Learning user interests across heterogeneous document databases

Bruce Krulwich
Center for Strategic Technology Research
Andersen Consulting LLP
100 South Wacker Drive, Chicago, IL 60606
krulwich@andersen.com

Abstract

This paper discusses an intelligent agent that learns to identify documents of interest to particular users, in a distributed and dynamic database environment with databases consisting of mail messages, news articles, technical articles, on-line discussions, client information, proposals, design documentation, and so on. The agent interacts with the user to categorize each liked or disliked document, uses significant-phrase extraction and inductive learning techniques to determine recognition criteria for each category, and routinely gathers new documents that match the user's interests. We present the models used to describe the databases and the user's interests, and discuss the importance of techniques for acquiring high-quality input for learning algorithms.

1. Heterogeneous document databases

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 services such as Mosaic 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.

Consider a sample set of Lotus Notes databases of this sort, shown in the icon squares in the center region of figure 1. The top row consists of large-volume discussions with a broad audience, similar to many bulletin boards. Documents in these databases are messages on particular topics, sometimes indexed according to the topics and technologies being discussed. The second row shows news feeds, which consist of articles from a variety of sources. These databases often index documents by topic or information source, or in other more domain specific ways. The third row shows databases containing more focused discussion among smaller groups of people, in which the messages often are not indexed in any way. The fourth row shows databases that are other relatively dynamic collections of documents, such as bulletins, memos, project descriptions, or reference information. The total number of new documents in this particular collection of databases can be more than 2000 on any single day, while the number that are relevant to any particular user is often less than a dozen. Clearly it is difficult or impossible to scan this high volume of complex and unstructured data each day. Furthermore, these databases are stored worldwide in a distributed fashion, so access may often be slow or intermittent.

This paper describes an intelligent agent currently under development to address this problem,[1] similar to research systems under development for 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]. The agent will maintain a representation of the user's interests, and search nightly for new documents that match these interests. They are then gathered into a database of documents that are of interest to the particular user.[2] To avoid requiring each user to define recognition criteria for his or her interests, we will have the agent learn this from sets of documents that the user indicates are and are not of interest. Our most significant finding is that effective results depend largely on extracting high-quality indicator phrases from the documents for input to the learning algorithm, and less on the particular induction algorithms employed.

2. Learning user interests

The interactions between the user and the agent are shown in figure 2. The user is scanning a set of documents that are organized into various topics, and upon finding one that is of interest, clicks on the "smiley face" button in the upper

[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. This is discussed in section 4.

[2] It should be noted that this approach also solves a number of other operational difficulties, such as efficient remote access and database replication.

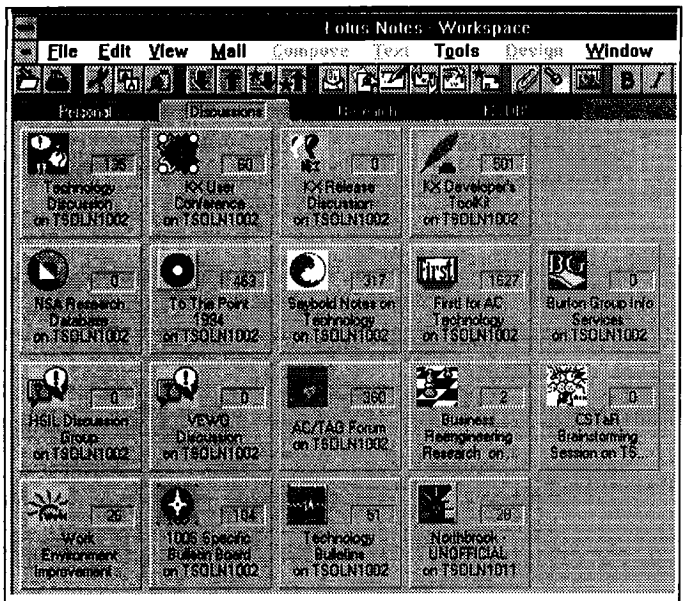

Figure 1: A sample collection of document databases

right corner. The agent then pops up a window that asks the user why he or she is interested in the selected document. More specifically, the agent asks the user to categorize the document in terms of the users own interest in the document, be it a subject area, project name, person, or something else entirely. This enables the learning system to generalize the documents into the specific categories that the user intends, without having to learn these as well (e.g., [Gil, 1994; Lieberman, 1994]). In our example, the user specifies that the document is interesting due to its being in the category "agents."

After the user has selected a number of documents as "interesting," the system will have a sample set of documents that are classified by the user. The system then learns classification criteria for each category using a three

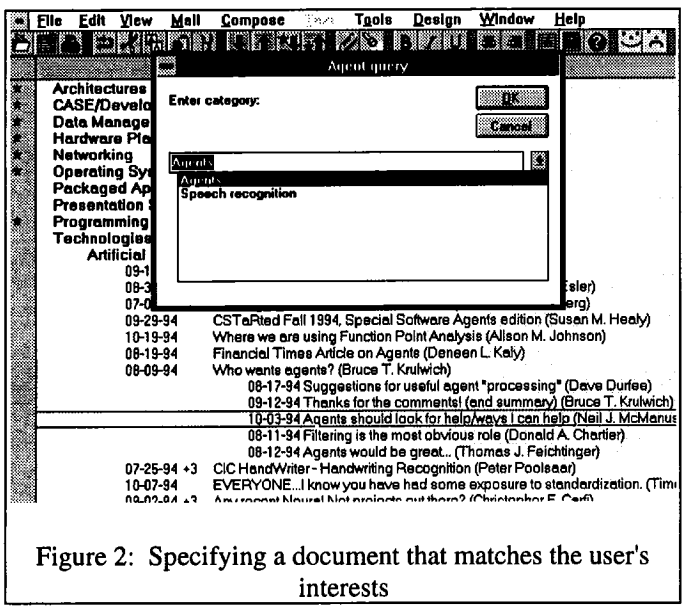

Figure 2: Specifying a document that matches the user's interests

step process:

1. Extract semantically significant phrases from each document.
2. Cross-check each document for the complete set of extracted phrases.
3. Learn which phrases serve as good indicators for each category.

The first step, phrase extraction, has proven to be the most crucial, despite the fact that most research in learning text document indicators has focused primarily on the induction algorithms themselves.[3] In situations where our system extracts good significant phrases from the documents, we have achieved effective learning with a very simple induction algorithm that counts phrase occurrences and searches for coverage of the sample set. Our exploration of alternate induction algorithms, including the NeuroAgent™ system and more standard decision-tree learning algorithms, confirms the primary importance of phrase extraction.

Figure 3 shows the learning process that would ordinarily be run in background. The system is learning indicators for a particular user's conception of a particular category, in this case the author's conception of the category "agent." On top is a list of the documents that the user has indicated fit this category. The system processes this list, first extracting a list of significant phrases that are present in the documents, using phrase extraction algorithms that are discussed in the next section. This process generates the list of phrases shown in the bottom of the figure. The system then goes back and searches all of the documents for occurrences of those phrases. This allows the system to compensate for the inevitable deficiencies in its phrase extraction algorithms. The system then learns a set of indicator phrases that covers the sample set and each of which appears in the largest number of documents. In our example, this list consists of the three phrases "agents," "EDI," and "artificial intelligence."

After the agent determines the set of indicators that best describes the user's interests in each category, it searches for new documents that match. Currently our agent simply searches for all documents containing at least one of the learned indicators. The documents that are found are copied into a customized database for the user, categorized according to the user's interests and sorted by the number of indicators present.

If any of these are false positives, and are in fact not interesting to the user, he/she would indicate this with the "frown face" button. The agent will then treat the document as a negative example, and prompt the user for the reason that this document is not a good member of the given

[3] A similar point has been made by [Holte, 1993].

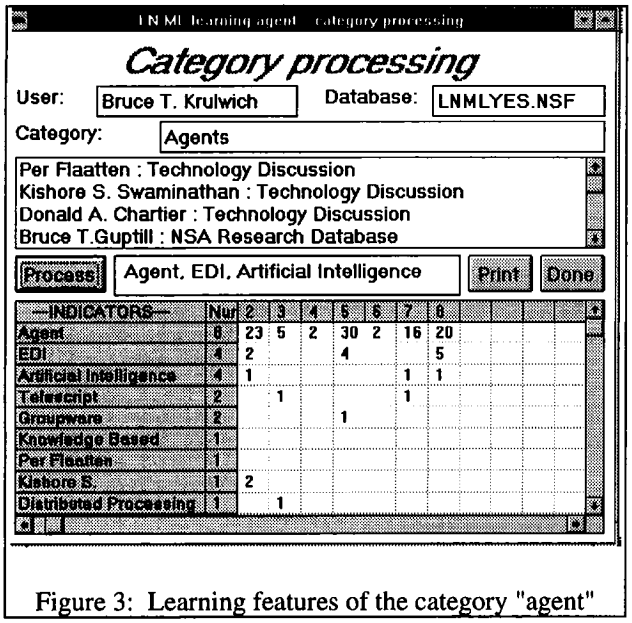

Figure 3: Learning features of the category "agent"

category.[4] Over time the agent will develop a set of indicators of inappropriate messages, which will be used to refine its search.

In our example, the indicator "EDI" (standing for "electronic data interchange") was not one that the user expected, nor is it one that appears on the surface to be appropriate as in indicator in its own right. It is found in the documents that address the notion of mobile agents moving around a network, in which it is a good description of the technology used, but it is probably too broad to serve as in indicator by itself. This clarification is learned subsequently by the agent when a document is presented to the user that refers to EDI but not to agents.

3. Modeling and processing document data

The most important step in the learning process described above is the extraction of sematically significant phrases. The learning algorithms themselves, however good they may be at generalizing patterns in data, will only be effective at learning document categories if they can be given high-quality input data. Previous research has attempted to perform induction on most or all of the words in a document (e.g., [Sheth, 1994]), but we are avoiding this approach for three reasons. First, very few of the words in a document reflect the underlying meaning and importance of the text, and moreover the distribution of words does not reflect the words or phrases that best characterize the document. Second, ideas discussed in a document can often be written using a wide variety of words, which will vary considerably across different authors and different organizations, but the catch-phrases and buzzwords are very often invariant across documents on the same category. Third, 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 more 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. The rich text can itself 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 an indicator
- Title or subject fields: Consider the field contents as an 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.

[4] Ultimately we would like to use the same learning process that we use for positive examples to learn indicators for rejection classes. The difficulties lie in sharing knowledge of the relevant phrases and categories between the two learning phases.

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.

As we said above, after our agent extracts significant phrases from each document, it rescans text blocks in all the documents for occurrences of all the phrases that have been extracted. Because of this, a phrase need only be extracted from one document by heuristic, and can then be found straightforwardly in those documents in which it was not easily recognizable.

In general, a key element of our research is developing heuristics for extracting significant phrases from documents. Ultimately, we feel that this contributes as much to high-quality machine learning of document categories than advances in induction algorithms.

4. Other issues in information agents

There are a number of issues that have arisen in considering systems such as the one we've described, that are the subjects of continuing research on this project. The most significant issues, we feel, concern the extraction of significant phrases from text, as we described in the previous section. We describe here other more general research issues in intelligent information management.

One significant focus of continuing research on this project is to extend the model of user preference learning to the level of working groups. The goal is to allow each user's agent to exchange learning results with other agents whose users share interests [Lashkari *et. al.*, 1994; Maes, 1994]. This will let each user's agent benefit from documents that other users have selected on the same topic. The difficulty in doing this is in knowing when two users have similar enough conceptions of their categories to share data.

Previous approaches to collaborative information filtering agents have had the agents share learned information about how to identify documents in various categories. The approach that we have taken here, of using heuristics to extract significant indicator phrases, opens up a new approach: have agents share the significant phrases that they extract, without sharing their particular user's feelings about the relevance of that phrase to the given category. This should enable a community of agents to build a shared set of phrases that are significant to a particular category, even if each user has a different conception of exactly which phrases are important to his or her interests in that category. This will enable each agent to use indicators that it may not be able to extract using its limited heuristics, if other agents have managed to extract them. This should allow collaboration between agents to be valuable while not rigidly forcing users to have similar conceptions of their categories.

Another focus is to extend our work to date, which has been in the context of Lotus Notes databases, to both World Wide Web (Mosaic) documents, and Usenet newsgroup articles [Sheth, 1994]. Both of these, we feel, contain a wealth of information that is largely untapped by all but the most avid users. These information sources also suffer from the signal-to-noise problem described above, that for any individual most documents or messages are uninteresting. An agent of the sort that we have described above would therefore be valuable.

A number of issues arise in considering these additional information sources. Most importantly, each of these sources impose additional requirements and opportunities for phrase extraction. In the case of Usenet articles, phrase extraction requires much more capacity for processing of raw text, due to the lack of formatting constructs. In the case of WWW documents, facilities must be established to extract not only the phrases in the body of documents, but also the phrases used in other documents to provide links to the document under consideration. In general, there are a number of relationships possible between two documents that are linked, and these relationships will give our system additional information to use in processing the documents. In addition, each of these information sources provides documents in groups, either by topic area in Usenet, or by provider and path in WWW. All of this additional knowledge will have to be brought to bear in processing documents from these information sources, and will have an impact on the nature of the learned categories.

Another issue that must be addressed in developing an information management agent is the support of exploration of new areas of potential interest [Sheth, 1994]. If a user were to use the agent described in this paper for all interactions with the information sources, new topics would never be seen. There must be a mechanism for the agent to determine that a document is sufficiently different from all previous documents to warrant offering it to the user as potentially interesting. The user could then indicate interest or disinterest through the mechanisms described above.

5. Summary and future work

We have described a system under development that learns user preferences in dynamic document databases. While most previous research in this area has focused on induction algorithms, we have instead focused on extracting high-quality learning inputs from the documents. This allows us to get results that better match the user's expectations, with less computational cost.

Future research will focus on four areas. First, better phrase extraction heuristics are necessary for handling rich text. Second, the system metaphor should be extended to support working groups. Third, more powerful induction methods will be considered and applied. Fourth, the approach will be applied to other information sources, such as World Wide Web documents or Usenet bulletin boards. With these advancements, we hope to have an effective system of agents for document gathering in information environments that are very large, dynamic, and of broad interest.

More generally, our approach raises the question of what other 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 this techniques, such as an agent that scans large quantities of messages to learn who the domain experts are in a variety of areas, an agent that learns the locations in which documents on various topics can be found, or an agent that monitors messages as they are composed and extracts summary and categorization information. We are also investigating the application of other core document processing techniques, such as complex 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.

Acknowledgements: I would like to thank Anatole Gershman, Larry Birnbaum, Kishore Swaminathan, and Chad Burkey for many useful discussions on the research presented here.

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.

Gil, Y., 1994. Trainable software agents. In *Working Notes of the 1994 AAAI Spring Symposium on Software Agents,* Stanford, CA, pp. 99-102.

Holte, R., 1993. *Very simple classification rules perform well on most commonly used datasets.* Machine Learning Journal 11(1), pp. 63-90.

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.

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.

Lieberman, H., 1994. Demonstrational techniques for instructible user interface agents. In *Working Notes of the 1994 AAAI Spring Symposium on Software Agents,* Stanford, CA, pp. 107-109.

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

Maes, P., 1994. Social interface agents: Acquiring competence by learning from users and other agents. In *Working Notes of the 1994 AAAI Spring Symposium on Software Agents,* Stanford, CA, pp. 71-78.

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.