Exploiting Information Access Patterns for Context-Based Retrieval

Travis Bauer and David B. Leake
Computer Science Department, Indiana University
150 S. Woodlawn Avenue, Bloomington, IN 47405
{trbauer, leake}@indiana.edu

ABSTRACT

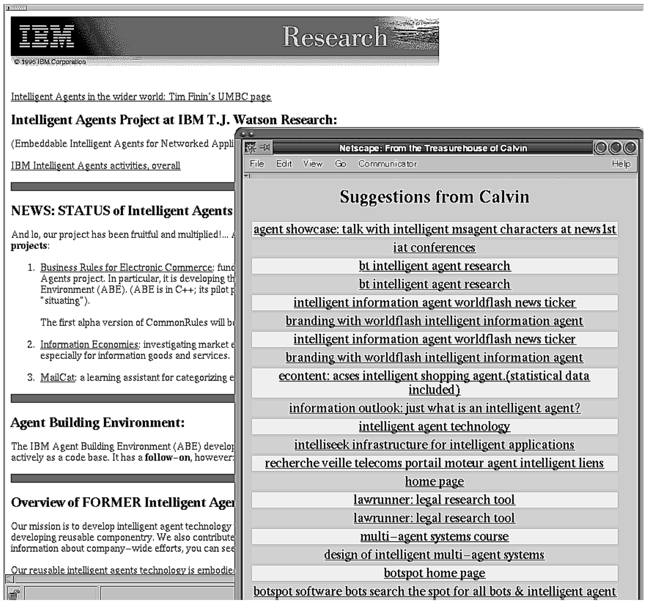
In order for intelligent interfaces to provide proactive assistance, they must customize their behavior based on the user's task context. Existing systems often assess context based on a single snapshot of the user's current activities (e.g., examining the content of the document that the user is currently consulting). However, an accurate picture of the user's context may depend not only on this local information, but also on information about the user's behavior over time. This paper discusses work on a recommender system, Calvin, which learns to identify broader contexts by relating documents that tend to be accessed together. Calvin's text analysis algorithm, WordSieve, develops term vector descriptions of these contexts in real time, without needing to accumulate comprehensive statistics about an entire corpus. Calvin uses these descriptions (1) to index documents to suggest them in similar future contexts and (2) to formulate context-based queries for search engines. Results of initial experiments are encouraging for the approach's improved ability to associate documents with the research tasks in which they were consulted, compared to methods using only local information. This paper sketches the project goals, the current implementation of the system, and plans for its continued development and evaluation.

Keywords

Recommender systems, information retrieval

INTRODUCTION

Personal information agents need to extract information about a user's task context and use this knowledge to provide assistance, such as suggesting relevant new documents to the user, or reminding the user of documents that have been useful in similar contexts in the past. To be acceptable to users, these agents need to learn unobtrusively, with relatively little explicit feedback from the user [1]. We are developing a recommender system,



Intelligent Agents in the wider world: Tim Finin's UMBC page

Intelligent Agents Project at IBM T.J. Watson Research:

(Embeddable Intelligent Agents for Networked Appli

IBM Intelligent Agents activities, overall

NEWS: STATUS of Intelligent Agents

And lo, our project has been fruitful and multiplied!... A
projects:

1. Business Rules for Electronic Commerce: func Agents project. In particular, it is developing th Environment (ABE). (ABE is in C++; its pilot p "situating").

   The first alpha version of CommonRules will be

2. Information Economies: investigating market e especially for information goods and services.
3. MailCat: a learning assistant for categorizing e

Agent Building Environment:

The IBM Agent Building Environment (ABE) develop actively as a code base. It has a **follow–on**, however:

Overview of FORMER Intelligent Agen

Our mission is to develop intelligent agent technology developing reusable componentry. We also contribute information about company–wide efforts, you can see

Our reusable intelligent agents technology is embodie

Netscape: From the Treasurehouse of Calvin
File Edit View Go Communicator Help

Suggestions from Calvin

agent showcase: talk with intelligent msagent characters at news1st
iat conferences
bt intelligent agent research
bt intelligent agent research
intelligent information agent worldflash news ticker
branding with worldflash intelligent information agent
intelligent information agent worldflash news ticker
branding with worldflash intelligent information agent
econtent: acses intelligent shopping agent.(statistical data included)
information outlook: just what is an intelligent agent?
intelligent agent technology
intelliseek infrastructure for intelligent applications
recherche veille telecoms portail moteur agent intelligent liens
home page
lawrunner: legal research tool
lawrunner: legal research tool
multi–agent systems course
design of intelligent multi–agent systems
botspot home page
botspot software bots search the spot for all bots & intelligent agent

Figure 1 Calvin's web based interface

Calvin, and a text analysis algorithm, WordSieve, to unobtrusively use the context of previous document accesses to suggest documents used in similar prior contexts.

Calvin monitors user browsing to capture and proactively provide task-relevant information. Other personal information agents, such as Letiza [6] and Watson [5] have similar aims. What distinguishes Calvin's approach is that rather than retrieving documents based solely on local information, Calvin extracts information about how the user tends to access groups of documents over time, and predicts document relevance based on similarity of the extended sessions within which various documents are accessed. To illustrate, consider the example of a person reading a web page about IBM's intelligent agents project. The user might be interested specifically in agents, in which case this page is probably being used as a source of information about agents. Alternatively, the user might be interested in jobs at IBM, in which case this page is probably being used as a place to learn more about the company. Studying document content alone will not enable distinguishing these

IUI'02, January 13-16, 2002, San Francisco, California, USA.

two situations. Sufficient local task information may enable making this distinction (e.g., if the user is writing a paper about agents [5]). However, such information may not be available if the user is simply performing a browsing task. On the other hand, if the user tends to do research on agents, accessing many documents about agents at a time, and not many documents about IBM, then this pattern of document accesses suggests that the user has come to this page to learn about agents. In this case, the system should present a page with information about agents—as Calvin does in the screen shot for its recommendations in this situation, shown in Figure 1.

WordSieve is a competitive learning text analysis algorithm that can choose dimensions for a term vector space. It runs fast enough to work in real time, while the user is accessing documents, and requires no explicit feedback from the user. Calvin uses WordSieve to automatically learn keywords that tend to partition a user's document access streams into different contexts, and uses these terms to characterize the user's current context for the purpose of indexing and searching. Initial tests are promising for its ability to identify useful context descriptions, and the system is now being scaled up and refined for larger-scale experiments.

CALVIN

Calvin observes users while they are browsing the WWW. It also provides a means for indexing consulted documents by the user's current task context and retrieving documents for suggestions. An early prototype of Calvin is discussed in [4]. The architecture is being redesigned as a dynamic multi-agent environment in which a variety of software tools can simultaneously observe, analyze, and support the user. The most recent version of Calvin is also totally web-accessible, supports multiple users concurrently, and conducts background Internet searches on automatically generated queries.

WORDSIEVE

WordSieve learns about a user's document access patterns to generate term vector indices for document storage and retrieval. The algorithm builds a persistent user profile corresponding to user access patterns, and uses this profile to index individual documents and identify similar contexts in the future for appropriate retrieval. To be acceptable to end users, such an algorithm needs to learn with relatively little explicit feedback. To be practical for use in an information retrieval agent, it does not have the luxury of pre-analyzing the entire text corpus. These considerations strongly influenced WordSieve's design.

By studying each document when the user accesses it, WordSieve identifies words that occur frequently for periods of time, but are absent at other times. WordSieve is built as a three layered competitive network in which words effectively compete to maintain positions. The first layer is a short-term memory that identifies the words that are currently occurring frequently in the document stream (whether or not significant). The second layer functions as a long-term memory that identifies which of these words tend to occur frequently over broader periods of time. The third layer identifies which of these words occasionally cease occurring, thus making them candidates for partitioning sets of related documents.

Words that achieve appropriate values in this network constitute a dictionary for for indexing, querying and retrieval. In our initial tests, WordSieve outperformed *Term Frequency/Inverse Document Frequency* at generating vectors that were closely correlated with a vector representation of the search task given to a user [3].

Status and Direction

Additional details on Calvin and WordSieve are available in [3, 4]. Initial experiments with WordSieve have been promising, so we are integrating the WordSieve context analysis mechanism with the Calvin architecture to enable full-scale tests of the algorithm as it performs "in the wild." We are also developing methods for more refined evaluations of Calvin, including studies of whether WordSieve generates more useful queries than other possible methods, and enhancing WordSieve to be responsive to co-occurrence of terms. This will help it ignore spurious terms and identify context-relevant terms that are not occurring during a specific browsing session, by looking at terms that have co-occurred in the past with relevant terms from the documents being accessed.

This research is supported in part by the Department of Education under award P200A80301-98, and by NASA under award NCC 2-1216.

REFERENCES

1. Goecks, J. and J. Shavlik. Learning users' interests by unobtrusively observing their normal behavior, in *Proceedings of IUI-2000, ACM Press,* pp. 129-132.

2. Berry, M, S. Dumais and T. Letsche. Computational methods for intelligent information access, in *Proceedings of Supercomputing '95,* pp. 1-38.

3. Bauer, T. and Leake D., WordSieve: A Method for Real-Time Context Extraction. *Modeling and Using Context: Proceedings of the Third International and Interdisciplinary Conference, Context 2001*, Springer-Verlag, 2001, pp. 30-44.

4. Leake, David, Travis Bauer, Anna Maguitman, and David Wilson. Capture, Storage and Reuse of Lessons about Information Resources: Supporting Task-Based Information Search, in *Proceedings of the AAAI-2000 Workshop on Intelligent Lessons Learned Systems, AAAI Press,* pp. 33-37.

5. Budzik, J., K. Hammond, and L. Birnbaum. Information access in context. *Knowledge based systems*. 14:37-53, 2001

6. Henry Lieberman. Letizia: An Agent That Assists Web Browsing. In *Proceedings of IJCAI*-95, Morgan Kaufmann, pp. 924-929.