Journal of Universal Computer Science, vol. 8, no. 6 (2002), 591-601
submitted: 4/2/02, accepted: 3/5/02, appeared: 28/6/02 © J.UCS

Instance Cooperative Memory to Improve Query Expansion in Information Retrieval Systems

Lobna Jéribi
(LISI (Laboratory of Information Science Engineering), INSA de Lyon, France
lobna.jeribi@lisi.insa-lyon.fr)

Béatrice Rumpler
(LISI (Laboratory of Information Science Engineering), INSA de Lyon, France
beatrice.rumpler@lisi.insa-lyon.fr)

Abstract: The main goal of this research is to improve Information Retrieval Systems by enabling them to generate search outcomes that are relevant and customized to each specific user. Our proposal advocates the use of Instance Based Reasoning during the information retrieval process.

When conducting a search, the system retrieves a previous similar search experience and traces back previous human reasoning and behavior and then replicates it in the current situation. Thus, user information retrieval experiences or instances are saved to be reused in future similar cases. The resulting cooperative memory is used for user query expansion.

In order to improve the information retrieval experience, we propose to conceptualize and model both the user profile, and the information retrieval process. This leads us to define some similarity functions between user profiles and information retrieval situations. The reuse of past experiences serves to enrich the initial user query by words from documents found in similar cases. Unlike the classical *Rocchio* method, these documents are those already judged as valid by users with similar profile and in similar search situation. The value this method brings to the user is an increasing relevance of the search outcomes while reducing user interaction with the system.

This method has been implemented in the COSYDOR (Cooperative System for Document Retrieval) prototype based on *Intermedia* (Oracle 8i). Tests and evaluations have been performed on the COSYDOR prototype using the test corpus of TREC (Text Retrieval Conference) and its standard procedures for performance analysis and benchmarking. The results of these analyses show a significant improvement of performance in the first search iterations compared to the *Intermedia* benchmark.

Key Words : cooperative memory, relevance feedback, information retrieval, query expansion, user modeling, instance based learning.
Categories: H, H.4.3 ; H, H.3.3 ; H, H.1.2

1 Introduction and Related Works

In a study carried out on intelligent system architectures and machine learning approaches for information retrieval, such as multi-agents systems [JER 2000], we were directed towards the reuse techniques and instance based reasoning [JER 2001]. This approach, based on the user reasoning cycle (intention, action, acquisition, evaluation), allows to carry out reasoning, to evaluate the document results automatically. This approach is relevant, because the user knowledge used is very

"reliable" since "lived" and "evaluated" by this user. Moreover, the reuse approach enables to minimize the user interaction. Furthermore, the field of experience reuse, offers methods to specify and build the experience *context*, so that the reuse is optimal [MIL 99].

We propose in this paper to build an instance memory of information retrieval instances. When the users search contexts are "similar", these instances are reused to improve user query formulation. The user profile is considered as a significant element of an instance search context. Thus we propose in this paper a formal representation of "user profile" model. We also define a representation of an information retrieval instance model, in order to evaluate the similarity between instances. The similarity evaluation enables to highlight candidate instances to be reused, from the instance memory.

Some related works of context definition and experience reuse were proposed in the early literature. RADIX project [COR 99] proposes the modeling of internet navigation sessions carried out by the user. These models are reused in order to suggest similar sessions to the user. CABRI-N [SMA 99] is a personalized image retrieval system. Smaïn proposes a modeling of user strategy during an information retrieval process. Retrieval sessions are memorized and reused to improve user strategy search.

In this paper, we present briefly a modeling study of the user during a search session, and a representation of a search instance or situation. Then, we present our approach of instance reuse for query expansion. Lastly, we present the results of our first tests and the prospects for evaluations.

2 Information Retrieval Instance Modeling

In this section, we firstly define the user profile model, representing a relevant feature of the retrieval instance. Next, a global information retrieval instance is proposed.

2.1 User Profile Modeling

Intelligent information systems aim to automatically adapt to individual users. Hence, the development of appropriate user modeling techniques is of central importance. Algorithms for intelligent information agents typically draw on work from the information retrieval and machine learning communities. Both communities have previously explored the potential of established algorithms for user modeling purposes [BEL 97] [WEB 98].

To define the specific user knowledge during a search session, we have used cognitive approach results [ALL 91]. We have classified the user knowledge in four *knowledge categories*, ranked according to their *evolution* degree.

1. Cultural knowledge (features having *little* or *no evolution*)
2. Professional knowledge (features having *long term evolution*)
3. System knowledge (features having *mean term evolution*)
4. Search knowledge (features having short term evolution), related to the current search session.

User knowledge features presented above constitute a *generic model* of user profile. S*pecific models* are related to the application cases used (documents, search system and users types).

2.1.1 Representation formalism

The chosen representation formalism is the vector model. It is the formalism commonly used in both communities: information retrieval [SAL 86] and instance based reasoning [KOL 88]. The vector model presents several advantages in processing similarity between vectors.
Let $U=<U_1, U_2, U_3, U_4>$, be the vector representation of U, where U_i represents the i^{th} category of user knowledge U.
$\boldsymbol{U_i = \{a_{i,j}\};\ \forall j \in [1, n];}$ $a_{i,j}$ represents the j^{th} attribute of the category U_i;
$\boldsymbol{a_{i,j} \in \{v_k\};\ \forall k \in [1, n];}$ v_k represents the k^{th} possible instance of $a_{i,j}$

2.1.2 Similarity function

We propose to memorize user retrieval experiences, in order to reuse them, when users have "similar" profiles. Thus, our first goal of formalizing the user model, is to define the "distance" between user profiles. The expression (1) shows the similarity function S_U, between two user profiles U^i and U^j

$$S_U(U^i, U^j) = \frac{\mu\, s_1(U_1^i, U_1^j) + \nu\, s_2(U_2^i, U_2^j) + \kappa\, s_3(U_3^i, U_3^j) + \lambda\, s_4(U_4^i, U_4^j)}{\mu + \nu + \kappa + \lambda}$$

Where:
U_p^i is the vector representing the p^{th} category of U^i, which is the profile of the user j.
s_p: similarity function between vectors of the p^{th} ***category*** of U
$s_p \in [0, 1]$; $\mu, \nu, \kappa, \lambda \in [0, 1]$
$\mu, \nu, \kappa, \lambda$, represent the parameters enabling to "contextualize" the similarity.
Details about this expression are presented in [JER 2001].

2.2 Search Instance Modeling

The results of various studies on search instance [JER 2001b], make highlight of following features of a search instance:
- User profile represented by ***U;***
- User information need expressed by a query, represented by ***Q;***
- Documents solutions represented by ***D***;
- Evaluations ***E*** of relevancy of the documents ***D,*** given by the user U.

Referring to the problem resolution field, the initial problem in information retrieval system is represented by the user profile ***U***, and his query ***Q***. Collected documents ***D*** represent the solution to the problem, and ***E*** the solution evaluation.

We propose a formal description of a search instance, carried out by a user, during an information retrieval session, (vector representation) as follows:

Instance = <U, Q, D, E>

We also define a similarity function in order to evaluate the distance between instances. Thus, the closest memorized instance to the current one would be reused.

In our context, we enrich the instance case, by a confidence degree that reflects the relevance of the instance.

3 Knowledge Base of Relevance Feedbacks to Improve Query Expansion

Before presenting our proposition of query expansion based on experience reuse, we introduce in the next paragraph the classical approaches of query expansion based on relevance feedback.

3.1 Relevance feedback for query expansion

3.1.1 Relevance feedback

An automatic relevance feedback system can be designed as following: the system processes the initial query and returns a list of documents ranked in order of their predicted relevance to the user. The user examines a few of the highest ranking documents, determines whether or not they are relevant, and sends this information back to the system. The system uses the analyzed documents to automatically construct a revised query and produces a new ranking of the remaining documents in the collection. The user can examine more documents and repeat the relevance feedback process as often as desired.

1.1.1 Feedback using the Vector Space Model: Rocchio

The VSM is ideally suited for automatic relevance feedback since it accepts free text input for the query. Therefore, we can simply incorporate some or all of the text from the relevant documents directly into the query. Non relevant documents can be utilized in a similar way. One of the most successful relevance feedback strategies was developed by *Rocchio* [ROC 71], and works as follows:

$$Q' = \alpha Q + \beta \left(\frac{1}{|D_r|} \sum_{d_i \in D_r} d_i\right) - \gamma \left(\frac{1}{|D_n|} \sum_{d_j \in D_n} d_j\right)$$

$\alpha, \beta, \gamma \in [0,1]$; d_i is weighted term vector representing a collected document
$|D_r|$ cardinality of relevant documents set
$|D_n|$ cardinality of non-relevant documents set

This strategy simply adds a weighted sum of the relevant document vectors and subtracts a weighted sum of the non relevant documents from the query. Relevance

feedback using this strategy produces a very large improvement in retrieval performance (from 20-80% or more, depending on the collection) [SAL 86] [HAR 92] [AAL 92].

Adding the full document text directly into the query does have its drawbacks. The number of terms in the query grows rapidly with the addition of evaluated documents, causing searches to take longer and longer. A full search must be repeated for each iteration of relevance feedback. Since relevance feedback is supposed to be an interactive process, the system must be able to return feedback results in a relatively short period of time.

3.1.2 Drawbacks of Rocchio method based on relevance feedback

Although relevance feedback has proven to be an effective way to improve information retrieval performance, it is rarely used in practice [NIE 96]. This mechanism has been incorporated into several online search engines, but few users actually use it. Nie thinks that an important reason is the short term effect of a relevance feedback. A user has to make great efforts in evaluating documents, but the evaluation only has an effect on a single query. Once a new query is input, new evaluations have to be made. From the user's point of view, it simply does not worth the efforts. However, we think users are ready to make the efforts when the effect is permanent. Therefore, we will suggest a way to adjust the system's knowledge according to relevance feedback.

3.2 Knowledge base for relevance feedback

3.2.1 Relevance feedback and Instance reuse

Our proposal is based on the *Rocchio* method of query expansion. The principle approach of the proposed solution, consists on "completing" the documents used for the query expansion issued from the current relevance feedback, by the evaluated documents extracted from the previous search instances. On the basis of these documents -coming from both sources-, we apply the *Rocchio* approach of query expansion. Thus, terms added to the user query could result from the instance base documents, evaluated previously by the user or other users being in similar search contexts, and having similar profiles.

However, these two document sources are not independent, since the documents evaluated during the previous search iteration (Relevance feedback) represent also an instance contained in the memory of instances.

The interest of our proposal is primarily justified when no relevance feedback is made by the user during his search session. In this case, the reuse of the instance base constitutes an interesting alternative for the query expansion. Moreover, this enables to give a certain "freedom" to the user, because he doesn't have to evaluate documents relevancy to obtain system help to express his query.

Nevertheless, the instances reused cannot contribute in the same way for query expansion. Thus, we propose to weight this contribution, according to the "confidence degree" of the reused instance. This concept will be detailed in the following paragraph.

3.2.2 Adapting Rocchio method for query expansion based on experience reuse

Giving the current instance represented by:

$I_{current} = <U_{current}, Q_{current}, D_{current}, E_{current}>$;
$Instance_{similar} = <U_{similar}, Q_{similar}, D_{similar}, E_{similar}>$: the most similar instance to the current one $(\boldsymbol{I_{current}})$, with the confidence degree $\varphi_{similar}$.
$D_{similar}$ is a set of documents d_i; $\boldsymbol{d_i}$ is the weighted term vector representing the document (or a part of the document) that the user has chosen to evaluate:
$\boldsymbol{d_i} = <b_{i,1}, b_{i,2}..., b_{i,n}>$
$E_{similar}$ represents the evaluation given by the user, on the relevancy of $D_{similar}$ according to $Q_{similar}$.
The proposed expression of query expansion of $Q_{current}$, consists on adapting the *Rocchio* approach, by reusing the evaluated documents $D_{similar}$ extracted from the instance $I_{similar}$.

$$Q'_{current} = \alpha \times Q_{similar} + \varphi_{similar} \times \beta \left[\frac{1}{|Dsimilar-r|} \sum_{d \in Dsimilar-r} die_i \right] + \varphi_{similar} \times \gamma \left[\frac{1}{|Dsimilar-n|} \sum_{d \in Dsimilar-n} die_i \right]$$

$d_i \in D_{similar}$; $e_i \in E_{similar}$; $e_i \in [-1, 1]$; e_i corresponds to the evaluation of d_i
$\alpha, \beta, \gamma \in [0,1]$; α, β, γ are coefficients defined in the *Rocchio* expression.
$\varphi_{similar} \in [0,1]$; $\varphi_{similar}$ *confidence degree of* $I_{similar}$
$D_{similar-r}$ represent relevant document set of the instance $I_{similar}$
$D_{similar-n}$ represent non relevant document set of the instance $I_{similar}$

The new expression for query expansion is based on documents extracted from the instance base. The added terms result from documents evaluated by users when they were in similar search situations. However, we give more importance to the contribution of the instance coming from a relevance feedback compared to those coming from the instance memory. Hence, we propose to decrease by φ *(confidence degree of the similar instance)* the contribution of this instance in the proposed expression.

Nevertheless, when the used instance corresponds to a relevance feedback (documents evaluated by the same user during the same session of search), the confidence degree of this instance is maximal $(\varphi = 1)$. In this case, the expression enables to apply the classical *Rocchio* approach.

3.2.3 Combining learning to the Rocchio approach

As presented above, our system allows two types of learning :
Long term learning thanks to the instance memory. The reuse approach and instance based learning allows the user to be given the aid of the system without having to interact and evaluate during each session, the collected documents (contrary to traditional methods of query expansion based on relevance feedback).

However, our solution is optimal when the number of memorized instances is significantly important. In the contrary case, the query expansion is based on classical *Rocchio* relevance feedback. The effectiveness of the instance base is well exploited when the users have common interests and carry out exploitable common searches. This is a classical constraint in the co-operative systems.
Short term learning thanks to the training by reward / penalty of the search instances. The system evolves according to the failure / success of the proposed solutions.

4 Experiments and Evaluations Using Test Corpus

4.1 COSYDOR Prototype presentation

Our prototype COSYDOR (Cooperative SYstem for DOcument Retrieval) is based on *Intermedia* of Oracle 8i. We enriched *Intermedia* by an intelligent layer (developed in java) enabling the users query expansion and the management of the instance base.

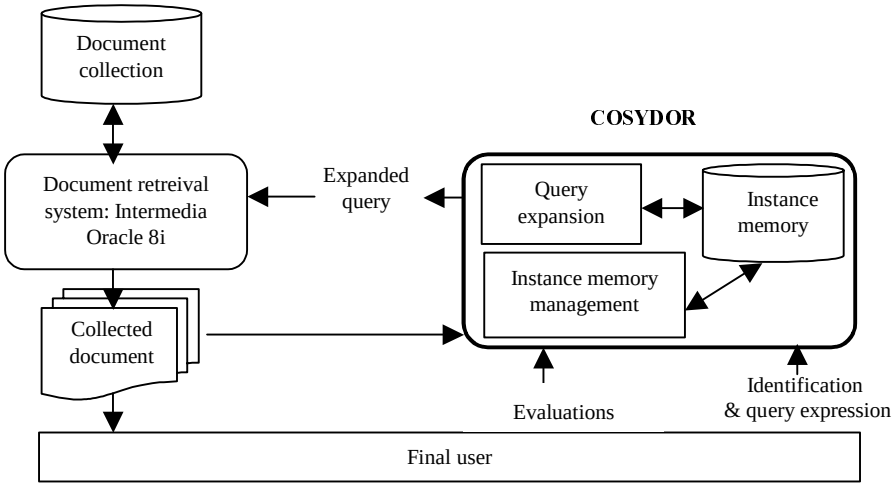

Figure 1: COSYDOR prototype

Intermedia is a textual DBMS, using linguistic tools (thesaurus, lexicon, etc.) for documents and queries representation. The choice of this tool results from a comparative survey on several information retrieval systems [JER 2001]. *Intermedia* proved to be most relevant in our context. One of its advantages, is to offer paragraph extraction functionality, enabling to present document “views” during the document restitution to the user. This makes the user evaluation more precise on the one hand, and makes easier the access to the long documents for the sight deficient users on the other hand.

4.2 Test and evaluations

4.2.1 TREC test corpus

In order to test and to evaluate the contribution of our system, we have used a TREC corpus of test. TREC (Text Retrieval Conference) is an American organization which provides a corpus of tests and common procedures of analysis of performance. Among this base of tests, we used and indexed a whole of 7000 documents, in HTML format, relatively long (approximately 600 word/document) and specialized in the biomedical field. We reused the examples of search provided by TREC, and calculated the various indices of performance of our system: the rate of precision (a number of collected relevant documents / a total number of collected documents) and the rate of recall (a number of collected relevant documents / a total number of relevant documents of the documentary corpus).

In this base of tests, we used the expansion possibilities of our system to note our work contribution and improvement, compared to Oracle *Intermedia* results. Our procedure of tests consists on iterations of retrieval, where query expansion are evaluated by "initiated" users. For each iteration, the rate of recall and the precision of the answers are processed.

4.2.2 Evaluations

In the carried tests, we note that the rates of precision/recall are improved comparing to iteration 0, which corresponds to *Intermedia* performance. Thus, our solution enables to optimize *Intermedia* results. The experimental average of precision/recall rates for information retrieval is generally around 0,3 [NIE 96]. The shape of the obtained curves (Figure 2) is justified by the antagonism of recall and precision rates. We noticed that this is related to the type of expansion carried out. Indeed, we noted that the positive query expansion (terms extracted from relevant documents) causes a significant increase in precision rate. However, negative expansions were less effective.

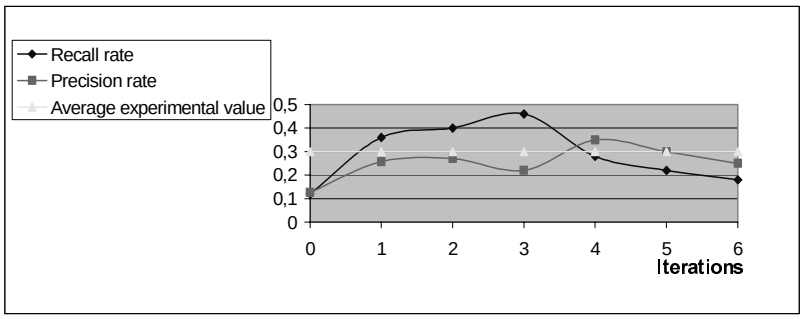

Figure 2: Performance evaluation of COSYDOR

Beyond the fourth iteration, we could note that the rates of recall / precision decrease. This is due to the length (a number of terms) of the query. Indeed, beyond a maximum number of added terms (5 terms in our context), the performance of expansion falls.

The first evaluation results, although very encouraging, must be moderated by the limited number of tests (ten queries). Moreover, the users who contributed to the tests were previously initiated to the system use. It would be then interesting to carry out these tests on a broader sample of users, having different profiles.

The second part of our evaluation consists on the comparison of COSYDOR performances versus other information retrieval systems using manual and automatic query expansion. The results of these systems were provided to us by TREC. First experimentation of these comparative tests are shown in next figure.

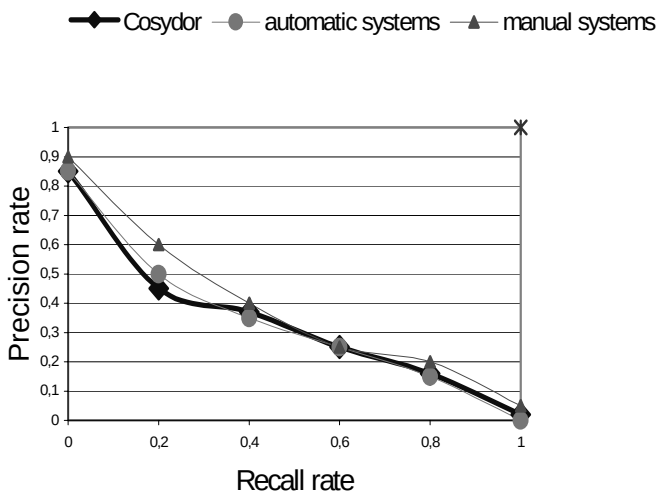

Figure 3: Comparison of COSYDOR to other queries expansion systems

The tests show that manual query expansion results remain always better (according to the recall/ precision rate performance) then automatic query expansion. However, the curves show that COSYDOR presents very close performances to those of the automatic systems. Our results are optimal for high recall rates. Our current work consists on handling more experiments in order to make better justifications of these comparison results.

The first evaluation results, although very encouraging, must be moderated by the limited number of tests (ten queries). Moreover, the users who contributed to the tests have very close profiles, and were previously initiated to the system use. It would be then interesting to carry out these tests on a broader sample of users, having different profiles and having a visual handicap.

5 Conclusion

The goal to build a knowledge base making “permanent” the user evaluations on search results, represents our first work motivation.

The cooperative instance memory offers several advantages compared to the *Rocchio* method of query expansion based exclusively on the relevance feedback.

The first advantage of the memorization consists to present an alternative solution to the *Rocchio* method, so that the user can be offered the system aid, without having to interact and to evaluate the collected documents, during each retrieval session. In fact, traditional methods of query expansion based on relevance feedback, are applied only after user interaction and after an evaluation of the retrieved documents relevancy. This approach “ignores” all knowledge chunks of the former search situations made by this user or by other users having “similar” profiles, and being in a close search situation.

The second advantage of the instance memory, is to enable the system to learn progressively, with experience acquisition. Indeed, the query expansion based on relevance feedback methods, constrain the system to make the same training during each session of search, from a search iteration to another, in order to define an optimal query. Whereas, it would be interesting to avoid these repetitive steps, by memorizing the search instances and their corresponding evaluations.

The third advantage is related to user psychology. Hence, users have more facilities to invest themselves in effort of evaluations, knowing that they will be permanent and reusable. This “reluctance” constitutes a real obstacle in the use of systems based on relevance feedback [NIE 96].

However, memorizing and reusing evaluations is a difficult task. It requires the ability to reproduce all the context of documents evaluation carried out by users, so that the memorized knowledge is representative of the moment, and the reuse is suitable. This justifies all the effort of contextual representation and modeling of search instance, presented in the first part of this paper.

These features were carried out in the COSYDOR system, implemented in Java, based on *Intermedia* (Oracle 8i). Tests and evaluations are carried out using the test corpus of TREC. The results show, for first search iterations, a significant improvement of performance compared to that of *Intermedia*. However, our sample of users is not sufficiently representative. Thus, an obvious direction for further research is to widen sample of users in the experimental tests.

References

[ALL 91] Allen N. Cognitive Research in information science: implication for design. *Annual review of information science and technology,* 1991, vol. 26, pp. 3-37

[BEL 97] Belkin, N., Kay, J., Tasso, C. Special issue on User Modelling and Information Filtering . *User modelling and User adapted Interaction*, 1997, vol 7(3), pp. 313-331.

[BIL 99] Billsus, D., Pazzani, M. A hybrid User Model for New Story Classification. *Proceedings of the Seventh International Conference on User Modelling (UM’99),* Banff, Canada, 20-24 Juin 1999.

[COR 99] Corvaisier, F., Mille, A., Pinon, J.M. – Recherche assistée de documents indexés sur l'expérience (RADIX): Mesures de similarité des épisodes de recherche sur le WEB. *IC'99 Ingénierie des connaissances*.

[JER 2001] Jéribi, L. Personalised Document Retrieval Aid : Experience Reuse Approach. PHD thesis, 7 December 2001, *Laboratory of Information Science Engineering, INSA de Lyon, France.*

[JER 2000] Jéribi, L., Rumpler, B., Pinon J.M. Personalised information retrieval in specialised virtual libraries. *New Library Worl review,* MCB press, VOL 101 N° 1153, 2000, p21-27.

[KOL 88] Kolodner J.L. (édité par). – *Workshop on case-based Reasoning, DARPA 88*. Clearwater, Florida, Morgan-Kaufmann, San Mateo, 1988.

[MIL 99] Mille, A., Fuchs, B., Chiron, B. Raisonnement fondé sur l'expérience en supervision industrielle. *Revue d'Intelligence Artificielle* 13, 1999, p. 97-128.

[NIE 96] Nie, J.Y., Brisebois, M., Lepage, F., (1996). Information retrieval as counterfactual, *The computer journal*, 38(8): 643-657.

[PAZ 97] Pazzani, M., Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine learning* 27: 313-331.

[ROC 71] Rocchio, J.J. Relevance feedback in formation retrieval. In Gerard Salton editor, *The SMART retrieval system: Experiments in Automatic Document Proceedings,* pages 313-323. Prentice Hall, 1971.

[SMA 99] Smaïl, M. Recherche de régularités dans une mémoire de sessions de recherche d'information documentaire, *InforSID 2-4 juin 1999, actes des conférences, XVIIème congrès*, 1999.

[WEB 98] Webb, G. Special issue on Machine Learning for User Modelling. *User Modelling and user Adapted Interaction*, vol8 (1-2), Kluwer Academic Publishers.