A Web Pages Automatic Filtering System

O. Nouali, A. Saidi, H. Chahrat, A. Krinah, B. Toursel

Abstract— This article describes a Web pages automatic filtering system. It is an open and dynamic system based on multi agents' architecture. This system is built up by a set of agents having each a quite precise filtering task of to carry out (filtering process broken up into several elementary treatments working each one a partial solution). New criteria can be added to the system without stopping its execution or modifying its environment.

We want to show applicability and adaptability of the multi-agents approach to the networks information automatic filtering. In practice, most of existing filtering systems are based on modular conception approaches which are limited to centralized applications which role is to resolve static data flow problems. Web pages filtering systems are characterized by a data flow which varies dynamically.

Keywords— Agent, Distributed Artificial Intelligence, Multi agents System, Web pages filtering.

I. Introduction

THE non stop increasing quantity of available information as electronic format induced of new information access needs. In order to save an invaluable time for the useful information search, the recourse to new tools seems to be inevitable. This need motivate the use of new mediators, between information sources and users, among whom information filtering systems.

These systems, positioned as a "third part" in communication between user and information source, must include methods and knowledge necessary to process, evaluate, filter, reach and extract relevant data, making user's role easier.

This article attempts to show applicability and adaptability of distributed artificial intelligence approach (multi-agents systems), to networks automatic information filtering. The use of a multi agents approach offers the following possibilities [1]-[3]:

- Modeling solutions using intelligent and independent entities, having each one a quite precise task of filtering to carry out and which interact according to co-operation, competition and coexistence modes'. This offers a considerable time saving compared to a sequential algorithmic solution. That also makes system more efficient thanks to intelligence of its entities;

- Building an open and dynamic system, this is of primary importance. Indeed, new criteria can be added dynamically, and the system must be able to integrate these new criteria in order to increase effectiveness, and this, without modifying what already exists. The need for opening is explained by the fact why the number of criteria which will intervene in filtering operation is not known in advance and new criteria can be constantly added to the system. The system dynamic owing to the fact that the criteria can be create and is destroyed dynamically;

- To conceive a distributed and reactive solution to the problem. The integration of a reactive environment allows the completude solutions partial worked out for construction of a solution total [4]-[6].

II. System Global Architecture

Our system is made up mainly of several agents types of which can be grouped under two main categories: permanent agents and non permanent agents.

The permanent agents are agents which after their creation always reside in the system.

The non permanent agents are agents which are create for a particular need and they will be destroyed at the end of their mission. In other words, the non permanent agents do not last, in the system.

The global architecture of the system is made up of following principal sub-systems (Fig. 1):

- "User interface";
- Documents recovery sub-system;
- Documents indexing sub-system;
- Similarity calculation sub-system;
- Control sub-system.

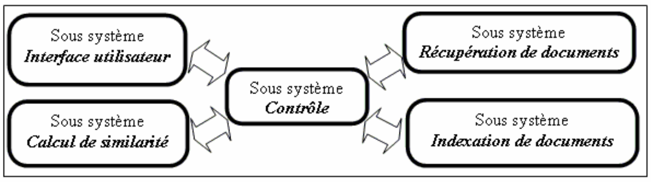

Fig. 1 System global architecture

A. User interface

This interface allows users to register them into the system, to express their interest by adding new profiles, to make updates or to remove them. It also makes it possible to

O. Nouali, Laboratoire Intelligence Artificielle, CE.R.I.S.T, Rue des 3 frères Aïssiou, Ben Aknoun, Alger, Algérie, Fax : (213) (0)21 912126 - Tél. : (213) (0) 21 916211 (e-mail : onouali@yahoo.fr).

A. Saidi, Laboratoire Intelligence Artificielle, CE.R.I.S.T, Rue des 3 frères Aïssiou, Ben Aknoun, Alger, Algérie.

H. Chahrat, Laboratoire Intelligence Artificielle, CE.R.I.S.T, Rue des 3 frères Aïssiou, Ben Aknoun, Alger, Algérie.

A. Krinah, Laboratoire Intelligence Artificielle, CE.R.I.S.T, Rue des 3 frères Aïssiou, Ben Aknoun, Alger, Algérie.

B. Toursel, LIFL, UPRESA CNRS 8022, University of Sciences and Technologies of Lille, 59655 Villeneuve d'Ascq Cedex, France, Fax : +33 (0)32877 8537 tel : +33 (0)32033 4733, (e-mail: toursel@lifl.fr).

manage the system learning while being based on user's judgments on the documents brought by the preceding filtering operations. It includes two types of agents, namely *Profiles agents* and *Profiles manager agent*.

a.1 Profile Agent

With the creation of a new profile by the user, the system associates him a proper permanent agent called *Profile agent*. Its role is of:

- Making a pre-filtering on the arrived documents which consists in eliminating, in first, the documents which have characteristics differing from those awaited by the user;
- Then, launch the filtering operation of documents retained after pre-filtering.

a.2 Profile manager Agent

It is a permanent agent which holds the identifiers of all existing profiles agents. It provides to the other agents (ex: Document agent), if requested, all necessary identification information.

A user profile contains a set of information, among which:

- A list of key word reflecting the requirements of the user. This last can assign an importance weight to each one of them;
- A list of advanced criteria representing the characteristics of the awaited documents (ex: author's name, language...);
- The research position in the document (ex: document title, body, abstract...).

We distinguish two interaction protocols:

- New profile of arrival protocol: on arrival of a new profile, the system associates him a proper agent called profile agent. This last takes note of all the information entered by user. Then, it informs an administrative agent about by sending him its address. After address reception confirmed, the administrative agent saves it in its knowledge base. This treatment allows documents agents to communicate with the existing profiles agents before launching filtering operation.
- Protocol of pre-filtering: on arrival of a new document, the system creates three agents of which a *Document* agent. This last starts by sending a message to the *manager* agent asking him all existing agents Profiles addresses. After receipting them, *Document* agent sends the characteristics, of the corresponding document (language, author name...) to Profiles agents from which it comes to receive the addresses. Each time a Profile agent receives these characteristics it compares them with the characteristics chosen by the user. If the characteristics are identical then the document in question will be retained by the pre-filtering operation. In this case, the Profile agent sends a positive acknowledge to the transmitting Document agent. After what, it launches association protocol and indexing protocol to carry out filtering itself. If not, it sends a negative acknowledge.

B. *Documents Recovery*

It is an agent which has as a role to recover new documents, then, to send them, possibly, to the users who their carry an interest.

C. *Documents Indexing*

It is an agents society which allows to analyze and to determine the relevant entities of a given document to send them to the similarity calculation subsystem. It includes six types of different agents:

- Permanent agent, **Tokeniser**, which role is to cut out the text in entities (Tokens) and to provide them to the Stoplist agent;
- Permanent agent, **Stoplist**, which have as a role to eliminate, by using a stop-list, general terms such as the pronouns, prepositions and certain adverbs and adjectives;
- Permanent agent, **Lemmatiseur**, which task consists of bringing back transmitted entities from Stoplist agent to their root. For example, "to filter", "filtering", "filter" are transformed into only one entity which is "to filter";
- Permanent agent, **Sauvegarde1**, which has as a role to save the result of an indexing operation in a file associated with the treated document. Each time this agent receives a keyword from Lemmatiseur agent, it saves it with its sequence number in the file which corresponding to the treated document. A keyword can exist several times in this file but with different sequence numbers;
- Permanent agent, **Sauvegarde2**, which has as a role to save the keywords resulting from Lemmatiseur agent with their number of occurrences in the text. This result is used in the process of learning which consists in improving the user's profiles;
- **Distributeur** agent whose role consists in sending to the *Tokens* agents keywords resulting from the indexing operation.

The protocol of indexing is described as follows: when a Profile agent sends to a Document agent a positive acknowledgement, this last will be accompanied by information indicating the document part to be carried out (title, body…).

Document agent sends to Tokenize agent the part of the document to be indexed. This last determines the various entities and sends them to the Stoplist agent. After what, stoplist agent checks if this entity belongs to the stop list. If it belongs, the entity is ignored. If not, it is sending to the Lemmatiseur agent who calculates its root and sends it to the Sauvegarde1 agent, the Sauvegarde2 agent and the Distributeur agent.

The entities received by the Sauvegarde1 agent are saved in a file used in certain case for occurrences number calculation.

The list of entities sent to the Sauvegarde2 agent allows building a second file used to improve the user profiles

The lemmatization disadvantage is that it transforms, sometimes, a word in another which does not have any sense. For example applying the rule which replaces the words which have the following form "a character string + age" by words of the form "a character string + er" to the word "filterage" will not cause problem since the word "filtrer" is correct, but if one applies this rule to the word "plage" one obtains the word "pler" which does not have any sense. This problem can be regulated by a traditional dictionary. But unfortunately the size of this last is very large. To cure this problem, we used a file which contains, only, the principal entries of a traditional dictionary instead of all its contents. Thus, if the term resulting from the lemmatization does not correspond in any term of this file then the word obtained does not have any sense. It must be rejected. And one tries to make another lemmatization with other possibilities.

D. *Similarity Calculation*

It is an agents society which allows calculating the similarity degree between profiles and new documents. It includes two types of agents:

- *Token* agent: to each keywords of a given document, is associated a non permanent agent called **Token** agent. It makes it possible to calculate the number of occurrences of this token in the document.

We can describe the protocol as follows: calculation of a profile keywords occurrence number in a document is carried out by Token agents. So that the latter can achieve their work, they must be initially registered at the Distributeur agent so that this last can inform it of the indexing operation resulting entities. Unfortunately, the Token agents cannot be registered directly at the Distributeur agent. By what the latter and the Token agents are created in a parallel way, consequently they do not know each other. So, we use another type of agents known at the same time by both Token and Distributeur agent. This agent is called *Enregistreur* agent. Two cases can arise:

- The first, it is the case where the Distributeur agent is creates before the Tokens agents. In this case, just after its creation, the agent distributor informs the Enregistreur agent of its address. Thereafter if the Tokens agents created want to be registered at the Distributeur agent, then they have to only enter in communication with the Enregistreur agent;
- The second case, it is the case where certain Tokens agents are create before the Distributeur agent. In this case, when an agent Token is created, it is registered initially at the Enregistreur agent. Thereafter, when the Enregistreur agent takes note of the creation of the Distributeur agent, it sends the addresses of the Tokens agents to him asking to be registered there.

Each time that a Distributeur agent receives an entity resulting from the indexing operation; it returns it to all the Token agents which were already registered there. Every Token agent compares the entities received with the associated keyword. If an entity is identical to the treated keyword, the number of occurrences of this last will be updated.

Another association protocol is used: when a Profile agent notes that a document has the same characteristics as those awaited by the user, it first creates a *Similarity agent*. Then, it sends a message to the *Manager* agent, containing corresponding profile terms, and address of the Similarity agent that it has just created. While receipting such message, the *Manager* agent checks for each received keyword if corresponding Token agent already created or not (after been requested by another Profile agent). In the negative case, the Manager agent creates a new Token agent and sends it the Similarity agent address (corresponding to the profile agent having transmitted the keyword). Then, the Manager agent saves, in an associated file, the received keywords and addresses of Tokens agents created. If the Manager agent receives, for a second time (from another Profile agent), a given keyword, it does not recreate him a second Token agent. It only recovers its address and sends him the address of the 2nd Similarity agent. Thus if several profiles contain the same keyword for a given document, will be created only one Token agent to treat this keyword. That permits to avoid redundancy.

- Similarity agent whose role consists in calculating the degree of similarity between a document and a profile.

The protocol is described as follows: the degree of similarity between a profile (P) and a document (D) is calculated by a specific Similarity agent to this couple (P, D). Calculation based on internal product between weights of terms in the profile and the weights of these terms in the document. After been created, the Similarity agent will be informed of all keywords used in the similarity calculation, and their weights in the user profile. Once a Token agent had finished occurrences number calculation of the corresponding term, it communicates it to the Similarity agents. After reception it, this one has to transform it into a weight (Bi). If last awaited weight, Similarity agent launches then, similarity degree calculation of couple (P,D) corresponding.

E. *Control*

A permanent agent used to ensure the system initialization when created and a good recovery after a possible breakdown.

III. Evaluation

We carried out tests to measure the filtering performances of the system considering response time, precision, recall and to show importance and of the learning, i.e. how learning operation affects filtering efficiency. We present, in what follows, our filtering system performances in two configuration cases: response time evaluation and a filtering session evaluation.

A. *Response time evaluation*

The user’s result latency is also an evaluation criterion of an automatic system. Thus system response time will be evaluated. We will simulate two machines (mono-processor) on which the agents operate:

- Machine1: sequential execution;
- Machine2: several agents’ simultaneous execution.

TABLE I
ESTIMATED EXECUTION TIME PER TASK

Task	Average execution time (ms)	
Document treatment	Tokenization	0.2
	Stop list	0.3
	Lemmatization	0.35
	save	0.05
	Total indexing time	0.9
Profile Treatment	0.09	
Similarity measure	0.08	
Communication time between two agents	0,02	

Table I presents the various tasks of the system as well as each task average time execution (estimated).

To evaluate time, we carry out several experiments by varying the both profiles and documents number.

TABLE II
SERVICE TIME MEASURED IN MACHINE 1 AND 2

execution time (ms)		1 profile	5 profiles	10 profiles
1 document	Machine1	1.07	5.35	10.7
	Machine2	1.15	1.52	1.87
20 documents	Machine1	21.4	107	214
	Machine2	2.67	10.07	17.7
50 documents	Machine1	53.5	267.5	535
	Machine2	5.07	23.57	41.07

Service time measured in the machine1 increased considerably when profiles and documents number arises, on the other hand, in the machine2 it increased slightly according to the number of agents (documents and profiles). By this experiment, the machine2 (our system) presents a better response time than the machine1 (sequential filtering), the multi-agents system is best adapted to a parallel environment.

B. Filtering session Evaluation

To evaluate a filtering system performance from efficiency point of view, we proceed to different experiments measuring both recall and precision rates. Then, we carry out an assisted learning (or feedback) to measure its efficiency and how it influences the two factors referred above. Indeed, user is invited to deliver his opinion on the behavior of the system (with aim of adapting the system to its needs). Thus, the training acts on the profiles, which consists to add or remove terms.

We consider five profiles; each one is defined by a list of keywords. Results are showed in Table III.

TABLE III
PERFORMANCES OF FILTERING

Profils	Performances (sans feedback)		Performances (avec feedback)	
	Précision	*Rappel*	*Précision*	*Rappel*
Profil 1	0.5	0.6	0.75	0.6
Profil 2	0.66	0.33	0.44	0.66
Profil 3	0.28	0.5	0.6	0.75
Profil 4	0.41	0.62	0.7	0.87
Profil 5	0.57	0.8	0.66	0.8

These experiments show an improvement of recall and precision rates due to regulation and adaptation of the profiles by training feedback technique.

IV. CONCLUSION

A Web pages automatic filtering system is about collecting and sending new documents to the users who being interested. It is dynamic and intelligent based on a multi agents architecture, which enables it to be more autonomous. This architecture makes it possible to remove, gradually, the personal element in the decision-making. Moreover, it allows processing time reduction, a better maintainability, and consequently, a greater capacity of evolution to be adapted to new uses developpement. The present work does not make it possible to treat the problems in all its aspects. If certain treated elements bring us some explanations, it is not unfinished. Many points are to be developed while new ways are to be explored. Each element of the system deserves a more thorough study. For that we will consider some prospects like:

- Enrich the system by filtering so that it can take into account the multi-terms and thesaurus (dictionary of synonyms);
- Improve system so that it can treat other types of documents, such as pdf, Word and XML files;
- Improve system so that it can make a pre filtering on languages, others that the French and English;
- Add other options, for example, inform users of new relevant documents arrival by sending them an e-mail.

REFERENCES

[1] O. Nouali, and B. Toursel, “An agent system for electronic mail filtering,” International Conference on Advances in Computer Science and Technology (ACST 2004), St. Thomas, Virgin Islands, USA, 2004.

[2] N.Jennings, N.Wooldridgr, and K.Sycara, “Application on intelligence agents,” Agent Technology Foundation, aplication and Maekets, Verlog, 1998.

[3] P. Balez, P. Bart, M. Beal, and E. Frigot, “Systèmes multi agents,” Spécialisation SCIA, 24 juin 2002. Available:http://etud.epita.fr:8000/□balez_p/rapports/rapports_SCIA_2002/SystemMultiAgents.pdf.D.

[4] Oulhadj, “les systèmes multi-agents, Etat de l’art,” rapport de mini projet de magister, CDTA, 1996.

[5] H. F. KARA, and HADJ Med Nabil, “Système multi-agents pour le contrôle d’exécution,” Mémoire d'ingénieur, INI, 1997.

[6] Y. Damazeau, “Fonctionnalité émergente dans une société d’agents autonome,” Thèse de doctorat, institut nationale polytechnique de Grenoble, 1998.