PERCIRS: a PERsonalized Collaborative Information Retrieval System

Hassan NADERI
Béatrice RUMPLER

INSA de LYON, Bâtiment Blaise Pascal, 7, Av. Jean Capelle
F69621 Villeurbanne Cedex
{hassan.nadery, beatrice.rumpler}@insa-lyon.fr

RÉSUMÉ. *Pendant que le volume d'information augmente, l'importance de la recherche d'information augmente. La CIR (Collaborative Information Retrieval) est l'une des approches conventionnelles dans les systèmes de recherche d'information. Un système de CIR enregistre les interactions des utilisateurs pour répondre aux questions suivantes plus efficacement. Mais les buts et les caractéristiques de deux utilisateurs peuvent être différents; ainsi quand ils envoient la même requête à un système de CIR, ils peuvent être intéressés par deux listes de documents différentes. Dans cet article nous traitons le problème de personnalisation dans les systèmes de CIR en construisant le profil pour chaque utilisateur. Nous proposons trois nouvelles approches pour calculer la similitude des profils d'utilisateurs que nous les emploierons dans notre algorithme personnalisé de CIR.*

ABSTRACT. *As the volume of information augments, the importance of the Information Retrieval (IR) increases. Collaborative Information Retrieval (CIR) is one of the popular social-based IR approaches. A CIR system registers the previous user interactions to response to the subsequent user queries more efficiently. But the goals and the characteristics of two users may be different; so when they send the same query to a CIR system, they may be interested in two different lists of documents. In this paper we deal with the personalization problem in the CIR systems by constructing the profile for each user. We propose three new approaches to calculate the user profile similarity that we will employ them in our personalized CIR algorithm.*

MOTS-CLÉS: *recherche d'information de collaboration, CIR personnalisé, profil d'utilisateur, similitude de profil d'utilisateur.*

KEYWORDS: *collaboration information retrieval (CIR), personalized CIR, user profile, user profile similarity.*

1. Introduction

The ultimate goal of IR is to find the documents that are useful to the user's information need expressed as a query. Much work has been done on improving IR systems, in particular in the Text Retrieval Conference series [TREC 2006]. In 2000, it was decided at TREC-8 that this task should no longer be pursued within TREC, in particular because the accuracy has plateaued in the last few years [Voorhees and Harman 1999]. We are working on a new system which learns to improve retrieval effectiveness by integrating:

1. The user characteristics (user model or user profile).
2. The characteristics in the interaction of the other users (social IR, stereotypes and collaborative information retrieval).
3. The context of the research (context modelling).

Such system may have the potential to overcome the current plateau in ad-hoc retrieval. This paper concerns to two first elements: the user profile and the Collaborative Information Retrieval (CIR).

CIR is an approach which learns to improve retrieval effectiveness from the interaction of different users with the retrieval system. Collaboration here assumes that users can benefit from search processes carried out at former times by other users although they may not know about the other users and their search processes. In other words, collaborative search records the fact that a result *d* has been selected for query *q*, and then reuses this information for similar queries in the future, by promoting results that were reliably selected in the past.

However the goals and the characteristics of two users may be different so when they send the same query to a CIR system, they may be interested in two different lists of documents (known as **personalization problem**). Personalization is a common problem which the CIR researchers often encounter in constructing their systems. For instance Armin, who has presented three important approaches toward a CIR system in [Armin 2004], confessed that:

> *"We are aware of the problems of "personalization" and "context", but in our first steps towards techniques we avoid further complexity of CIR by ignoring these challenges. Personalization means that different users may have different preferences on relevant documents, because of long-term interests; context means that different users may have different preferences on relevant documents, because of short-term interests."*

Recently Barry S. et al. implemented a significant collaborative web search technique as a robust and scalable Meta search engine architecture in the form of I-SPY (*http://ispy.ucd.ie*) search engine [Smyth 2005]. They define collaborative web search as exploiting repetition and regularity within the query-space of a community of like-minded individuals in order to improve the quality of search results. However they state that: "*the precise nature of a community's shared interests may not be so easy to characterise*". Because of this difficulty I-SPY can't associate a user to a suitable community automatically. So I-SPY explicitly ask the users to recognize their community among a set of predefined communities at the time of inscription. This method has several restrictions which some of them are as:

1. Finding an appropriate community is a tedious task for a user especially when the number of communities multiplies rapidly (another search problem!).
2. These predefined communities are not exclusive. Thus in most of the times the user can't find an appropriate communities.
3. The interests of the user change over the time while assigning a user to a predefined community is a static task.
4. A user might want to search the different topics while he/she has just a community.
5. The communities are either very general or extremely specific to be helpful in retrieval process.

In the previous paragraphs we have shown that personalization is a serious problem in CIR systems. In this paper we create a PERsonalized CIR System (called PERCIRS) to overcome the personalization problem in CIR as like as in IR. Our personalized system is the first attempt toward resolving the problem of personalization in the CIR systems by incorporating the user profiles.

In section 2 we will present some further related works in the CIR. We explain the contradiction between the motivation of CIR and the personalization problem in the section 3. In section 4 we present our three methods for calculating the similarity between two profiles. In section 5 we present a personalized collaborative information retrieval algorithm which is based on our three profile similarity methods. Finally we conclude our paper in section 6.

2. Related work

[Fitzpatrick and Dent, 1997; Glance, 2001; Raghavan and Sever, 1995; Wen, 2002] have all demonstrated how query logs can be mined to identify useful past queries that may help the current searcher. In [Freyne et al., 2004; Smyth et al., 2003], a novel approach to Web search—collaborative Web search— was introduced. It combined techniques for exploiting knowledge of the query-space with ideas from social networking to develop a Web search platform capable of adapting to the needs of (ad-hoc) communities of users. In brief, the queries submitted and the results selected by a community of users are recorded and reused in order to influence the results of future searches for similar queries. Results that have been reliably selected for similar queries in the past are promoted. For example, users of an AI-related Web site might have a tendency to select case-based reasoning results in response to vague queries such as ‘CBR’, while largely ignoring alternatives such as Google’s higher-ranking ‘Central Bank of Russia’ or ‘Comic Book Resources’ results. In this instance collaborative search will gradually adapt its result-lists to emphasise case-based reasoning results, for searches that originate from the same community.

3. CIR motivations vs. personalization problem

The observations show that the users prefer to use only a few keywords (about 2 or 3 keywords) in making the queries [Yates and Riberio 1999]. In other words, the users give a general query to an IR system and subsequently try to find the desired documents with navigating the returned list by the system. As the users characterize their needs very general by constructing the short queries, it’s very probably that two different users make the same or similar queries for their needs. Thus CIR can be promising so in such cases the later users can use the history of interaction of the former users. In addition when a user sends a query to an IR system, he spends much of the time and effort to find the desired documents which are relevant to his query. In the current IR systems the user's efforts will be lost. We believe that these queries and their corresponding documents are the rich resources in order to make a more intelligent IR system.

The above discussion encourages the IR researcher to construct the efficient Collaborative Information Retrieval systems [Armin et al 2002 and 2004]. In a CIR the user interaction will be registered in a query-base, in order to avoid losing the user efforts. This query-base is composted of the *(q, Dq)* pairs where *q* is a sent query by the user and *Dq* is the set of selected documents as relevant by the query sender. In such an intelligent system when another user sends an equal or similar query to *q*, the system will retrieve the registered relevant documents to *q* in order to respond to this new query. As the user judgement of relevance of a document to a query is more accurate than machine judgement of relevance; we believe that methods which take advantage of the user experiences can improve the performance of IR systems in term of efficiency. Therefore the CIR systems tend to exploit these user-judgements to make reliable the machine-judgements.

A CIR system helps us to recognize the requirements of the users (which is one of the main problems in the IR systems) because as we know understanding the needs of a user by another user is extremely easier than by a machine. Thus a CIR system profits the fact that query-document relevancy recognition by a human is more precise than a machine. The researchers in the IR domain try to make the systems which would be able to recognize the users’ needs such as a human being (intelligent IR [Croft 1987, Giorgio 1987]). It seems that this goal is so difficult to achieve. We think that CIR is an indirect way to achieve this purpose by recognizing the needs of the users implicitly. In this approach, the system doesn't recognize the need of a user by analysing his query directly. Accordingly a CIR will be able to benefit from the human abilities in order to bring the machine judgement closer to the human judgement.

As we have mentioned early sending the short queries by the users are one of the CIR system’s motivations. However as the length of the queries were shorter the side effect of personalization problem would be greater. In other words the probability of sending the same queries by two different users for two different needs will increase. Personalisation problem means that when two different users send the same query to an IR system, they may be interested in two different documents (consider two users who send the same query “java”, one as an island and the other as a programming language). In fact personalization problem is a well-known problem in the IR systems. However its impact on a CIR system is greater because CIR systems rely entirely upon the judgements of the previous users, and if the needs of the previous users were very different, then the efficiency of the systems would be extremely decreased. As the diversity of the users in a search engine is high, using a pure CIR system such as I-SPY can be problematic.

According to the previous paragraphs, CIR systems are powerful in recognizing the requirements of the users but they suffer from the personalization problem. So if we can resolve the personalization problem in the CIR, then we can look forward to more efficient IR systems. In this paper we try to resolve the problem of personalization in the CIR by finding the users with the same requirements (or profile). In PERCIRS we exploit the history of interaction of a user for another user if and only if their profiles are similar to each other. So our main goal in this paper is to calculate the similarity between two user profiles. As a starting point to calculate the similarity between two user profiles we have proposed three methods that we will explain them in the following section.

4. Profiles similarity calculation methods

In this section we present some initial methods for calculating the similarity between user profiles. In this paper, for the reason of simplicity, we restrict ourselves to the history of the user queries as a user profile. So a user profile can be presented as a set of pairs *(q,D_q)* in which q is a query and D_q is a set of documents that the user has marked as relevant to q. Thus our ultimate goal is to calculate the similarity between two following sets in order to obtain the similarity between the users X and Y:

$$P(X)=\{(\mathrm{q}_1^{\mathrm{x}},D_{q1}^{\mathrm{x}}),(\mathrm{q}_2^{\mathrm{x}},D_{q2}^{\mathrm{x}}),...,(\mathrm{q}_{\mathrm{N}}^{\mathrm{x}},D_{qN}^{\mathrm{x}})\}$$
$$P(Y)=\{(\mathrm{q}_1^{\mathrm{y}},D_{q1}^{\mathrm{y}}),(\mathrm{q}_2^{\mathrm{y}},D_{q2}^{\mathrm{y}}),...,(\mathrm{q}_{\mathrm{M}}^{\mathrm{y}},D_{qM}^{\mathrm{y}})\}$$

Where q_i^x is the i-*th* query of the user X, and D_{qi}^x are the relevant documents to q_i^x according to X's judgments. A query *q* is a set of keywords which can be weighted or unweighted. In what follow we describe our three Profile Similarity (PS) calculation methods: query based PS, document based PS and query-document based PS.

4.1. Query based profile similarity

In this approach we merely consider the queries in the user profile. We believe that the users' queries can partially represent the needs and the preferences of the users because the users express their requirements formally with the queries. If we consider only the user queries in our calculation, the PS calculation problem will reduce to the following problem:

What is $S(P_q(X), P_q(Y))$ where $P_q(X) = \{q_1^x, q_2^x, ..., q_N^x\}$ and $P_q(Y) = \{q_1^y, q_2^y, ..., q_M^y\}$?

In the above problem we have two set of queries of two different users X and Y. Thus we are required to estimate the closeness of these two sets of queries to compute the similarity between the profiles of their corresponding users. There have been some research studies since the time this paper was written to calculate the similarity between two different queries. With regard to the type of queries (weighted or unweighted) there are two methods to calculate the similarity between two queries [Wen 2001]:

A- If the queries are not weighted: This measure directly comes from IR studies. Keywords are the words, except for words in the stop-list. All the keywords are stemmed using the Porter algorithm [Yates and Riberio 1999]. The similarity between two queries q_1 and q_2 is proportional to their common keywords:

$$S(q_1, q_2) = \frac{KN(q_1, q_2)}{Max(kn(q_1), kn(q_2))} \qquad (1)$$

Where *kn(.)* is the number of keywords in a query, *KN(q_1, q_2)* is the number of common keywords in two queries.

B- If the queries are weighted: If query terms are weighted, the following modified formula can be used:

$$S(q_1, q_2) = \frac{\sum_{i=1}^{K} cw_i(q_1) \times cw_i(q_2)}{\sqrt{\sum_{i=1}^{S} w_i^2(q_1)} \times \sqrt{\sum_{i=1}^{T} w_i^2(q_2)}} \qquad (2)$$

Where $cw_i(q_1)$ and $cw_i(q_2)$ are the weights of the *i-th* common keywords in the queries q_1 and q_2 respectively, and $w_i(q_1)$ and $w_i(q_2)$ are the weights of the *i-th* keywords in the queries q_1 and q_2 respectively. S and T are the number of the keywords in the queries q_1 and q_2 respectively and K is the number of common words in two queries.

In the following paragraphs we will use two preceding query similarity methods to calculate the similarity between two **set** of queries in order to obtain the PS between their corresponding profiles. In what follow we explain two different methods for PS calculating based on their queries.

The first formula considers the queries as inseparable object. So the similarity between two queries is 1 if they are exactly equal or 0 otherwise. The main idea this formula is that: the similarity between two set of queries is proportional to the number of common queries in them.

$$\begin{aligned}&S(P_q(X),P_q(Y)) = S(\{\mathrm{q}_1^{\mathrm{x}},\mathrm{q}_2^{\mathrm{x}},\ldots,\mathrm{q}_{\mathrm{N}}^{\mathrm{x}}\},\{\mathrm{q}_1^{\mathrm{y}},\mathrm{q}_2^{\mathrm{y}},\ldots,\mathrm{q}_{\mathrm{M}}^{\mathrm{y}}\}) = \\ &\frac{|\{\mathrm{q}_1^{\mathrm{x}},\mathrm{q}_2^{\mathrm{x}},\ldots,\mathrm{q}_{\mathrm{N}}^{\mathrm{x}}\} \cap \{\mathrm{q}_1^{\mathrm{y}},\mathrm{q}_2^{\mathrm{y}},\ldots,\mathrm{q}_{\mathrm{M}}^{\mathrm{y}}\}|}{\log(\mathrm{N}+\mathrm{M})}\end{aligned} \tag{3}$$

According to this method if the number of common queries in the two sets of queries increases, then the similarity between these profiles will also increase. We have used *log* in the above formula in order to normalize the impact of N+M on the PS calculation. We will use the function *log* for the next formulas too.

However two profiles may be similar while they have not any common queries. Thus the above formula may not be so efficient. In such cases the second formula (formula 4) which considers the queries as separable objects could be more efficient. Here the similarity between two queries is between 0 and 1 that can be calculated from formula 1 and 2 according the type of queries (weighted or not).

$$S(P_q(X),P_q(Y)) = \frac{\sum_{i=1}^{\mathrm{N}}\sum_{j=1}^{\mathrm{M}} s(q_i^{\mathrm{x}},q_j^{\mathrm{y}})}{\mathrm{N}\times\mathrm{M}} \tag{4}$$

This formula is based on the similarity between the queries which calculate the average similarity between the queries in two profiles. The $s(q_i^{\mathrm{x}},q_j^{\mathrm{y}})$ can be computed from the formula 1 or 2.

4.2. Document based profile similarity

In this approach we absolutely consider the documents that the user has studied or has marked as pertinent to his requirements. These marked documents lead us to determine the users' needs. When a user reads a particular document it can be judged that the user's need is related to the content of this document. Thus the marked documents in a user profile can be useful to estimate the similarity between two profiles. This approach is very similar to the former approach (query based) except that instead of queries it deals with the documents the user has marked before. By regarding purely the available documents in a profile, the problem of similarity calculation between two profiles reduce to the following problem:

What is $S(P_d(X),P_d(Y))$ where
$P_d(X)=\{D_1^x,D_2^x,...,D_N^x\}$ and $P(Y)=\{D_1^y,D_2^y,...,D_M^y\}$?

Where D_i^x and D_j^y are the i-*th* and j-*th* document in the profiles of users X and Y respectively.

In the documents the terms (keywords) are often (unlike the queries) weighted. We represent a document d_1 as a vector $\vec{d}_1$ in which the weight of each dimension is the *tf*idf* score of corresponding word. Cosine formula is one of the most common formulas for calculating the similarity between two documents:

$$S_d(d_1,d_2)=\cos ine(\vec{d}_1,\vec{d}_2)=\frac{\vec{d}_1.\vec{d}_2}{\left\|\vec{d}_1\right\|\times\left\|\vec{d}_2\right\|} \tag{6}$$

We will use the above formulas in order to calculate the similarity between two set of documents. In the first method we consider each document inseparable. So the similarity between two set of documents is proportional to the number of their common documents. Thus:

$$S(P_d(X),P_d(Y))=\mathrm{S}(\{D_1^x,D_2^x,...,D_N^x\},\{D_1^y,D_2^y,...,D_M^y\})= \frac{\left|\{D_1^x,D_2^x,...,D_N^x\}\cap\{D_1^y,D_2^y,...,D_M^y\}\right|}{\log(\mathrm{N}+\mathrm{M})} \tag{7}$$

In the formula (7) we calculate the similarity between two sets of documents by counting the number of common documents. In other words this formula is in the level of documents and it doesn't deal with the content of each document. One of the drawbacks of this formula is that if two sets of documents are very similar but they don't have many documents in common; this formula won’t be able to determine precisely their similarity. In order to overcome this difficulty we should consider the content of the documents in PS calculation as the following formula:

$$S(P_d(X),P_d(Y))=\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}s_d(D_i^x,D_j^y)}{\mathrm{N}\times\mathrm{M}} \tag{8}$$

In this formula, $s_d(D_i^x, D_j^y)$ is the similarity between two documents D_i^x and D_i^y that can be calculated by the cosine formula.

4.3. Query-document based profile similarity

In the last two approaches we computed the PS based on the users' queries and their marked documents respectively. Each of these approaches has some advantages. We think that both query based and document based approaches can partially capture somewhat the users' interests. Therefore, it seems to be better to use both of them with together. A simple way to do it is to combine both measures linearly as follows:

$$S(P(X), P(Y)) = \alpha \times S(P_q(X), P_q(Y)) + \beta \times S(P_d(X), P_d(Y)) \qquad (9)$$

Where $\alpha + \beta = 1$. There is an issue concerning the setting of parameters α and β which we have planed to estimate them experimentally in our subsequent investigations.

In the above formula the mutual connection between a query and its corresponding documents has not been considered because query similarity and document similarity have been calculated separately. However we believe that there is a semantic behind the relationship between a query and its corresponding documents which can be useful in enhancing the precision of PS calculation. In the following paragraphs we describe how we can calculate the similarity between two profiles exhaustively based on the queries, documents and their relationship. We called such approaches **Partially Complete Profile Similarity (PCPS)**, because they consider all of the available information in a user profile: query, document and the relationship between queries and documents. We have called these methods *partially* because in these calculations we deal with a simplified version of the user profile. Because of exhaustively PCPS methods, we believe that the best method for PS calculation between our proposed methods is a PCPS method. In what follows we will first explain our method for calculating the similarity between two pairs (q_1, D_{q1}) and (q_2, D_{q2}) with regard to relationship between the queries and their corresponding documents. Then we will represent our PCPS method.

4.3.1. The (q,Dq) similarity calculation

[Gui 2004] stated that in many IR systems, similarities between two objects of the same type (say, queries) can be affected by the similarities between their interrelated objects of another type (say, documents), and vice versa. They have introduced a new framework called *similarity spreading* to take into account the mutual interrelationship between a query and his corresponding documents, to enhance the efficiency of similarity calculation formula. In their calculation the similarity of the documents is a function of similarity of queries and vise versa (formula 10).

$$S_q(q_1,q_2) = f(S_d(D_{q_1},D_{q_2})) \\ S_d(d_1,d_2) = g(S_q(q_1,q_2)),\ d_1 \in D_{q_1}\ and\ d_2 \in D_{q_2} \tag{10}$$

D_{q_1} is a list of documents, the user has marked as relevant to the query q_1. They resume similarity calculation iteratively until values converge (figure 1).

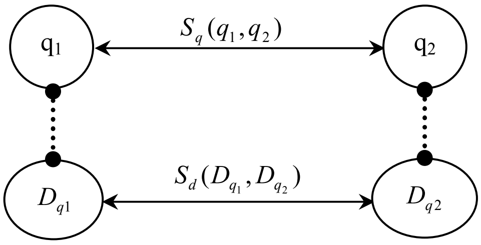

Figure 1: query and document similarity calculation separately and iteratively.

Their method is not very effective because they consider the queries and their associated documents as two separated objects, and convergence the query and documents similarity is difficult to reach. In our new method we consider the queries and their associated documents as two parts of an individual compound object. In what follows we explain the similarity calculation between these compound objects.

In the figure 2 we have represented a compound object with *qd* in order to represent its components: query (*q*) and document (*d*). In our *qd* similarity calculation method we consider a compound object as a point in a two dimensions space in which the query and the document are two axes. We use the distance formula for two points in a Cartesian space in order to calculate the distance between two *qd* objects; and after that the similarity between two *qd* objects can be calculated from their distance. The relation between the distance and the similarity between two objects (query, document and etc.) can be written as:

$$dis\tan ce_q(q_1,q_2) = \frac{1}{S_q(q_1,q_2)} \\ dis\tan ce_D(D_{q_1},D_{q_2}) = \frac{1}{S_d(D_{q_1},D_{q_2})} \\ dis\tan ce_{qd}(qd_1,qd_2) = \frac{1}{S_{qd}(qd_1,qd_2)} \tag{10}$$

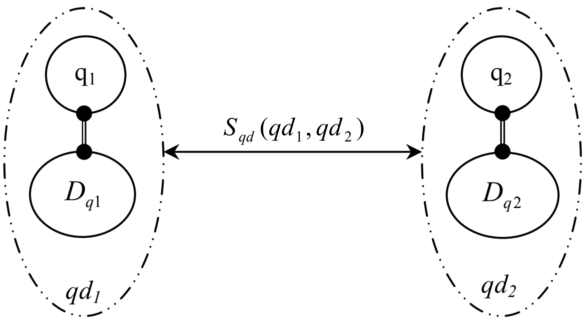

Figure 2: The (q,d) pairs similarity

With regard to the above discussion, the distance between two compound objects can be calculated as the following:

$$dis\tan ce_{qd}(qd_1, qd_2) = \sqrt{\alpha_q \times dis\tan ce_q^2(q_1, q_2) + \alpha_d \times dis\tan ce_D^2(D_{q_1}, D_{q_2})}$$

Where $\alpha_q + \alpha_d = 1$ (11)

Due to the difference between the units of axes *x* and *y*, we have incorporated the coefficients α_q and α_d in the distance calculating formula. Finally according to formulas 10 and 11, we will have:

$$S_{qd}((q_1, D_{q1}), (q_2, D_{q2})) = \frac{1}{dis\tan ce_{qd}(qd_1, qd_2)} = \frac{1}{\sqrt{\frac{\alpha_q}{S_q^2(q_1, q_2)} + \frac{1-\alpha_q}{S_d^2(D_{q1}, D_{q2})}}} \qquad (12)$$

Where $0 \le \alpha_q \le 1$.

4.3.2. PCPS calculation method

Now we are ready to calculate the PCPS between two profiles $P(X)$ and $P(Y)$ based on their *(q,D$_q$)* pairs as:

$$S(P(X), P(Y)) = S(\{(\mathrm{q}_1^{\mathrm{x}}, D_{q1}^{\mathrm{x}}), ..., (\mathrm{q}_{\mathrm{N}}^{\mathrm{x}}, D_{qN}^{\mathrm{x}})\}, \{(\mathrm{q}_1^{\mathrm{y}}, D_{q1}^{\mathrm{y}}), ..., (\mathrm{q}_{\mathrm{M}}^{\mathrm{y}}, D_{qM}^{\mathrm{y}})\})$$
$$= \frac{\sum_{i=1}^{N}\sum_{\mathrm{j}=1}^{M} S_{qd}((\mathrm{q}_{\mathrm{i}}^{\mathrm{x}}, D_{qi}^{\mathrm{x}}), (\mathrm{q}_{\mathrm{j}}^{\mathrm{y}}, D_{qj}^{\mathrm{y}}))}{\mathrm{N} \times \mathrm{M}} \qquad (13)$$

D_{qi}^{x} is the set of documents the user X has marked as relevant to query $q_{\mathrm{i}}^{\mathrm{x}}$. The similarity between two compound objects (S_{qd}) can be calculated from the formula 12.

5. The personalized CIR algorithm

In this section we describe our personalized collaborative information retrieval algorithm. When a user U sends a query q to PERCIRS, the system uses the following procedure to create a pertinent list of documents to q (figure 3). In this algorithm (U_i, q_i, D_{qi}) is a registered triple in U_i's profile in which U_i is the sender of q_i and D_{qi} is the set of relevant documents to q_i according to U_i. In the first step, PERCIRS selects those triples whose corresponding query has a similarity to q_i that is above some specified threshold; typically $\theta = 0.5$ according to [Smyth 2005]. In the current PERCIRS ω is equal to 0.5 (we will compute the optimal value of ω in our subsequent experiments).

```
//finding the similar queries to q that their sender is similar to U.
1. set $\mathrm{A} = \{(\mathrm{U}_1, q1, D_{q1}), (U_2, q2, D_{q2}), \ldots, (U_\mathrm{m}, qm, D_{qm})\}$ where
   $s(q, q_i) > \theta \quad \& \quad s(P(U), P(U_i)) > \omega \qquad 1 \le i \le m$

//calculating the set of all documents which can be relevant to q.
2. set $\mathrm{D}_\mathrm{q} = D_{q1} \cup D_{q2} \cup \ldots D_{qm}$

3. for each $d \in D_q$ calculate the PCIR rank:
   $R_{PCIR}(U, d, q) = \sum_{d \in D_{qi}} s(q, q_i) \times s(P(U), P(U_i))$

4. for each $d \notin D_q$ :
   $R_{PCIR}(U, d, q) = 0$

5. for each d in the corpus compute $R(d, q)$ with a traditional
   IR algorithm such as cosine similarity measure.

//calculate the final rank of each document in the corpus.
6. $R(U, d, q) = a \times R(d, q) + b \times R_{PCIR}(U, d, q)$

7. sort the documents by their final rank decreasingly for
   constructing the output list.
```

Figure 3: the personalized CIR algorithm.

D_q is the set of all documents which can be pertinent to q. In the steps 3 and 4, we give a personalized collaborative rank to each document d in the D_q. $R_{PCIR}(U, d, q)$ is the rank of d based on the judgments of the other users who are similar to U. In the step 5, the rank of each document in the corpus is calculated by an efficient content-based retrieval algorithm such as Okapi [Robertson 94]. Finally in the step 6 the collaborative rank and the content-based rank are combined to obtain the final rank of each document. The value of parameters *a* and *b* are dependent on the situation of the research. For example in the first steps of exploiting the system, the value of *a* will be much greater than the value of *b*.

6. Conclusion and the future works

In this paper we explained the problem of personalization in the IR systems. We have also expressed that the impact of this problem on the CIR systems is more serious than on the non collaborative IR systems due to the nature of CIR systems. We proposed to integrate the similarity of users in the process of CIR in order to make a more intelligent CIR system. Such a system can resolve the personalization problem in the CIR and more generally in the IR. We proposed to calculate the similarity between two user profiles based on their included queries, documents and more precisely based on the correlation the queries and the relevant documents. Subsequently we initiated three different methods to calculating the similarity between two user profiles.

We believe that a personalized CIR system such as PERCIRS could be successful in retrieving the more pertinent documents for a given query because:

1. A CIR system saves the efforts of the previous users.
2. Human-judgement is more reliability than machine-judgement.
3. The probability of posing the same or similar queries by different users is relatively high in the search engines.
4. And more importantly PERCIRS is able to resolve the personalization problem in the IR.

In the next step toward a personalized CIR system, we will study the efficiency of the proposed PS calculation methods in this paper. We will integrate the best PS calculation method in our personalized CIR algorithm to construct the final version of PERCIRS. We are interested to evaluate the performance of PERCIRS in relation to the other similar systems such as I-SPY [ISPY 2006] that are not personalized!

7. Acknowledgement

This research is partially supported by the French Ministry of Research and New Technologies under the ACI program devoted to Data Masses (ACI-MD), project #MD-33.

8. References

[Armin et al 2002] Armin H., Stefan K., Markus J., Andreas D., "*Towards Collaborative Information Retrieval: Three Approaches"*. In: Text Mining - Theoretical Aspects and Applications. 2002.

[Armin 2004] Armin H., "Learning Similarities for Collaborative Information Retrieval"*, Proceedings of the KI-2004 workshop "Machine Learning and Interaction for Text-Based Information Retrieval"*, TIR-04, Germany, 2004.

[Croft 1987] Croft W.B., "Approaches to intelligent information retrieval". *Information Processing and Management,* 23 (4): 249-254, 1987.

[Fitzpatrick and Dent, 1997] Larry Fitzpatrick and Mei Dent., "Automatic Feedback using Past Queries: Social Searching?" In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 306–313. ACM Press, 1997.

[Freyne *et al.*, 2004] Jill Freyne, Barry Smyth, Maurice Coyle, Evelyn Balfe, and Peter Briggs., "Further Experiments on Collaborative Ranking in Community-Based Web Search" *Artificial Intelligence Review*, 21(3–4):229–252, 2004.

[Giorgio 1987] Giorgio Brajnik, Giovanni Guida, and Carlo Tasso., "User modeling in intelligent information retrieval", *Information Processing and Management*, 23(4):305-320, 1987.

[Glance, 2001] Natalie S. Glance., "Community Search Assistant". In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 91–96. ACM Press, 2001.

[Gui 2004] Gui-Rong X. et al., "Similarity spreading: a unified framework for similarity calculation of interrelated objects", *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, 2004, New York, USA

[ISPY 2006] the site of I-SPY search engine., Available on: *http://ispy.ucd.ie*, 25/02/2006.

[Raghavan and Sever, 1995] Vijay V. Raghavan and Hayri Sever., "On the Reuse of Past Optimal Queries" In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 344–350. ACM Press, 1995.

[Robertson 94] Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M. M., Gatford, M., "Okapi at TREC-3", *NIST Special Publication 500-225: the Third Text REtrieval Conference (TREC-3),* pp. 109-126.

[Smyth *et al.*, 2003] Barry Smyth, Evelyn Balfe, Peter Briggs, Maurice Coyle, and Jill Freyne., "Collaborative Web Search" In *Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI-03*, pages 1417–1419. Morgan Kaufmann, 2003. Acapulco, Mexico.

[Smyth *et al.*] Barry Smyth, Evelyn Balfe, Jill Freyne, Peter Briggs, Maurice Coyle, and Oisin Boydell., "Exploiting Query Repetition & Regularity in an Adaptive Community-based Web

Search Engine" *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*.

[Smyth 2005] Smyth, B., Balfe, E. Boydell, O., Bradley, K., Briggs, P., Coyle, M., Freyne, J., "A Live User Evaluation of Collaborative Web Search", In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*. Edinburgh, Scotland, 2005.

[TREC 2006] the site of TREC: *Text REtrieval Conference*., Aavailable on : *http://trec.nist.gov/*, 25/02/2006.

[Yates and Riberio 1999] R. Baeza-Yates and B. Ribeiro-Neto., *Modern Information Retrieval,* Addison-Wesley, 1999.

[Voorhees and Harman]. Ellen M. Voorhees and Donna K. Harman., "Overview of the eighth text retrieval conference (TREC-8)" *NIST Special Publication 500-246*, pages 1–23, 1999.

[Wen 2001] Wen J., Nie J., and Zhang H., "Clustering user queries of a search engine", *In Proc. at 10th International World Wide Web Conference*, pages 162–168. W3C, 2001.

[Wen, 2002] Wen J., "Query clustering using user logs". *ACM Transactions on Information Systems*, 20(1):59–81, 2002.