Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2000 Proceedings

Americas Conference on Information Systems (AMCIS)

2000

A Multi-Agent Framework for Web Based Information Retrieval and Filtering

Hamid Nemati
University of North Carolina at Greensboro, nemati@uncg.edu

Marc Boumedine Montaner
ITESM-CCM, mboumedi@campus.ccm.itesm.mx

Minghe Sun
University of Texas at San Antonio, msun@utsa.edu

A Multi-Agent Framework for Web Based Information Retrieval and Filtering

Hamid Nemati, The University of North Carolina at Greensboro, ISOM Department, nemati@uncg.edu
Marc Boumedine Montaner, ITESM-CCM, Computer Science Department,
mboumedi@campus.ccm.itesm.mx
Minghe Sun, The University of Texas at San Antonio, Division of Management and Marketing,
msun@utsa.edu

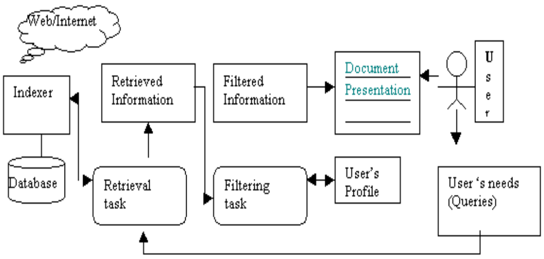
Abstract

Searching for information on the Web is a time consuming task. To help users and to speed up searching for relevant documents efficient retrieval and filtering techniques are needed. To increase the efficiency of information retrieval and filtering tasks, intelligent agents have been widely studied and deployed. In this paper, we present a general agent framework for retrieving and filtering relevant/irrelevant documents.

Introduction

Searching for information on the Web is a time consuming task. This task becomes more tedious due to the exponential growth of information available on the Internet. The information retrieval and filtering problems consist of selecting from a collection of documents those that are the most relevant to a user's profile or query and pruning those that are not relevant. This collection of documents may be found in a corporate database, a corporate Intranet or the World Wide Web. Either case, information retrieval and filtering tasks can be divided into four main steps: 1) Indexing, 2) Query formulation, 3) Retrieving and Filtering and 4) Feedback. (See Figure 1).

Figure 1: Information Retrieval and Filtering tasks



To help users and to speed up searching for relevant documents efficient retrieval and filtering techniques are needed. This task can be automated according to the user profile and queries. Traditional methods for text retrieval such as full text scanning (Boyer and Moore, 1977), inversion (Salton and McGill, 1983), signature files (Files and Huskey, 1969) and clustering (Croft 1980) does not satisfy user's needs for Web searching. Most of these techniques use only a small portion of document content as the basis for classifying and filtering the collection of documents. Other approaches use parsing, syntactic information and natural language processing techniques (Lewis 1993). The main focus of information retrieval based on natural language is to match the semantic content of queries with the semantic content of documents. Although these techniques have been applied with some success on narrow domain such as large Text Retrieval Conference (TREC) corpus they have limited performance on a broad domain (Strzalowski 1994).

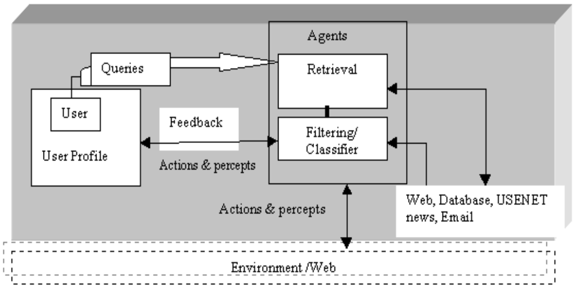
To assist users with browsing, retrieving and filtering information efficiently intelligent agents are widely studied and deployed. These agents use statistics techniques as well as machine learning techniques such as neural networks, symbolic learning and genetic algorithms (Quinlan 1979). The main advantage of these methods is that they have the abilities to acquire knowledge, recognize pattern and self-adapt automatically taking into account examples, experiences or training data. These techniques when combined with the traditional models provide an agent with a great potential for enhanced performance (Wilkinson 1991).
In this paper, we present a general agent framework for retrieving and filtering relevant/irrelevant documents. The main idea of the paper comes from the following two observations: 1) classical techniques for information retrieval and filtering are fast but not accurate because the do not take into account user's feedbacks and they can not learn. 2) In order to obtain better results we need to use learning techniques. However, these techniques are time consuming and can not be applied in real-time. So, in order to tackle these problems (processing time vs. accuracy), we need to design an architecture that combines both advantages. Since the behavior of the agent should vary according to its learning task, user's needs and feedback as well as the environmental constraints (real-time or batch processes) a two levels-learning architecture has been proposed. This design is adaptive because it allows for combining fast learning and slow learning information retrieval/filtering approaches. In addition, the proposed architecture is flexible and reusable by extending intelligent agent behaviors such as learning capabilities, communication with the users, communication with other agents, mobility.

This architecture uses classical techniques in real-time (level 0) while in the background a learning process acquires knowledge during user's interactions with the system (feedbacks). We can represent graphically the knowledge as a state space where each node represent the current knowledge and arc represent the learning process which allows to update the knowledge obtained from the user interactions with the system. The real time technique (level 0) uses the current knowledge (leaf) i.e. the most updated information to run either retrieval or filtering processes while the batch technique (level 1) is used to generate a new knowledge state in the background. This new state will be used for the next query by the real time technique.

Basic issues of our Agent System

Based of the basic issues, we propose the following framework for developing our agent system (see Figure 2). The proposed architecture is based of the following components: The user interface, the user profile, the retrieval and filter agent system. The user interface permits the user to express his needs (queries) and describes his initial profile that are provided to the agent system. The user profile describes the user's preferences defined or learned during the previous searches.

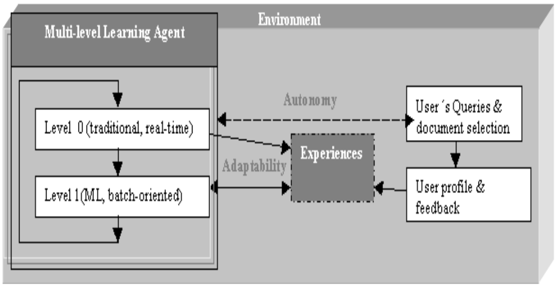
Figure 2. Agent System Components



The agent system is composed of two agents: the retrieval agent component and the filtering agent component. The retrieval component collects document sources from databases (structured information) or the Web (non-structured information). Besides gathering information from information sources, it is in charge of displaying the most relevant information first according to some classification models. The initial query is considered as an ideal relevant document using the vector space model (Salton and McGill 1983) as well as the document sources. A similarity measure between the query vector and each document vector of the source (database or Web) is derived. Documents with the highest similarities are displayed first. Based on the user's relevance feedback and the user's query, it is possible to build a learning model by modifying the vector representing the ideal document (Salton 1983).

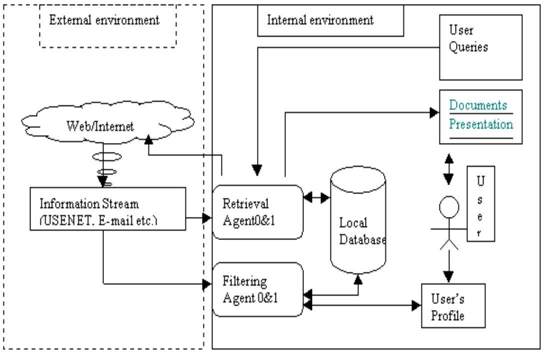
The filtering component has a dual functionality. It can be used for selecting information stream relevant to the user according to his profile (by discarding irrelevant information), or can operate in conjunction with the retrieval component in order to discard the lowest ranked document taking into account the specific user's query. The choices for the retrieval and filtering approaches in order to implement the agents features such as learning, adaptability and autonomy will be discussed later on.

Figure 3. IREFIA Architecture


Environment
Multi-level Learning Agent
Autonomy
Level 0 (traditional, real-time)
Experiences
Adaptability
Level 1(ML, batch-oriented)
User´s Queries & document selection
User profile & feedback

The goal of the filtering is different from the retrieval agent. Its task is to generate an hypothesis about the user's needs based monitoring the user actions on the environment. Using this hypothesis, the agent will discard the documents that do not the verify the hypothesis (documents classified as irrelevant) and present the ones are the most relevant (according to the hypothesis). This process is performed iteratively by refining the hypothesis, observing the user behavior on the documents that have been retrieved. The vector space model can also be used by adapting the representation (Lewis and Smeaton 1993). Numerous approaches have been taken for solving learning problems, and some authors have shown that the different machine learning techniques achieve quite similar performances for solving general problems. In order for our agent system to be useful for solving information retrieval/filtering problem, the following features are highly desirable: response efficiency, accuracy, adaptability and autonomy (see Figure. 4). Since, our agent system is designed to be deployed in an interactive environment, the user is expecting fast responses from the system. Thus, the computation time is also a crucial feature for our system. The accuracy is measured in terms of three statistics: ***precision***, **recall** and ***fallout***. Recall is the percentage of relevant documents in the data sources that are retrieved/filtered. Precision is the percentage of documents retrieved that are relevant. Fallout is the percentage of the non-relevant documents that are selected by the filtering agent. The learning models are not our main concern for designed our agent system since a set of models, provided as a library (Java API) can be chosen according to the specific usage, application and interests of the user. The models are based on statistics, symbols, neural networks, genetic algorithms or hybrids and are largely described in the literature. In the following section we show how the model can be incorporated in our system.

Figure 4: Information Retrieval and Filtering Information IREFIA agent system

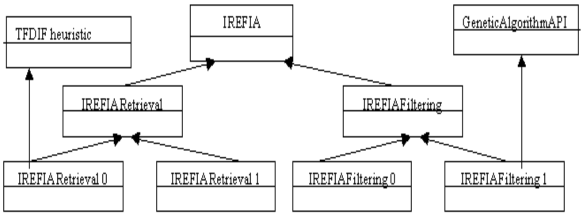


In order to implement the previous characteristics the Information REtrieval and Filtering Agent (IREFIA) has been designed (see Figure 3). The main basic idea is to build an architecture that couples fast real time information retrieval/filtering techniques (level 0) and slow learning techniques (level 1). This architecture is based on a generic two-levels (level 0&1) learning scheme. At level 0, the agent uses a non-learning technique to access to data sources collected and stored in a local data base by a slow learning agent (level 1). The level 1 agent is running as a batch process and its main task is to constantly improve its representation knowledge about the user's query (retrieval agent) or the user information needs (filtering agent). The overall system is depicted on Figure 4. The main objective of this architecture is to makes a trade off computational time for accuracy of results. A similar concept can be accomplished using an contract algorithm (Dean 1988). This algorithm can be suspended at anytime and be able to return an good enough answer. In addition and the algorithm should improve its performance over time. This process is iterative and dynamic. The documents that are filtered and consequently the documents that are presented to the user should be more accurate with the time.

Design goals and implementation of the system

The system has been designed to meet the following goals: adaptativity, evolutivity, reusability, portability, and extensibility. Figure 5 shows an example of class diagram for system. The root class is based of the IREFIA agent which is reused to build the Retrieval (IREFIARetrieval) and the Filtering agent (IREFIAFiltering). In turns, retrieval agent of level 0 can be derived from a class of level 0 such as the TFDIF heuristic and the superclass IREFIARetrieval class. In the same way, filtering agent of level 1 can be derived from a class of level 1, i.e. a machine learning technique such as Genetic Algorithm (GeneticAlgorithAPI) and the superclass IREFIARetrieval class. This class diagram can be extended to incorporate new techniques as well as more levels.

Figure 5. Class Structure for IREFIA Implementation



References

Boyer R. S:, and Moore J.S. A fast string searching algorithm. CACM, 20(10), pp. 762-772, October, 1997.

Croft W. B. A model of cluster searching based on classification. Information Systems, 5, pp. 189-195, 1980.

Files J.R. and Huskey H.D. An information retrieval system based on superimposed coding, Proc. AFIPS FJC, 35, pp. 423-432, 1969.

Lewis D. and Smeaton A., Workshop on: Use Natural Language Processing at TREC, in D.K. Harman, editor, The first Text Retrieval Conference (TREC-1), pp. 365-366. Gaithersburg, MD, March 1993.

Quinlan J.R. Discovering rules by induction from large collection of examples. In Expert Systems in the Micro-electronic Age, pp. 168-201, Michie D. , Editor, Edinburgh University Press, Edinburgh, Scotland, 1979.

Salton G. and McGill. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.

Strzalowski T. and Jose Perez Carballo, Recent Developments in natural language text retrieval, Workshop on Use Natural Language Processing at TREC, in D.K. Harman, editor, The first Text Retrieval Conference (TREC-2), pp. 123-136. Gaithersburg, MD, March 1994

Wilkinson R. and Hingston. Using the Cosine measure in neural network for document retrieval. In Proceedings of the 14th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, pp. 202-210, Chicago, IL, October, 1991.