An Expertise Recommender using Web Mining

Purnima Chandrasekaran, Anupam Joshi, Michelle Shu Yang, Ramya Ramakrishnan

Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
1000 Hilltop Circle
Baltimore, Maryland 21250
[pchand1, joshi, syang3, rrmaka1]@cs.umbc.edu

Abstract

In this paper we explore techniques to mine web pages of scientists to extract information regarding their expertise, build expertise chains and referral webs, and semi automatically combine this information with directory information services to create a recommender system (Resnick and Varian 1997) that permits query by expertise. We experimented with some existing techniques that have been reported in research literature in recent past (including our own prior work), and adapted them as needed. We developed software tools to capture and use this information.

Introduction

Large organizations often need to build systems that gather information about the expertise of employees from existing documents and use it to recommend specialists in targeted areas. Present methods used by organizations, mostly in annual "directory update" exercises are essentially static. Individuals describe their capabilities using some "keywords" that are matched in response to a query. For a large organization, the exercise of gathering this information is fairly massive and time consuming. Moreover, once entered, the keywords remain unchanged until explicitly altered. More importantly, keywords are often too restrictive to capture the expertise of the individual, and people end up mentioning their primary expertise. Narrative descriptions of research in web pages or titles/abstracts of papers often provide a far richer description of a person's overall expertise, both primary and others. We present a method using text-mining techniques to semi-automatically extract expertise-related information from individual web pages. This not only saves human effort, but also allows automatic updates via web spiders that re-mine a page when it changes beyond a certain degree.

Studies (Kraut, Galegher, and Edigo 1990) have shown that in most organizations, the informal network of colleagues is one of the most important sources of disseminating expertise related information. In other words, when looking for a person with a particular expertise, people tend to ask friends and colleagues, who in turn might refer them to other colleagues and so on, thus building a *referral chain* (Kautz, Selman, and Shah 1995). Kautz et al. (Kautz, Selman, and Shah 1995) have shown how such "social networks" can be built by analyzing web pages for author lists or mention of particular names, or email headers for sender and recipient fields. Effectively, this is looking for structured data patterns in the web pages. We postulate that some of the newer research in text mining can be used to extract similar information from unstructured information (research descriptions etc) in the web pages as well.

Related to the expertise chains is the notion of collaborative filtering (Kautz, Selman, and Shah 1995). In collaborative filtering, people rate "information", and these ratings are used by the system to "recommend" appropriate information to other users. For example, our W3IQ system (Kavasseri et al. 1996; Joshi, Weerawarana, and Houstis 1997; Joshi, Punyapu, and Karnam 1998) uses ratings provided by users to present appropriate URLs in response to queries by other users who share a similar interest. We have also shown (Joshi and Krishnapuram 1998) how people's information access behavior can be mined to form groups of people with similar interests. In the present work, however, we pursued other techniques in a intrusive way.

We explored classic text mining mechanisms to identify keywords from a document corpus, such as Term Frequency Inverse Document Frequency (TFIDF) (Frakes and Baeza-Yates 1992) that we show to be ineffective for this system. However, by intelligently using domain constraints, we could improve the performance of such methods by exploring the semi-structured nature of the employee web pages and their publications mentioned therein.

We implemented this system for the NASA Goddard Space Flight Center (NASA GSFC). Issues studied involve scalability, complexity of the system, ease of maintenance, efficiency in query execution and linkage sought with their existing X.500 infrastructure etc.

This work is supported in part by the NASA Goddard Space Flight Center, Greenbelt, Maryland and an NSF award No: 9801711.

System Implementation

Approach

The existing approaches to obtain a person's expertise use static data from large databases that are updated quite infrequently. Moreover this method is very restrictive as an individual describes his expertise in terms of a few keywords that are matched to return a response to a query.

Our approach to the problem is to mine narrative text, web pages, departmental reports, progress reports, etc., which are rich sources of expertise related information. This method also allows for automatically updating expertise information via web spiders that re-mine pages that have changed beyond a certain degree. This system architecture is further discussed in the following subsection.

System Architecture

We have developed a system, called Expertise Recommender, that mines web pages of scientists at NASA and extracts information regarding their expertise, builds expertise chains and referral webs, and links this information with the existing X.500 directory services to create a recommender system that permits query by expertise. Our Expertise Recommender consists of the following five main components as shown in fig. 1.

1. **Web Crawler:** A web crawler crawls the specified domain (in our case, gsfc.nasa.gov domain) and downloads text and HTML web pages referred to from this domain and sub-domains recursively to a local disk. This forms the document corpus used as input to the *Expertise Extractor.*
2. **Expertise Extractor:** The Expertise Extractor is a set of programs and scripts that identify documents of interest from the document corpus and extracts sections that potentially describe employee expertise. It then generates the keywords from these sections using various techniques that we describe in the *Implementation Details* section.
3. **Referral Chain Builder:** This builds referral chains using information contained in the publications section of the documents.
4. **Knowledge Base:** The knowledge base maintains the expertise keywords extracted by the Expertise Extractor and also the referral chains built by the Referral Chain Builder. This is hosted by an mSQL server and may be queried from the web interface of the Expertise Recommender system.
5. **Web Interface:** A web server interfaces between the user and the knowledge base – providing an HTML forms based query interface and displaying the results. An interface to the NASA X.500 service is also provided.

We discuss the implementation of these components in detail in the next section.

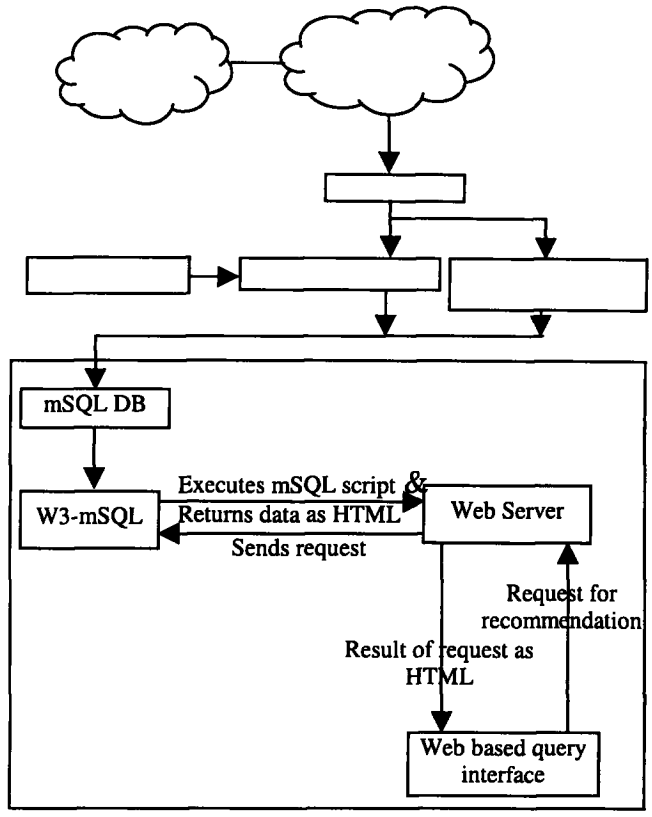

Figure 1 : Expertise Recommender Architecture

Implementation Details

We started with Webcrawl (Hall 2000), a public domain tool that takes a given URL, downloads the page, and follows all the URLs referenced in that page. It repeats this process on all the downloaded pages, recursively, until no more pages are left to download or no more links to follow. Initially, we limited the search to the web pages in the gsfc.nasa.gov domain. However, we soon found that quite a few NASA GSFC scientists maintain some of their personal web pages in other institutions they are affiliated to. They use URLs to those web pages from the NASA web pages than maintain a local copy under the gsfc.nasa.gov domain. In order to mine those external web pages, we updated Webcrawl to record those external references. Another script that we developed mines those external links to other domains but does not recurse beyond a configurable search depth.

The next step is to identify key words in the archived content. We used TFIDF, a classical text mining approach. In this approach, a set of stop words is first removed from the archived documents. Stop words are those words that occur very commonly and can be straightaway rejected as non-key words. Example stop words include "a", "and",

"the", "that", "where", etc. The remaining words in each document are 'stemmed' by removing root word extensions, so multiple related words map on to their word root. This further reduces the number of candidate key words. TFIDF then determines the occurrence of the words in various documents and prioritizes the words, so that a word occurring in least number of documents with highest frequency are picked over other words. The first *'N'* number of words is returned as the keywords, where *'N'* is user selectable.

However, we found that in the web accessible employee resume, which is a major source of the areas of interest/expertise, the expertise terms occurred just a few times. Thus, both TF and DF for the expertise information are low and hence, not selected as keywords by the above algorithm. This scheme is likely to work better for descriptive texts like annual reports, progress reports, etc. across many groups in GSFC because then the TF increases due to the multiple occurrence of potential keywords in the said texts while the DF remains the same. These documents are more likely to satisfy the underlying principles of TFIDF.

As a workaround, we studied the format of the web resumes and saw that they are not unstructured text narratives as assumed by TFIDF. We exploited the semi-structured nature of the resumes by looking for particular ways the expertise is described (in research interests and publication titles).

The document corpus returned by Webcrawl is analyzed to determine employee resumes. Then, keyword extraction in these resumes is restricted to particular sections like the "Research Interests" and the "Selected Publications". The HTML tags delimiting the text following the heading "Research Interests/Research Area/Expertise" is determined and extracted for further processing. This text is specified either as a narrative or as a list of interests. If there is a "Selected Publications" section, then again the text within the tags delimiting the section is extracted. Stop words are eliminated from the extracted text and the expertise of the employee returned is updated into an mSQL database that has been provided with a Web based query interface.

The Referral chains mentioned earlier in the section has been built from the co-authorship of publications in the "Selected Publications" section of the web resumes. We found that the names of co-authors are specified following the same conventions as any technical publications. We look for these patterns in the "Selected Publications" section and extract the data. The extracted data are the co-authors who have published with the employee whose resume is being parsed. This data is updated into the mSQL database to build the referral chain. This data is also used for stop word elimination from the text in the Publications section. When a user of the Expertise Recommender queries the system with a key word mined from the web pages, it will return the employee as the principal researcher and her collaborators as referrals.

We have also built a distributed version of the expertise extractor. In this mode, the Webcrawl and associated expertise extractor software are installed in a set of workstations. They are invoked from shell scripts running on a workstation that acts as a master. We used the Secure Shell, called ssh, to invoke these programs remotely in a secure way to crawl different web domains to speed up information extraction. Since the backend of the Expertise Recommender is a centralized database, it can be populated with the information extracted from the crawled web pages at individual workstations independently of every other workstation. This obviates the need for any synchronization across the multiple distributed processes, and also results in linear performance improvements with the number of participating workstations/processes. Thus the scripts running on individual workstations simultaneously crawl multiple domains, create the document corpus locally, extract expertise information, build referral chains and update the central database.

A web-based query interface to this central database has been developed with links to NASA's X.500 directory services. This has been developed using W3-mSQL v 2.0, which is the WWW interface package to the mSQL database. Lite (a programming language provided by W3-mSQL) code to query the database has been embedded into HTML documents. The HTML document is processed through the W3-mSQL program that generates HTML on the fly for the embedded Lite code. This web-based interface can be used to query by name or as an Expertise Recommender by querying on expertise area. When queried on expertise area, the system returns a list of employees who have the queried expertise, links to their X.500 data and a referral list for each employee sorted so as to give priority to co-authors who share the expertise being queried.

Conclusions and Future Work

In this paper, we present a prototype system that dynamically extract and update expertise information from currently existing semi structured web documents. The effectiveness of traditional text mining techniques such as TFIDF has been evaluated and found to be inappropriate. We developed a system that uses more customized methods to extract expertise information and build referral chains.

Automated discovery of expertise information from existing unstructured data sources such as web pages can be done more efficiently if the creator of the web page could annotate it to provide expertise-related information. This can be facilitated by the employees using an XML Scheme that allows such annotations, and development of tools that permit the annotations to be easily added to web pages.

The system can be enhanced by further research in the following areas:

- Keywords extracted from employee resumes may be mapped to NASA "standard" terminology to build a

hierarchical ontology so that finer levels of expertise areas from resumes map to broader fields of expertise. This would allow queries based on broader and more generally known terminology than those directly present in the resumes mined.

- Create seed fuzzy thesauri (Klir and Yuan 1995) that show the degree of similarity between various terms that describe similar expertise and thus solving the "synonym" problem.
- Use of association rule type approaches to find key phrases as well as words that occur together. This would enable the capture of more complex expertise descriptions than simple keywords.

References

Resnick P. and Varian H. 1997. Recommender Systems. *Communications of ACM* 40(3):56-58.

Joshi A.; Punyapu C.; and Karnam P. 1998. Personalization & Asynchronicity to support mobile web access. In Proceedings of the Workshop on Web Information and Data Management. ACM Conference on Information and Knowledge Management.

Joshi A.; Weerawarana S.; and Houstis E. N. 1997. Disconnected Browsing of Distributed Information. In Proc. Seventh IEEE International Workshop on Research Issues in Data Engineering, pages 101-108. IEEE, April 1997.

Joshi A. and Krishnapuram R. 1998. Robust fuzzy clustering methods to support web mining. In S. Chaudhuri and U. Dayal, editors, Proc. ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, June 1998.

Kautz H.; Selman B.; and Shah M. 1997. Referral Web: Combining social and collaborative filtering. *Communications of ACM* 30(3):63-65.

Kavasseri R.; Keating T.; Wittman M.; Joshi A.; and Weerawarana S. 1996. Web Intelligent Query - Disconnected Web Browsing using Cooperative Techniques. In Proc. 1st. IFCIS Intl. Conf. On Cooperative Information Systems, pages 167-174. IEEE, IEEE Press, 1996.

Klir G. and Yuan B. 1995. Fuzzy Sets and Fuzzy Logic. Prentice Hall, 1995.

Kraut R.; Galegher J.; and Edigo C. 1990. Intellectual Teamwork: Social and Technological Bases for Cooperative Work. Lawrence Erlbaum, Hillsdale, NJ, 1990.

Frakes W. B. and Baeza-Yates R. 1992. *Information Retrieval: Data Structures & Algorithms*. pp 131. Murray Hill, NJ: Prentice Hall.

Hall J. 2000. ftp://ftp.freebsd.org/pub/FreeBSD/branches/-current/ports/www/webcrawl.tar