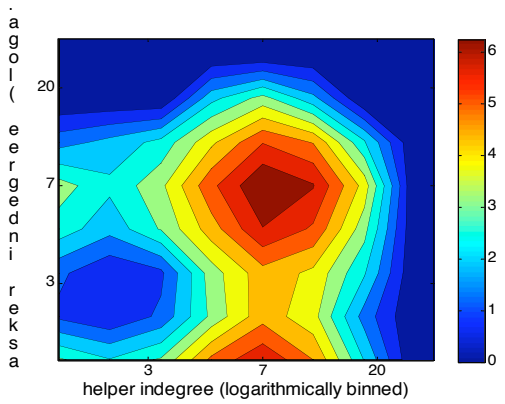


Figure 16: Correlation profile of the 'just-better network'

Figure 17 displays the performance comparisons of the various ranking algorithms in this new network: ExpertiseRank and Z_score perform the best, and HITS_authority is the worst. Since hubs and authorities reinforce one another in the iterative HITS algorithm, in the 'best preferred' network, the newbies who have their questions answered by the best experts reinforce the scores of those experts. However, in the 'just better' algorithm, the newbies who are asking the most questions are often helped by users with only slightly higher expertise. Therefore HITS identifies individuals with medium expertise as the highest experts. Similarly Figure 18 shows an example of a high expert user who is helping other expert users. Since experts have low HITS hub scores, they thus impart a low HITS authority score to the expert helping them. On the other hand, ExpertiseRank propagates the expertise score from the newbies to the intermediate users who answer their questions and from the intermediate users to the best experts. Thus we expect that PageRank-based algorithms such as ExpertiseRank will in general outperform other algorithms when the askers' and helpers' expertise is correlated. The Java Forum did not display this behavior (in fact, it is already very well described by our first model). But, as mentioned, such a scenario is plausible where users make the best use of their time by being more selective in choosing questions that are challenging to them yet they are still capable of answering.

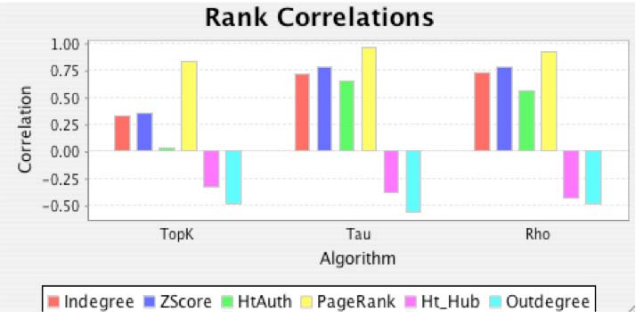


Figure 17: Performance of expertise ranking algorithms in the 'just better' network

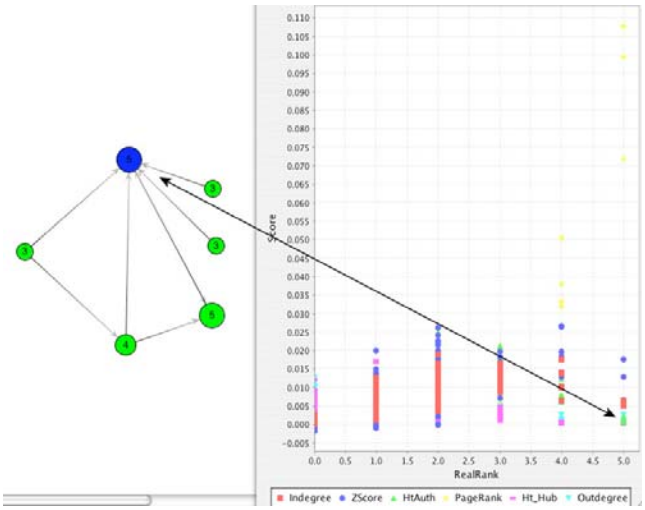


Figure 18: A case where a high expertise node has low authority

7. SUMMARY AND FUTURE WORK

In summary, we wanted to augment how people can help one another in online communities, particularly help-seeking or technical support communities. To do this, we wished to augment what we call the expertise network here – the way that expertise is distributed and deployed in practice.

To do this, we went through three steps. First, we wanted to know what went on socially in a typical help-seeking community. We analyzed the network representing asker-helper interactions in an online community, the Java Forum. Among them were highly skewed degree distributions, much like the graph of the World Wide Web. But unlike the Web, specific dynamics governing this particular forum produce a different bowtie structure and degree correlation profile.

We then ran an evaluation of expertise ranking algorithms – algorithms to analyze the relative expertise of different users – in this community.

To understand the results, we simulated these dynamics and produced networks that not only matched the observed aggregate network characteristics but also allowed us to understand why automated expertise ranking algorithms perform differently in differently structured networks. This understanding should help us weigh the tradeoffs in algorithm design and use for networks we encounter in the future. In fact, it is critical to do so.

In this work, then, we found:

- Structural information *can* be used for evaluating an expertise network in an online setting, and relative expertise can be automatically determined using social network-based algorithms. We also found, however, that the network's structural characteristics matter.
- These algorithms did nearly as well as human raters. However, there were significant tradeoffs among the algorithms. Sometimes a relatively simple measure was as good as more complex algorithms, such as an adaptation of PageRank.

- We believe, and have tested with simulations, that the structural characteristics of the online communities lead to differences in the performance of these algorithms.
- Indirectly, we also determined that simulation is a useful method for the analysis of expertise networks and expertise finding. We were able to tie the performance of the algorithm directly back to the dynamics of the communities. The simulations indicated under what structural conditions, or in what kind of networks, those algorithms will perform best. And we were able to do this without requiring interventions in real organizations, experimental conditions which we cannot obtain.

Work remains to be done. First, we would like to look at several other help-seeking communities (such as an intranet community) and compare it with our results and simulations. This would enable us to gain more insights about the tradeoffs in using these algorithms as well as in modeling online communities. Second, we will explore algorithms that combine content information (to differentiate specific knowledge) and structural information in order to develop more advanced online community based expertise finders.

8. ACKNOWLEDGMENTS

This work has been funded, in part, by the National Science Foundation (IRI-9702904). We also wish to thank Steve Morrow and Michael Duffy at Java Forum for their contribution, and Volker Wolf, Jodi Tyron, George Furnas, Michael Cohen, TJ Guili, Ye Du, Xiaomu Zhou, Kevin Nam, Jina Yoo, Ben Congleton, and the anonymous reviewers for feedback and suggestions.

9. REFERENCES

1. Barabasi, A.L. and Albert, R. Emergence of Scaling in Random Networks. *Science*, *286*, 509-512.
2. Ackerman, M.S. and McDonald, D.W. Answer Garden 2: merging organizational memory with collaborative help. In *Proceedings of CSCW '96*, Boston, MA, 1996, ACM Press, 97-105
3. Ackerman, M.S., Wulf, V. and Pipek, V. (eds.). *Sharing Expertise: Beyond Knowledge Management.* MIT Press, 2002.
4. Berkhin, P. A Survey on PageRank Computing. Internet Math. 2 (1), 2005, 73-120
5. Bollen, J., de Sompel, H., Smith, J. and Luce, R. Toward alternative metrics of journal impact: A comparison of download and citation data. I*nformation Processing & Management, 41* (6). 1419-1440.
6. Borodin, A., Roberts, G.O., Rosenthal, J.S. and Tsaparas, P. Link Analysis Ranking Algorithms Theory And Experiments. *ACM Transactions on Internet Technology*, *5* (1). 231-297.
7. Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A. and Wiener, J. Graph structure in the Web. *Computer Networks*, *33* (1-6). 309-320.
8. Campbell, C.S., Maglio, P.P., Cozzi, A. and Dom, B., Expertise identification using email communications. In *the twelfth international conference on Information and knowledge management*, New Orleans, LA, 2003, 528-231
9. Dom, B., Eiron, I., Cozzi, A. and Zhang, Y., Graph-based ranking algorithms for e-mail expertise analysis. In *DMKD*, New York, NY, 2003, ACM Press, 42-48.
10. Fagin, R., Kumar, R. and Sivakumar, D., Comparing top k lists. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, Baltimore, MA, 2003, Society for Industrial and Applied Mathematics, 28-36
11. Fisher, D., Smith, M. and Welser, H., You Are Who You Talk To. In *HICSS '06*, Hawaii, http://www.hicss.hawaii.edu/HICSS39/Best%20Papers/DM/03-03-08.pdf
12. Foner, L.N. Yenta: a multi-agent, referral-based matchmaking system. In *Proceedings of Agents '97*, ACM Press, Marina del Rey, CA, 1997, 301-307
13. Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J.T. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, *22* (1). 5-53
14. Kautz, H., Selman, B. and Shah, M. Referral Web: combining social networks and collaborative filtering. *Commun. ACM*, *40* (3). 63-65
15. Kleinberg, J.M. Hubs, authorities, and communities. *Acm Computing Surveys*, *31*. U21-U23
16. Kollock, P. The economies of online cooperation: gifts and public goods in cyberspace. In Smith, M.A. and Kollock, P. eds. *Communities in Cyberspace*, Routledge, London, 1999.
17. Krulwich, B. and Burkey, C., ContactFinder agent: answering bulletin board questions with referrals. In *the 13th National Conference on Artificial Intelligence*, Portland, OR, 1996, 10-15
18. Lakhani, K. and von Hippel, E. How open source software works: "free" user-to-user assistance. *Research Policy*, *32* (6), 923-943
19. Littlepage, G.E. and Mueller, A.L. Recognition and utilization of expertise in problem-solving groups: Expert characteristics and behavior. *Group Dynamics: Theory, Research, and Practice*, *1*. 324-328
20. Liu, X., Bollen, J., Nelson, M.L. and Sompel, H.V.D. Co-authorship networks in the digital library research community. *Information Processing and Management*, *41* (6). 1462-1480
21. Newman, M.E.J. The structure and function of complex networks. *Siam Review*, *45* (2). 167-256
22. Page, L., Brin, S., Motwani, R. and Winograd., T. The Pagerank Citation Ranking: Bringing Order to the Web, Stanford Digital Library Technologies Project, 1998
23. Sergei Maslov, K.S., Alexei Zaliznyak. Pattern Detection in Complex Networks: Correlation Profile of the Internet *eprint arXiv:cond-mat/0205379*, 2002
24. Streeter, L. and Lochbaum, K., Who Knows: A System Based on Automatic Representation of Semantic Structure. In *Proceedings of RIAO*, 1988, 380-388
25. Wasserman, S. and Faust, K. *Social Network Analysis: Methods and Applications.* Cambridge University Press, Cambridge, 1994
26. Zhang, J., Ackerman, M.S. and Adamic, L. CommunityNetSimulator: Using Simulation to Study Online Community Network Formation and Implications, In Proceedings of C&T '07, East Lansing, MI, 2007