AISTA 2004 in Cooperation with the IEEE Computer Society Proceedings / ISBN : 2-9599776-8-8
15-18 November, Kirchberg - Luxembourg

PAWebSearch: An Intelligent Agent for Web Information Retrieval

Rachid Arezki, Pascal Poncelet, Gérard Dray
LGI2P, école des Mines d'Alès
Site EERIE, Parc Scientifique Georges Besse
30035, Cedex 01, Nimes, France
Email: {rachid.arezki, pascal.poncelet, gerard.dray}@ema.fr

David William Pearson
EURISE, Jean Monnet University of Saint-Etienne
23, rue du docteur Michelon
42023 Saint-Etienne, France
Email:david.pearson@univ-st-etienne.fr

***Abstract*— With the strong growth of the World Wide Web and the development of storage device, the amount of available information is so great that finding the right and useful information becomes a very hard task for an end user. In this paper, we introduce *PAWebSearch* a personal agent for Web information retrieval. It is based on dynamic choice of indexing terms depending on the user request but also on his profile. The general idea is to consider that the need of a user depends on his request but also on his knowledge acquired through time on the thematic of his request.**

***Keywords*. User Modeling, Information Retrieval, Software Agents.**

I. Introduction

With the strong growth of the World Wide Web and the development of storage device, the amount of available information is so great that finding the right and useful information becomes a very difficult task. The end user, generally overloaded by information, can't efficiently perceive such information. In order to help the user in his task, search engines available on the Web propose, through requests expressed by user in form of key words, a set of documents. Unfortunately, the quantity of returned results is also very large. Moreover, some relevant documents are often badly ranked and thus rarely consulted. $Blair\&Maron$ showed that the poor performance of IR systems is mainly due to the incapacity of users to formulate adequate requests [1]. Indeed, requests only formulated by key words express badly user information needs. In fact, these needs depend of course on the formulated request but also on the knowledge acquired by the user in his search domain: two users can formulate the same requests for different needs, and the same user for the same request may expect different answers in different periods of time [2]. For example, the results expected by an expert in java language who formulates the request "java course" are different from the results expected by a non expert which formulates the same request. A possible solution of this problem is to take into account the user profile in order to refine the ranks of the results returned by the Web search engines. In other words, the personalized Web information retrieval consists in finding a model able to consider efficiently user interests.

In this paper, we introduce *PAWebSearch* a personal agent for Web information retrieval. It is based on dynamic choice of indexing terms depending on the user request but also on his profile. The general idea is to consider that the need of a user depends on his request but also on his knowledge acquired through time on the thematic of his request.

The article is organized as follows: Section 2 describes the functional architecture of *PAWebSearch* agent. Then, Section 3 describes our approach for modelling and learning user profile. Section 4 presents our information retrieval model. Experimental results are presented in Section 5, we show that our method allows to improve significantly quality of Web information retrieval. Section 6 gives an overview of related work. Finally, section 7 provides some concluding remarks and directions of future research.

II. PAWebSearch Architecture

Personalization of the information retrieval on the Web consists in adapting documents returned by the Web search engines according to user profile. In this framework, we have developed *PAWebSearch* a personal agent for Web information retrieval. It carries out the two following tasks:

1) *Learning user profile*: according to various actions of user and to each consulted document (Web page), *PAWebSearch* automatically updates the user profile.
2) *Personalization of Web information retrieval*: for each information retrieval request carried out via a Web search engine (*Google, Yahoo, ..*), *PAWebSearh* considers user request and results provided by the Web search engine in order to rank these results according to the user profile.

As shown in figure 1, *PAWebSearch* is composed of the three following sub-systems:

1) *A proxy*: is an interface between the user browser, the Web, the user learning sub-system and the filtering sub-system. All Http transactions between browser and the Web pass through the proxy which informs user learning sub-system and filtering sub-system. The results of information retrieval requests, carried out through the Web search engines, are initially sent to the filtering sub-system which is in charge of adapting results according to the user profile.
2) *User learning sub-system*: learns and models user interests according to his actions. Indeed, it supervises

the user's actions by updating his profile each time a document is consulted.

3) *Filtering sub-system*: filters results of information retrieval requests according to the user profile.

The general principle of *PAWebSearch* is as follows: From a request q carried out by a user on a Web search engine, *PAWebSearch* agent recovers all the results through the proxy. Then, an analysis of user profile (user knowledge) is performed in order to obtain a set T of indexing terms, which are constituted by key words of the initial request, enriched by the terms in correlated with these key words. The construction of the indexing terms set T depends both on the user profile and on the user request q. We thus index all documents returned by search engine and request q according to the indexing term set T (documents and requests are represented by vectors). Then, to better adapt to the user's needs, the initial request vector q is transformed into q'. Proposing documents to the user is done by the calculation of similarities between the documents returned by the Web search engine and the request q'.

In the following sections, we present in detail how user profile is represented and how it is considered to improve the quality of the answers of search engines.

III. User profile representation

A user is defined by a tuple $P =< id, G >$ where id stands for an unique user identifier and G is a graph representing documents consulted by the user. The general idea is to analyze the content of the different documents and to store in the graph G co-occurrence frequency between various terms (words) of a document, as well as occurrence frequency of these terms. More precisely, $G =< V, E >$ is a labelled graph such as:

1) $V = \{(t_1, f_{t_1})..(t_n, f_{t_n})\}$ is a set of vertices of G, where each vertex (t_i, f_{t_i}) is represented by a term t_i and its frequency f_{t_i}.
2) $E = \{(t_i, t_j, fco(t_i, t_j))/t_i, t_j \in V\}$ is a set of edges of G, where $fco(t_i, t_j)$ represents co-occurrence frequency between the terms t_i and t_j.

The co-occurrence frequency (or co-frequency) between two terms is defined as the frequency of both terms occurring within a given textual unit. Textual unit can be k words windows, sentences, paragraphs, sections, or whole documents [3][4]. In the framework of our user model, $fco(t_i, t_j)$ represents co-occurrence frequency between terms t_i and t_j in the set of documents consulted by the user.

Thus, the user profile is built through the set of the documents consulted by user. For each new consulted document d, a graph of co-occurrence G_d associated to d is built, according to the following steps:

1) Identification of terms (lexical segmentation),
2) Elimination of the stop words, that is, terms that are not interesting (by using a preset list),
3) Stemming, that is, the reduction of terms to their root,
4) Construction of the graph G_d.

As shown in algorithm 1, for each new consulted document d, a graph G_d is built, then G_d is added to the graph G representing the user profile.

Algorithm 1: User Profile Learning
Input:
consulted document d,
the user profile $p =< id, G >$
Output:
updated user profile $p =< id, G >$
begin
1. construction of the co-occurrence graph G_d
2. **for** *each term* t_i *of* G_d **do**
   if $t_i \in G$ **then**
   $f_{t_i}^G = f_{t_i}^G + f_{t_i}^{G_d}$
   else
   create a new vertex (t_i, f_{t_i}) in the graph G such as
   $f_{t_i}^G = f_{t_i}^{G_d}$
3. **for** *each edge* $(t_i, t_j, fco(t_i, t_j))$ *of* G_d **do**
   $fco_G(t_i, t_j) = fco_G(t_i, t_j) + fco_{G_d}(t_i, t_j)$
end
$fco_G(t_i, t_j)$ stands for the frequency of co-occurrence between terms (t_i, t_j) in the graph G.

IV. Information Retrieval Model

We consider in this section that a request q was sent to a Web search engine, and that we have a set X of returned documents, and let p be a user profile. Our information retrieval model can be presented as a tuple $< X, Q, P, T, s, f >$, where X represents the set of documents (i.e. document collection), Q stands for the set of requests, P is the set of user's profiles, T represents the term set indexing, s is a similarity or distance function and f is the term set construction function. For a given request q and a profile p we have $T = f(p, q)$.

Our motivation is to integrate effectively the user interests in the information retrieval process. Thus, the construction of the indexing term set T is done in a dynamic way and depends both on the user profile p and on the user request q (i.e. $T = f(p, q)$). For each new user request q, a new term set T is rebuilt. After the determination of the indexing term set T, the request q and each document of the collection X are represented by vectors according to the indexing term set T. Then, the initial request vector q is transformed into q'. The transformation of q to q' requires the construction of the profile-request matrix (Sect. 4.A).

A. Indexing Term Set Construction

The choice of the indexing terms takes into account user profile as well as information retrieval request. Our motivation is to choose indexing terms reflecting the knowledge of the user in the domain of his search. As shown by the algorithm 2, the indexing terms are selected among the terms of the user profile which are in co-occurrence with the terms of the initial request.

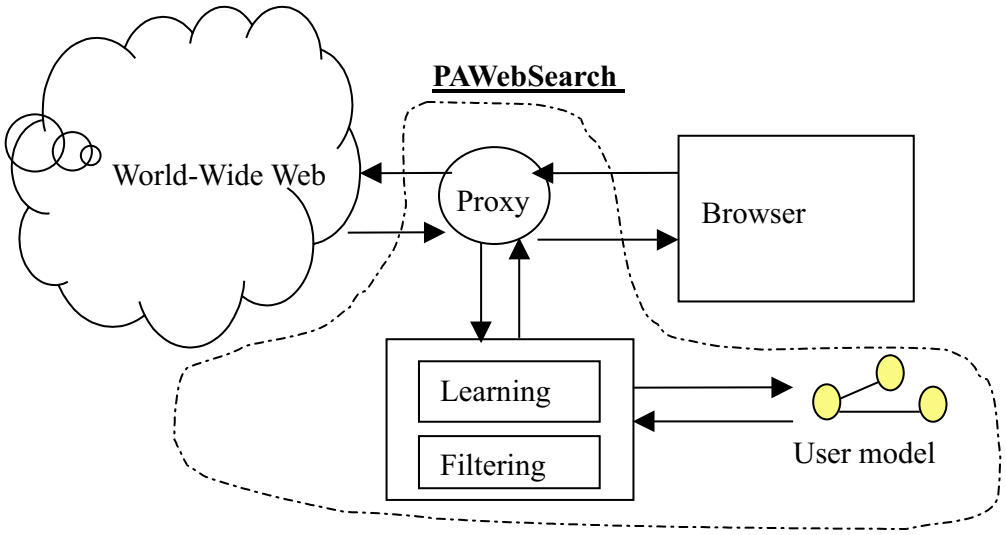

Fig. 1. Functional Architecture

Algorithm 2: Indexing Term Set Construction
Input: user request q,
the user profile $p =< id, G >$
Output: indexing term set T
begin
1. $T \leftarrow$ terms contained in the request q;
2. **for** *each term* t_i *of* q **do**
   for *each term* t_j *of* G *such as* $fco(t_i, t_j) > 0$ **do**
   if $\frac{(fco(t_i,t_j))^2}{f_{t_i} \times f_{t_j}} > \beta$ **then**
   $T = T \cup \{t_j\}$

end
β : constant representing the threshold of term selection.

B. Profile-request matrix

From the indexing terms obtained previously, we extract from the user profile p, the co-occurrence frequency matrix of the indexing term set T. This matrix represents semantic bonds between the various indexing terms.
Let T_p be the set of terms contained in the user profile $p =< id, G >$. We call matrix *profile-request*, noted M_T, the square matrix of dimension $\mid T \times T \mid$ such that $T \subset T_p$, where each element m_{ij} of M_T is defined by:

$$m_{ij} = fco(t_i, t_j) \quad \text{where } (t_i, t_j) \in T^2$$

C. Request and document representation

From the matrix profile-request M_T, we can calculate the new request q' in order to adjust it according to user profile. This request aims to reflect, as well as possible, the user interest in his search domain.

$$q' = \alpha \times \frac{q}{|q|} + (1 - \alpha) \times \frac{q \times M_{T'}}{|q \times M_{T'}|}$$

q: initial request, indexed on the term set T,
$|q|$: Euclidean length of vector q,
$|q \times M_T|$: Euclidean length of vector $|q \times M_T|$,
M_T: profile-request matrix,
α: threshold such that $0 \leq \alpha \leq 1$, allowing hybridation between initial request $\frac{q}{|q|}$ and the enriched request $\frac{q \times M_T}{|q \times M_T|}$, the higher α is, the more the user profile is considered.

Documents of the corpus are represented in the traditional vector space model [5]. They are indexed on the set of terms T. The information retrieval is done by the calculation of similarity between the new request q', and the documents of the collection. We use, to measure the similarity, the cosine formula [5]. Let d_i and d_j be two documents, the similarity of the cosine between these two documents is formulated by:

$$SIM(d_i, d_j) = \frac{d_i \bullet d_j}{\mid d_i \mid \times \mid d_j \mid}$$

V. Experimentation

An evaluation was made to measure the ability of the *PAWebSearch* agent to personalize in a relevant way the information retrieval. The experimentation is carried out in two steps, at the first step, we compare results provided by *PAWebSearch* to those of *Google*. In the second step we compare *PAWebSearch* with other information retrieval systems on a local Database.

A. First step

The evaluation was made on 10 users (real and simulated). We asked each user to formulate a search request corresponding to its personal interest on *Google* and to evaluate the results provided by this last one. Then, it reformulates the same request by using the *PAWebSearch* agent. Starting with an empty profile, the user consults documents on the Web, and at each 10 consulted documents, it reformulates the same request and evaluates the results obtained. For this first step

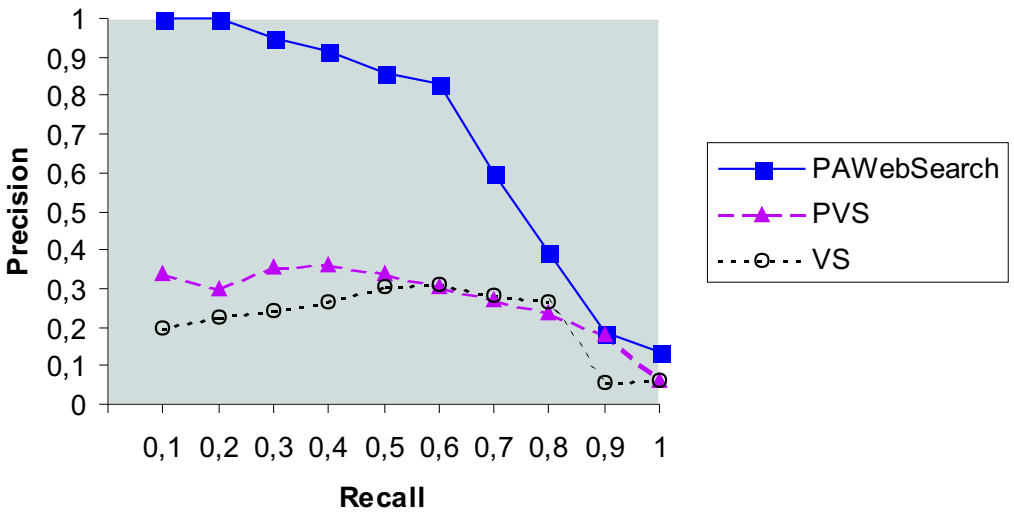

Fig. 2. PAWebSearch, PVS, VS: Precision/Recall after 100 documents consulted

the optimal values of the constants α and β are respectively at 0.1 and 0.005 (where it gives highest average results).

B. Second Step

The documents used for this experiment step are press articles, collected from 5 different online newspapers in different periods. Our collection contains 1200 documents on different thematic (Cinema, Data Processing, Economy, Football, International policy, ..).

We compare *PAWebSerach* agent with the an information retrieval system based on the standard vector space model *VS* and with system based on the model presented in [2] (we call it *PVS*). In *PVS*, user profile has the same structure as a request or a document in the system and is represented by a vector in the vector space, for a given document d, a request q and profile p, a retrieval function $f(q, p, d)$ is defined by:

$$f(q, p, d) = \alpha.s(q, d) + (1 - \alpha).s(p, d)$$

where s is the similarity function (we use the the similarity of the $cosine$).

By varying the value of α, we found that the optimal value is between 0.2 and 0.6, for this experiment α is fixed to 0.5 (where it gives highest average accuracy).

The mechanism for updating the user profile in *PVS* is based on a linear adding of vectors (of documents consulted by the user).

The evaluation was made on 5 users (real and simulated). as in the first step, we asked each user to formulate request corresponding to its personal interest on the three systems and to evaluate the results provided. Starting with an empty profile, the user consults documents and at each 10 consultations he formulates the same request and evaluates the results obtained. For this step the optimal values of the constants α and β are respectively at 0.3 and 0.01 (where it gives highest average results).

C. Results

The evaluation of the IR systems is usually done with the standard measures of precision (P) and recall (R), where:

$$P = \frac{Number\,of\,relevant\,documents\,retrieved}{Total\,number\,of\,documents\,retrieved}$$

$$R = \frac{Number\,of\,relevant\,documents\,retrieved}{Total\,number\,of\,relevant\,documents}$$

Table 1 shows the precision of the documents returned by *PAWebSearch* and *Google*. p(10), p(20), p(30) represent successively the relevance of the 10, 20 and 30 first returned documents. From the table below, we see that in the majority of the cases, the documents returned by *PAWebSearch* are distinctly more relevant than those returned by *Google*. Indeed starting from the 10^{th} document consulted by the user, the relevance reaches 42% in the top 10, and 35% in the top 20, and 31% in the top 30, whereas the relevance of *Google* is only of respectively 35%, 27%, 21%. As expected, we also note that the more user consults documents the more relevance of the documents returned by *PAWebSearch* increases. Indeed starting from the 30^{th} consulted document the precision exceed the 55% in the top 10, to reach 73% with the 100^{th} document consulted.

Table 2 shows the precision of the documents returned by each system *(PAWebSearch, PVS, VS)* according to the number of documents consulted by the user. These results show significant improvement in the precision score when using *PAWebSearch* rather than *VS* or *PVS*. We also note that more user consults documents more the relevance of

Documents consulted	Google P(10)	PAWeb-Search P(10)	Google P(20)	PAWeb Search p(20)	Google P(30)	PAWeb Search P(30)
10	0.350	0.420	0.270	0.350	0.210	0.310
20	0.350	0.420	0.270	0.360	0.210	0.413
30	0.350	0.550	0.270	0.465	0.210	0.440
40	0.350	0.610	0.270	0.550	0.210	0.490
50	0.350	0.650	0.270	0.550	0.210	0.510
60	0.350	0.690	0.270	0.650	0.210	0.366
70	0.350	0.700	0.270	0.685	0.210	0.640
80	0.350	0.720	0.270	0.700	0.210	0.640
90	0.350	0.730	0.270	0.700	0.210	0.633
100	0.350	0.730	0.270	0.700	0.210	0.630

TABLE I

PAWEBSEARCH, GOOGLE: PRECISION VALUES PER DOCUMENT CONSULTED

Documents Consulted	PAWebSearch			PVS			VS		
	P(10)	P(20)	P(30)	P(10)	P(20)	P(30)	P(10)	P(20)	P(30)
10	0.733	0.733	0.688	0.366	0.333	0.277	0.100	0.116	0.222
20	0.900	0.816	0.788	0.366	0.300	0.266	0.100	0.116	0.222
30	0.900	0.850	0.800	0.400	0.300	0.255	0.100	0.116	0.222
40	0.966	0.900	0.855	0.400	0.300	0.255	0.100	0.116	0.222
50	0.966	0.916	0.833	0.366	0.316	0.266	0.100	0.116	0.222
60	0.966	0.850	0.822	0.333	0.300	0.288	0.100	0.116	0.222
70	1.000	0.916	0.855	0.333	0.316	0.300	0.100	0.116	0.222
80	1.000	0.916	0.866	0.300	0.266	0.300	0.100	0.116	0.222
90	1.000	0.933	0.877	0.300	0.266	0.300	0.100	0.116	0.222
100	1.000	0.933	0.877	0.433	0.333	0.322	0.100	0.116	0.222

TABLE II

PAWEBSEARCH, PVS, VS: PRECISION VALUES PER DOCUMENT CONSULTED

the documents returned by *PAWebSearch* increases, and it increases more than *PVS*.

In order to illustrate further the comparison between *PAWebSearch* and *PVS*, Figures 2 presents the precision/recall graphs. The results show that the precision of *PAWebSearch* is very hight (greater than 0.83) for recall values less than 0.6. For high recall values (> 0.7) the precision decreases (between 0.13 and 0.6) and these values are however good. We note also that the precision of *PAWebSearch* is more important than *PVS* and *VS* for all values of recall.

VI. RELATED WORK

In traditional information retrieval systems, users express their needs by formulating requests which are often insufficient to obtain relevant documents. $Blair\&Maron$ showed that the poor performance of IR systems is mainly due to the incapacity of users to formulate adequate requests [1]. Indeed, experiments have proved that different users may expect different answers for the same request, and the same user for the same request may expect different answer in different periods of time [6]. Thus, information retrieval models taking into account user profile were proposed [6][2] [7]. Different methods for learning user interests for information filtering and information retrieval were proposed [8][5][9][10][11][12][13]. Thus, *Chen* models the user by a multiple TF-IDF vectors [9]. In [14] the authors represent a profile as Boolean features using a Naive Bayesian classifier to determine whether a Web page is relevant or not. In [15][16], the authors use neural networks for learning user profiles. Contrary to the existing information retrieval models, our model integrates semantic information in the representation of the user profile but also in the choice of the indexing terms.

VII. CONCLUSION

We have introduced *PAWebSearch* an intelligent agent for Web Information retrieval. This agent analyzes user behavior and build automatically user profile. It adapts documents returned by the Web search engines according to user profile. The model proposed allows a better consideration of the user's interests in information retrieval process by:

- A choice of indexing terms reflecting as well as possible the user knowledge in his search domain.
- An enrichment of the user request by the matrix of profile-request.

In the systems where the user is represented by key-word vectors, an iterative process of user profile re-indexing is necessary to take into account of new indexing terms. In our case no re-indexing of user profile is necessary, therefore it is very adapted to the Web, where information are very heterogeneous. One of the prospects for research is the application of the indexing term set construction method in the framework of a standard information retrieval model.

REFERENCES

[1] D. Blair and M. Maron, "An evaluation of retrieval effectivness for a full-text document retrieval system," *Communication of the ACM*, vol. 28, no. 3, pp. 289–299, 1985.

[2] C. Danilowicz and H. Nguyen, "Using user profiles in intelligent information retrieval," in *Proceedings of the 13th International Symposium on Foundations of Intelligent Systems*. Springer-Verlag, 2002, pp. 223–231.

[3] R. Besançon, M. Rajman, and J.-C. Chappelier, "Textual similarities based on a distributional approach," in *Proceedings of the Tenth International Workshop on Database And Expert Systems Applications (DEXA99), Firenze, Italy*, 1999, pp. 180–184.

[4] A. Mokrane, R. Arezki, G. Dray, and P. Poncelet, "Cartographie automatique du contenu d'un corpus de documents textuels," in *Proceeding of the 7th international conference on the statistical analysis of textual data JADT, 12-15 mars 2004, Louvain-La-Neuve, Belgique*, 2004.

[5] G. Salton and M. M. Gill, "Introduction to modern information retrieval." *New York: McGraw-Hill*, 1983.

[6] S. Myaeng and R. Korfhage, "Integration of user profiles: Models and experiements in information retrieval," *Information Processing & Management*, vol. 26, pp. 719–738, 1990.

[7] P. Chen and F. Kuo, "An information retrieval system based on user profile," *The journal of Systems and Software*, vol. 54, pp. 3–8, 2000.

[8] J. Rocchio, "Relevance feedback in information retrieval," *In G. Salton, the SMART Retrieval System : Experiments in Automatic Document Processing*, pp. 313–323, 1971.

[9] L. Chen and K. Sycara, "Webmate: Personal agent for browsing and searching," in *Proceeding of the Second International Conference on Autonomous Agents*, 1998, pp. 132–139.

[10] R. Arezki, A. Mokrane, G. Dray, P. Poncelet, and D. Pearson, "Luci : A personnalization documentary system based on the analysis of the history of the user's actions," in *Proceedings of the 6th International Conference On Flexible Query Answering Systems (FQAS 2004)*. Lecture Notes in Artificial Intelligence, Springer Verlag, 2004.

[11] C. Danilowicz and H. Nguyen, "User profile in information retrieval systems," in *Proceedings of the 23rd International Scientific School (ISAT 2001)*. PWr Press, 2001, pp. 117–124.

[12] D. Widyantoro, T. Ioerger, and J. Yen, "Learning user interest dynamics with a three-descriptor representation," *Journal of the American Society of Information Science*, vol. 52, no. 3, pp. 212–225, 2001.

[13] H. Lieberman, "Letizia: An agent that assists web browsing," in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, C. S. Mellish, Ed. Montreal, Quebec, Canada: Morgan Kaufmann publishers Inc: San Mateo, CA, USA, 1995, pp. 924–929.

[14] M. Pazzani and D. Billisus, "Learning and revising user profiles: The identification of interesting web sites," *Machine Learning Journal*, pp. 313–331, 1997.

[15] M. McElligot and H. Sorensen, "An evolutionary connectionist approach to personnal information filtering," in *Proceedings of the Fourth Irish Neural Network Conference, Dublin, Ireland*, 1994, pp. 141–146.

[16] A. Tan and C. Teo, "Learning user profiles for personalized information dissemination," in *Proceedings, 1998 IEEE International Joint Conference on Neural Networks, Alaska*, 1998, pp. 183–188.