http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (1): 117-122, 2007
ISSN 1812-5638

Research and Filtering Informations from Internet with Agents

Sid Ahmed Rahal and Djamila Mokeddem
Laboratoire Bases de Données et, Intelligence Artificielle Université des Sciences et de la Technologie, Mohamed Boudiaf, Oran (USTO) Algerie, BP 1505 El M'naouer Oran Algérie

Abstract: This research presents a conceptual framework of multiagent system destined to information-rich environment such as the Internet. The system has three types of agents: Interface Agent, Execution agent and Information Agents. We particularly describe the Interface Agent Filter-Agent that interact with the user receiving user specifications and delivering results according to user preferences (user profile). Filter-Agent is an information filter for Internet news groups using a neural prediction model to score articles based on relevance. The filtering process uses the technique called the backward propagation of errors to adjust the connection weights of a neural network with three layers. The elements of network input vector are not simply the occurrences of terms in documents, but we use the TFIDF formula. The user can construct his profile in a particular domain, providing a keywords set and selecting a sample of documents which he evaluates with feedback values. The trained network uses the article profile to predict expected usefulness as represented by the feedback value. Documents are sorted in descending order of the output value computed.

Key words: Multiagent system, interface agent, information filtering, neurone network, TFIDF

INTRODUCTION

Information-rich environments are the modern, open and large scale environments with autonomous heterogeneous information resources. They are a major target for systems based on agents and multiagent systems (MAS) technologies. Workers involved in agent research have offered a variety of definitions. There is no real agreement even on the core question of exactly what is an agent. However, we believe that most researchers would find themselves in broad agreement with the following definition (Jennings and Wooldridge, 1998). Intelligent Agent is a software-based computer system that enjoys the following properties: Autonomy, social, ability, reactivity, pro-activeness. The presence of all the attributes in a single software entity provides the power of the agent paradigm and distinguishes agent systems from related software paradigms such as object oriented systems, actors, distributed systems and expert systems.

A Multiagent System (MAS) can be defined as loosely coupled network of problem solvers that work together to solve problems that are beyond the individual capabilities or knowledge of each problem solver (Bensaid, 1998; Feber, 1995). Some reasons for increasing interest in MAS research include:

- The ability to provide robutness and efficiency;
- The ability to allow inter-operation of existing legacy systems
- The ability to solve problems in which data, expertise or control is distributed.

MULTIAGENT SYSTEMS IN INFORMATION RICH ENVIRONMENTS

Internet is an example of information-rich environments where information is becoming increasingly more difficult for a person or machine system to collect, filter, evaluate and use in problem solving (Sycara and Zang, 98).

The notion of intelligent agents and multiagent systems has been proposed to address this challenge. A survey of several projects in this target: RETSINA (Sycara, 1998), NETSA,UMDL (Côté, 1999), CARNOT (Huhns and Singh, 1998), MACRON (Decker *et al*., 1995), BASAR (Thomas, 1997) has showed the use of three agent types:

- Interface agents tied closely to an individual human's goals;
- Execution agents (or task agents) involved in the process associated witm problem solving tasks;
- Information agents that are closely tied to a source or sources data.

Corresponding Author: Sid Ahmed Rahal, Laboratoire Bases de Données et, Intelligence Artificielle, Université des Sciences et de la Technologie Mohamed Boudiaf, Oran (USTO) Algerie, BP 1505 El M'naouer Oran Algérie Tel: 21341465686

It is a sign of their maturity that this systems are beginning to evolve a standard set of agent types. They are developed generaly also with standard tools such as the communication language between agents KQML (Finin *et al*., 1994) and Java Language.

PRESENTATION OF THE SYSTEM

In our project, we propose a multiagent system that includes likewise the three levels concept. We focus on the Interface Agent FilterAgent whose main task is information filtering to alleviate the user's cognitive overload (Fig. 1).

Interface agent: Interface Agent interacts with the user receiving user specifications and delivering results. The main functions of FilterAgent include:

- Collecting relevant information from the user to initiate a task;
- Receiving the user request and transforming it in KQML message that it sends to Execution Agent;
- Constructing the user profile;
- Presenting relevant information including results and explanations.

Execution agent: The Execution Agent have knowledge of the task domain. This knowledge is represented by a rule production system for formulating problem solving plans and carrying them out through querying and exchanging information with other software agents. The execution Agent performs the following tasks:

- Receives the request send by the Interface Agent.
- Constructs the Execution Plan of the request using the base knowledge. The generated plan is the shape of executing actions.
- Receives results from Information Agents and sends proposed solution to Interface Agent
- Performs some actions that does not need other agents, such as constructing digital library of collected documents.

Information agent: The Information Agent is responsible to information retrieval in different information sources that can be a local database (access by SQL language) or

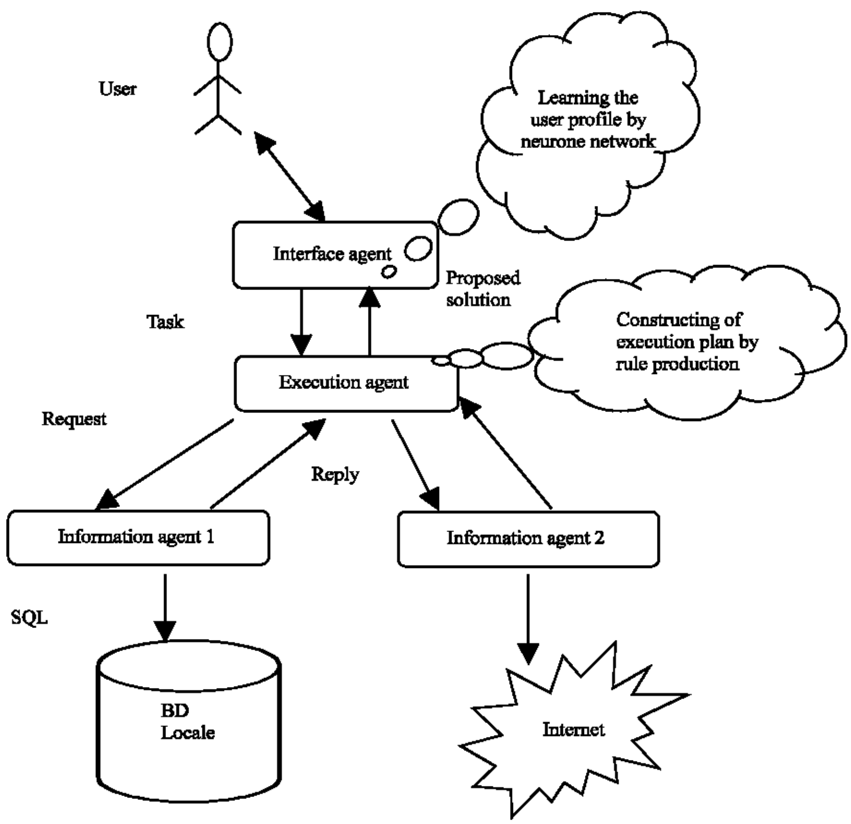

Fig. 1: The conceptual model of the system

the internet network (access by different protocols such as NNTP for news groups).

The execution cycle of Information Agent is:

- Receiving the request from Execution Agent;
- Executing the information request from the appropriate Information Resource.

The agents in our system communicate using KQML

THE CONCEPTUAL MODEL OF INTERFACE AGENT FILTERAGENT

The multiagent system that we are proposing can be used in documents filtering for Internet news group. The Interface Agent performs the task of profile constructing using backward propagation technique (Bigus and Bigus, 1998) and TFIDF formula (Van Risbergen, 1979). The documents retrieved in news group will be filtered according to the user profile.

Information filtering: Information filtering systems are typically designed to sort through large volumes of dynamically generated information and present the user with sources of information that are likely to satisfy his or her information requirement (Lieberman *et al*., 1995). The objective is to automate the process of examining documents by computing comparisons between the representation of the information need (the profile) and the representation of the documents. The Fig. 2 shows the representation and comparison process implemented by these systems (Oard and Marchionini, 1996; Wooldridge and Jennings, 1995).

In an ideal text filtering: C (P (Information Need), T(Document)) = J (Information Need, Document)
∀ (Information need) Є I, ∀ (Document) Є D

The traditional models destined for information retrieving (and filtering) are search engines that operate on bug database (Muller, 1999). In these systems it is difficult to find the wanted information because a user can't accurately express what he wants and search engines don't adapt their search strategies according to different users. Moreover, the problem is exacerbated because the information sources have high noise, i.e., most of the pages are irrelevant to a particular user's interests. Intelligent software agents are being developed to deal with these issues (Bjurn, 1997).

Information filtering methods: Every approach to information filtering (text filtering) has four basic components (Oard and Marchionini, 1996):

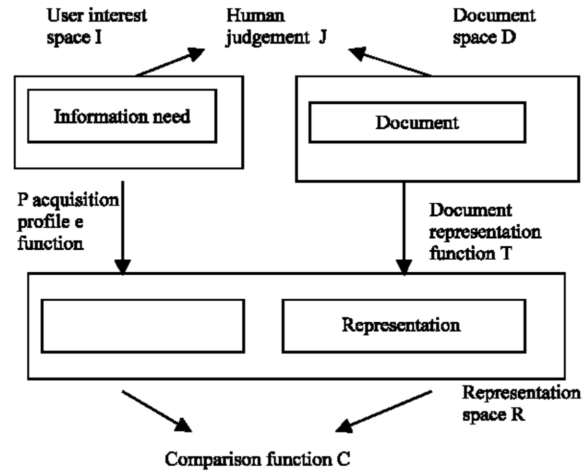

Fig. 2: Information filtering

- Some technique for representing the documents,
- Some technique for representing the information need (i.e., profile construction),
- Some way of comparing the profiles with the document representations,
- Some way of using the results of that comparison.

In (Delgado, 2000) Information filtering approaches are classified in three categories:

- Content-based filters
- Social-based filters
- Event-based filters

Content-based filters: In these systems, it is common to see resources and users profiles being represented under the same model. After the profile has been obtained, it is then compared with the instances being filtering using similarity measure that evaluates relevance. There are several techniques in this approach such as (Sycara, 1998):

- Bayesian classifier
- Nearest Neighbour
- Decision Trees
- Neural Networks
- Vector Space model TFIDF

Social-based filters: These systems share users' impression over items in order to produce predictions on the user's previously capturated rates and the rating over the same item of other similar users in the system.

Event-based filters: This technique consists in tracking and following the surfing habits of people in the web. The agent Letizia for example represents typically browsing

Table 1 : The training data of neural network

Terms Wi		W_1	W_2	......	W_n	Feedback value (desired)
	D_1	$\frac{TFIDF_{W1,D1}}{Max(TFIDF_{W1,D1})}$	$\frac{TFIDF_{W2,D1}}{Max(TFIDF_{W2,D1})}$	....	$\frac{TFIDF_{Wn,D1}}{Max(TFIDF_{Wn,D1})}$	VF (D_1)
Documents selected by the user	D_2			....		VF(D_2)
	. .					
	D_k	$\frac{TFIDF_{W1,Dk}}{Max(TFIDF_{W1,Dk})}$	$\frac{TFIDF_{W2,Dk}}{Max(TFIDF_{W2,Dk})}$	....	$\frac{TFIDF_{Wn,Dk}}{Max(TFIDF_{Wn,Dk})}$	VF(D_k)

Table 2: Attribution of feedback values to documents

Article's relevance	Interesting	Midly interesting	Neutral	Not very useful	Useless
Feedback value	1.0	0.75	0.5	0.25	0.0

process as follows: to examine the current document, decide which links to follow, or to return a document previously encountered. The different events have different weights in the decision process.

Description of user interests: In order to initiate the user profile, the agent FilterAgent asks the user to specify keywords set characterizing his interests and the news group address. The user will can select a sample of documents, see their content and estimate the quality of every document providing feedback values.

The calculation of weights keywords using TFIDF formula: The occurrence of items is in common use in information filtering as a criteria of document 's relevance. Experience has shown that the representation effectiveness can be improved substantially by transforming term frequency in ways which amplify the influence of words which occur often in a document but rarely in the whole collection (Oard and Marchionini, 1996). This measure known as Term Frequency Inverse Document Frequency weighting (TFIDF), assigns term w in document d a value TFIDF:

$$TFIDF = TF_{W,d} \,.\, IDF_{W,d}$$

$$IDF_{W,d} = \log_2\left(\frac{N}{DF_W}\right)+1$$

W : term
d : document
$TF_{W,d}$: occurrence of term w in document d
N : number of documents
DF_W : number of documents with term w

In the TFIDF technique, both the profile and the documents are represented as vectors and a similarity measure is calculated (e.g., the angle between the two vectors). Our system attempts to benefit by this technique in the part of weights computing. The training set is represented by a vector whose elements are the weights TFIDF of terms in the document.

Learning user profile by back propagation technique: Back propagation features a feed forward connection topology, meaning that flows through the network in a single direction, and uses a technique called the backward propagation of errors to adjust the connection weights (Champciaux, 1997).

Our interface Agent FilterAgent uses a three layers neural network to learn user profile. The input data is presented by a vector whose elements are TFIDF weights of terms in the documents selected by the user (Table 1) and the feedback value that represents the relevance's document according the user (Table 2). The training data is normalized (transforms to a value in the range of 0.0 to 1.0).

Training algorithm of neural network (Fig. 3):

- Selection of documents and attribution of feedback values by the user,
- Computation of TFIDF weights, normalization of the training vector,
- Start the training process.
- Initialize the control parameters (weights and thresholds by random values; Learn rate r, momentum, and number of iterations by corresponding values);
- While number of iterations is not attained
- For every document selected do:

$$C_j = \frac{1}{1+e^{\theta j-\sum_{i=1}^{nb} Ei\ \ Pij}}$$

Cj : Value of hidden neurone j
θj : Value of threshold of hidden neurone j
nb : Number of input neurones
Ei : Value of input neurone i
Pij : Value of weight from E to C

Compute values of output layer;

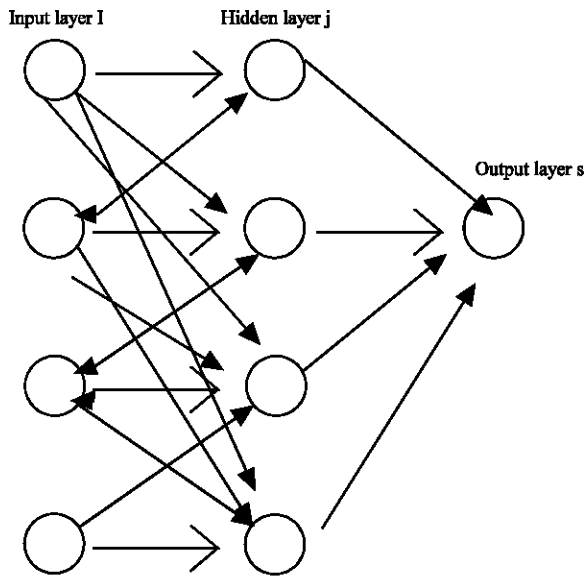

Fig. 3: The neural network for constructing user profile

$$S = \frac{1}{1 + e^{\theta j - \sum_{j=1}^{nb} Ci \ \ Pij}}$$

S : Value of output neurone
θs : Value of threshold of output neurone
Pjs : Value of weight from Cj to S

Compute the error of output layer

$$\Delta S = S.\ (1\text{-}S).\ (VF - S)$$

Δ S : Error signal of output layer
VF : Feedback value assigned by the user

Compute the error of hidden layer

$$\Delta J = C_j\ .\ (1 - C_j).\ (P_{js}\ .\ \Delta S)$$

ΔJ : Error signal of hidden layer j

Compute the weights and thresholds changes from the prior step:

$$\Delta p_{ij}(n{+}1) = \alpha.\ \Delta P_{ij}(n) + r.\ \Delta J.\ E_i$$

$$\Delta p_{js}(n{+}1) = \alpha.\ \Delta P_{js}(n) + r.\ \Delta S.\ C_j$$

$$\Delta\ \theta j(n{+}1) = \alpha.\ \Delta\ \theta j(n) + r.\ \Delta J$$

$$\Delta\ \theta s(n{+}1) = \alpha.\ \Delta\ \theta s(n) + r.\ \Delta S$$

n : Step number
α : Momentum
r : Learn rate parameter

Adjust weights and thresholds:

$$P_{ij}(n+1) = P_{ij}(n) + \Delta P_{ij}$$

$$P_{js}(n+1) = P_{js}(n) + \Delta Pjs$$

$$\theta j\ (n+1) = \theta j\ (n) + \Delta\theta j$$

$$\theta s\ (n+1) = \theta s\ (n) + \Delta\theta s$$

θj : Threshold value for output neurone
θs : Threshold value for hidden neurone j

The neural network is trained for several passes (2500 in our implementation) over the data.

IMPLEMENTATION

A first prototype of FilterAgent has been realised in Java Language (JDK1.2) representing the documents by term frequency. The future work on FilterAgent has the following directions:

- Representing the user profile and documents of news server by vectors whose elements are the weights TFIDF and feedback values.
- Automatic detection of changes in news servers.
- Integrating FilterAgent in the multiagent system previously described.

CONCLUSIONS

In this article, we have presented a three layer multiagent architecture for information rich-environments such as the Internet.In order to have a modular open system, we have opt for distributing the tasks over three agent types:

- Interface Agent whose the primordial task is constructing the user profile.
- Execution Agent which has the ‘reasoning’ capability over his knowledge base in order to construct the execution plan.
- Information Agents which are responsible of information retrieval in different information sources that can be a local database.

In a first step, we have focused on the Interface Agent FilterAgent: It is able to filtering documents from news servers, according to the user profile previously constructed using two approaches:

- A neural network with three layers, trained by backward propagation technique,
- The TFIDF formula.

The user can construct his profile in a particular domain, providing a keywords set and selecting a sample of documents which he evaluates with feedback values. The trained network uses the article profile to predict expected usefulness as represented by the feedback value. Documents are sorted in descending order of the output value computed. The multiagent system is implemented in Java Language.

REFERENCES

Bensaid, N., 1998. Contribution to the realisation of multi-agent hierarchique architecture, Thesis of doctorat, University of Lille.

Bigus, J.P. and J. Bigus, 1998. Constructing Intelligent Agents with Java: A programmer's Guide to Smarter Applications, Wiley Computer Publishing.

Bjurn, H., 1997. Intelligent software agent on the Internet: An inventory of currently offered functionality in the information society and a prediction of (near) future developments. M.Sc. Tthesis, Tiburg University, Tiburg, TheNetherlands.

Champciaux, L., 1997. Learning and declarative modelisation, Thesis of Doctorat, Ecole des Mines de Nantes, pp: 79-89.

Côte, M., 1999. NETSA: A multiagent architecture and his application to financial services, Mémory of Maître ès Sciences (M.Sc.), Faculty of high studies of University de Laval, Canada.

Decker, K., V. Lesser, M.V. Nagendra, Prasad and T. Wagner, 1995. MACRON: An Architecture for Multiagent Cooperative Information Gathering, Technical Report Department of Computer Science University Massachusets.

Delgado, R., 2000. Agent based Information Filtering And Recommender Systems on the Internet, M.Sc. Thesis, Nagoya Institute of Technology.

Feber, J., 1995. Multi-Agent Systems, Toward collective intelligence, France Inter Edition.

Finin, T., R. Fritzon D. McKay and R. McEntire, 1994. KQML as an agent communication language, Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM '94), ACM Press.

Huhns and M.P. Singh, 1998. Multiagent systems in Information Rich Environments, technical Report, Department of Electrical and Computer Engineering University of South Carolina Columbia, SC 29208, USA.

Jennings, N.R. and M. Wooldridge, 1998. Applications of intelligents agents, Queen Mary and Westfield College, University of London.

Lieberman, H., 1995. Letizia: An agent that assists web browsing, Proceedings of the 1995 International Joint Conference on Artificial Intelligence, Montreal, Canada.

Muller, M.E., 1999. An Intelligent Multiagent Architecture for information Retrieval from Internet, Technical Report. Institute for Semantic Information Processing University of Osnabruck, D-49069 Osnabruck.

Oard and G. Marchionini, 1996. A conceptual Framework for text filtering, technical report, University of Maryland, College Park.

Sycara, K., 1998. In context information management through adaptative collaboration of intelligent agents, Technical Report. The Robotics Institute, Carnegie-Mellon University.

Sycara, K. and D. Zeng, 1998. Multiagent Integration of Information Gathering and Decision Support. In the Proceedings of ECAI.

Thomas, C., 1997. Using agents to improve the usability and usefulness of the www, Technical Report, German National Research Center for Information Technology Germany.

Van Risbergen, 1979. Information Retrieval, Butterworths, 2nd Edn.

Wooldridge, M. and N.R. Jennings, 1995. Intelligent Agents: Theory and practice, the knowledge engineering review, pp: 115-152.