# Intelligent Mobile Agents for Information Retrieval and Knowledge Discovery from Distributed Data and Knowledge Sources

Jihoon Yang, Vasant Honavar, Les Miller and Johnny Wong
Artificial Intelligence Research Laboratory, Department of Computer Science
226 Atanasoff Hall, Iowa State University, Ames, IA 50011. U.S.A.

*Abstract*— **Tools for selective proactive as well as reactive information retrieval and knowledge discovery constitute some of the key enabling technologies for managing the data overload and translating recent advances in automated data acquisition, digital storage, computers and communications into fundamental advances in decision support, scientific discovery and related applications. This paper describes an implementation of intelligent, customizable mobile software agents for information retrieval and knowledge discovery from distributed data sources. These tools are part of the *Distributed Knowledge Network* (DKN) toolbox that is being developed at the Iowa State University's Artificial Intelligence Laboratory. Experiments with retrieval of journal paper abstracts demonstrate the feasibility of using machine learning to design mobile intelligent agents for customized information retrieval. A similar approach has been successfully employed for knowledge discovery (using machine learning) from distributed data collections.**

## I. Introduction

Recent advances in high throughput data acquisition technologies, digital storage technologies, computers and communications have made it possible to gather and store scientific (e.g., genome data), business, and military data (e.g., intelligence data) in electronic form in databases and computerized information systems. In order to translate the advances in our ability to acquire and store data in increasing volumes and at increasing rates into gains in our understanding of the respective domains and new capabilities for effective decision-making, sophisticated tools are needed for information retrieval, knowledge discovery, and decision support [1].

Several applications (e.g. military command and control, law enforcement, scientific discovery), require the use of multiple, geographically distributed, heterogeneous data and knowledge sources (e.g., sensors, satellites, intelligence reports, etc.). This calls for the use of *information assistants* or *software agents* for intelligent, selective, and context-sensitive data gathering and data assimilation prior to large scale data analysis. Such tools have to monitor different data sources and route the appropriate information selectively to relevant sites or specific users. Since the information of interest is user and context-dependent, such tools have to be customizable to specific users and information contexts.

Given the large volumes of data that are involved, it is desirable to perform as much analysis as feasible at the sites where the data is located and transmit only the results of analysis rather than flooding the network with data. This calls for the use of *mobile* software agents that can transport themselves to appropriate sites, carry out the computation on site, and return with useful results.

Intelligent agents, mobile agents, and multi-agent systems provide an attractive approach to the design of *distributed knowledge network* (DKN) tools for information retrieval, information extraction, information assimilation, and knowledge discovery using heterogeneous, distributed data and knowledge sources. (See [1] for an overview of distributed knowledge networks).

This paper focuses primarily on the design and implementation of customizable intelligent mobile agents for document retrieval from distributed document collections. We have applied a similar approach to perform knowledge discovery using machine learning algorithms from distributed data and knowledge sources.

The rest of the paper is organized as follows: Section II introduces mobile agents and briefly describes the mobile agent infrastructure that was used in our implementation. Section III describes our approach to customized information retrieval and summarizes the results of some of our experi-

Vasant Honavar's research was partially supported by the National Science Foundation (through grant IRI-9409580) and the John Deere Foundation.

ments with the system. Section IV concludes with a summary and discussion of some directions for future research.

## II. Mobile Agents

Mobile agent technology [2], [3], [4], facilitated by recent advances in computers, communications, and artificial intelligence, provides an attractive framework for the design and implementation of *communicating applications* in general [4], [5], and distributed knowledge networks in particular [1]. A mobile agent can be defined as a named object which contains code, persistent state, data and a set of attributes (e.g., movement history, authentication keys) [2], [3], [4] and can move about or transport itself from one host to another as needed for accomplishing its tasks. Mobile agents provide a potentially efficient framework for performing computation in a distributed fashion at sites where the relevant data is available instead of expensive shipping of large volumes of data across the network. They provide an attractive paradigm for design and implementation of scalable, flexible, and extensible systems for selective information retrieval and knowledge discovery from multiple, geographically distributed, heterogeneous, data sources.

Mobile agent infrastructures support the creation, deployment, and management of mobile software agents. There is considerable ongoing research on such infrastructures [4], [6], [7], [8]. Most of these efforts share a similar architecture consisting of at least three components: agent servers, agent interface, and agent brokers (service directory). Agent servers support basic agent migration mechanisms, authentication, and sometimes provide other services. Agent brokers provide addresses of agent servers and support mechanisms for uniquely naming agents and agent servers. The agent interface is used by application programs to create and interact with agents. The different mobile agent infrastructures differ from each other in terms of detailed implementation (e.g., choice of agent transport mechanism, agent programming languages, etc.). Recently, a set of standards for *Mobile Agent System Interoperability Facilities* (MAF) [9] has been proposed by a consortium of several companies and research groups including the GMD FOKUS and IBM. MAF standardizes several key aspects of mobile agent infrastructure including: agent management, agent transfer, agent and agent system names, agent system types, and location (address) syntax. This will facilitate interoperability among different mobile agent systems that are based on different architecture, design, and implementation choices. (See [7] for a discussion of an MAF-compliant Java-based implementation of a mobile agent infrastructure).

For the experiments described in this paper, we used ObjectSpace's Voyager [8] mobile agent infrastructure. Voyager is implemented entirely in Java and uses the Java language object model. Voyager allows regular message syntax to construct remote objects, send them messages and move them between programs. The Voyager *Object Request Broker (ORB)* provides services for mobile objects and autonomous agents. It also provides services for persistence, scalable group communication, and basic directory services.

An attractive feature of Voyager is that it seamlessly integrates distributed computing with agent technology. An agent in the Voyager system is a special kind of object that can move independently, can continue to execute as it moves, and otherwise behaves like any other Java object. Voyager enables objects and other agents to send standard Java messages to an agent even as the agent is moving. In addition, Voyager allows us to remote-enable any Java class, even a third party library class, without the need to modify the class source. (See [8] for further details of the Voyager platform).

## III. Customizable Information Retrieval Agents

A primary focus of this paper is on the design of customizable agents for information retrieval from distributed data sources. We illustrate our approach to this problem using the customized document retrieval task. However, the proposed approach can be easily adapted to handle a wide range of selective information retrieval tasks (e.g., image retrieval, DNA sequence retrieval, etc.).

The recent proliferation of computers and communication networks has made it possible for scientists, decision makers, etc. to be able to access a wide variety of geographically distributed information sources. However, effective use of these information sources (e.g., documents, articles, electronic mail messages, news, and the like) requires fairly sophisticated tools for locating, classifying, and retrieving only those items that are of interest to a given user of a group of users. For instance, a researcher might be interested in selectively obtaining recently published papers related to his or her research from diverse sources. Similarly, an individual might be interested in selectively retrieving and reading news articles on topics that interest him or her. A military intelligence analyst might similarly benefit from being made aware of reports that might be relevant in specific decision-making contexts. All of these tasks present us with the task of classifying data (e.g., documents) and selectively retrieving the items of interest. This is just one of many similar tasks that need to be automated in order for people to be able to make effective use of the emerging computing, communications, and information infrastructure.

Document retrieval has been the subject of study for several decades [10]. However, work on personalized document retrieval agents is relatively new. Some examples include WebWatcher [11], Personal WebWatcher [12], Fab [13], [14] which learn user interests using user feedback and recommend/prefetch web pages; and software agents for mail handling and electronic news filtering [15], and software agents for classifying journal abstracts and news articles [16].

### *A. Design of Customizable Document Classifiers*

Classification of documents necessarily has to involve some analysis of the contents of a document. In the absence of a satisfactory solution to the natural language understanding problem, most current approaches to document retrieval use a *bag of words* representation of documents [17], [18]. A document is represented as a vector of weights for terms (or words) from a *vocabulary*. Although a variety of approaches can be used to design document classifiers using the bag of words representation, to keep the discussion focused, we restrict ourselves to a relatively simple yet effective approach based on the TFIDF (*term frequency – inverse document frequency*) classifier [17], [18].

The TFIDF approach to document classification works as follows: Let $\mathcal{V}$ be the vocabulary used. Let $d$ be a document. The document is processed using *stemming* and *stopping* procedures [17], [18] to obtain a bag of words for document $d$. Let $w_i$ be the $i$th word in the vocabulary $\mathcal{V}$. The *term frequency* of $w_i$, $TF(w_i, d)$, is the number of times $w_i$ occurs in $d$. The *document frequency* of $w_i$, $DF(w_i)$, is the number of documents in which $w_i$ occurs at least once. The *inverse document frequency* of $w_i$, $IDF(w_i)$, is defined as $IDF(w_i) = log(\frac{|D|}{DF(w_i)})$, where $|D|$ is the total number of documents. Then, the *term frequency – inverse document frequency* of $w_i$, $TFIDF(w_i, d)$ is given by $TF(w_i, d) \cdot IDF(w_i)$ [17], [18]. The vector representation $\vec{d}$ of a document $d$ is given by $\vec{d} = \left[TFIDF(w_1, d)\, TFIDF(w_2, d)\ \cdots TFIDF(w_{|\mathcal{V}|}, d)\right]$.

A TFIDF document classifier is constructed as follows: Let $\mathcal{C}$ be a collection of document classes of interest. A prototype vector $\vec{c}$ is generated for each class $c \in \mathcal{C}$ as follows:

$$\vec{c} = \sum_{d \in c} \vec{d}$$

A document $n$ to be classified is assigned to the class whose prototype is the closest to it (as measured by some suitable distance function). The cosine function is a commonly used distance measure. Thus, the classification of $n$ is given by

$$\arg\max_{c \in \mathcal{C}} cos(\vec{n}, \vec{c})$$

$$= \arg\max_{c \in \mathcal{C}} \frac{\vec{n} \cdot \vec{c}}{\| \vec{n} \| \cdot \| \vec{c} \|}$$

A simple version of the document classification task is to classify documents into two categories: interesting or uninteresting. This is accomplished by training a document classifier (i.e., constructing the class prototypes) using a set of pre-classified documents.

### *B. Experiments*

The TFIDF classifier is incorporated into mobile agents on the Voyager mobile agent platform. First, a mobile agent is generated for searching and retrieving a set of documents from a remote site that matches with the query given by the user. The query is used to retrieve documents that match the query. The agent is shipped to the remote site, and the agent retrieves matching documents and sends them to the local site. Then the agent dies. The user then classifies the retrieved documents as interesting or not interesting. This provides a dataset for training the document classification and retrieval agent using the approach outlined above. Then, a TFIDF based document classification and retrieval agent is designed using the training data and the resulting agent is sent to a remote site to retrieve *relevant* documents. Relevant documents are determined by the classifier at the remote site and returned to the local site. Then the agent dies.

We experimented with the use of customizable information retrieval agents on a number of document classification and retrieval tasks including the classification of news articles and journal abstracts. The interested reader is referred to [16] for a detailed of the experiments and the results. A trainable agent is initially provided with a query that results in the retrieval of a collection of documents which are then used to customize the behavior of the agent using the classification provided by the user. Once the classifier is constructed, it is used by mobile agents to selectively retrieve documents from remote collections. It was found that the TFIDF based approach to the design of customized information retrieval agents worked quite well (in most cases, assigning correct classification in close to or exceeding 90% of the documents that were not used for training the classifier).

Instead of downloading all documents from the distributed databases, the agents worked on the remote databases, retrieved only a subset of relevant documents and sent them to the local site thereby minimizing the duration of the expensive network connection. In our experiments, the amount of data transferred in the mobile agents (the classifier and the relevant information) was much less than that in a conventional system (the entire data). The savings of network connection in mobile agents will be even greater for very large data.

Systematic experiments with distributed data sources, under a variety of network conditions and data source characteristics (e.g., percentage of relevant documents) are in progress.

## IV. Summary and Discussion

Intelligent mobile agents offer an attractive paradigm for the design of modular, flexible, robust, scalable, and adaptive information systems for a variety of applications, including customized information retrieval. Machine learning appears to be the most practical approach to designing customizable software agents. This paper has presented the design of mobile agents for customized document classification and retrieval using the commercially available Voyager mobile agent infrastructure. The experiments reported here demonstrate the effectiveness of machine learning as a viable and practical approach to the design of such agents.

Mobile agents offer a significant performance advantage over conventional remote procedure calls when we deal with very large remote data collections only a small fraction of which is of interest to the user. We have also used mobile agents to perform knowledge discovery by applying machine learning algorithms on distributed remote data collections.

Some interesting and promising directions for further research include: design and implementation of multi-agent systems in which agents collaborate in retrieving and analyzing data from distributed data and knowledge sources and provide decision support services in complex real world applications; systematic studies to explore the relative performance advantages and disadvantages of alternative designs and implementations of such systems (including the mix of mobile and static agents); design of customizable information retrieval agents for a broad range of semi-structured and unstructured data sources e.g., genome data, image data, etc.

## References

[1] V. Honavar, L. Miller, and J. Wong, "Distributed knowledge networks," in *IEEE Information Technology Conference*, Syracuse, NY, 1998.

[2] C. Harrison, D. Chess, and A. Kershenbaum, "Mobile agents: Are they a good idea?," Tech. Rep., T. J. Watson Research Center, IBM, 1995.

[3] R. Gray, "Agent tcl: A flexible and secure mobile-agent system," in *Fourth Annual TCL/TK Workshop (TCL'96)*, 1996.

[4] J. White, "Mobile agents," in *Software Agents*, J. Bradshaw, Ed. MIT Press, Cambridge, MA, 1997.

[5] B. Hermans, "Intelligent software agents on the internet: An inventory of currently offered functionality in the information society and a prediction of (near-)future developments," 1996, http://www.hermans.org/agents.

[6] J. Kiniry and D. Zimmerman, "A hands-on look at java mobile agents," *IEEE Internet Computing*, July/August 1997.

[7] J. Wong, V. Honavar, L. Miller, and V. Naganathan, "Design and implementation of mobile agent infrastructure based on mobile agent interoperability facilities (maf)," Submitted for publication.

[8] ObjectSpace, Inc., Voyager *Core Package Technical Overview. Agent Enhanced Distributed Computing for Java*, 1997, http://www.objectspace.com/voyager.

[9] GMD FOKUS, IBM, and *et al.*, "Mobile agent system interoperability facilities specification," OMG TC Document orbos/97-10-05.

[10] G. Salton, *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, 1982.

[11] T. Joachims, D. Freitag, and T. Mitchell, "Webwatcher: A tour guide for the world wide web," in *International Joint Conference on Artificial Intelligence*, 1997.

[12] D. Mladenic, "Personal webwatcher: Design and implementation," Tech. Rep., J. Stefan Institute, Ljubljana, Slovenia, 1996.

[13] M. Balabanovic, "An adaptive web page recommendation service," in *Proceedings of the First International Conference on Autonomous Agents*, 1997.

[14] M. Balabanovic and Y. Shoham, "Combining content-based and collaborative recommendation," *Communications of the ACM*, March 1997.

[15] P. Maes, "Agents that reduce work and information overload," in *Software Agents*, J. Bradshaw, Ed. MIT Press, Cambridge, MA, 1997.

[16] J. Yang, P. Pai, V. Honavar, and L. Miller, "Mobile intelligent agents for document classification and retrieval: A machine learning approach," in *14th European Meeting on Cybernetics and Systems Research. Symposium on Agent Theory to Agent Implementation*, Vienna, Austria, 1998.

[17] G. Salton, *Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, Reading, Massachusetts, 1989.

[18] R. Korfhage, *Information Storage and Retrieval*, Wiley, New York, 1997.