Published in the Proceedings of the ACM 2001 International Conference on Supporting Group Work (GROUP'01), Boulder, CO, September 30 – October 3, 2001.

Evaluating Expertise Recommendations

David W. McDonald
FX Palo Alto Laboratory, Inc.
3400 Hillview Avenue, Bldg. 4
Palo Alto, CA 94304
mcdonald@pal.xerox.com

ABSTRACT

Finding a person who has the expertise to solve a specific problem is an important application of recommender systems to a difficult organizational problem. Prior systems have made attempts to implement solutions to this problem, but few systems have undergone systematic user evaluation. This work describes a systematic evaluation of the Expertise Recommender (ER), a system that recommends people who are likely to have expertise in a specific problem. ER and the organizational context for which it was designed are described to provide a basis for understanding this evaluation. Prior to conducting the evaluation, a baseline experiment showed that people are relatively good at judging coworkers' expertise when given an appropriate context. This finding provides a way to demonstrate the effectiveness of ER by comparing ER's performance to ratings by coworkers. The evaluation, the design, and results are described in detail. The results suggest that the participants agree with the recommendations made by ER, and that ER significantly outperforms other expertise recommender systems when compared using similar metrics.

Categories and Subject Descriptors

H.5.3 [**Information Interfaces and Presentation**]: Group and Organizational Interfaces – *computer-supported cooperative work, evaluation/methodology, organizational design.* H.4.1 [**Information Systems Applications**]: Office Automation – *groupware.*

General Terms

Design, Human Factors, Experimentation, Management.

Keywords

Expertise location, recommendation systems, user evaluation, Computer-Supported Cooperative Work, CSCW.

1. INTRODUCTION

Expertise is in demand. Workers need expertise to solve difficult day-to-day problems and organizations seek to manage it. Knowledge management and organizational memory systems have improved the management and access to important knowledge resources, but expertise is still elusive. The problem of identifying and recommending individuals who have expertise is one organizationally relevant application of recommender systems.

Commonly, recommender systems have been used to solve problems of information overload. In such situations a plethora of choice overwhelms an individuals' ability to choose [2, 10, 14, 22, 26]. Recommendation systems have been commercially employed to assist users in choosing web pages, books, movies, compact discs, and restaurants. In a sphere of public participation and where issues of taste are preeminent, recommender systems work.

Applying recommender systems to organizationally relevant tasks has been more problematic. Recommender systems have addressed some organizationally relevant problems such as recommending documents [7, 8] and finding people [5, 12, 16, 19, 25, 27]. Finding people who have the expertise to solve specific problems is important to many organizations. Despite this importance, few 'people finding' systems have been systematically evaluated.

This work describes a systematic evaluation of a portion of the Expertise Recommender (ER) [19]. This work makes three specific arguments with regard to the evaluation:

- When given a specific context, people are reasonably good at judging each others' level of expertise. This is established through an experiment that compares human performance on an objective test with estimated performance from all test participants.
- The heuristics implemented in ER are effective at identifying individuals who are likely to have expertise. This is based on an evaluation that isolates the performance of the heuristics by having participants rank order randomized lists that contain people recommended by ER.
- ER's level of performance is better than that of prior systems when using similar metrics for comparison. Few prior systems have provided any user evaluation data and there are no standard metrics for evaluating expertise recommender performance.

The paper concludes by summarizing these arguments and contrasting the prevalent approaches to expertise recommendation with the approach used in ER which blends factual aspects of an individuals' expertise with the social and contextual aspects of making a good recommendation.

The discussion first turns to a small study that establishes a baseline for human judgements of expertise.

2. A BASELINE FOR EXPERTISE JUDGEMENTS

The effectiveness of any expertise recommendation system is evaluated by users relative to some context. The most likely context is their own experience in the workplace. Users compare the recommendations they receive from a system with their experience and suggestions from their coworkers. But experience and recommendations from coworkers can be limited and wrong in some cases. Before evaluating ER, it is important to digress slightly to characterize the degree to which people are a good judge of each other's expertise. This was done by comparing performance on an objective test to that of the estimates of performance by all of the test participants.

The study was conducted at a medium sized software development company called Medical Software Company[1] (MSC). MSC has about 100 employees at its headquarters and approximately 60 additional employees who work in wholly owned subsidiaries and remote offices around the country. The study focused on technical development and technical support. These two departments are central to MSC's core business and comprise about half of the employees at headquarters. This study was not an isolated interaction with the company. MSC has been an on-going participant in this and other expertise field studies.

The study relied on a Knowledge Mapping Instrument (KMI). The KMI was originally designed as a lightweight method of cataloging expertise in the MSC workplace. The KMI is a short objective inventory of a persons' knowledge specific to MSC's processes and products. The questions on the KMI ranged from simple trivia to problems requiring a detailed understanding of MSC's products and processes (see, Lutters, Ackerman, Boster and McDonald [15] for a description of the KMI development).

The KMI was administered to 26 participants who volunteered for the study. The participants were provided lunch at a local eatery as an inducement to participate. In addition to the KMI, the participants were asked to provide a social score evaluation (SE score). Specifically, each participant was given a list of everyone else who would take the KMI and asked to "guess" the percentage score that the other participants would receive. Because of the sensitive nature of this type of evaluation, participants were told that they could "opt-out" of up to five evaluations. That is, a participant could skip guessing the scores of up to five others.

The actual scores on the KMI ranged from 86% to 32% with a fairly flat distribution in that range. The average SE scores ranged from 90.0% to 34.1%. The plot of KMI score versus SE score is shown in Figure 1.

The level of agreement between actual scores on the KMI and SE scores as estimated by other participants was exceedingly high (Pearson $r = 0.8823$). This correlation demonstrates that individuals at MSC can assess the expertise that other individuals have when given a relatively specific context, like the KMI. The plot shows very few outliers, with many of the estimations laying very near the regression line. The slope of this regression line is 0.95, suggesting that the population is as accurate at predicting low scores as it is at predicting high scores on the KMI. In general, the group tended to slightly overestimate the actual scores, but by 5% or less on average.

The motivating question, how accurately can people judge each others' level of expertise, was answered by considering the amount of error in each participants' estimations relative to the actual KMI score. This analysis directly engages the accuracy of the "expertise about expertise" judgement by each participant.

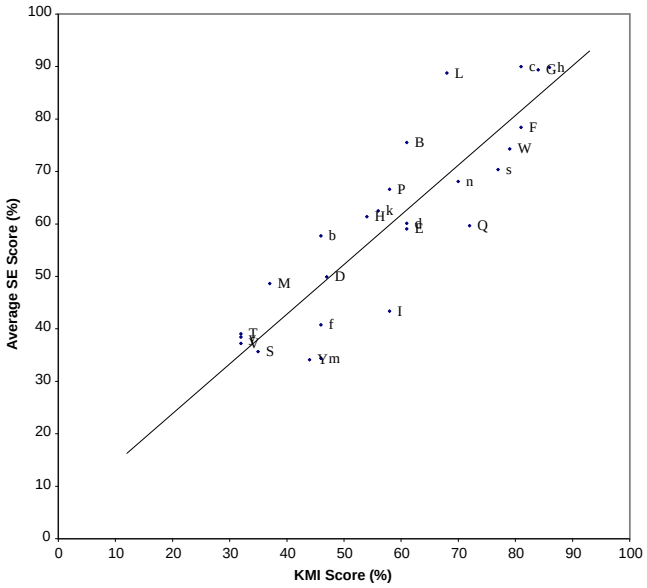

Figure 1. KMI score versus SE score.

The root-mean square (RMS) error was calculated for each participants' SE estimations and the actual scores of the other participants on the KMI. The RMS errors, as estimations of percentage scores, ranged from 12.35% to 37.98% with a mean of 20.89% and median of 20.59%. Figure 2 presents a histogram of the number of participants who scored within 2.0 unit intervals of RMS error. The figure shows a break near the median, with a small cluster of scores above and a larger cluster below.

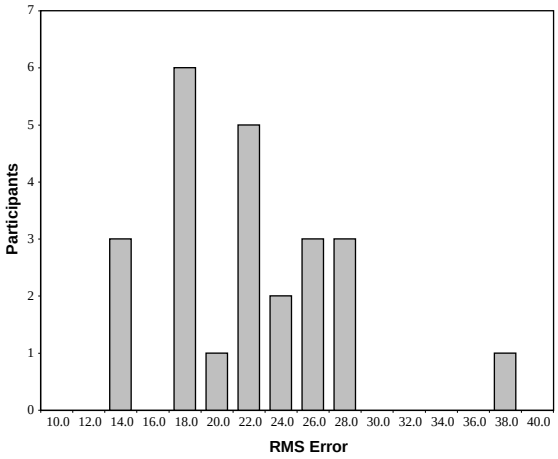

Figure 2. Histogram of RMS error in KMI estimation.

Interpreting these data requires understanding derived from prior fieldwork at MSC [18, 20]. There are nine individuals who perform above the median. Four of them are cited by their

[1] This is not the real name of the company.

peers as being particularly good at making referrals. These four people fill an organizational role that is a type of technological gatekeeper [1]. At MSC, gatekeeping is a specialized role. Individuals who filled the role were described as being an "expertise concierge" [20]. Three others who performed above the median include a technical support manager, a senior support representative, and a programmer; all of whom have eight or more years of service with MSC. The last two people, one from technical support and one from development, had a little over a year of service at the time of the KMI and SE administration. The individuals who fell below the median included a mixture of senior and junior staff members. There were no expertise concierges who performed below the median. A test to see whether there was any correlation between the number of months of service and either the score on the KMI or the SE evaluation, revealed no correlation.

The expertise concierges are an important group of participants because they are seen as being particularly good at making referrals when others need help. As well, they seem to effectively judge the other participants levels of expertise. The concierges have an RMS error of 14.87%, which is considerably better than both the mean and median of the regular population.

Summarizing briefly, these results provide two important starting points. First, the strong correlation between KMI and SE scores show that the people at MSC are relatively good at making judgements about each others' expertise. Second, the accuracy of the expertise concierges provides an important baseline for understanding how well an expertise recommendation system should perform. With this important baseline established and with evidence that judgements of expertise among the participants can be reasonably trusted, the discussion shifts to the organizational context of the recommender system and, in turn, the evaluation.

3. THE EXPERTISE RECOMMENDER (ER) CONTEXT

The Expertise Recommender (ER) was designed based on the results of an ethnographic study of MSC [20]. MSC builds, sells, and supports medical and dental practice management software. Practice management is considered the "back-office" or business side of medical and dental practice. Practice management software can include patient scheduling, billing, and validating insurance coverage for specific treatments. These functions are considered distinct and separate from the actual clinical treatment of a patient.

The original field study included the same groups used for the KMI-SE study. These groups are responsible for the design, implementation, and primary customer support for the majority of MSC's products. The study found that participants solve the problem of locating expertise through several behavioral and cognitive practices. These practices form a framework for expertise location that include expertise identification, expertise selection and escalation.

At a high level this expertise locating framework seems clear. During *expertise identification* participants rely on topic specific heuristics to identify other candidates who are likely to have the required expertise. After identifying some possible candidates, participants use *expertise selection* to pick one (or a small number) of candidates to ask for help. In expertise selection, participants use social and organizational norms and cues to guide their selection. Finally, *escalation* is the reiteration of expertise identification and expertise selection to repair breakdowns and account for unforeseen circumstance.

At a low level, the social and cognitive details of expertise location are actually quite messy. The heuristics that support identification and selection are not always clear and individuals are often incapable of describing why they are acting in a particular way. Two heuristics that guide the participants when looking for expertise were implemented in ER. The details of these heuristics will be covered later.

We now turn to a description of the ER system to tie together the expertise location framework, how the heuristics fit within the system and the way the evaluation design addresses the effectiveness of those heuristics.

3.1 The ER System

ER is a system for recommending individuals who are likely to have expertise in a specific problem area. A user garners a recommendation from ER by picking a relevant identification heuristic, selecting a matching technique, and entering a description or terms related to a problem. ER responds with a list of individuals who are likely to have expertise with the problem and who are a good social match for the person making the request.

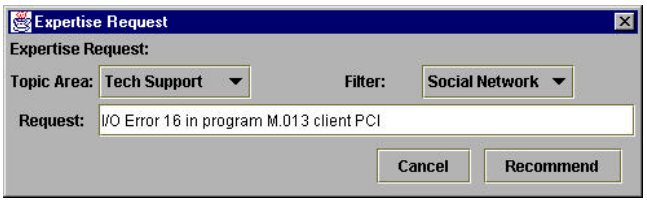

Figure 3. The new request dialog.

A user makes a recommendation request through the new request dialog in Figure 3. The user picks a "Topic Area", telling the system which expertise identification heuristic to apply. The user then picks a "Filter", which indicates the type of social matching the user would like the system to perform. Lastly, the user provides ER a brief description, some keywords or terms related to the problem.

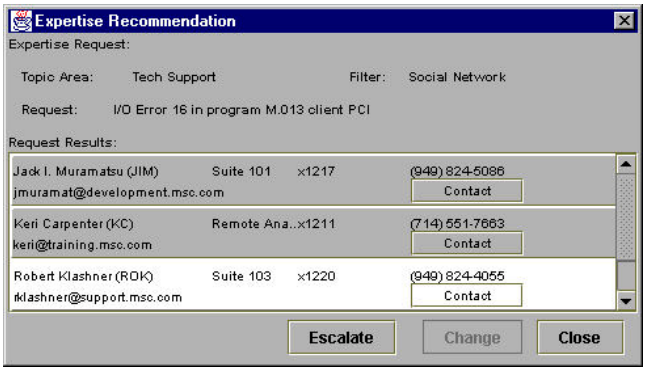

Figure 4. The recommendation response dialog.

ER generates recommendations and returns a list of potential people to contact in the recommendation response dialog in Figure 4. The list of recommended people is not strictly ordered based on the possible expertise that the individuals possess. Instead, ER relies on both the likely expertise and social aspects of the matching criteria to create and order the recommendation list.

ER currently implements two heuristics, one from technical development and one from technical support. These heuristics were chosen because they cover the majority of the day-to-day work activity of their respective workgroups.

3.1.1 Change History Heuristic

Change History is a heuristic designed to mimic an expertise identification behavior common to MSC's technical development group. Programmers call this behavior the "Line 10 Rule." On the surface the "Line 10 Rule" is a simple heuristic. Given a problem with a module a programmer looks into the version control system to see who last modified the code and then approaches that person for help.

The effectiveness of the "Line 10 Rule" is facilitated by the work practices at MSC. For each software change, a programmer checks the appropriate module out of the version control system, makes the change, test the change, and then checks the change back into the version control system. Each change is annotated with the module name, the module version, the programmer responsible for the change, the check-in date, and a short text description of the change. Since many changes are the result of a contractual obligation with a customer, an administrative assistant monitors the changes, which serves to reinforce the work practice.

Programmers at MSC do not specialize in any part of the system. In practice, however, there is some attempt to assign a programmer work in an area of the code where she has worked in the past. Yet, the size of the code base, the number of programmers, and programmer turnover will result in a programmer being assigned work in portions of the code where she has not worked before. Programmers use the "Line 10 Rule" in an attempt to overcome the problems that arise when working with unfamiliar code.

Programmers say that this rule works because the person who last made a change has the code "freshest" in mind. The version control system is not specifically designed to support this exact use. As a result, the rule is not strictly followed, but most programmers state that the heuristic is "good enough" to often get the help they need.

3.1.2 Tech Support Heuristic

The tech support identification heuristic is not known by a specific name within MSC. When faced with a difficult problem, a support rep will perform multiple, separate queries over the support database using the symptoms, customer, or program module involved. The rep then scans the records sequentially looking for similarities between the current problem and any past problems as returned by the different queries. In scanning records a support rep looks to identify people who have previously solved similar problems.

This heuristic is used with the technical support database. The support database mediates the majority of the work performed by technical support representatives. New problems ("calls") can be entered by a support rep or by customers via email. Incoming calls are automatically assigned to an appropriate tech support rep. The primary interface of the support database facilitates tracking active calls, modifying call status and establishing communications between a support rep and the customer.

The support database was not designed to facilitate the activity of the tech support heuristic. Each query (symptom, customer or program) must be completed separately. Finding similarities among the three primary characteristics is mostly done in the support representative's head.

The tech support identification heuristic consists of attaching the symptoms, customers, and program modules of solved problems to the person who solved them. As a simple rule of thumb this scanning behavior works fairly well. However, the process can be time consuming and is only profitable when applied to more difficult support problems.

3.1.3 Making Recommendations

ER does not require users to rate each other's expertise. ER incrementally mines work and work byproducts of an employees day-to-day activities. These can be very different from the careful presentation of self that employees display through their self-maintained ability inventories, resumes, white papers and publications. By using day-to-day work products and byproducts ER reflects the expertise-in-practice rather than simply topics of interest. As well, through an incremental approach, as the nature of work and the concomitant expertise changes, so does each persons' profile in ER.

Adding new people or new employees to ER is quite simple, they just go to work for MSC and perform their normal duties. Like at many organizations, few people who begin working for MSC are immediately marked as experts in the system and the organizational practice. As a new employee performs her day-to-day duties, her profile in ER grows and expands. As her profile grows and so does the likelihood that she will be identified as having some expertise.

ER mines the data sources that are relevant to each heuristic, extracting the appropriate features and associating those features with an individual to create a profile. In turn, the heuristics operate on the profiles to identify individuals who are likely to have expertise. MSC provided eight years of change history data (7316 changes) and four years of data from the technical support database (over 200,000 records) to populate ER. The heuristics that operate over these user profiles are the focus of this evaluation.

ER generates a recommendation in two distinct phases. In the first phase, expertise identification, ER relies on a user selected heuristic to generate a list of individuals who are possible recommendations. In the second stage, expertise selection, ER relies on matching heuristics to tailor recommendations to the individual who made the request. The ER implementation, the architecture, the heuristics are further described in [19].

This two stage approach to recommendation is a problem for any evaluation. It is not enough to simply evaluate the final recommendations that are returned from the system. Either the first or second phase could have serious problems that would result in inaccurate recommendations. The following evaluation considers the effectiveness of the two identification heuristics described above.

4. EVALUATING THE HEURISTICS

The current implementation of ER is designed to support MSC. ER implements expertise identification heuristics similar to those which MSC employees use on a day-to-day basis. Through their work and interactions MSC employees have detailed knowledge and expectations about each other, including who is likely to have expertise in a given problem area. The question which motivates this evaluation is:

- Do the identification heuristics, as implemented in ER, identify individuals who others expect to have expertise?

This evaluation compares each participants' judgement about who has expertise with the recommendations generated by the identification heuristics. This evaluation separates the expertise identification and expertise selection phases of ER's recommendation process.

4.1 Evaluation Design

The basic design compares how the two expertise identification heuristics perform in the context of the two workgroups from which the heuristics were derived. Scenarios provided a contextualized problem statement for which the participants were given a list of possible recommendations. Participants were asked to rank-order the list of possible recommendations. The design of the ranking instrument supports a comparison between the recommendations generated by the heuristics and participants' agreement with those recommendations.

The use of scenarios to prompt participants in an evaluation is not entirely uncommon. Scenario based design is often used to project the future use of a design rather than the evaluation of a current design [3, 28]. In contemporary views of system development (e.g. in Participatory Design and Computer-Supported Cooperative Work), design, implementation, and evaluation are just portions of an ongoing cycle. In this context, a methodology for developing or choosing scenarios appropriate for an evaluation is not prevalent in the prior literature.

The scenarios balanced several criteria. Scenarios were chosen to be representative of the type of work problems that occur with reasonable frequency. This differs depending on whether the problem is relevant to the technical support or technical development. The scenarios were then used to generate a characteristic expertise request that could be used with ER. Scenarios and the resulting request were chosen such that any individual request identified a minimum of three people and identified no more than 30 people. Scenarios that were too difficult or too rare, identify no one, or perhaps, only one person. Likewise, scenarios that were very common identify almost everyone. The prior fieldwork [20] guided the selection of appropriate scenarios.

4.2 Instrument

The ranking instrument consists of instructions on how to complete the ranking task and a set of questions. Each question is composed of a brief scenario (to provide context for a recommendation request), the topic area appropriate for the request, the request text, a list of possible recommendations (people), and a numbered list of seven blank spaces. For each question the participant transforms the list of possible recommendations into a rank-ordered list. The instructions ask the participant to order the list from "most likely to know" to "least likely to know" as best they can. The instrument contains a total of 12 questions, six questions appropriate to each identification heuristic.

The instrument was specifically designed as a type of cue based recall task. These tasks are easier and more approachable than a pure recall task. This is an important consideration in an organizational context. Based on the prior qualitative fieldwork [20], it is clear that only a small number of people are very good at a pure recall task with regard to expertise. As demonstrated by the results of the baseline experiment described at the outset of this paper, individuals who are good at this task are often expertise concierges. The cue based recall task in this design provides a challenge while allowing participation by many people at MSC.

For each scenario the top three ranked individuals as recommended by ER were designated "targets" for the scenario. The entire list of individuals recommended by ER and the population at MSC was then intersected to find a set of individuals who ER would never recommend for the specific scenario. Three people were chosen from this set as "distracters."

The three targets and three distracters where composed into a list and randomized. A blank space was added to the bottom of the target/distracter list, allowing a participant to include one extra individual if she felt that a particularly knowledgeable person was missing from the list. The target/distracter list thus contained seven items.

Allowing a participant to add a person who they feel is particularly knowledgeable, provides a check against a complete misidentification for a scenario. In a recommendation system like ER, based on live data and heuristics, it is possible that particularly knowledgeable people might never be recommended. People who have key expertise might never appear in the live data or the heuristics identified in the fieldwork may not apply to them. The blank space provides a mechanism to see whether there is any consensus among the participants about people who the system failed to identify.

4.3 Instrument Administration

Participants were asked to volunteer from technical development and technical support. The participant was given a brief overview of ER including a description of what it does and the topic domains of the two identification heuristics. The participant was told about the data sources used by the heuristics, the general types of queries appropriate to each heuristic, and the way the recommendations are presented. The participant was asked if she had any questions. If there were no questions then the participant was asked to do her best to complete the ranking instrument, ranking all people for the 12 questions. The participant was told that there was no time limit for the exercise.

Most participants thought the instrument was entertaining. The time commitment stated by the participants ranged from 5 minutes to 30 minutes. After completing the instrument, some participants expressed concern about the quality of some recommendations. They were concerned that some of the individuals listed in the recommendations were quite inaccurate. When participants indicated concern or some interest in the recommendations, they were carefully debriefed and asked not to share the extra information with other people at MSC. During the debriefing participants were informed of the ruse, that not all of the people in the supposed recommendation list were actually generated by the recommendation system. The participants were never told whether any individual was a target or a distracter.

The instrument responses were scored such that each target appearing in answer list positions 1, 2, or 3 and any distracter in positions 4, 5, or 6 were scored one point. If the participant added an additional knowledgeable person and ranked them amongst the targets and distracters, then the scoring scheme

was appropriately adjusted to account for the list position where the additional person was placed in the ranking. This resulted in a ranking score in the range zero to six for each question. With 6 points possible on each of 12 questions, the instrument maximum score is 72.

4.4 Results and Discussion

The hypothesis is that for each scenario and request the people recommended by the identification heuristic are "most likely to know" something about the problem relative to the other individuals in the list. In other words, the hypothesis is that the targets will be ranked higher in the list (assigned lower list numbers) than the distracters.

In total 23 individuals completed the ranking instrument. This was a little more than half of the two groups. One individual's data was discarded because the rankings were inconsistent and could not be scored in any reasonable way (e.g., the same target placed into two different rank positions). This resulted in 22 scores, 11 from technical development and 11 from technical support.

There was only one question where the participants consistently wrote in a person who ER missed. This person was consistently ranked as the first or second most knowledgeable person. Several participants were asked about why this person was ranked highest and they consistently responded that the ranking was based on the amount of work the person had performed in the topic for a recent release of MSC's software. Since the time period in question was more recent than the data that MSC provided for ER, this was not considered a misidentification. The person would have appeared and would have been identified given more recent change history data. The responses for this question were included in the analysis.

The sample mean was 51.82 or 72%. A t-test compared the sample mean to an expected mean that would result from random performance by the identification heuristics. In situations where there is no *a priori* comparison mean, comparison to a random expected mean is a reasonable starting point. The heuristics, overall, perform much better than random ($16.52 >> t_{0.01,21}$).

The 72% agreement and the significance of the t-test does not completely describe how well the participants agree with the heuristics. The t-test was solved to find a confidence interval; means (a high and a low) around which the sample mean would still be significant. These potential comparison means were then converted to a percentage score. The range, 68% to 75%, is the effective interval for the participants' agreement with the recommendations made by ER.

One way to characterize ER's performance is to compare the participants' level of agreement with each other to their level of agreement with ER. Ultimately, this is a very difficult comparison to make. Pearson r and a percentage agreement score are not the same measures. As well, the KMI/SE correlation is a function of the KMI, which is a more general measure of expertise than that which is presented in the scenarios of the ranking instrument. The KMI/SE correlation (Pearson $r = 0.88$) is very high for sociometric agreement data. This level of correlation is very near a ceiling value for this type of evaluation data. However, the participants' level of agreement with ER does not appear to be a ceiling. One conclusion is that the participants agree more with each other than they agree with the recommendations made by ER. Basically, ER performs well, but has room for improvement.

The RMS error calculation that compares the absolute error in SE scores to actual scores can be used as another comparison. The expertise concierges have an RMS error of 14.87%. If the concierges represent the best that humans can do at gauging one another's expertise, then an error rate of about 15% forms another type of baseline for considering ER's performance. Against this baseline, the aggregate evaluation shows that ER's overall error rate is about 28%. Again, based on this metric, ER has room for improvement.

The t-test comparing sample means of the programmers to the support representatives was not significant. The support reps and the programmers, as groups, equally agree with the identification heuristics. Another t-test compared the sample means between the individual heuristics. The participants agree more with Change History than with Tech Support ($4.29 > t_{0.01,21}$).

That the participants agree more with the results of the Change History heuristic ("Line 10 Rule") should be somewhat surprising for two reasons. First, the heuristic is probably the simplest heuristic found in the MSC workplace. It is encouraging that a workplace can have expertise identification heuristics that are very simple to implement and that perform reasonably well when compared to people's expectations of expertise. Secondly, in the community of software metrics, within software engineering, there is a strongly held belief that change history data has little or no value when trying to understand programmer performance [13, 23]. A portion of Grinter's [9] study of the use of configuration management tools and the results reported here should begin to refute this belief.

		Workgroup	
		Programmers	Support Reps.
Identification Heuristic	Change History	Mean: 28.91 S.D.: 2.88	Mean: 26.91 S.D.: 2.58
	Tech. Support	Mean: 23.64 S.D.: 3.32	Mean: 23.71 S.D.: 3.16

Figure 5. Participant means and standard deviations.

The agreement of each workgroup with each identification heuristic was compared. This analysis attempts to understand if the heuristics identify any workgroup specific expertise. The sample means and standard deviations for this evaluation are provided in Figure 5. Pair wise tests compared the sample means in each quadrant of the chart. The differences are significant at the $t_{0.05,10}$ level, with the exception that there is no significant difference in agreement between programmers and support rep workgroups using the tech support heuristic. Figure 6 shows a graph of the sample means relevant to each workgroup.

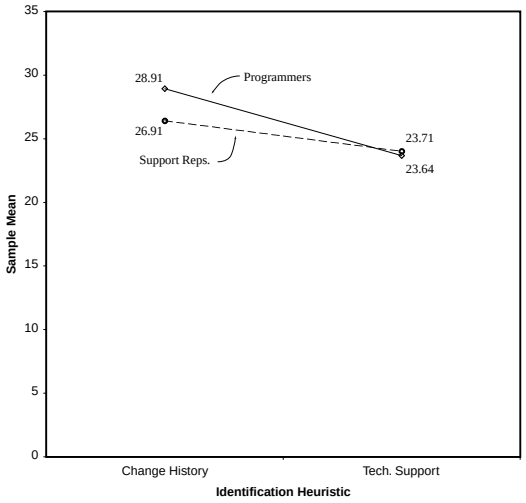

Figure 6. Workgroup versus identification heuristic.

An interaction would be an important finding for this evaluation. Consider this question: Who is most likely to be the best judge of another person's expertise? A reasonable answer is the individuals who know that person best. An argument that relies on social networks suggests that support representatives know one another best, and likewise, the programmers know each other best. Given that, the programmers are likely the best judge of who has expertise in programming and support representatives are likely the best judge of who has support expertise.

The data demonstrates that the programmers agree more with the Change History heuristic. So, if programmers are a better judge of who has expertise in programming, and they agree more with the results of ER's identification heuristic, then the heuristic is likely capturing some local aspect of the expertise-in-practice. The same argument can be made for the performance of the tech support identification heuristic, but the difference is so slight that it cannot be statistically supported. This suggests that the heuristics are identifying some workgroup specific expertise.

5. PRIOR WORK

Several prior systems support expertise location, through recommendations, matchmaking or some kind of mapping. The earliest of these systems is Who Knows [6, 24, 25]. Who Knows uses latent semantic indexing to generate profiles based on work products that are submitted by the individuals being profiled. A user could then perform an unstructured query of Who Knows and get a list of people whose profiles most closely matched the text space of the query. There was no systematic user study of Who Knows. Anecdotal results of its use at Bellcore suggest that the system was highly regarded by Bellcore management, but that profile maintenance was deemed overly burdensome.

Yenta [5] was not explicitly built as an expertise recommendation system, but rather as a matchmaking system. Yenta analyzes communication, email or perhaps news postings, and creates user profiles based on those communications. When queried, Yenta attempts to find other individuals who have profiles similar to the query. As an expertise recommender, Yenta confounds interest in a topic with expertise in a topic. People communicate about a great number of topics with which they have very little expertise. The subtle problem for Yenta is that a matchmaking system only needs to pair up people with similar interests. In matchmaking there is no attempt to warrant that one of the parties has any expertise. The discussion of Yenta does not include a systematic user evaluation.

Referral Web [11, 12] uses social networks as a type of relational map to assist a user with expertise location. Referral Web identifies relevant individuals by their participation in co-authoring relationships and presents users with a chain of relationships that need to be traversed from the person seeking expertise to the person who might have the desired knowledge. While this is not explicitly an expertise recommendation, this is an important part of locating expertise. Referral Web has not had a systematic user evaluation.

5.1 Prior User Evaluations

The systems described above have not had systematic evaluations. However there are two expertise recommender systems that have reported some evaluation data. These user evaluations provide another way to view the performance of ER and provide insight into other approaches to expertise recommendation. The two expertise recommendation systems described below were both, unfortunately, named Expert Finder.

5.1.1 Expert Finder$_1$

Vivacqua and Lieberman [27] report on a system that recommends individuals who are likely to have expertise in Java programming. Vivacqua and Lieberman's Expert Finder analyzes Java code and creates profiles based on a model of significant features in the Java programming language and class libraries. These features are defined by a domain model which was created to support the system. To avoid confusion in the following discussion we will call this system Expert Finder$_1$.

Vivacqua and Lieberman had 10 participants perform 20 queries each with their system. The queries where taken from a publicly available database and represent common problems that people have with the Java language. For each query the participant self-reports whether she knows the answer immediately, knows where to look for the answer or does not know the answer. This data and the recommendations from the system are then used to generate a one-in-three metric.

The one-in-three metric represents the number of times at least one of the top three recommended people self-reports that she knows the answer or knows where to look for the answer over the total number of trials. Vivacqua and Lieberman report a best case one-in-three metric of 85%. As the queries become "more specific" the metric drops to 71%. In the case of "more abstract" queries the metric was 75%. The methods used to classify the queries into the categories of "more specific" and "more abstract" were not disclosed.

ER performs substantially better than Expert Finder$_1$ when compared using Vivacqua and Lieberman's metric of choice. ER's best case is 99.6% using the one-in-three metric (based on 22 participants and 12 scenarios). To make this metric completely clear, there was only one instance in all trials where the recommendations from ER were not judged as finding at least one out of three people with the necessary expertise to solve a problem. Unlike Expert Finder$_1$, ER does not rely on self-report data. The ER evaluation uses aggregate data across a larger number of participants. Comparisons to the "more

specific" and "more abstract" levels of performance cannot be made without the method of classifying the queries.

Ultimately, the one-in-three metric is a poor metric for reporting effectiveness of expertise recommendations. As the performance of ER demonstrates, the one-in-three metric is entirely too easy. As in the case of ER and the next system, a comparison to human performance or the use of other established metrics is probably a better approach to measuring performance.

5.1.2 Expert Finder$_2$

A group at MITRE has also developed an expertise recommendation system called Expert Finder [16, 17] (Expert Finder$_2$ for this discussion). Expert Finder$_2$ generates its results by performing a query over a MITRE wide corporate database that includes 4500 MITRE employees. The entries in the database are maintained by each individual employee. After performing the query, Expert Finder$_2$ filters the results and presents a list of employees who are likely to have some expertise in the queried topic.

Expert Finder$_2$ is characterized by two metrics which are compared to a baseline of human performance. The researchers at MITRE had 10 participants (human resource managers) perform a ranked recall of the top five "experts" in the entire organization for five topic domains. They calculated the percentage with which the same named "expert" was listed by the participants in the same rank for the 1st, 2nd and 3rd ranks. These average agreements ranged from 63% for the first ranks down to 11% for the third rank.

Two measures, "precision" and "recall," are defined and calculated for Expert Finder$_2$. Precision, as defined in the MITRE study, is how many of the top five Expert Finder$_2$ recommendations were also identified as expert by the participants. The precision for Expert Finder$_2$ across all topics domains is 41%. The MITRE researchers define recall as how many of the top five experts pre-identified by the participants were included in the top five results of the Expert Finder$_2$. The average recall across all topics is 29%.

Comparing the precision of ER and Expert Finder$_2$ is problematic. The design details of the Expert Finder$_2$ study are not clear. Precision and recall statistics could have been generated by either a within participants or between participants design. Expert Finder$_2$ precision is based on the best out of five, where equivalent numbers for ER can only be based on the best out of three. Lastly, the average precision for Expert Finder$_2$ is based on five topic domains where ER only covers two domains.

An alternative way to compare ER and Expert Finder$_2$ is to consider the apparent error rates. Given the precision and recall numbers for Expert Finder$_2$, the system has an error rate between 59% and 71%. ER performs substantially better with an error rate of only 28% across all topics.

A number of things standout with regard to the MITRE study. It is unfortunate that it relied on judgements of only 10 participants. It is unlikely that all 10 participants actually know which of the 4500 employees are experts in the five topic domains. The possible exception to this would be people who fill a special organizational role known as the technological gatekeeper. Allen [1] and subsequent fieldwork in information seeking [4, 21] and expertise location [18, 20] have pointed out that individuals who fill technological gatekeeper roles often have highly connected social networks. However, the MITRE study does not make clear if the individuals picked as participants meet the criteria for being technological gatekeepers. The participants in the MITRE study are unlikely to agree on the corporate wide experts because they cannot know them all.

Sociometric data often has high variance, and the MITRE data demonstrates this fact. The variance in the data precludes finding any significance in the level of agreement among the participants. This makes it hard or impossible to compare Expert Finder$_2$ performance with human performance. However, comparing Expert Finder$_2$ to the results of the baseline study describe at the outset of this paper, Expert Finder$_2$ seems to be substantially below the accuracy of humans.

Lastly, there is a disconnect between the judgement of "expertise" and an expert's actual performance of expertise. The MITRE study fails to establish any link between the ability of the participants to judge expertise and the target recommendations actual abilities. In the Expert Finder$_1$ study, Vivacqua and Lieberman relied on self-report data as a (weak) link between actual expert performance and judgement of expertise. In this evaluation of ER, the link between expert performance and judgement of expertise was established through the use of the KMI, an objective independent instrument.

6. SUMMARY

The Expertise Recommender (ER) is a system designed to facilitate identifying individuals who have the necessary expertise to solve a specific problem. Heuristics, discovered through fieldwork, were implemented in a version of ER tailored to the organization involved in the study. These heuristics were systematically evaluated and their performance characterized by comparison to human performance.

Prior to evaluating ER's identification heuristics, this study found that the people at MSC have an uncanny ability to judge each others' level of expertise. Additionally, a small group of individuals, which included all of the expertise concierges found during the qualitative fieldwork, were found to be much more accurate than the majority of people when estimating others' level of expertise.

The evaluation of ER suggests that ER performs well when compared to human performance, but that there is room for improvement. Additionally, compared to other expertise recommendation systems ER shows superior performance when using metrics that are similar to the ones used to describe those other systems.

Most of the prior work in expertise recommendation promotes a false dichotomy. That is, the prior work is concerned with finding the expert or the small set of experts for the whole organization or a whole community. The prior work discounts the importance of local knowledge (context) and the inherently social aspects of expertise locating. In short, the same "expert" is not a perfect match for every seeker of expertise. Each recommendation must be tailored to the user who solicits the recommendation.

This evaluation did not highlight the social filtering aspects of ER. However, ER is designed to blend the characteristics of who would be the best factual recommendation (expertise identification) with who would be a best social or contextually appropriate recommendation (expertise selection). This

evaluation validates that the identification heuristics implemented in ER are accurate enough to begin either an adoption study or some other evaluation of ER's social and organizational matching techniques.

7. ACKNOWLEDGEMENTS

This project has been funded, in part, by grants from National Science Foundation (IRI-9702904), the UCI/NSF Industry/University Cooperative Research Center at the Center for Research on Information Technology and Organizations (CRITO), the University of California MICRO program and a University of California Regents' Dissertation Fellowship.

This work has benefited from conversations with Mark Ackerman, Jack Muramatsu, Kate Ehrlich, Saul Greenberg, Joe Konstan, John Riedl, Lynn Streeter, Loren Terveen, and Elizabeth Churchill. Wayne Lutters deserves special recognition for KMI development and data collection efforts. We thank the participants at MSC for their patience and insights over the last few years.

8. REFERENCES

[1] Allen, T.J. *Managing the Flow of Technology*. MIT Press, Cambridge, 1977.

[2] Balabanovic, M. and Shoham, Y. Fab: Content-Based, Collaborative Recommendation. *Communications of the ACM*, *40* (3). (1997). 66 - 72.

[3] Carroll, J.M. and Rosson, M.B. Getting Around the Task-Artifact Cycle: How to Make Claims and Design by Scenario. *ACM Transactions on Information Systems*, *10* (2). (1992). 181 - 212.

[4] Ehrlich, K. and Cash, D., Turning Information into Knowledge: Information Finding as a Collaborative Activity. in *Digital Libraries '94*, (College Station, TX, 1994), 119 - 125.

[5] Foner, L.N., Yenta: A Multi-Agent, Referral-Based Matchmaking System. in *First International Conference on Autonomous Agents (Agent'97)*, (Marina del Rey, CA, 1997), ACM Press.

[6] Furnas, G.W., Deerwester, S., Dumais, S.T., Landauer, T.K., Harshman, R.A., Streeter, L.A. and Lochbaum, K.E., Information Retrieval Using a Singular Value Decomposition Model of Latent Semantic Structure. in *Proceedings of the 11th International Conference on Research and Development in Information Retrieval (SIGIR '88)*, (1988), 465 - 480.

[7] Glance, N.S., Aregui, D. and Dardenne, M., Making Recommender Systems Work for Organizations. in *Practical Application of Intelligent Agents and Multi-Agents (PAAM'99)*, (London, UK, 1999).

[8] Goldberg, D., Nichols, D., Oki, B.M. and Terry, D. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, *35* (12). (1992). 61-70.

[9] Grinter, R.E. Supporting Articulation Work Using Software Configuration Management Systems. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, *5*. (1996). 447-465.

[10] Hill, W., Stead, L., Rosenstein, M. and Furnas, G., Recommending and Evaluating Choices in a Virtual Community of Use. in *CHI '95*, (Denver, CO, 1995), ACM Press, 194-201.

[11] Kautz, H.A., Selman, B. and Shah, M. The Hidden Web. *AI Magazine* (Summer). (1997). 27 - 36.

[12] Kautz, H.A., Selman, B. and Shah, M. Referral Web: Combining Social Networks and Collaborative Filtering. *Communications of the ACM*, *40* (3). (1997). 63 - 65.

[13] Kitchenham, B. *Software Metrics: Measurement for Software Process Improvement*. Blackwell Publishers, Inc., Cambridge, MA, 1996.

[14] Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R. and Riedel, J. GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, *40* (3). (1997). 77 - 87.

[15] Lutters, W.G., Ackerman, M.S., Boster, J.S. and McDonald, D.W., Mapping Knowledge Networks in Organizations: Creating a Knowledge Mapping Instrument. in *Americas Conference on Information Systems, AMCIS'00*, (Long Beach, CA, 2000), AIS, 2014-2018.

[16] Mattox, D., Maybury, M. and Morey, D. Enterprise Expert and Knowledge Discovery. The MITRE Corporation (McLean, VA), September, 2000. http://www.mitre.org/support/papers/tech_papers99_00/maybury_enterprise/maybury_enterprise.pdf

[17] Maybury, M., D'Amore, R. and House, D. Awareness of Organizational Expertise. The MITRE Corporation (MacLean, VA), October, 2000. http://www.mitre.org/support/papers/tech_papers99_00/maybury_awareness/maybury_awareness.pdf

[18] McDonald, D.W. Supporting Nuance in Groupware Design: Moving from Naturalistic Expertise Location to Expertise Recommendation. University of California, Irvine. Ph.D. Thesis, 2000.

[19] McDonald, D.W. and Ackerman, M.S., Expertise Recommender: A Flexible Recommendation System and Architecture. in *ACM 2000 Conference on Computer-Supported Cooperative Work (CSCW'00)*, (Philadelphia, PA, 2000), 231-240.

[20] McDonald, D.W. and Ackerman, M.S., Just Talk to Me: A Field Study of Expertise Location. in *CSCW'98*, (Seattle, WA, 1998), ACM Press, 315 - 324.

[21] Paepcke, A. Information Needs in Technical Work Settings and Their Implications for the Design of Computer Tools. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, *5*. (1996). 63 - 92.

[22] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J., GroupLens: An Open Architecture for Collaborative Filtering of Netnews. in *CSCW '94*, (Chapel Hill, NC, 1994), ACM Press, 175-186.

[23] Shepperd, M. *Foundations of Software Measurement*. Prentice Hall, London, 1995.

[24] Streeter, L.A. and Lochbaum, K.E., An Expert/Expert-Locating System Based on Automatic Representation of Semantic Structure. in *Fourth Conference on Artificial Intelligence Applications*, (San Diego, CA, 1988).

[25] Streeter, L.A. and Lochbaum, K.E., Who Knows: A System Based on Automatic Representation of Semantic Structure. in *RIAO '88*, (Cambridge, MA, 1988), 380 - 388.

[26] Terveen, L.G., Hill, W., Amento, B., McDonald, D. and Creter, J., Building Task-Specific Interfaces to High Volume Conversational Data. in *CHI'97*, (1997), ACM Press, 226 - 233.

[27] Vivacqua, A. and Lieberman, H., Agents to Assist in Finding Help. in *ACM Conference on Human Factors in Computing Systems (CHI 2000)*, (2000), 65-72.

[28] Young, R.M. and Barnard, P., The Use of Scenarios in Human-Computer Interaction Research: Turbocharging the Tortoise of Cumulative Science. in *Human Factors in Computing Systems and Graphics Interface, CHI+GI '87*, (Toronto, Canada, 1987), ACM Press, 291-296.