ContactFinder: Extracting indications of expertise and answering questions with referrals

Bruce Krulwich and Chad Burkey
Center for Strategic Technology Research
Andersen Consulting LLP
100 South Wacker Drive, Chicago, IL 60606
{ krulwich, burkey } @ cstar.ac.com

Abstract

This paper presents a novel approach to utilizing large heterogeneous information repositories such as the World Wide Web, Lotus Notes™, or Usenet. Rather than extracting knowledge to be used directly in problem solving, our approach is to extract key contacts for specific technical areas that can then be contacted for help in those areas. We discuss this in the context of ContactFinder, an intelligent agent that extracts key contacts and answers discussion questions with referrals.

1. Heterogeneous information repositories

A growing number of businesses and institutions are using distributed information repositories to store large numbers of documents of various types. The growth of Internet sub-systems such as the World Wide Web and Gopher, as well as the emergence on the market of distributed database platforms such as Lotus Notes™, enables organizations of any size to collect and organize large heterogeneous collections of documents, ranging from working notes, memos and electronic mail to complete reports, proposals, design documentation, and databases. However, traditional techniques for identifying and gathering relevant documents become unmanageable when the organizations and document collections get very large.

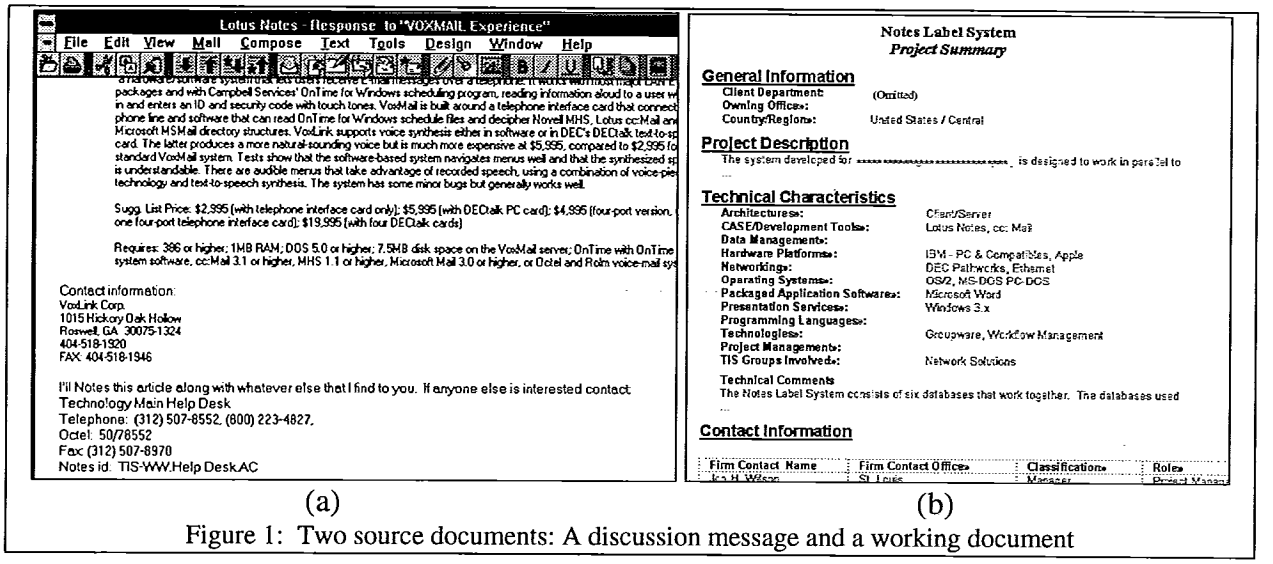
This paper describes an intelligent agent called ContactFinder, that is currently under development to address this problem.[1] ContactFinder is similar to research systems under development for question answering [Hammond *et. al.,* 1995], e-mail filtering [Maes and Kozierok, 1993; Lashkari *et. al.,* 1994], event scheduling [Dent *et. al.,* 1992; Maes and Kozierok, 1993; Kautz *et. al.,* 1994], Usenet message filtering [Sheth, 1994], or other information search and retrieval domains [Holte and Drummond, 1994; Knoblock and Arens, 1994; Levy *et. al.,* 1994]. Like these other systems, ContactFinder extracts information from a large number of documents in order to present it to users in a more focused and productive fashion.

Unlike these previous approaches, however, our goal is not to present the user with a subset of the information that can be used directly in problem solving. ContactFinder instead extracts key human contacts for different topic areas, and suggests contacts that can help users solve problems that arise.

This is a very valuable function for an intelligent agent to perform for several reasons. First, an agent that attempts to provide information that is directly relevant to the user's goals will always be limited by the information that is available. While this is not a problem in solving problems that are very basic or frequently asked [Hammond *et. al.,* 1995], it may make it difficult to be helpful in novel or very focused situations. In such a situation, however, a referral to a human expert can prove very useful. Second, extracting information from a repository that can aid problem solving relies heavily on correct processing of the details of document contents. Extracting key contacts, on the other hand, relies only on processing of more general topic areas, and can provide a useful contact without correct handling of the details. This makes our approach valuable in the short term (while document processing heuristics are still under development), as well as in novel or emerging areas for which the agent does not have a lot of detailed knowledge. Third, extracting contacts and facilitating human expertise transfer fits very well into current work styles, which will (hopefully) enable easy field tests of actual use.

ContactFinder's processing happens in two phases. The first phase scans the new documents in the information

[1] While we present a solution in the context of Lotus Notes, our solution is equally applicable to both Usenet newsgroups and World Wide Web documents.



(a) (b)

Figure 1: Two source documents: A discussion message and a working document

repositories and searches for indications of key contacts in any technical area. It extracts the contacts and their technical areas, and stores them in its own database. In the second phase, ContactFinder scans on-line discussions for questions. It extracts the topics of the questions and checks if it has a contact to give as a referral on those topic areas. If it does, it responds to the question with a referral.[2] This referral gives the name and contact information, along with quotes from the previous documents that served as the basis for the referral.

2. Extracting key contacts

Figure 1 shows two documents that can be used to extract key contacts. The first is a document in a discussion group, which gave a technical description of a product and ended with contacts for further information. The second is a document in a database of project summaries, which gives the project client, description, technical characteristics, and contact information.

These two documents several aspects of extraction of contacts from heterogeneous information repositories. The first document is likely to be a more useful contact, because it was given specifically as a contact for questions on the topic under discussion. The second, on the other hand, was given as a contact for a project, and may or may not be a good contact for particular aspects (technical or otherwise) of that project. On the other hand, even if the second document contact is not actually an expert in the area, he or she will very likely know of an expert who worked on the project, so the "six degrees of separation" principle applies. Such a person is still a good contact.

Another aspect of contact extraction is the likelihood that the contact can be extracted accurately by ContactFinder. The second document is trivial to parse for contacts, since they're specified in a document field designated for that purpose. The first document, however, is more difficult, because extraction requires processing raw formatted text. ContactFinder approaches this problem by using heuristics that are specifically designed for extracting contacts from text documents. Rather that attempt to process the document in a general fashion, it simply searches for indications of contacts, and looks locally at that point in the document for a name and contact information, again using specialized heuristics. This approach, very focused information extraction instead of general document understanding, has proven effective in this domain.

Another task that ContactFinder carries out in phase one is to extract topic indicators from each document, to serve as a description of the content areas for the extracted contact. In some cases, such as the structured project description in figure 1(b), this is easy. In other cases, such as raw text documents, this can be difficult. This process is used in both phases of ContactFinder, as well as in other agents under development at CSTaR [Krulwich, 1995], and is discussed in detail in section 4.

Figure 2 shows ContactFinder extracting contact information from the discussion document from figure

[2] For our discussion in this paper we are omitting logistical details, such as human confirmation of contact accuracy and topic area prior to public referral.

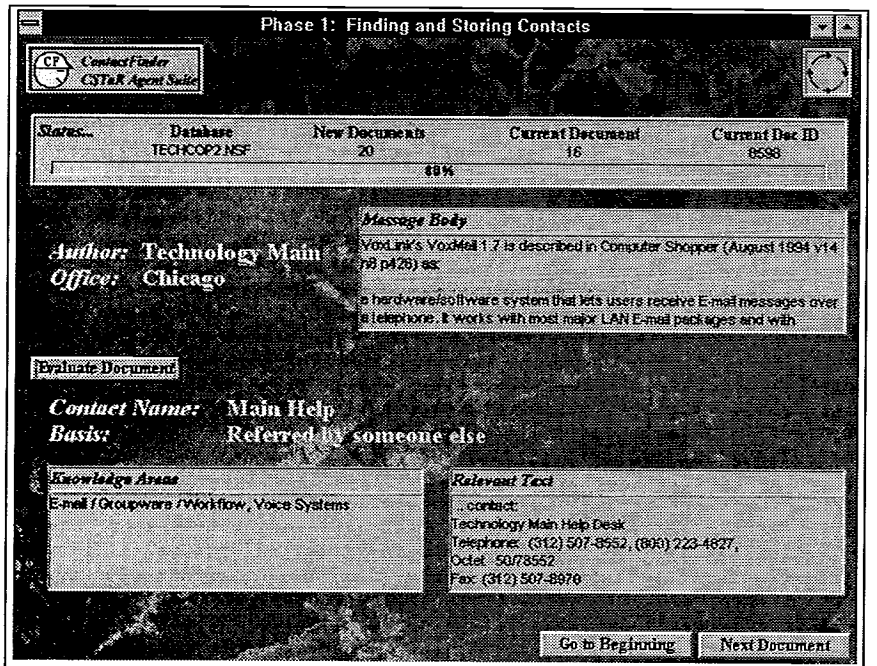


Figure 2: Extracting a third party contact

1(a).[3] The top of the screen shows the document information and its contents. The bottom shows the contact that was extracted, the topic areas, and the basis and relevant text of the extraction. In this case, the document in figure 1(a) ended with two contact names, one for the manufacturer of the product being discussed and one for people within the company who have used the product. ContactFinder uses third-party contact extraction heuristics to extract the second name and contact information, including the name of the contact ("Main help desk") and the phone numbers and internal phone numbers (shown in the relevant text).

There are a number of types of contacts that ContactFinder will extract from discussion documents. Besides third party referrals, ContactFinder will consider anyone who answers a question, without including a third party referral or an indication of lack of expertise, to be a contact for that area. As we said above, even if the person is not a direct technical expert, he or she is likely to have enough exposure to the topic area to provide good direction towards finding help. ContactFinder also searches for specific indications

ContactFinder was also able to extract areas of expertise for the extracted contact. In this example there were subject areas associated with the discussion document, which ContactFinder extracted and transformed into a canonical form for storage. These canonical forms have been hard-coded for ContactFinder, but could in the future be based on topic hierarchies used by library services.

3. Answering questions with referrals

ContactFinder's second phase is to find questions in on-line discussions, extract their topic areas, and search for previously-extracted contacts to give as referrals. Figure 3 shows such a question in a discussion group, which asks about the same product mentioned in figure 1(a). ContactFinder should find this document, extract indicators that it is a question, extract the topic areas, and find the expert extracted in the previous section.

In the current example, ContactFinder realizes that this is a question based on the phrase "Does anyone," with the question mark at the end of the sentence. It extracts the topic indicators using the same methods as are used in phase one (discussed in section 4), augmented with a search of the document text for all indicator phrases that it knows about. It then proceeds to search its database of

[3] Note that the display shown in figure 2 will never be seen by a user, since the process is run on the information repositories in background. This display is used for explanation and demonstration only.

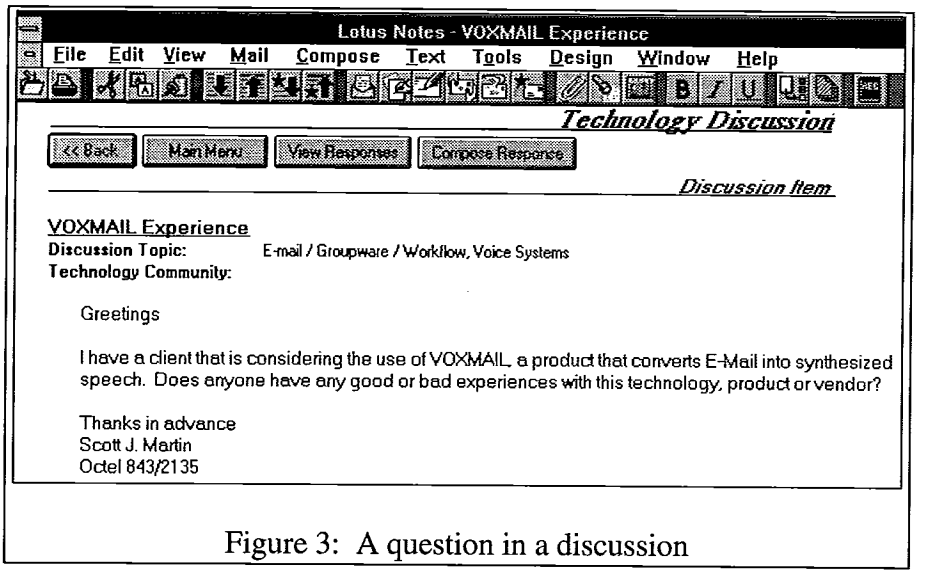


Figure 3: A question in a discussion

key contacts. It finds the Main Help Desk, which was extracted as a contact previously, and sends the referral.

A key point of this research initiative is that the documents that provide the original indication of expertise in phase one need not actually address the questions that are handled in the second phase. All that is required is that the topic areas that are extracted match close enough to make it likely that the contact person would be able to help the question asker, or minimally be able to refer the question asker to a third person. Because of this, our primary focus has been on the extraction of topic areas, not details of the questions being asked and the expertise being provided.

4. Extracting topic indicators

The most important step in both phases of the process described above is the extraction of semantically

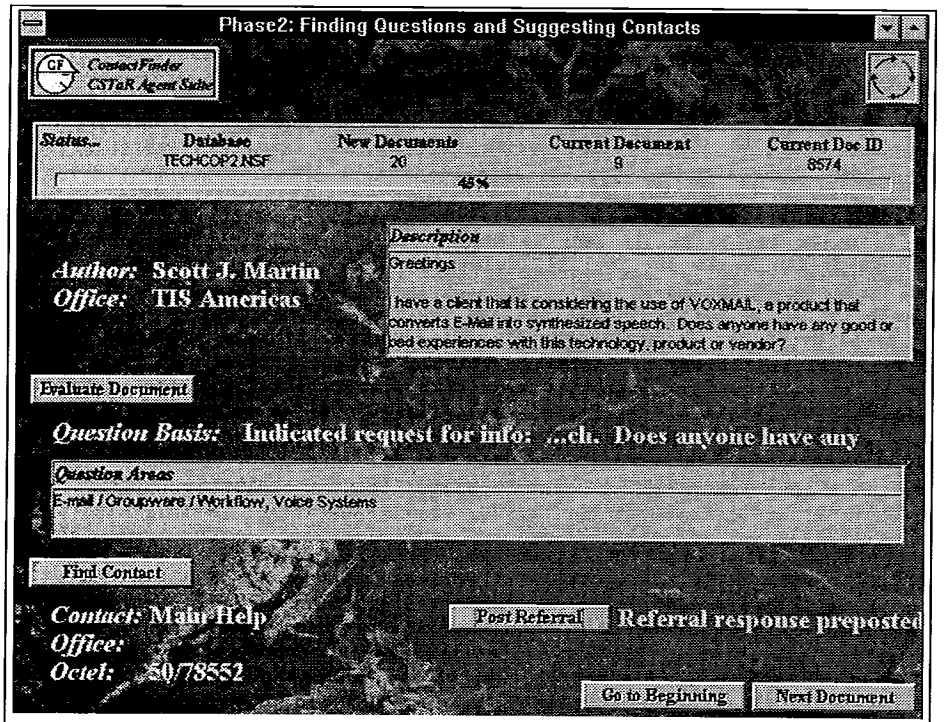


Figure 4: ContactFinder processing a question

significant phrases. Previous research has attempted to perform document comparison using most or all of the words in a document (e.g., [Sheth, 1994]), but we are avoiding this approach for two reasons. First, very few of the words in a document reflect the topic areas that are addressed by the text, given our goal of characterizing contacts, not content. Second, processing the entire text of a document is extremely costly in computational terms, and can be prohibitive for very large sample sets, while extracting semantically significant phrases and learning from them is quite tractable.

The document databases that we are using represent documents as a collection of fields, each of which contains keywords, names, raw text, or rich text. Rich text can contain, in addition to formatted text, imbedded objects such as pictures, other documents, spreadsheets, and so on. More importantly, each field is given a fixed semantic meaning within the context of a particular database. Our system extracts significant phrases from a document by treating each field using one of the following methods:

- Keyword list fields: Simply read the keywords from the field
- Name field: Consider the name itself as a possible contact
- Title or subject fields: Consider the field contents as a topic indicator if it's short
- Raw text or rich text fields: Extract visually or semantically significant phrases using *phrase extraction heuristics*

The critical step for extracting high quality phrases for documents is the set of heuristics for processing blocks of text. This is especially true for highly unstructured documents, which don't have many structured fields or keyword classifications. Even if a set of documents does have categorization keywords associated with each document, it is necessary to augment them with other significant phrases that the authors include in the document text.

To accomplish this we are in the process of integrating and building upon the heuristics found in the TAU system [Swaminathan, 1993] for extracting visually significant features from documents (see also [Rus and Subramanian, 1994]). This approach is built upon the observation that document authors often use a variety of visual techniques to convey significant pieces of information to readers, such as key points, lists of significant items, document structure, synopses, logical progression, and so on. Recognizing some of these visual patterns allows our agent to extract semantically meaningful phrases from bodies of text.

For example, a simple heuristic is to extract any single word that is fully capitalized. Such a word is most likely an acronym, or in some cases a proper technical name. In addition, there are a number of ways to find a definition of an acronym, such as looking for a parenthesized phrase immediately after the acronym, or at the words before the acronym if the acronym is itself in parentheses, or in the sentence or two preceding the acronym if neither is an parentheses.

Another simple heuristic is to extract any short phrase, of length 1-5 words, which appears in a different format from surrounding text, and which is not a complete sentence. This heuristic takes advantage of the convention of italicizing (or underlining) significant phrases the first time that they're used, or of capitalizing the first letters of proper names.

A further condition for both of these heuristics to be applicable is that the phrase not appear on a fixed list of non-significant words and phrases. For example, the first heuristic should not extract the acronym TM that may follow a product name, and the second should not extract words such as "not" or "certainly," which are often italicized for emphasis.

Other heuristics of this sort include recognition of lists of items (with bullet points or numbers), section headings, diagram labels, row and column headers in tables, and heavily repeated phrases. We have also explored heuristics such as extracting compound noun phrases (made up of three or more nouns in a row), which are frequently domain-specific phrases. Additionally, we are investigating the integration of a thesaurus, either commercial or domain-specific, to allow the agent to recognize that two words or phrases that have been extracted should be treated as equivalent.

A key aspect of these heuristics is that they are completely free of domain and contextual knowledge, and rather focus entirely on the syntactic structure of the text. This allows them to be widely applicable, without relying on background knowledge, and to be computationally efficient. There will be situations, however, in which such knowledge is necessary to perform effectively. Types of knowledge that could be added include topic areas and their relationships. The next section discusses a particular situation in which this is necessary, and undoubtedly more such cases will be uncovered as experimentation progresses. For the most part, however, ContactFinder will operate using

knowledge-free heuristics of the sort described in this section.

5. Preliminary results

To date, ContactFinder has operated for several weeks on an internal bulletin board for discussion of technical issues. Out of 3280 total documents on that bulletin board, ContactFinder extracted 1933 key contacts on various topics. This reflects our desire that ContactFinder operate relatively conservatively and error on the side of false negative contact extractions (failing to extract contacts) rather than false positives (extracting people as contacts who in fact are not). Many messages that may be indications of expertise, such as those that are top-level (not responses) but are not questions, are skipped by ContactFinder for lack of certainty.

Out of the same total set of messages, the system extracted 631 questions, for which it found 72 potential referrals. This rate of success (11.4%) reflects a number of aspects of the system's operation. First, the system will never post a referral to someone who has already responded to the question, which will often be the case during early operation when most of the system's set of key contacts have been extracted from the same set of messages. Second, the system requires a fairly strong match between topic areas of the question and the contact (90%) before considering a referral.

Of these 72 referrals, 21 related to a particular technical topic (a system named SAP) that posed difficulties for ContactFinder's approach. SAP is a very large system composed of many sub-systems, and for the most part any individual will only work with a small number of these sub-systems. It's necessary, therefore, for ContactFinder to correctly determine the relevant sub-systems for every contact and question. Unfortunately, this has proven difficult for a number of reasons. First, the sub-systems are often named as two-letter acronyms, without the use of punctuation as seperators, such as SD, FL, HP, MM, PS, DB, and PP. Many of these two-letter names are also used in English messages for other purposes, such as PP being used for page number references, or FL being used in addresses in Florida. For this reason it has been impossible for ContactFinder to extract these topic indicators in a knowledge-free and context-free fashion. Second, these sub-systems are sometimes referred to by their expanded names, requiring that ContactFinder know that the two-letter codes are synonyms for their expansions.

In general, this problem is with the knowledge-free nature of ContactFinder's topic extraction heuristics. Were ContactFinder to have specific knowledge of SAP and its sub-systems, it could look for the two-letter names only in the context of SAP, and could know the relevant synonyms. While we have in fact included knowledge of some synonyms in ContactFinder, we have not yet explored broader domain knowledge such as system components and sub-systems. Future research will determine the degree to which knowledge of this sort is necessary.

Out of the 51 remaining referrals, 28 of them have been approved by the contacts themselves, and 2 of them have been refused, giving us a 93% success rate (after excluding SAP-related referrals). Continuing testing of the system will determine how this rate holds up over larger numbers of documents.

Anecdotal feedback concerning ContactFinder has been generally positive. Some bulletin board users have feared that the system will reduce the number of on-line responses, moving the flow of knowledge off-line as people call contacts directly instead of waiting for them to respond on-line. In practice, however, it appears that just the opposite is true. In several cases the contacts referred by ContactFinder have posted information on-line, having not seen the questions until ContactFinder contacted them. If this trend continues, it appears that ContactFinder will in fact increase the amount of information on-line, as people who do not have a chance to read the bulletin board regularly are encouraged to respond to particular messages that relate to their areas of expertise.

6. Summary and discussion

We have described an intelligent agent prototype that mines a heterogeneous information repository for key contacts in specific subject areas. This approach allows the agent to assist people seeking information without requiring deep understanding of the information source documents. It also allows the agent to fit well with typical work styles by facilitating transfer of expertise between people. Lastly, the advice and the reasoning behind it is very easily understood by the people involved, because the referral can include a reference to the document that provided the contact.

The agent has been designed to operate by responding to questions on discussion groups. This allows it to answer only those questions for which it has referrals, and to

operate in a background fashion, appearing to users as simply another source of messages.

The system currently leaves open a number of issues that will serve as the basis for our continuing research. How can a large variety of types of documents be successfully mined for indications of contacts? How can documents consisting of plain formatted text be processed effectively to extract contacts, questions, and indications of subject area? What types of background will be needed to operate effectively in a variety of domain areas?

More generally, our approach raises the question of what other intelligent agent functionality can be achieved using document processing techniques such as significant phrase extraction, inductive learning, and document search. We are beginning development of several agents based on these techniques, such as an agent that learns the information interests of various users along with how to find new documents matching those interests [Krulwich, 1995], an agent that interacts with on-line internet services, and an agent that browses on-line documents to extract summary information. We are also investigating the application of other core document processing techniques, such as schema matching and message sequence modeling, to intelligent agent tasks. Future research will determine the range and effectiveness of intelligent agents that can be built on core document processing techniques such as these.

Acknowledgments: We would like to thank Anatole Gershman, Larry Birnbaum, Kishore Swaminathan, and Ed Gottsman for many useful discussions on the research presented here, and the members of CSTaR's Human-Systems Integration Lab for comments on the ContactFinder prototype.

References

Dent, L., Boticario, J., McDermott, J., Mitchell, T., and Zabrowski, D., 1992. A personal learning apprentice. In *Proceedings of the 1992 AAAI Conference,* San Jose, CA, pp. 96-103.

Hammond, K., Burke, R., and Martin, C., 1995. FAQ Finder: A case-based approach to knowledge navigation. To appear in *The Working Notes of the 1995 AAAI Spring Symposium on Information Gathering in Distributed Environments,* Palo Alto, CA.

Holte, R. and Drummond, C., 1994. A learning apprentice for browsing. In *Working Notes of the 1994 AAAI Spring Symposium on Software Agents,* Stanford, CA, pp. 37-42.

Kautz, H., Selman, B., Coen, M., Ketchpel, S., and Ramming, C., 1994. An experiment in the design of software agents. In *Proceedings of the 1994 AAAI Conference,* Seattle, WA, pp. 438-443.

Knoblock, C. and Arens, Y., 1994. An architecture for information retrieval agents. In *Working Notes of the 1994 AAAI Spring Symposium on Software Agents,* Stanford, CA, pp. 49-56.

Krulwich, B., 1995. Learning user interests across heterogeneous document databases. In *Working Notes of the 1995 AAAI Spring Symposium on Information Gathering in Distributed Environments,* Palo Alto, CA.

Lashkari, Y., Metral, M., and Maes, P., 1994. Collaborative interface agents. In *Proceedings of the 1994 AAAI Conference,* Seattle, WA, pp. 444-449.

Levy, A., Sagiv, Y., and Srivastava, D., 1994. Towards efficient information gathering agents. In *Working Notes of the 1994 AAAI Spring Symposium on Software Agents,* Stanford, CA, pp. 64-70.

Maes, P. and Kozierok, R., 1993. Learning interface agents. In *Proceedings of the 1993 AAAI Conference,* Washington, DC, pp. 459-465.

Rus, D., and Subramanian, D., 1994. Designing structure-based information agents. In *Working Notes of the 1994 AAAI Spring Symposium on Software Agents,* Stanford, CA, pp. 79-86.

Sheth, B., 1994. *A learning approach to personalized information filtering.* M.S. Thesis, EECS Department, MIT.

Swaminathan, K., 1993. *Tau: A domain-independent approach to information extraction from natural language documents.* DARPA workshop on document management, Palo Alto.