Enterprise Expert and Knowledge Discovery

Dave Mattox[1]*, Mark Maybury[2], Daryl Morey[2]
The MITRE Corporation
[2]202 Burlington Road
Bedford, MA 01730
[1]1820 Dolley Madison Blvd
McLean, VA 22102
{mattox, maybury, dmorey}@mitre.org

Abstract

In this paper we describe two systems designed to connect users to distributed, continuously changing experts and their knowledge. Using information retrieval, information extraction, and collaborative filtering techniques, these systems are able to enhance corporate knowledge management by overcoming traditional problems of knowledge acquisition and maintenance and associated (human and financial) costs. We describe the purpose of these two systems, how they work, and current deployment in a global corporate environment to enable end users to directly discover experts and their knowledge.

Keywords

User-expert interfaces, expertise modelling, collaborative filtering.

Introduction

We have all experienced having a burning question that could be answered if we could just find the right person to ask. *Expertise on-demand* has been a dream for many years which has become more closely attainable with the emergence of on-line computing, distributed collaboration, and user and domain modelling. We consider this problem in the context of MITRE, with 4,500 technical staff distributed between Washington, D.C., Bedford, MA, and dozens of sites worldwide supported by a sophisticated corporate intranet called the MITRE Information Infrastructure (MII).

This paper describes two systems in use on the MII that connect users with experts, exploiting the knowledge experts access, learn from, and create. Whereas traditional intelligent user interfaces rely on explicit representations and reasoning about user, discourse, and domain models (Maybury and Wahlster, 1998), in contrast the systems we describe exploit the by-products of expert activity (e.g., web browsing, publishing) using relatively simple algorithms that connect users with experts or the knowledge they create. We first describe the ExpertFinder system which mines corporate databases to build models of experts which users can then search on demand. We present a preliminary evaluation of its performance and informally contrast this with chance and the performance of human resource managers. We next describe a collaborative knowledge filtering system, KEAN, for resource discovery, which builds aggregate models of user's assessments of web pages. We conclude by underscoring lessons learned in corporate use of these tools, and outline evaluation metrics of an ongoing knowledge sharing experiment.

ExpertFinder

Distribution of staff, decreasing project size, and cost/time pressure are driving a need to leverage enterprise expertise and quickly discover who knows what and to quickly find and form expert teams. Those in need typically have little or no means of finding experts other than by recommendation. Unfortunately, busy experts do not have time to maintain adequate descriptions of their continuously changing specialized skills. Past experience with "skills" databases at MITRE indicate they are difficult to maintain and quickly outdated.

MITRE's ExpertFinder (Mattox, Smith and Seligman 1998) fills this gap by mining information and activities on the MII related to experts and providing this in an intuitive fashion to end users. Figure 1 illustrates the system in action. In this example, a user is trying to find chemical experts in the corporation. When the user searchers using the term "chemical", the system ranks employees by the number of mentions of a term or phrase and its statistical association with the employee name either in corporate communications (e.g. newsletters) or based what they have published in their resume or document folder (a shared, indexed information space). Integrated with our corporate employee database, employees are ranked by frequency of mentions, pointing to sources in which they appear.

Figure 1. ExpertFinder "chemical" example

ExpertFinder exploits the MII and thus avoids maintaining information internally. By doing so, ExpertFinder operates in real-time, using the most recent information available to locate experts. The MII contains many different sources of information that can be used to locate relevant expertise. Staff members can easily and quickly (and are encouraged to) publish documents in individual staff "document" folders on the MII. These include technical papers, presentations, resumes and home pages. Also, information is published about MITRE employees in project descriptions, announcements, and (internal and external) newsletters. At MITRE, all of these documents are indexed by a common text search engine.

ExpertFinder works by linking documents found through queries to the search engine with MITRE employees. In and of itself, each source of information mentioned above is not sufficient to determine if an employee is an "expert" in a particular topic. ExpertFinder relies on the combination of evidence from many sources. The ExpertFinder system considers someone an expert in a particular topic if they are linked to a wide range of documents and/or a large number of documents about that topic.

ExpertFinder works by taking a keyword phrase (e.g. "chemical weapons") and passing it to the underlying search engine which then returns a set of documents as a set of hyperlink pointers. As described above, the documents used by ExpertFinder fall into two main categories, documents about a topic which are published by an employee and documents which mention employees in conjunction with a particular topic. For the self-published case ExpertFinder relies on the number of documents published by an employee about a given topic to provide an "expert score" for that employee. The only exception is that of an employee's resume which is given additional weight.

Managing the second class of documents, those which mention employees and topics, is more complicated. Whereas with the self-published documents there is an explicit linkage between the documents and employee (they are indexed by employee number), with documents that mention employees, this linkage must be derived from the underlying text. The first step after the documents have been returned from the search engine is to locate the proper names within the text. This is done using a commercial product that tags names within a text document[1]. Once the names have been located the next step is to associate them with the query topic. All the documents returned by the search engine contain the query string somewhere but there are several distinct types of documents and each type has a structure that must be exploited differently. For example, MITRE publishes an internal newsletter which contains short paragraphs describing accomplishments by MITRE staff e.g. 'Dr. John Smith presented a paper titled "Intelligent Agents and Data Management" at the Tenth International Con-

---

[1] NameTag from IsoQuest Corporation.

ference on Autonomous Agents.' In this case ExpertFinder uses proximity as a metric for expertise (i.e. the name "John Smith" is five words away from the keyword phrase "Intelligent Agents"). In addition, since the document has been formatted in HTML, the HTML tags can be used to further refine the score. In this case a paragraph marker (<p>) breaks the linkage between a name and a topic since the announcements are always contained within a single paragraph. However, the heuristics used for the newsletter don't work well with other documents. For example, in technical reports the authors names appear at the top of the document and may be several paragraphs away from the relevant keywords and therefor require different heuristics to determine the linkage between names and keywords.

Once each document has been examined the evidence gathered about each person found is combined into a single score for that person. The person names are then matched against a database of MITRE employees, ordered by their final score and displayed. This means that the people that are most likely to be experts are displayed at the top of the list (see example in Figure 1). The user can then find detailed information on each expert including the source documents that were used to locate the expert.

Performance

Overall, the results obtained by the ExpertFinder system are quite good. The original goal was to place a user within one phone call of an expert. That is, even if the persons listed as the result of an ExpertFinder query weren't the experts, they would be able to provide the name of someone who was. However, in the majority of the cases tested, reasonable candidates for the title "expert" are listed as the top three or four candidates, where the likelihood of randomly selecting a correct expert is the total number of experts divided by 4600 total staff, often significantly less than a one percent chance of getting any right. Table 1 illustrates preliminary results contrasting the performance of ten technical human resource managers, professionals at finding experts, with ExpertFinder for the task of identifying the top five corporate experts in speciality areas listed in the table. The first column in Table 1 shows the degree of inter-subject variability in reporting experts (measuring percentage of agreement of first, second, and third of five experts). Columns 2 and 3 show results for *precision*, the degree to which a staff found by ExpertFinder is considered expert by humans, and *recall*, the degree to which apriori human-designated experts are found by ExpertFinder. In Table 1, we use harsh measures where precision measures how many of the top five expert finder results were also identified as expert by humans. In contrast, recall measures how many of top five experts that humans identified were included in the top five ExpertFinder results. In spite of human variability (e.g., note the difficulty humans had in identifying chemical and network security experts), ExpertFinder works remarkably well except in network security and collaboration (ironically a result of few expert collaboration staff publishing on the MII perhaps because they use specialized collaboration environments which were not instrumented) .

Table 1. Human and ExpertFinder Performance

Expert Area	Human Agreement (1^{st}, 2^{nd}, 3^{rd})	ExpFinder Precision	ExpFinder Recall
Data mining	70%, 49%, 24.5%	60%	40%
Chemical	40%, 8%, 0.8%	60%	40%
HCI	90%, 36%, 11%	60%	40%
Network Security	50%, 10%, 0.4%	20%	20%
Collaboration	70%, 35%, 17.5%	5%	5%
AVERAGE	**63%, 28%, 11%**	**41%**	**29%**

Problems and Anomalies

The speed at which ExpertFinder operates requires improvement. The average response time for the above 5 queries was approximately 30 seconds. The majority of the time is spent in the name tagging process. More significant, the use of the pre-existing MII is a double-edged sword. It makes it easy to maintain since ExpertFinder doesn't actually maintain any information. However it also means that ExpertFinder is totally dependant on outside entities to maintain the necessary information. In the case of the MII it is the employees themselves that maintain a majority of the information though self-publication. Of course, when some person doesn't publish, they don't show up as experts. Another problem is when people publish documents that they don't create. For example, the secretary for the MITRE vice-president in charge of the research program regularly publishes the research overview presentations given by the VP. Because these presentations contain a wide variety of technical information, the secretary is routinely listed as an expert in a great number of areas. This problem generalises to one of "buzzword pollution". People that make liberal use of technical jargon can show up as experts even when they are not.

Future Work

There are several improvements that are planned for ExpertFinder. The major one is the addition of a Bayesian network to manage the evidence combination from the various sources. The current method of evidence combination is fairly arbitrary. The addition of a Bayesian net would allow the system to learn over time by using the results of each query to adjust the weights of each term. For example, we assume that someone can only really be an expert in one or two subjects. So the probability that someone is an expert is inversely proportional to the number of times they show up for distinct queries. So as ExpertFinder operates over time it would adjust the prior probabilities in the network based on empirical evidence. Other improvements include adding additional sources of information, incorporating more standard information retrieval techniques such as query expansion and Boolean operations and adding the capability to define standing queries so that a user can be notified when new experts appear.

Knowledge Evaluation and Annotation (KEAN)

Whereas finding and connecting to experts is an important function, it is equally important to benefit from their knowledge and to enable experts to benefit from one another even when you cannot or choose not to connect users to them (e.g., because they are a scarce resource). Knowledge Exchange and Annotation (KEAN) is a recommender system that has been deployed at The MITRE Corporation to enable experts to benefit from one another's knowledge. For example, just as we were able to find an expert on chemicals using ExpertFinder in Figure 1, let's say this expert is called away on a high priority task. They might point us to another expert or to their work. KEAN enables us to do this automatically. Let's say our chemical expert has suggested we look at what knowledge another expert, Stanley Boykin, has captured with respect to chemicals. We bring up KEAN as shown in Figure 2a and search on "boykin" and "chemical" and obtain the results as shown in Figure 2b. KEAN acts like a window onto the web for each source, listing individual assessments (ranking from 1-10), classification of content into pre-specified categories, and associated textual comment. We quickly scan these sites and discover University of Sheffield's ChemDex (http://www.shef.ac.uk/~chem/chemdex/), a web site that catalogues academic, industrial, and government organizations involved in research or manufacture of chemicals.

How KEAN Works

KEAN operates on top of a flexible metadata architecture that allows generic metadata types on URLs to be stored, transmitted, and filtered. KEAN creates a notion of an instance for each knowledge collection. Each instance has its own set of value added metadata that is determined by the instance creator. In the instance that is described above that is being used internally at MITRE, the metadata types used are classification, keyword, date, utility rating, submitter, and textual annotation. There are also derived metadata types that are used by the MITRE system and supported by KEAN. One derived attribute used by the MITRE system is the average rating that has been compiled on a URL. Content, in the form of metadata and URLs, can be contributed by anyone to the collection. To guarantee content and bootstrap the collection, however, knowledge stewards (experts) are defined in each of the classification categories. The knowledge stewards are under agreement to contribute a minimum amount of content to the collection each month, or a replacement steward is assigned that role.

The focus of the KEAN tool is to give the user access to the high utility information that an expert would normally be able to provide through pointers and advice. The metadata filters allow the user to drill down to high-utility information in the manner most effective the user. For example, by filtering the metadata in the above instance, KEAN can handle all of the following types of queries:

- What information does Chris (expert in data mining) think is useful for data mining?
- What information do people in the data mining community of practice find useful on data mining?
- What information does everyone think is useful on data mining in the past few weeks?
- What information on data mining have I found to be useful in the past?

In addition, an important value added of KEAN through its support of textual annotations is that you can not only *find* information, but you can see *why* that information was useful to each individual.

The current version of KEAN uses an Oracle RDBMS to store the metadata. PL/SQL and Oracle Web Server generate the web pages. Java is used for the search and submit interfaces and Javascript glues the HTML frame based interface together.

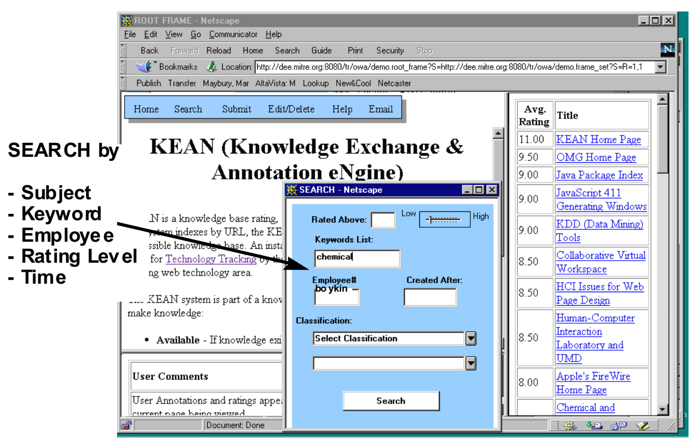

Figure 2a. KEAN Search

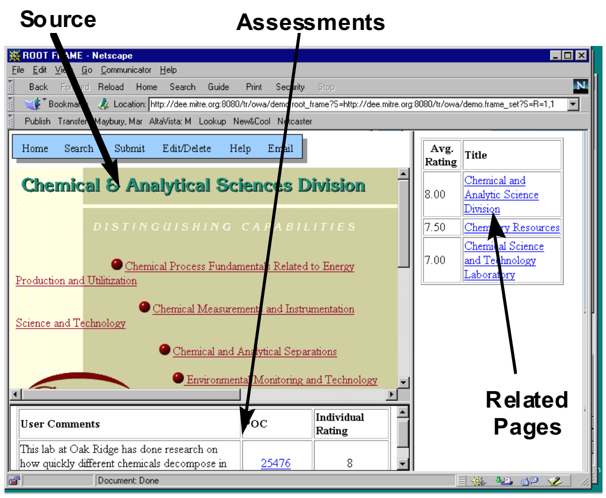

Figure 2b. Source, Assessment, Related Pages

Lessons Learned and Future Plans

Through the use of the KEAN system, we have found that people value having utility ratings available but most of the time do not want to take the time to contribute a rating. This phenomenon is well known in the user interface community and has been described by Grudin [6] among others. To tackle this problem, we are exploring implicit methods of discovering utility. Following up on work in [7], we tested 26 individuals on 295 URLs visited to see how the length of time reading a document on the web correlated to their explicit utility rating.

Each individual in the experiment was given a focused web-based task where they had to answer questions posed by experts in the area of directory services and find the answers on the web. The questions were, “Which standards organization defines the X.500 specification?”, “How does LDAP differ from X.500?”, and “Name some of the data types that can be stored in an LDAP attribute.” The individuals in the experiment did not know that the length of time reading each URL on the web was important to the experiment. They only thought they were being timed for reasons of recording how long it took them to get a response on each question. After the experiment, each person was taken back through each of their URLs and asked to give an explicit utility from 1-10 where 10 being the highest.

A preliminary regression test for a linear relationship on the data found that there was a positive correlation (explicit utility = .0113*time read) between time read and explicit utility. The R^2 was rather low, however, with only 13.5% of the explicit utility explained by the time read and the rest error. Focusing on the intended goal of finding a threshold for time read that would indicate high explicit utility, we then searched for association rules in the data. By defining high utility as the explicit utility between 6-10, we found that 66% of all URLs read for greater than 78 seconds were defined as high utility by the user. From this and previous findings [7], it appears that length of time reading a document can be used as a measure of utility. Future versions of KEAN will incorporate the ability to add time utility metadata.

We are also in the process of moving KEAN to a distributed object architecture that will allow much greater flexibility in how information can be filtered, accessed, processed and viewed. JavaBeans components will be used for the GUI, logic and interface layers. The metadata, which will include information object relationship metadata will be stored in an object database. With this new architecture, if you don't like how KEAN displays information or orders the information displayed, you can build a custom view by simply tweaking a few dis

play parameters in a component or by building your own custom component that operates on the same data. Instead of a single place for gathering information and metadata, the architecture will support multiple components in higher leverage locations, such as one already built that plugs into Microsoft Word, PowerPoint, and Excel (See figure 3).

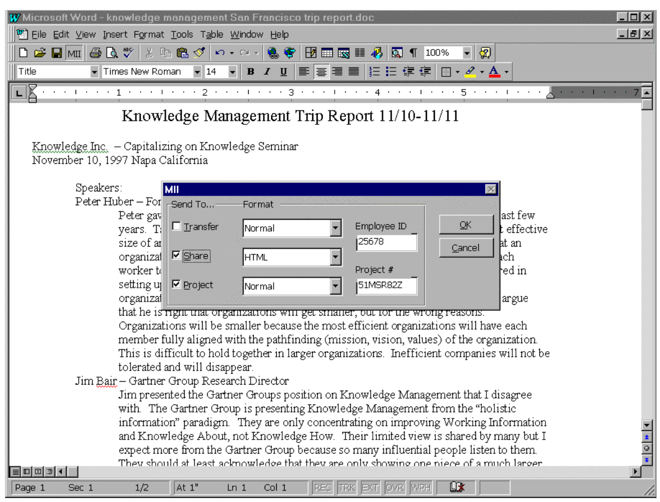

Figure 3. Info. sharing & metadata gathering Word plug in.

The Word plug in component makes it very easy to share information and also provides a reminder to share by popping up on file save operations.

Evaluation

We are currently evaluating the application of knowledge recommendation systems as well as methods to measure and subsequently reward knowledge sharing behaviour. We are tracking knowledge dissemination in staff through traditional means such as publications, citations, volume and frequency of publishing and access of content on each employee's home page and "publish" and "transfer" folder. We have begun to measure the following quantitative elements such as the volume and frequency of:

- documents published by staff on the corporate intranet
- documents accessed by staff on the corporate intranet

We also will measure the volume and distribution of new consulting engagements as a measure of how employees are spending their time (e.g., working on native projects or sharing their time and expertise across the corporation). In addition to these quantitative measures, we intend to measure qualitative measures such as:

- actions taken as a consequence of a staff's knowledge sharing
- informal feedback of tacit knowledge sharing via a "thank you" email address

Summary

Effectively implemented and deployed, intelligent user-expert interfaces promise several benefits. These include:

- More *efficient* interaction -- enabling users to more rapidly find experts with no more effort or time than they would in an electronic yellow pages.
- More *effective* interaction – providing better support by ensuring more valid recommendations which are tailored to user needs, based on up to date information.

Our experience in a corporate setting is yielding important lessons regarding how to take advantage of both implicit technical instrumentation as well as social behavior to enhance organizational agility and efficiency.

Acknowledgements

We are indebted to David House for his design and implementation of ExpertFinder. We would also like to thank Inderjeet Mani, whose research efforts supported the development of this application and to Chris Elsaesser for the original idea for an expert finder. We thank Tim Frangioso for KEAN usage studies.

References

1. Maybury, M. and Wahlster, W. (eds. 1998). *Readings in Intelligent User Interfaces*. Morgan Kaufmann: Menlo Park, CA.

2. Mattox, D., Smith, K., and Seligman, L. (1998). Software Agents for Data Management. In Thuraisingham, B. *Handbook of Data Management*, CRC Press: New York. 703-722.
3. Morey, D. 1997. Enterprise Knowledge Management. In Thuraisingham, B. (ed.) *Data Management Systems*, CRC Press: New York, 723-729.
4. Grudin, J. (1994), Groupware and social dynamics: eight challenges for developers, *Communications of the ACM*, 37(1), 92-105.
5. Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R. and Riedl, J. (1997), Applying collaborative filtering to usenet news, *Communications of the ACM*, 40(3), 77-87.
6. Fenn, J. 1999. Skill Mining: An Emerging KM Technology. Garnter Group Report.