Information Retrieval Model Based on User Profile

R. AREZKI[1], P. PONCELET[1], G. DRAY,[1] D.W. PEARSON[2]
[1] Centre LGI2P EMA, Site EERIE Parc Scientifique Georges Besse
30035 Nimes Cedex 1, France
{rachid.arezki, pascal.poncelet, gerard.dray }@ema.fr
[2] IUT de Roanne, 20 Avenue de Paris 42334 Roanne, France
david.pearson@univ-st-etienne.fr

Abstract. With the development of internet and storage devices, online document servers abound with enormous quantities of documents, so 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. It became urgent to propose new information retrieval systems able to apprehend efficiently these enormous quantities of documents. In this paper we present *PQIR* an information retrieval model based on user profile. The originality of our approach is a choice of indexing terms depending on the user request but also on his profile. An empirical study confirms the relevance of our approach.
Keywords. User Modeling, Information Retrieval

1 Introduction

With the development of internet and storage devices, online document servers abound with a very large number of documents. The end user, generally overloaded by information, can't efficiently perceive such information. Today, one of the tackled issue is thus to provide the end user with the right and useful information. Indeed, the objective of an information retrieval system is to answer efficiently to a user request often expressed in natural language. Whatever the effectiveness of information retrieval system used, relevant documents can be omitted because the user request does not refer correctly to these documents. Indeed, requests only formulated by key words express badly the user information needs. Of course, these needs depend on the formulated request but also on the knowledge acquired by the user in his search domain. In other words, two users can formulate the same requests for different needs. For example, the results awaited by an expert in Java language formulating the request "Java course" are different from the results awaited by a non expert with the same request. One, solution consists of taking into account the user profile in the information retrieval process in order to increase the relevance of the answered documents.

In this paper, we present *PQIR* an information Retrieval model based on user profile. The originality of our approach is a choice of indexing terms depending on the user request but also on his profile. That is to say that we consider 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 presents information retrieval in the standard vector space model. Then, Section 3 presents our information retrieval model. Experimental results are presented in Section 4. Section 5 gives an overview of related work. Finally, section 6 provides some concluding remarks and directions of future research.

2 Information Retrieval Based on Standard Vector Space Model

In the vector space model, each document d is represented by a n-dimensional vector $(w_1, .., w_n)$, where w_i is the weight of the term t_i in the document d. A term can be a word, a stem, a lemma, or a compound. For each pair (u, v) of vectors, a similarity function (or distance function) $s(u, v)$ should be defined.
For a given request vector q (a request is also just a text and can be converted into a vector), retrieval is achieved by measuring similarity between a document and request in the underlying vector space. Thus, the most similar documents to the request will be proposed to the user.
More formally, the standard vector space model can be presented as a tuple $< X, Q, T, s, f >$, where X represents the set of documents (i.e. document collection), Q stands for the set of requests, T represents the term set indexing, s is a similarity or distance function and f is the term set construction function, with $T = f(X)$.

Term set construction The indexing term set T is built from the collection X. Its elements are chosen to be as discriminative as possible. Thus, there are various methods: for example the choice of indexing term set can be based on term frequency, where terms that have both high and low frequency within a document are considered to be function words [Luh58][SG83][VR79].

Term Weighting The weight of a term represents the degree of its importance in a document. There are three main factors term weighting: term frequency factor, collection frequency factor and length normalization factor.
TF-IDF weighting is one that has been well studied in the information retrieval, where the importance of a term is proportional to the occurrence frequency of this term in each document, and inversely proportional to the total number of documents to which this term occurs in a given document collection [SG83]. More precisely, let TF_i be the frequency of occurrence of term t_i in a document d, and let DF_i be the corresponding document frequency. The importance of word i in document d, denoted by w_i, is expressed as follows:

$$w_i = \frac{TF_i}{\sum_j TF_j} \times log(N/DF_i)$$

where N represents the number of documents in the collection.

Similarity Measure They are differents similarity measures, the most popular one is the cosine coefficient. It measures the angle between a document vector and the request vector. Let d_i, d_j be two documents vectors, the similarity of the *cosine* between these two documents is formulated by:

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

where $\mid d_i \mid$ represents the euclidean length of the vector d_i.

3 PQIR Model

$PQIR$ is an extended vector space model, it 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 request user 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. To better adapt to the user's needs, the initial request vector q is transformed into q'. The transformation of q to q' requires the construction of the profile-request matrix (Section 3.3). The algorithm below describes the information retrieval process associated with a new user request q.

Algorithm 1: Retrieval Algorithm
Input:
q: user request, p: the user profile
Output: proposal of documents to the user
begin
1. construction of the Indexing Term Set T
2. calculation of the profile-request matrix M_T
3. vectorial representation of the request q and of each document of the collection X
4. calculation of the new request vector q'
5. calculation of the similarity between q' and the whole documents of the collection X
6. propose to the user the documents most similar to the request q'

end

3.1 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 this user. The general idea is to analyze the content of the different documents and to store in the graph G the co-occurrence frequency between various terms of a document, as well as occurrence frequency of these terms. More formally, $G =< V, E >$ is a labelled graph such as:

1. $V = \{(t_1, f_1)..(t_n, f_n)\}$ is a set of vertices of G, where each vertex (t_i, f_i) is represented by a term t_i and its frequency f_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. A textual unit can be k terms windows, sentences, paragraphs, sections, or whole documents [BRC99][MADP04]. In the framework of our user model, $fco(t_i, t_j)$ represents co-occurrence frequency between the terms t_i and t_j in the set of the 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,
3. Stemming, that is, the reduction of terms to their root,
4. Construction of the graph G_d.

For each new consulted document d, a graph G_d is built, then G_d is added to graph G. User profile is thus represented by the graph G (see Algorithm 2).

Algorithm 2: User Profile Learning Algorithm
Input: d: consulted document, the user model $p =< id, G >$,
Output: updated user model $p =< id, G >$
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 edge (t_i, f_{t_i}) in the graph G such as
      $f_{t_i}^{G} = f_{t_i}^{G_d}$
  3. **for** *each vertex* (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

where $fco_G(t_i, t_j)$ represents the frequency of co-occurrence between the terms (t_i, t_j) in the graph G.

3.2 Indexing Term Set Construction

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

Algorithm 3: Indexing term set construction
Input:
q: user request
the user model $p =< id, G >$,
Output:
T: indexing term set
begin
  1. $T \leftarrow \emptyset$;
  2. $T \leftarrow$ terms contained in the request q;
  3. **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

where β is a constant representing the threshold of term selection.

3.3 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 $T_p = \{t_1, .., t_n\}$ 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 as $T \subset T_p$, where each element m_{ij} of M_T is defined by:

$$m_{ij} = fco(t_i, t_j)$$

3.4 Request and document representation

The initial request q is indexed on the set of terms T. Then from the matrix profile-request M_T and the initial request q, a new request vector q' is calculated in order to take into account the user profile. This request aims to reflect, as well as possible, the user interest in his search domain.

$$q' = (1 - \alpha) \times \frac{q}{|q|} + \alpha \times \frac{q \times M_T}{|q \times M_T|}$$

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

The documents of a collection are represented in the traditional vector space model [SG83], and 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. To measure the similarity, we use the cosine formula [SG83] as described in Section 2.

3.5 Example

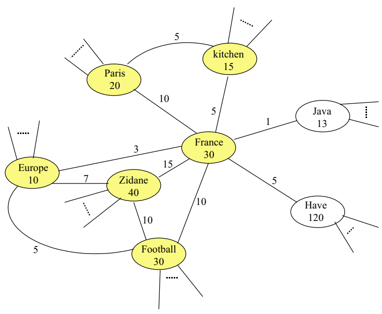

Fig. 1. A part of the user profile graph

In order to illustrate our approach, let us consider the following example. Let a user be defined by his profile $p = < id, G >$. Let us consider that this user retrieves a documents on France through the request "France". The Figure 1 shows a part of the user profile graph.
The indexing terms are selected among the neighbor of the term "France " in the graph G modelling the user profile. Thus, with for example a threshold $\beta = 0.01$, by applying the algorithm of indexing term set construction (Algorithm 3), we obtain:

$$\frac{fco("france","football")^2}{f_{"france"} \times f_{"football"}} = \frac{10^2}{30\times30} = 0.111 > \beta \Rightarrow \text{will be selected}$$
$$\frac{fco("france","zidane")^2}{f_{"france"} \times f_{"zidane"}} = \frac{15^2}{30\times40} = 0.187 > \beta \Rightarrow \text{will be selected}$$
$$\frac{fco("france","europe")^2}{f_{"france"} \times f_{"europe"}} = \frac{3^2}{30\times10} = 0.030 > \beta \Rightarrow \text{will be selected}$$
$$\frac{fco("france","paris")^2}{f_{"france"} \times f_{"paris"}} = \frac{10^2}{30\times20} = 0.166 > \beta \Rightarrow \text{will be selected}$$
$$\frac{fco("france","kitchen")^2}{f_{"france"} \times f_{"cuisine"}} = \frac{5^2}{30\times15} = 0.055 > \beta \Rightarrow \text{will be selected}$$
$$\frac{fco("france","java")^2}{f_{"france"} \times f_{"java"}} = \frac{1^2}{30\times13} = 0.002 < \beta \Rightarrow \text{will be rejected}$$
$$\frac{fco("france","have")^2}{f_{"france"} \times f_{"avoir"}} = \frac{10^2}{30\times30} = 0.006 < \beta \Rightarrow \text{will be rejected}$$

The indexing term set selected is:

$$T = \{"france", "europe", "kitchen", "paris", "football", "zidane"\}$$

The request vector indexed on the set of term T is:

$$q = (1, 0, 0, 0, 0, 0)$$

The profile-request matrix will be:

$$M_T = \begin{bmatrix} 0 & 3 & 5 & 10 & 10 & 15 \\ 3 & 0 & 0 & 0 & 5 & 7 \\ 5 & 0 & 0 & 5 & 0 & 0 \\ 10 & 0 & 5 & 0 & 0 & 0 \\ 10 & 5 & 0 & 0 & 0 & 10 \\ 15 & 7 & 0 & 0 & 10 & 0 \end{bmatrix}$$

For example, element $M_T[1][3]$ corresponds to the frequency of co-occurences between the terms "France" and "kitchen".

Thus, the transformed request becomes:

$$q' = (1-\alpha)\times\frac{q}{|q|}+\alpha\times\frac{q \times M_T}{|q \times M_T|} = (1-\alpha)\times(1, 0, 0, 0, 0, 0)+\frac{\alpha}{\sqrt{459}}\times(0, 3, 5, 10, 10, 15)$$

With $\alpha = 0.5$, we obtain : $q' = (0.5, \frac{1.5}{\sqrt{459}}, \frac{2.5}{\sqrt{459}}, \frac{5}{\sqrt{459}}, \frac{5}{\sqrt{459}}, \frac{7.5}{\sqrt{459}})$

The information retrieval is done by the calculation of similarity between the new request q', and the documents of the collection indexed on the set of terms T. Thus, the most similar documents to q' will be proposed to the user.

4 Experimentation

An evaluation was made to measure the capacity of the *PQIR* model to personalize in a relevant way the information retrieval.

4.1 Method

Data The documents used for our empirical study are press articles, collected from 5 different online newspapers in different periods. Our collection contains 1200 documents on different thematics (Cinema, Data Processing, Economy, Football, International policy, ..).

Comparaison We have chosen to compare *PQIR* model with the standard vector space model *VS* and with a model similar to the one presented in [DN02] (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.
The mechanism for updating the user profile in *PVS* is based on a linear adding of vectors (of documents consulted by the user).

Procedure The evaluation was made on 5 users (real and simulated). 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 experimental study the constants α and β are fixed respectively to 0.3 and 0.01.

Consulted Documents	PQIR			PVS			VS		
	P(10)	P(20)	P(30)	P(10)	P(20)	P(30)	P(10)	P(20)	P(30)
10	73.33	73.33	68.88	36.66	33.33	27.7	10	11.66	22.2
20	90	81.66	78.88	36.66	30	26.66	10	11.66	22.2
30	90	85	80	40	30	25.55	10	11.66	22.2
40	96.66	90	85.55	40	30	25.55	10	11.66	22.2
50	96.66	91.66	83.33	36.66	31.66	26.66	10	11.66	22.2
60	96.66	85	82.22	33.33	30	28.8	10	11.66	22.2
70	100	91.66	85.55	33.33	31.66	30	10	11.66	22.2
80	100	91.66	86.66	30	26.66	30	10	11.66	22.2
90	100	93.33	87.77	30	26.66	30	10	11.66	22.2
100	100	93.33	87.77	43.33	33.33	32.22	10	11.66	22.2

Table 1. Experiment Results

4.2 Results

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

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

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

The table 1 shows the precision of the documents returned by each system *(PQIR, PVS, VS)* according to the number of documents consulted by the user. Thus, P(10), P(20) and P(30) represent successively the relevance of the 10, 20 and 30 first returned documents. These results show significant improvement in the precision score when using the *PQIR* model rather than *VS* or *PVS* model. We also note that more user consults documents more the relevance of the documents returned by *PQIR* increases, and it increase more than *PVS* model.

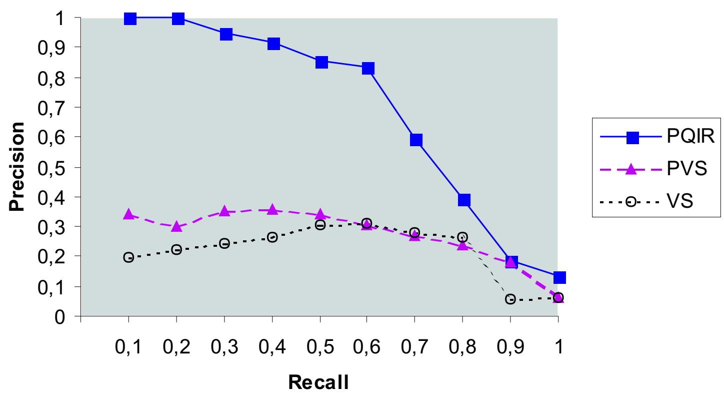

Fig. 2. Precision/Recall of the differents systems after 100 documents consulted

In order to illustrate further the comparison between *PQIR* and *PVS* model, Figures 2 presents the precision/recall graphs. The results show that the precision of the *PQIR* model 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 the *PQIR* model is more important than *PVS* and *VS* for all values of recall.

5 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 the users to formulate adequate requests [BM85]. Indeed, experiments have proved that different users may expect different answers for the same request. Furthermore, the same user, for the same request, may expect different answer in different periods of time [MK90]. Thus, information retrieval models taking into account user profile were proposed [MK90,DN02,CK00]. Different methods for learning user interests for information filtering and information retrieval were proposed [Roc71,SG83,CS98,AMD+04,DN01,WIY01,Lie95,FGP04]. Thus, *Chen* models the user by a multiple TF-IDF vectors [CS98]. In [PB97] the authors represent a profile as Boolean features using a Naive Bayesian classifier to determine whether a Web page is relevant or not. In [MS94,TT98], 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.

6 Conclusion

In this paper, we proposed a new approach for a personalized information retrieval. The model proposed allows a better consideration of the user's interests in information retrieval process by:

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

In the information retrieval models where the user is represented by vectors of terms, an iterative process of user profile re-indexing is necessary to take into account of new indexing terms. In our model no re-indexing of user profile is needed. Experimental results confirm the relevance of our approach. 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

[AMD+04] R. Arezki, A. Mokrane, G. Dray, P. Poncelet, and D.W. 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)*. LNAI, Springer Verlag, 2004.

[BM85] D.C. Blair and M.E. Maron. An Evaluation of Retrieval Effectivness for a Full-Text Document Retrieval System. *Communication of the ACM*, 28(3):289–299, 1985.

[BRC99] 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*, pages 180–184, 1999.

[CK00] P.M. Chen and F.C. Kuo. An Information Retrieval System Based on User Profile. *The journal of Systems and Software*, 54:3–8, 2000.

[CS98] L. Chen and K. Sycara. WebMate: Personal Agent for Browsing and Searching. In *Proceeding of the Second International Conference on Autonomous Agents*, pages 132–139, 1998.

[DN01] C. Danilowicz and H.C. Nguyen. User Profile in Information Retrieval Systems. In *Proceedings of the 23rd International Scientific School (ISAT 2001)*, pages 117–124. PWr Press, 2001.

[DN02] C. Danilowicz and H.C. Nguyen. Using User Profiles in Intelligent Information Retrieval. In *Proceedings of the 13th International Symposium on Foundations of Intelligent Systems*, pages 223–231. Springer-Verlag, 2002.

[FGP04] W. Fan, M.D. Gordon, and P. Pathak. Effective Profiling of Consumer Information Retrieval Needs: A Unified Framework and Empirical Comparison. *Decision Support Systems*, to appear, 2004.

[Lie95] H. Lieberman. Letizia: An Agent That Assists Web Browsing. In Chris S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 924–929, Montreal, Quebec, Canada, 1995. Morgan Kaufmann publishers Inc: San Mateo, CA, USA.

[Luh58] H.P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2:159–165, 1958.

[MADP04] 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.

[MK90] S.H. Myaeng and R.R. Korfhage. Integration of user profiles: Models and Experiements in Information Retrieval. *Information Processing & Management*, 26:719–738, 1990.

[MS94] M. McElligot and H. Sorensen. An Evolutionary Connectionist Approach to Personnal Information Filtering. In *Proceedings of the Fourth Irish Neural Network Conference, Dublin, Ireland*, pages 141–146, 1994.

[PB97] M. Pazzani and D. Billisus. Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning Journal*, pages 313–331, 1997.

[Roc71] J.J Rocchio. Relevance Feedback in Information Retrieval. *In G. Salton, the SMART Retrieval System : Experiments in Automatic Document Processing*, pages 313–323, 1971.

[SG83] G. Salton and M.J Mc Gill. Introduction to Modern Information Retrieval. *New York: McGraw-Hill*, 1983.

[TT98] A. Tan and C. Teo. Learning User Profiles for Personalized Information Dissemination. In *Proceedings, 1998 IEEE International Joint Conference on Neural Networks, Alaska*, pages 183–188, 1998.

[VR79] C. J. Van Rijsbergen. *Information Retrieval, 2nd edition.* Dept. of Computer Science, University of Glasgow, 1979.

[WIY01] D.H. Widyantoro, T.R Ioerger, and J. Yen. Learning User Interest Dynamics with a Three-Descriptor Representation. *Journal of the American Society of Information Science*, 52(3):212–225, 2001.